

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
УЗБЕКИСТАН**

**ШАДМАНОВА ГУЛЧЕРА
КАРИМОВА ХАБИБАХОН ХАМДАМОВНА
КЕНДЖАЕВА ДИЛДОРА ХУДАЙБЕРГАНОВНА**

**«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ»**

УЧЕБНОЕ ПОСОБИЕ

5630200 - «Экологическая безопасность в водном хозяйстве»

Ташкент – 2020

Аннотация

Учебное пособие дает обширную информацию о структуре и возможностях современных компьютеров последнего поколения для решения специализированных задач, и их технического и программного обеспечения, операционной системе Windows, программы редактирования текстов, программы электронных таблиц, программы для создания базы данных, использования ресурсов Internet и электронной почты, экологической программы «Эко центр», алгоритмы и программы на языке C++ для решения инженерных и специализированных задач. А также охватывает общую концепцию построения экономико-математических моделей, в том числе, математические методы, используемые при решении экологических и экономических задач водного хозяйства. Учебное пособие предназначено для направления 5630200 - «Экологическая безопасность в водном хозяйстве».

Annotatsiya

Ushbu o'quv qo'llanmada mutaxassislik masalalarini echishda muhim bo'lgan zamonaviy kompyuterlarning tuzilishi va imkoniyatlari, texnik va dasturiy ta'minoti, Windows operatsion tizimi, matn muxarrirlaridan foydalanish, tahrirlash, elektron protsessor, ma'lumotlar bazasi bilan ishlash, maxsus ekologik dasturi "Eko sentr", Internet resurslaridan, elektron pochta xizmatidan to'g'ri va oqilona foydalanish haqidagi mavzular, ekologik masalalarini echish algoritmi va dasturlarini tuzish kabi masalalar o'z aksini topgan. Shuningdek, ekologik va iqtisodiy-matematik modellarni qurishning umumiy konsepsiyasini, suv xo'jaligida ekologiya va iqtisodiy masalalarini yechishda ishlatiladigan matematik usullarini o'z ichiga olgan. O'quv qo'llanma 5630200 - «Suv hojaligida ekologik xavfsizlik» bakalavriat ta'lim yo'nalishi talabalari uchun mo'ljallangan.

Annotation

The manual provides extensive information about the structure and capabilities of the latest generation of computers for solving specialized tasks, and their technical and software, Windows operating system, text editing software (MS Word), spreadsheet software (MS EXCEL), database creation software (MS Access), the use of Internet resources and e-mail, the "Eco Center" ecological program, C++-language algorithms and programs for solving engineering and specialized tasks. It also covers the general concept of constructing economic and mathematical models, including mathematical methods used in solving ecological and economic problems of water management. The manual is intended for areas 5630200 - "Ecological safety in water management".

Рецензенты: **М.Якубов** – д.т.н., профессор кафедры «Информационные технологии»
Ташкентского университета информационных технологий
Б.Худоёров – д.ф.-м.н., профессор кафедры «Высшая математика»,
ТИИИМСХ

ВВЕДЕНИЕ

Переоснащение всех отраслей Республики новейшей техникой и технологией, развитие компьютерной и телекоммуникационной системы, отвечающих высоким международным требованиям, стали одной из актуальнейших задач современности. Общество стало называться информационным в прямом смысле слова. В информационном обществе процесс компьютеризации даёт людям доступ к надёжным источникам информации, избавляет их от рутинной работы, обеспечивает высокий уровень автоматизации обработки информации в производственной и социальной сферах. В обществе изменяется не только производство, но и весь уклад жизни, система ценностей, возрастает значимость культурного досуга по отношению к материальным ценностям. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе производятся и потребляются интеллект, знания, что приводит к увеличению доли умственного труда. От человека потребуется способность к творчеству, возрастет спрос на знания. Усложнение индустриального производства, социальной, экономической и политической жизни, изменение динамики процессов во всех сферах деятельности человека привели, с одной стороны, к росту потребностей в знаниях, а с другой – к созданию новых средств и способов удовлетворения этих потребностей. Для решения данных вопросов началось проведение реформы в образовательной системе.

Реформы в образовательной системе коснулись также такой дисциплины, как «Экологическая безопасность в водном хозяйстве и математическое моделирование процессов».

Цель курса «Экологическая безопасность в водном хозяйстве и математическое моделирование процессов» – дать студентам фундаментальные и практические знания по теоретическим основам работы современных компьютеров последнего поколения для решения специализированных задач, и их технического и программного обеспечения,

операционной системе Windows, программы редактирования текстов MS Word, программы электронных таблиц MS Excel, программы для создания базы данных MS Access, использования ресурсов Internet и электронной почты, экологической программы «Эко центр», алгоритмы и программы на языке C++ для решения инженерных и специализированных задач. А также охватывает темы основы моделирования, моделирование эколого-экономических процессов, использование различных методов при решении экологических и экономических задач водного хозяйства.

Учебное пособие состоит из трёх глав, списка использованной литературы. В конце каждой главы приведены ключевые слова, задачи для самостоятельной работы, методические указания для выполнения самостоятельных работ на компьютере, контрольные вопросы.

ГЛАВА I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Лекция 1. ПОНЯТИЕ ИНФОРМАЦИИ. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ. ГЕОИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1.1. Понятие информации. Виды и качество информации.

Термин *информация* происходит от латинского *information*, что означает разъяснение, осведомление, изложение. С позиции материалистической философии информация есть отражение реального мира с помощью сведений (сообщений). Сообщение – это форма представления информации в виде речи, текста, изображения, цифровых данных, графиков, таблиц и т.п. В широком смысле информация – это общенаучное понятие, включающее в себя обмен сведениями между людьми, обмен сигналами между живой и неживой природой, людьми и устройствами.

Информация – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Информатика рассматривает информацию как концептуально связанные между собой сведения, данные, понятия, изменяющие наши представления о явлении или объекте окружающего мира. Наряду с информацией в информатике часто употребляется понятие *данные*. Покажем, в чем их отличие.

Данные могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, а только хранятся. В том случае, если появляется возможность использовать эти данные для уменьшения неопределенности о чем-либо, данные превращаются в информацию. Поэтому можно утверждать, что информацией являются используемые данные.

Современное общество характеризуется широким внедрением информационных технологий во все сферы деятельности, как отдельного индивидуума, так и промышленного производства, финансовой деятельности и торговли, государственного управления, здравоохранения, одним словом - общества в целом. Поэтому с полным правом можно утверждать, что в настоящее время завершается переход от индустриального общества к информационному обществу.

К информационным технологиям, или как в настоящее время принято говорить - ИТ относится совокупность методов и средств сбора, хранения, обработки, передачи и ввода/вывода информации. Целью информационных технологий является снижение трудоемкости процессов использования информационных ресурсов, повышение надежности и оперативности выполнения информационных процессов.

Информация - (от латинского informatio - разъяснение, изложение), первоначальное понятие - сведения, передаваемые людьми устным, письменным или другим способом (с помощью условных сигналов, технических средств и т. д.); с середины XX века. общенаучное понятие, включающее обмен сведениями между людьми, человеком и автоматом, автоматом и автоматом; одно из основных понятий кибернетики. Законодательно термин «информация» определен следующим образом: «Информация - сведения (сообщения, данные) независимо от формы их представления».

Сегодня, в век информатизации и компьютеризации, информация является таким же ресурсом, как трудовые, материальные и энергетические.

Информация - ценнейший ресурс наряду с такими традиционными видами ресурсов, как нефть, газ, полезные ископаемые и др., а значит, процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию.

Общеизвестно, что исторически первым носителем человеческих информации, знаний была речь, представлявшая изначально кодированные

звуки для координации действий в человеческом сообществе. Затем появилось наскальное письмо каменного века, далее пиктограммы (“иконы”) бронзового века, иероглифическое письмо (сохраненное до сих пор, например, в Китае) и письмо “обычное” - конкатенацией букв алфавита в слоги (“слоговое письмо”) и т.д.

Объединение систем, процессов, связанных с понятиями “информация”, “управление” привело к появлению нового предмета “кибернетика (или науки об управлении в живых организмах и автоматах)” (40-ые годы XX века), изучающей информационные процессы в живых организмах и машинах (автоматах). Кибернетика явилась одной из важных предпосылок появления и развития информатики. В последнее время, предмет кибернетики понемногу, видимо, “поглощается” предметом информатики. Но при этом информатика не зачеркивает кибернетику, которая теперь может развиваться сильнее, используя результаты, методы и технологии информатики.

Подлинная информационная революция связана, прежде всего, с созданием электронно-вычислительных машин в конце 40-х годов XX века, и с этого же времени исчисляется эра развития информационной технологии, материальное ядро которой образует микроэлектроника. Микроэлектроника формирует элементную базу всех современных средств приема, передачи и обработки информации, систем управления и связи. Сама микроэлектроника возникла первоначально именно как технология: в едином кристаллическом устройстве оказалось возможным сформировать все основные элементы электронных схем.

Документированная информация (документ) - зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать;

Экономическая информация – (от англ. economic information) - сведения, уменьшающие неопределенность, недостаток знаний, дополняющие представление о социально-экономических процессах, как в

сфере производства, так и в непроеизводственной сфере на всех уровнях и во всех органах. Экономической информации отображает факты производственно-хозяйственной деятельности с помощью системы натуральных и стоимостных показателей. Экономическая информация, как правило, передается и обрабатывается в форме знаков, фиксируемых на различных вещественных носителях. Экономическую информацию следует отличать от понятия «экономические данные», которые представляют собой отображение экономических явлений, записанных на определенном языке на конкретных материальных носителях (показатели, записанные в документах, текстах или таблицах), но не связанных с определенной задачей или потребителем. Экономические данные только тогда становятся экономической информацией, когда заключенные в них сведения используются для решения определенных задачи и уменьшают неопределенность условий этой задачи.

Как и любая информация, экономическая информация исследуется в трех аспектах: синтаксическом, семантическом, прагматическом.

Синтаксический анализ исследует отношения между знаками, представляющими экономическую информацию, отвлекаясь от их содержания и ценности для получателя. Он устанавливает важнейшие параметры информационных потоков, включая количественные характеристики, необходимые для выбора технических средств, осуществляющих сбор, регистрацию, передачу, хранение и переработку экономической информации.

Семантический анализ рассматривает содержание экономической информации и способы его языкового представления (язык человека, компьютерные языки), но также не учитывает ее ценность (т.е. фактически рассматривает экономические данные).

Прагматический анализ осуществляется для определения полезности экономической информации в процессе ее использования получателем для различных целей и принятия решений, т.е. именно прагматический анализ

применяется в тех условиях, когда экономические данные превращаются в экономическую информацию.

Для каждого вида анализа существуют свои количественной оценки и критерии. Наиболее существенными для практического применения являются оценки полезности экономической информации, поскольку они позволяют определить потребительскую ценность поступившей информации с точки зрения возможности ее использования в процессах прогнозирования и управления экономикой. Ценность экономической информации определяется такой совокупностью ее свойств, как своевременность, достоверность, содержательность, удобство восприятия и т.п. Экономическая информация имеет ряд особенностей, облегчающих применение компьютерных методов анализа. Большинство экономических показателей измеряется в количественном выражении (стоимость, объем продукции, прибыль и т.п.); цикличность, характерная для большинства производственных и хозяйственных процессов позволяет многократно использовать созданную программу; объемность информационных потоков (детальная информация об экономических процессах позволяет применять математические модели для прогнозирования и управления ими); возможность контроля за достоверностью информации (передача и обработка экономической информации осуществляется лишь на юридически оформленном носителе - документе с подписью, электронном сообщении с электронной цифровой подписью т.п.).

Информационные процессы - процессы сбора, обработки, накопления, хранения, поиска и распространения информации;

Информационная система - организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы; *Информационные ресурсы* - отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах

(библиотеках, архивах, фондах, банках данных, других информационных системах);

Информация о гражданах (персональные данные) - сведения о фактах, событиях и обстоятельствах жизни гражданина, позволяющие идентифицировать его личность; *Конфиденциальная информация* - документированная информация, доступ к которой ограничивается в соответствии с законодательством региона;

Средства обеспечения автоматизированных информационных систем и их технологий - программные, технические, лингвистические, правовые, организационные средства (программы для электронных вычислительных машин; средства вычислительной техники и связи; словари, тезаурусы и классификаторы; инструкции и методики; положения, уставы, должностные инструкции; схемы и их описания, другая эксплуатационная и сопроводительная документация), используемые или создаваемые при проектировании информационных систем и обеспечивающие их эксплуатацию;

Собственник информационных ресурсов, информационных систем, технологий и средств их обеспечения - субъект, в полном объеме реализующий полномочия владения, пользования, распоряжения указанными объектами;

Владелец информационных ресурсов, информационных систем, технологий и средств их обеспечения - субъект, осуществляющий владение и пользование указанными объектами и реализующий полномочия распоряжения в пределах, установленных Законом;

Пользователь (потребитель) информации - субъект, обращающийся к информационной системе или посреднику за получением необходимой ему информации и пользующийся ею.

Информатизация общества – совокупность научных, социально-экономических и политических факторов, обеспечивающих каждому члену общества к любым источникам информации. Поэтому к основным

проблемам формирования информационного общества относится реализация права доступа к информации, защиты персональных данных, прав и свобод личности, идентификации информации, повсеместное использование электронной цифровой подписи.

Глобализация информационной сферы – современное общество, его информационные потоки становятся интернациональными и «не знают границ». Можно сказать, что любая информация, созданная в какой-либо стране, может быть практически мгновенно распространена в сети Интернет (причем, совсем не обязательно с помощью средств именно данной страны) и далее – доступна всему миру.

Информационный сервис представляет собой вид деятельности, направленный на удовлетворение потребностей клиента путем оказания услуг информационного характера. Информационный сервис обеспечивает познавательную деятельность общества и может быть представлен в виде двух уровней деятельности: эмпирическом (предоставление информации об отдельных фактах и событиях) и теоретическом (анализ информации, выявление закономерностей в данной сфере явлений).

На эмпирическом уровне информационного сервиса предоставляется информация (справки) о компаниях (местонахождение, телефонах, адресах электронной почты e-mail, Web - сайта), о наличии товара, о предоставлении услуг, о расписаниях, о событиях политической, культурной жизни, юридическая информация и т.п. Следует выделить в данном уровне сервиса получение клиентом справки (выписки) о состоянии банковского счета и т.п. информации, очевидно, что данная информация является конфиденциальной и может быть предоставлена только владельцу счета или его доверенному лицу.

На теоретическом уровне информационного сервиса осуществляется анализ ситуации в экономике (региона, компании и т.п.), изучение общественного мнения, перспектив развития спроса на товары, услуги и т.п.

Качество информации

Возможность и эффективность использования информации обуславливается такими основными ее потребительскими *показателями качества*, как репрезентативность, содержательность, достаточность, доступность, актуальность, своевременность, точность, достоверность, устойчивость.

Репрезентативность информации связана с правильностью ее отбора и формирования в целях адекватного отражения свойств объекта. Важнейшее значение здесь имеют:

- правильность концепции, на базе которой сформулировано исходное понятие;
- обоснованность отбора существенных признаков и связей отображаемого явления.
- Нарушение репрезентативности информации приводит нередко к существенным ее погрешностям.

Содержательность информации отражает семантическую емкость, равную отношению количества семантической информации в сообщении к объему обрабатываемых данных, т.е. $\tilde{N} = \frac{I_c}{V_A}$.

С увеличением содержательности информации растет семантическая пропускная способность информационной системы, так как для получения одних и тех же сведений требуется преобразовать меньший объем данных.

Наряду с коэффициентом содержательности *С*, отражающим семантический аспект, можно использовать и коэффициент информативности, характеризующийся отношением количества синтаксической информации (по Шеннону) к объему данных $Y = \frac{I}{V_A}$.

Достаточность(полнота) информации означает, что она содержит минимальный, но достаточный для принятия правильного решения состав (набор) показателей. Понятие полноты информации связано с ее смысловым содержанием (семантикой) и прагматикой. Как неполная, т.е. недостаточная

для принятия правильного решения, так и избыточная информация снижает эффективность принимаемых пользователем решений.

Доступность информации восприятию пользователя обеспечивается выполнением соответствующих процедур ее получения и преобразования. Например, в информационной системе информация преобразовывается в доступную и удобную для восприятия пользователя форму.

Актуальность информации определяется степенью сохранения ценности информации для управления в момент ее использования и зависит от динамики изменения ее характеристики и от интервала времени, прошедшего с момента возникновения данной информации.

Своевременность информации означает ее поступление не позже заранее назначенного момента времени, согласованного с временем решения поставленной задачи.

Точность информации определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления близости получаемой информации цифровым кодом. Известны четыре классификационных понятия точности:

- формальная точность, измеряемая значением единицы младшего разряда числа;
- реальная точность, определяемая значением единицы разряда числа, верность которого гарантируется;
- максимальная точность, которую можно получить в конкретных условиях функционирования системы;
- необходимая точность, определяемая функциональным назначением показателя.

Достоверность информации определяется ее свойством отражать реально существующие объекты с необходимой точностью. Измеряется достоверность информации доверительной вероятностью необходимой точности, т.е. вероятностью того, что отображаемое информацией значение

параметра отличается от истинного значения этого параметра в пределах необходимой точности.

Устойчивость информации отражает ее способность реагировать на изменения исходных данных без нарушения необходимой точности. Устойчивость информации, как и репрезентативность, обусловлена выбранной методикой ее отбора и формирования.

В заключение следует отметить, что такие параметры качества информации, как репрезентативность, содержательность, достаточность, доступность, устойчивость, целиком определяется на методическом уровне разработки информационных систем.

Параметры актуальности, своевременности, точности и достоверности обуславливаются в большей степени также на методическом уровне, однако на их величину существенно влияет и характер функциональности и точности.

1.2. Информационные технологии. Методы и средства информационных технологий.

Истоки и этапы развития. Информационные технологии (ИТ) можно представить совокупностью трёх основных способов преобразования информации: хранения, обработки и передачи. Вся история становления ИТ неразрывно связана со становлением и развитием этих трёх способов, проходивших в несколько этапов, которые можно сгруппировать в три революционных периода.

Предварительные этапы развития ИТ. На раннем этапе развития общества профессиональные навыки передавались в основном личным примером по принципу «делай как я». В качестве способа передачи информации использовались ритуальные танцы, обрядовые песни, устные предания и т.д., которые реализовывались человеком.

Первый этап развития ИТ связан с открытием способов длительного хранения информации на материальном носителе. Это и пещерная живопись,

сохраняющая наиболее характерные зрительные образы, связанные с охотой и ремеслами (примерно 25 – 30 тыс. лет назад); и гравировка по кости, обозначающая лунный календарь, а также числовые нарезки для измерения (выполненные примерно 20 – 25 тыс. лет назад). Способы хранения информации подверглись совершенствованию, а период до появления инструментов для обработки материальных объектов и регистрации информационных образов на материальном носителе составил около 1 млн. лет или 1% времени существования цивилизации. Становится понятно, почему при решении абстрактных информационных задач эффективность человека резко возрастает при представлении информации в виде изображений материальных объектов (использование графических интерфейсов). В этом случае включаются в работу те области человеческой интуиции, которые развивались в первые 99% времени существования цивилизации.

Второй этап развития ИТ начал свой отсчёт около 6 тыс. лет назад и связан с появлением письменности. Эра письменности характеризуется появлением новых способов регистрации на материальном носителе символической информации. Применение этих технологий позволяет осуществлять накопление и длительное хранение знаний. В качестве носителей информации на втором этапе развития ИТ использовались: камень, кость, дерево, глина, папирус, шёлк, бумага. Этот ряд можно продолжить: магнитные покрытия (лента, диски, цилиндры и т.д.), жидкие кристаллы, оптические носители, полупроводники и т.д. В этот период накопление знаний происходит достаточно медленно и обусловлено трудностями, связанными с доступом к информации (недостаток второго этапа развития ИТ). Знания, представленные в виде рукописных изданий, хранятся в единичных экземплярах. Причём доступ к ним существенно затруднён, так как они охранялись специальной кастой – жрецами, которые наделялись исключительным правом монопольного доступа к фонду человеческого опыта и являлись посредниками между накопленными

знаниями и заинтересованными людьми. Этот барьер был разрушен на следующем этапе.

Первая информационная революция. Начало третьего этапа датируется 1445 г., когда Иоганн Гуттенберг изобрёл печатный станок, и подводит итог становлению способов регистрации информации. Появление книг открыло доступ к информации широкому кругу людей и резко ускорило темпы накопления систематизированных по отраслям знаний. За три столетия после изобретения печатного станка оказалось возможным накопить ту «критическую массу» социально доступных знаний, при которой начался лавинообразный процесс развития промышленной революции. Печатный станок сыграл роль информационного ключа, резко повысив пропускную способность социального канала обмена знаниями.

Характерным признаком первой информационной революции является то, что с этого момента началось необратимое поступательное движение технологической цивилизации. Книгопечатание – это первая информационная революция.

Вторая информационная революция. В 1946 году начинается четвёртый этап развития ИТ, который обусловлен появлением электронной вычислительной машины (ЭВМ) для обработки информации. Этой машиной является первая ЭВМ (типа ENIAC), запущенная в эксплуатацию в Пенсильванском университете. Данная машина не имела хранимой программы, которая задавалась путём шнуровой коммутации (аналог табуляторов – счётно-решающих машин). Электронно-вычислительная машина UNIVAC (1949) уже использовала общую память и для программ, и для данных, что обеспечивало сохранение программ на носителе (магнитных лентах, магнитных барабанах). К этому времени уже значительная часть населения была занята в информационной сфере.

Третья информационная революция. Совершенствование способов обработки информации вызвало развитие способов передачи информации –

появление информационно-вычислительных (компьютерных) сетей. В 1983 году (пятый этап) Международная организация по стандартизации (International Standard Organization – ISO) разработала систему стандартных протоколов, получившую название модели взаимодействия открытых систем (Open System Interconnection – OSI) или эталонной модели взаимодействия открытых систем. Модель OSI представляет самые общие рекомендации для построения стандартных совместимых сетевых программных продуктов, служит базой для разработки сетевого оборудования. Появление этого стандарта сыграло важную роль при формировании различных компьютерных сетей, в том числе Internet.

Характерным признаком третьей информационной революции является то, что некоторые авторы, анализируя ИТ, которые используются в сети Internet, сравнивают его с нейронной сетью и обсуждают вопрос о возникновении и развитии нейронной сети планеты и становлении планетарного разума.

В целом можно выделить основные особенности информационных технологий:

- целью информационного технологического процесса является получение информации;
- предметом технологического процесса (предметом обработки) являются данные;
- средства, которые осуществляют технологический процесс – это разнообразные вычислительные комплексы (программные, аппаратные, программно-аппаратные);
- процессы обработки данных разделяются на операции в соответствии с выбранной предметной областью;
- управляющие воздействия на процессы осуществляется руководящим составом организации;
- критериями оптимальности информационного технологического

процесса являются своевременность доставки информации пользователям, её надёжность, достоверность и полнота.

Информационная технология направлена на целесообразное использование информационных ресурсов и снабжение ими всех элементов организационной структуры. Информационные ресурсы являются исходным «сырьём» для системы управления любой организации, учреждения, предприятия, а конечным продуктом является принятое решение. Принятие решения в большинстве случаев осуществляется в условиях недостатка информации, поэтому степень использования информационных ресурсов во многом определяет эффективность работы организации.

Таким образом, основная цель автоматизированной информационной технологии – получать посредством переработки первичных данных информацию нового качества, на основе которой вырабатываются оптимальные управленческие решения.

Основная цель информационной технологии достигается за счёт:

- интеграции информации;
- обеспечения актуальности и непротиворечивости данных;
- использования современных технических средств для внедрения и функционирования качественно новых форм информационной поддержки деятельности аппарата управления.

Информационная технология справляется с существенным увеличением объёмов перерабатываемой информации, ведёт к сокращению сроков её обработки и является наиболее важной составляющей процесса использования информационных ресурсов в управлении.

Автоматизированная информационная технология непосредственно связана с особенностями функционирования предприятия или организации.

Выбор стратегии организации автоматизированной информационной технологии определяется следующими факторами:

- областью функционирования предприятия или организации;
- типом предприятия или организации;
- производственно-хозяйственной или иной деятельностью;
- принятой моделью управления организацией или предприятием;
- новыми задачами в управлении;
- существующей информационной инфраструктурой и т.д.

В информационном обществе производство и потребление информации является важнейшим видом деятельности; возрастает значимость информационной составляющей в структуре ресурсного обеспечения различных сфер человеческой деятельности; информационные технологии и техника определяют прогрессивное развитие производственных и социальных процессов; информационная среда наряду с социальной и экологической становится новой средой обитания человека. Информационные технологии играют определяющую роль в обеспечении информационного взаимодействия между людьми; являются технологическим фундаментом интеллектуализации общества, развития системы образования и культуры, играют ключевую роль в процессах получения и накопления новых знаний.

По мнению специалистов, в определении ИТ подчёркивают широкий и узкий смысл понятия. *В широком смысле* ИТ – это наука, научная дисциплина о методах описания, разработки и создания информационных технологий. *В узком смысле* – это целенаправленная деятельность по реализации информационных процессов с помощью современных методов и средств, с одной стороны, с другой стороны, ИТ – это представленное в проектной форме концентрированное выражение научных знаний и практического опыта, т.е. не что иное, как технологическая документация, технологическая карта и т.п.

Следует отметить, что механический перенос понятия «технология» имеет положительные и отрицательные стороны. Положительное – широкое использование и популяризация этого термина в теории и практике

информатики. Отрицательные стороны связаны с отождествлением материальной (вещественной) и идеальной (интеллектуальной) природы информации. Второй важной составляющей понятия ИТ является понятие «информация». В последнее время термин «информация» приобретает всё большую популярность и привлекает к себе внимание многих учёных и специалистов. Однако, несмотря на это, единый взгляд на познание сущности «информация» в настоящее время не выработан.

В социальной среде информация часто проявляется в форме сообщения, данных, семантической информации и знания. Информация, предназначенная для передачи, называется *сообщением*. Сообщение представляется в виде знаков и символов на определённом языке и носителе. *Данные* – это организованные по определённым правилам знаки (символы) для представления различных изменений состояний окружающей нас действительности. Представляя информацию в виде данных, потребитель не учитывает её содержание (смысл).

Семантический аспект рассмотрения информации связывают с организацией данных, которые способны изменять состояние потребителя информации на осведомленное (информированное), т.е. несущее определённый смысл. Состояние потребителя информации характеризуется семантической структурой, которая называется *тезаурусом*. При получении информации тезаурус может изменяться, и степень этого изменения характеризует воспринятое количество информации. Такую информацию будем называть *семантической информацией*.

Интерпретация полученных данных, семантической информации предполагает использование познавательных процессов более высокого уровня – это анализ, синтез, систематизация, классификация и т.п., в результате которых наступает понимание и способность распространять его на различные ситуации. В этом случае принято говорить, что потребитель обладает *знанием*.

Таким образом, термин «информация» используется как обобщённое понятие, включающее различные формы её проявления и функционирования – в виде сообщения и данных, сведений и семантической информации, знаний. Все они между собой тесно взаимосвязаны и их выделить можно только условно.

Элементный аспект предполагает описание состава системы, количественную и качественную характеристику частей, компонентов, их координацию и субординацию, приоритетную (лидирующую) часть системы. ИТ, как наука и целенаправленная деятельность по переработке информации, делится на технологическую и обеспечивающую части.

Объектом исследований информационной технологии как научной дисциплины являются информационные технологии (в узком понимании этого термина), т.е. способы рациональной организации информационных процессов. *Предметом* исследований должны стать теоретические основы и методы создания информационных технологий, а также их проектирование и эффективная реализация.

Колин К. К. выделяет следующие первоочередные задачи теории информационной технологии:

1. Разработка методов структуризации и классификации информационных технологий различного вида и назначения по их характерным признакам.
2. Разработка критериев эффективности информационных технологий, методов их оптимизации и сравнительной количественной оценки.
3. Определение перспективных направлений развития информационных технологий на ближайшие годы, а также научных методов, которые должны лежать в их основе.
4. Определение принципов построения перспективных средств для реализации высокоэффективных информационных технологий нового поколения.



Рис. 1.1. Модель информационной технологии

В технологическую подсистему входят компоненты, осуществляющие преобразования информации в соответствии с характеристиками разработанными теорией ИТ. Это объект преобразования (операнд) – данные, семантическая информация, знания, технологический принцип (способ) преобразования, действия над операндом, структура технологического информационного процесса, конечный результат (операнд с требуемыми свойствами).

Средства обеспечения ИТ представлены методами, техническими средствами (аппаратные средства ЭВМ, оргтехника и др.), алгоритмическими и программными средствами, информационным и методическим обеспечением, компьютерными сетями и телекоммуникациями, персоналом и др.

Для достижения целей ИТ необходимо выполнить определённые функции. Главными из них являются формирование концептуальной модели ИТ, преобразование данных, информации и знаний (сбор, обработка,

хранение, передача, распространение и др.) и функции по обеспечению ИТ. Внешние функции реализуются для удовлетворения потребностей в качественных и эффективных ИР разнообразных элементов: государства, политической среды, социальной и производственной сферы, науки, экономики, технологии и др.

Свойства и основные направления развития информационных технологий.

Информационная технология предполагает умение грамотно работать с информацией, программными продуктами и вычислительной техникой. Эффективность функционирования информационной технологии определяется её основными свойствами, представленными ниже.

1. Целесообразность – состоит в повышении эффективности производства за счёт внедрения современных средств вычислительной техники, распределённых баз данных, различных вычислительных сетей, что позволяет обеспечить эффективную циркуляцию и переработку информации.

Функциональные компоненты – это конкретное содержание процессов циркуляции и обработки данных (информационная база ИТ).

Структура информационной технологии – это внутренняя организация, представляющая собой взаимосвязанные компоненты ИТ.

2. Наличие компонентов и структуры. В состав информационной технологии должны входить:

- комплекс технических средств (КТС), состоящий из средств вычислительной, коммуникационной и организационной техники;
- программные средства, состоящие из общего (системного), прикладного (программ для решения функциональных задач специалистов) и инструментального программного обеспечения (алгоритмических языков, систем программирования, языков спецификаций, технологии программирования и т.д.);
- система организационно-методического обеспечения, включающая нормативно-методические и инструктивные материалы по организации

работы управленческого и технического персонала конкретной ИТ.

3. Взаимодействие с внешней средой предполагает организацию взаимосвязи информационной технологии с объектами управления, внешними предприятиями, организациями, включая потребителей и поставщиков продукции, финансово-кредитные органы и т.д. Взаимодействие информационных технологий различных объектов организуется посредством программных и технических средств автоматизации.

4. Целостность. Информационная технология является целостной системой, способной решать задачи, не свойственные ни одному из её компонентов.

5. Развитие во времени – это обеспечение динамичности развития информационной технологии, возможность её модернизации и модификации, изменение структуры, включение новых компонентов, возможность решения новых задач и т.д.

Применение автоматизированных информационных технологий позволило представить в формализованном виде, пригодном для практического использования, концентрированное выражение научных знаний и практического опыта для реализации и организации социальных процессов. При этом предполагается экономия затрат труда, времени и других материальных ресурсов, необходимых для осуществления этих процессов. Поэтому автоматизированные информационные технологии играют важную стратегическую роль, которая постоянно возрастает. Можно выделить семь основных направлений, по которым информационная технология оказывает непосредственное влияние на развитие экономики и общества.

1. Информационные технологии позволяют активизировать и эффективно использовать информационные ресурсы общества, которые сегодня являются наиболее важным стратегическим фактором его развития.

Опыт показывает, что активизация, распространение и эффективное использование информационных ресурсов (научных знаний, открытий,

изобретений, технологий, передового опыта) позволяют получить существенную экономию других видов ресурсов: сырья, энергии, полезных ископаемых, материалов и оборудования, человеческих ресурсов и т.д.

2. Информационные технологии позволяют оптимизировать и во многих случаях автоматизировать информационные процессы, которые в последние годы занимают всё большее место в жизнедеятельности человеческого общества.

Общеизвестно, что развитие цивилизации происходит в направлении становления информационного общества, в котором объектами и результатами труда большинства занятого населения становятся уже не материальные ценности, а, главным образом, информация и научные знания. В настоящее время в большинстве развитых стран большая часть занятого населения в той или иной мере связана с процессами подготовки, хранения, обработки и передачи информации и поэтому вынуждена осваивать и практически использовать соответствующие этим процессам информационные технологии.

3. Информационные технологии выступают в качестве компонентов соответствующих производственных или социальных технологий.

Объясняется это тем, что информационные процессы являются важными элементами других более сложных производственных или же социальных процессов. При этом они, как правило, реализуют наиболее важные, «интеллектуальные» функции этих технологий. Характерными примерами являются системы автоматизированного проектирования промышленных изделий, гибкие автоматизированные и роботизированные производства, автоматизированные системы управления технологическими процессами и т.п.

4. Информационные технологии сегодня играют исключительно важную роль в обеспечении информационного взаимодействия между людьми, а также в системах подготовки и распространения массовой информации.

В дополнение к ставшим уже традиционными средствами связи (телефон, телеграф, радио и телевидение) в социальной сфере всё более широко используются системы электронных телекоммуникаций, электронная почта, факсимильная передача информации и другие виды связи. Эти средства быстро ассимилируются культурой современного общества, так как они не только создают большие удобства, но и снимают многие производственные, социальные и бытовые проблемы, вызываемые процессами глобализации и интеграции мирового общества, расширением внутренних и международных экономических и культурных связей, миграцией населения и его всё более динамичным перемещением по планете.

5. Информационные технологии занимают сегодня центральное место в процессе интеллектуализации общества, развития его системы образования и культуры.

Практически во всех развитых и во многих развивающихся странах компьютерная и телевизионная техника, учебные программы на оптических дисках и мультимедиа -технологии становятся привычными атрибутами не только высших учебных заведений, но и обычных школ системы начального и среднего образования. Использование обучающих информационных технологий оказалось весьма эффективным методом и для систем самообразования, продолженного обучения, а также для систем повышения квалификации и переподготовки кадров.

6. Информационные технологии играют в настоящее время ключевую роль также и в процессах получения и накопления новых знаний.

На смену традиционным методам информационной поддержки научных исследований путём накопления, классификации и распространения научно-технической информации, приходят новые методы, основанные на использовании, вновь открывающихся возможностей информационной поддержки фундаментальной и прикладной науки, которые предоставляют современные информационные технологии.

Современные методы получения и накопления знаний базируются на теории искусственного интеллекта, методах информационного моделирования, когнитивной компьютерной графики, позволяющих найти решения плохо формализуемых задач, а также задач с неполной информацией и нечёткими исходными данными.

7. Использование информационных технологий может оказать существенное содействие в решении глобальных проблем человечества и, прежде всего, проблем, связанных с необходимостью преодоления переживаемого мировым сообществом глобального кризиса цивилизации.

Именно методы информационного моделирования глобальных процессов, особенно в сочетании с методами космического информационного мониторинга, могут обеспечить уже сегодня возможность прогнозирования многих кризисных ситуаций в регионах повышенной социальной и политической напряженности, а также в районах экологического бедствия, в местах природных катастроф и крупных техногенных аварий, представляющих повышенную опасность для общества.

1.3. Информационные системы.

Классификация и состав информационных систем

Для предоставления услуг в сфере информационного сервиса любого уровня необходимо на первом этапе обладать и уметь управлять необходимой информацией. Данная информация должна содержаться в информационных системах.

Информационная система. Под информационной системой организации следует понимать совокупность организационных, программных и технических средств, использующих автоматизированные информационные технологии для поддержки информационных методов управления, предоставления управленческому персоналу методов и средств работы с информацией для реализации функций управления.

Информационные системы могут значительно различаться по типам объектов, характером и объемом решаемых задач и рядом других признаков.

Общепринятой классификации информационных систем по настоящее время не существует, поэтому их можно классифицировать по разным признаками, что приводит к существованию нескольких различных классификаций ИС.

Классификация информационных систем. Информационные системы (ИС) подразделяются на следующие категории:

➤ по масштабам применения:

❖ *изолированные ИС* – информационная система установлена на компьютере, либо не включенном в локальную или иную сеть компании, либо ИС в своей работе не взаимодействует с ресурсами сети;

❖ *офисные ИС* – информационная система установлена в локальной сети офиса, количество компьютеров от 2 до нескольких десятков; ИС использует выделенный сервер (при количестве компьютеров больше 5-10), технологии файл-сервер (при количестве компьютеров-рабочих станций до 20) или «клиент-сервер» (при большем количестве рабочих станций). Данные ИС для связи компьютеров используют технологию локальной сети;

❖ *ИС предприятий (или корпоративные)* – информационная система установлена на предприятии, как правило, размещена в нескольких зданиях, в том числе – расположенных в разных городах, включают в себя локальные ИС. Данные ИС для объединения компьютеров используют собственные выделенные линии связи либо ресурсами Интернет;

❖ *глобальные ИС* – включают в себя корпоративные ИС, размещены нескольких странах (континентах), для объединения компьютеров используют ресурсы Интернет.

➤ по функциональному признаку – *производственные, маркетинговые*

- (анализа рынка, рекламные, снабженческие и т.п.), финансовые (бухгалтерские, статистические, и т.п.), кадровые;
- по квалификации персонала и уровням управления – *стратегические (топ-менеджеров), функциональные (менеджеров среднего звена) и оперативные (специалистов);*
 - по характеру обработки информации: *системы обработки данных, системы управления, система поддержки принятия решений;*
 - по оперативности обработки данных – *пакетной обработки и оперативные;*
 - по степени автоматизации - *ручные, автоматические, автоматизированные;*
 - по характеру использования информации - *информационно-поисковые, информационно-справочные, информационно-решающие, управляющие, советующие и т.п.;*
 - по степени централизации обработки информации — *на централизованные, децентрализованные, информационные системы коллективного использования;*
 - по характеру использования вычислительных ресурсов – *на локальные и распределенные;*
 - по сфере деятельности – *государственные, территориальные (региональные), отраслевые, объединений, предприятий или учреждений технологических процессов;*
 - по классу реализуемых технологических операций - *на системы с текстовыми редакторами, системы с табличными редакторами, системы управления базами данных (СУБД), системы управления базами знаний (СУБЗ), системы с графикой, мультимедиа, гипертекстом;*
 - по месту в процессе управления предприятия – *на автоматизированные рабочие места (АРМ) специалиста, ИС руководителя, ИС внешнего контролера, интегрированные системы,*

объединяющие в себе часть или все из этих функций;

- по концепции построения – *файловые, автоматизированные банки данных, банки знаний.*

Общие представления об ИС, их компонентах и свойствах

ИС называется объект информатизации, который с одной стороны рассматривается как единое целое, а с другой - как множество взаимосвязанных и взаимодействующих между собой составных частей.

Понятие ИС охватывает комплекс взаимосвязанных элементов, действующих как единое целое. В систему входят следующие компоненты:

- структура - множество элементов ИС и взаимосвязей между ними;
- входы и выходы - потоки сообщений, поступающие в ИС и выводимые ею;
- закон поведения ИС - функция, связывающая изменения входа и выхода ИС;
- цель и ограничения на функционирование ИС.

Для ИС характерны такие свойства как сложность, делимость, целостность и структурированность.

Сложность определяется множеством входящих в ИС компонентов, изменяющих внутренние и внешние связи и отношения.

Делимость предполагает, что ИС состоит из подсистем, выделенных по определенному признаку и отвечающих конкретным целям и задачам.

Целостность означает, что функционирование множества элементов ИС подчинено единой цели.

Структурированность предполагает распределение элементов ИС по уровням иерархии.

Создание автоматизированных информационных систем (АИС) - сложное и трудоемкое дело, требующее значительной подготовки и организации. Эффективность функционирования разработанной АИС в значительной мере зависит от научно-обоснованных методов ее создания.

Создание и функционирование ИС основывается на следующих принципах:

- *Системность*. Позволяет четко определить цели создания АИС и общие свойства, присущие системе как единому целому; выявляет критерии декомпозиции системы и многообразные типы связей между ее элементами.
- *Модульность*. Предусматривает построение АИС в виде взаимосвязанных и взаимно дополняемых модулей; причем замена одного модуля другим не нарушает целостность системы.
- *Адаптируемость* (гибкость). Обеспечивает приспособление системы к новым условиям функционирования при сохранении ее работоспособности.
- *Непрерывность развития* (открытость). Предопределяет АИС как систему, способную к развитию и совершенствованию при использовании новейших технологий процесса обработки данных.
- *Стандартизация и унификация*. Для проектирования АИС следует использовать в разумной мере типовые решения.
- *«Новые задачи»*. Предусматривает решение новых задач, которые ранее не рассматривались.
- *Надежность*. Предполагает устойчивость работы системы в условиях сбоя отдельных ее элементов.
- *Совместимость*. Заключается в способности взаимодействия различных АИС, имеющих информационное, техническое и технологическое сопряжение.
- *Однократность ввода*. Предусматривает одноразовый ввод информации и многократное, многоцелевое ее использование.
- *«Дружелюбность»*. Система должна быть простой и доступной для установки, изучения и эксплуатации.
- *Эффективность* (окупаемость). АИС не должна разорять пользователя и окупаться как материально, так и морально.

- *Автоматизация.* Предполагает безбумажную технологию, состоящую в комплексном использовании технических средств на всех стадиях технологического процесса обработки информации.
- *Безопасность.* Это сохранность и целостность конфиденциальной информации в системе.

1.4. Геоинформационные технологии (ГИТ). Геоинформационные системы (ГИС)

Геоинформационные технологии (ГИТ) — это технологический комплекс, интегрирующий и объединяющий многие информационные технологии. Их специфика состоит в ориентации на обработку пространственных данных. Пространственные данные могут интегрироваться с другими видами данных, что определяет ГИТ как многоцелевое средство, применяемое не только в науках о Земле, но и в общественных науках, экономике, информатике, медицине, управлении и т. п.

Геоинформационные технологии (ГИТ) — разновидность информационных технологий, связанных со сбором, обработкой, хранением, представлением и передачей геоинформации и геоданных. Ресурсами ГИТ являются геоданные, геоинформация, геоинформационные системы. Геоданные (пространственно-распределённые данные, пространственные) вводятся в компьютеры с устройств ввода в результате технологически переработанных пространственно-распределённых сигналов. Под термином «геоданные» обычно понимают пространственно-временные данные, отражающие свойства объектов, процессов и явлений, происходящих на Земле. Однако, действия могут проходить не только на Земле, а и в космосе и на других планетах, поэтому с нашей точки зрения, более удачный термин пространственно-распределённые данные. Геоинформация является результатом дальнейшего преобразования геоданных. Это соответствует известной модели DICW. На этом этапе преобразования уменьшаются

избыточные данные и погрешности. Тесно связана с геоинформацией и ГИТ геоинформационная система. *Геоинформационная система* (географическая информационная система, ГИС) — система сбора, хранения, обработки, анализа и графической визуализации пространственных данных (геоданных) и связанной с ними информации о необходимых объектах. ГИС могут быть статистическими и динамическими. В последнем случае геоданные должны также иметь временную составляющую.

Объектами обработки в геоинформационных технологиях и геоинформационных системах являются геоданные и геоинформация. Объектами исследования в геоинформационных технологиях и геоинформационных системах являются социально экономические объекты, пространственные объекты, природные объекты, геотехнические объекты. Между объектами могут быть установлены устойчивые и неустойчивые связи. Устойчивые связи позволяют создать структуру. Структура может быть различной: матричной, сетцентрической, иерархической, сетевой, бинарной, гомогенной и гетерогенной. Важным элементом технологии является цель. Цель преобразования диктует направление развития технологии, хотя технологии могут развиваться и в силу самосовершенствования сложных систем.

Первыми информационными технологиями были технологии хранения текстов, обработки текстов, информационно-поисковые системы, графические редакторы (векторные и растровые). Объединение этих технологий с системами автоматизированного проектирования привело к созданию геоинформационных технологий. Позднее в ГИТ стали входить и технологии обработки пространственной информации и изображений. Геоинформационная система (ГИС) и анализ пространственных данных сформировались как две отдельные области научных исследований, однако последние время они демонстрируют заметное сближение друг к другу, так что сегодня уже можно утверждать, что эти две области включаются в геоинформатику, поддерживая и дополняя друг друга. Развитие

геоинформатики — GIScience было предопределено разработками в области ГИС и ГИТ. Исследования в ГИТ продвинули техническую способность обрабатывать пространственно распределённые данные, стимулируя реализацию отражения отношения между тем, что называется «пространственной действительностью», и формированием концептуального представления этой действительности в конечных цифровых формах, то есть в виде исчисляемого количества точек, линий, и областей в двумерном пространстве.

Возможность принятия руководством предприятия, района, города, региона единственно верного решения и эффективность интеллектуального труда работников повышается наибольшими темпами в том случае, когда удаётся собрать воедино и быстро проанализировать большие объёмы разнообразной информации, не увеличивая в той же пропорции инвестиции и численность персонала.

Для эффективного управления имеющимися ресурсами, планирования развития и оперативного управления всеми сферами жизни необходима автоматизированная система сбора, хранения и анализа информации, пригодная для выработки верных управленческих решений. Такую роль выполняют *географические информационные системы (геоинформационные системы)*, интегрирующие разнородную информацию в единый информационно-аналитический комплекс на основе географических и пространственных данных.

Геоинформационная система (ГИС) – комплекс средств создания и обработки различных видов данных, включая пространственно- временные, и представления их в виде системы электронных карт.

Географические данные – это данные, которые описывают любую часть поверхности земли или объекты, находящиеся на этой поверхности. Они показывают объекты с точки размещения их на поверхности Земли, т.е. представляют собой «географически привязанную» карту местности.

Пространственные данные – это данные о местоположении, расположении объектов или распространении явлений. Они представляются в определённой системе координат, в словесном или числовом описании.

В основе любой геоинформационной системы лежит информация о каком-либо участке земной поверхности: стране, городе или континенте. База данных организуется в виде набора слоев информации. Основной слой содержит географические данные (топографическую основу). На него накладывается другой слой, несущий информацию об объектах, находящихся на данной территории: коммуникации, промышленные объекты, коммунальное хозяйство, землепользование и др. Следующие слои детализируют и конкретизируют данные о перечисленных объектах, пока не будет дана полная информация о каждом объекте или явлении. В процессе создания и наложения слоев друг на друга между ними устанавливаются необходимые связи, что позволяет выполнять пространственные операции с объектами посредством моделирования и интеллектуальной обработки данных.

Особенности геоинформационных систем:

1. Основой интеграции данных в ГИС является географическая информация, однако большинство задач, решаемых в геоинформационных системах, далеки от географических систем.

2. Основой интеграции технологий в ГИС являются технологии автоматизированного проектирования, но решаемые задачи далеки от проектных.

3. По определению ГИС относится к системам хранения информации, но по своему функциональному назначению геоинформационные системы можно отнести к классу систем обработки данных и управления.

Для работы геоинформационных систем требуются мощные аппаратные средства:

- запоминающие устройства большой ёмкости;
- системы отображения;

- оборудование высокоскоростных сетей.

Программное ядро географической информационной системы состоит из ряда компонентов. Они обеспечивают:

- ввод пространственных данных;
- хранение данных в многослойных базах данных;
- реализацию сложных запросов;
- пространственный анализ;
- просмотр введенной ранее и структурированной по правилам доступа информации;
- преобразование растровых изображений в векторную форму;
- моделирование процессов распространения загрязнения, геологических и других явлений;
- анализ рельефа местности и др.

Выделяют основные возможности, которые предоставлены геоинформационными системами:



1. *Формирование пространственных запросов и анализ данных.* ГИС помогает сократить время получения ответов на запросы клиентов; выявлять территории, подходящие для требуемых мероприятий; выявлять взаимосвязи между различными параметрами (например, почвами, климатом и урожайностью сельскохозяйственных культур); выявлять места разрывов электросетей.

Пример. Экологи используют ГИС для поиска воздушной загрязненной территории и дальнейшего вычисления ущерба предприятия данной. Запрос может быть уточнён введением дополнительных параметров, например стоимостных. Можно получить список всех территорий, находящихся на определённом расстоянии от определённой магистрали, лесопаркового массива или места работы.

2. *Улучшение интеграции внутри организации.* Одно из основных преимуществ ГИС заключается в новых возможностях улучшения управления организацией и её ресурсами на основе географического объединения имеющихся данных и возможности их совместного использования и согласованной модификации разными подразделениями. Возможность совместного использования и постоянно наращиваемая и исправляемая разными структурными подразделениями база данных позволяет повысить эффективность работы, как каждого подразделения, так и организации в целом.

Пример. Компания, занимающаяся экологическими работами (расчётами, планированием и др.), может чётко спланировать работы, начиная с получения полной информации и отображения на экране компьютера (или на бумажных копиях) соответствующих участков, на которых эти работы повлияют, и уведомлением их о сроках предполагаемого окончания данных работ.

3. *Помощь в принятии обоснованных решений.* ГИС – это не инструмент для выдачи решений, а средство, помогающее ускорить и повысить эффективность процедуры принятия решений, обеспечивающее ответы на запросы и функции анализа пространственных данных, представления результатов анализа в наглядном и удобном для восприятия виде.

Пример. ГИС помогает в решении таких задач, как предоставление разнообразной информации по запросам органов планирования, разрешение территориальных загрязнений, выбор оптимальных (с разных точек зрения и по разным критериям) мест для размещения объектов и т.д. Требуемая для

принятия решений информация может быть представлена в лаконичной картографической форме с дополнительными текстовыми пояснениями, графиками и диаграммами. Наличие доступной для восприятия и обобщения информации позволяет ответственным работникам сосредоточить свои усилия на поиске решения, не тратя значительного времени на сбор и осмысливание доступных разнородных данных. Можно достаточно быстро рассмотреть несколько вариантов решения и выбрать наиболее эффективный вариант.

4. *Создание карт.* Процесс создания карт в ГИС намного более прост и гибок, чем в традиционных методах ручного или автоматического картографирования. Он начинается с создания базы данных. В качестве источника получения исходных данных можно пользоваться и оцифровкой обычных бумажных карт. Основанные на ГИС картографические базы данных могут быть непрерывными (без деления на отдельные листы и регионы) и не связанными с конкретным масштабом.

Пример. Можно создавать карты (в электронном виде или как твердые копии) на любую территорию, любого масштаба, с нужной нагрузкой, с её выделением и отображением требуемыми символами. В любое время база данных может пополняться новыми данными (например, из других баз данных), а имеющиеся в ней данные можно корректировать по мере необходимости. В крупных организациях созданная топографическая база данных может использоваться в качестве основы другими отделами и подразделениями, при этом возможно быстрое копирование данных и их пересылка по локальным и глобальным сетям.

Области применения ГИС

Задачи управления муниципальным хозяйством - одна из крупнейших областей приложений ГИС. В любой сфере деятельности местной администрации (обследование земель, управление землепользованием, замена существующих бумажных записей, управление ресурсами, учёт состояния собственности (недвижимости) и дорожных магистралей)

применимы ГИС. Они могут использоваться также на командных пунктах управления центров по мониторингу и в службах быстрого реагирования. ГИС - неотъемлемый компонент (инструментальный, технологический, программный) любой муниципальной или региональной информационной системы управления.

Организации, обеспечивающие коммунальные услуги, наиболее активно используют ГИС для построения базы данных об основных средствах (трубопроводы, кабели, насосы, распределительные станции и т.п.), которая является центральной частью в их стратегии информационной технологии. Обычно в этом секторе доминируют ГИС, обеспечивающие моделирование поведения сетей в ответ на различные отклонения от нормы. Наибольшее применение находят системы автоматизации картографирования и управления основными средствами для поддержки "внешнего планирования" в организации: прокладка кабелей, расположение задвижек, щитов обслуживания и др. (рис. 2).

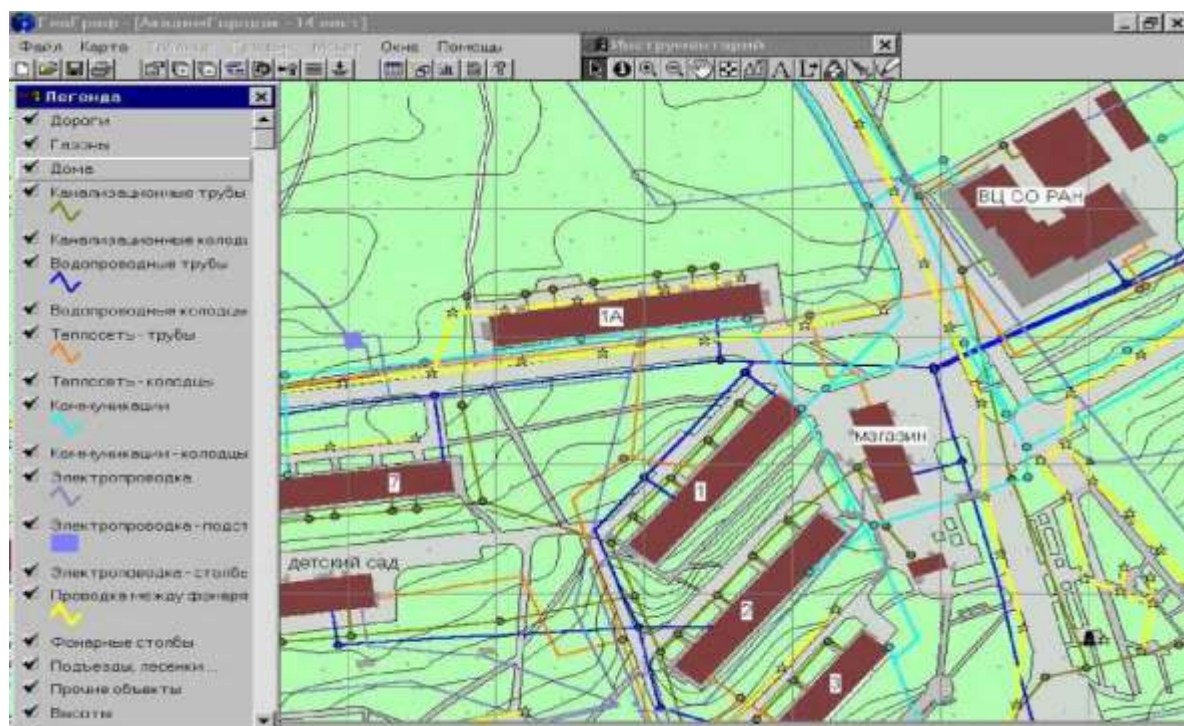


Рис.1.2. Фрагмент плана инженерных коммуникаций в Академгородке - составная часть проекта «ГИС Академгородок»

Охрана окружающей среды

Наиболее ранними пользователями ГИС были организации, заинтересованные в охране окружающей среды. На простейшем уровне - для исследования состояния окружающей среды (например, расположение и состояние лесов, рек). Более сложные приложения используют аналитические возможности ГИС для моделирования процессов в окружающей среде, таких как эрозия почв или разлив рек в случае большого количества осадков, распространение выбросов загрязняющих веществ промышленных предприятий в атмосфере (рис. 3). После сбора исходных картографических данных производится их аналитическая обработка в ГИС.

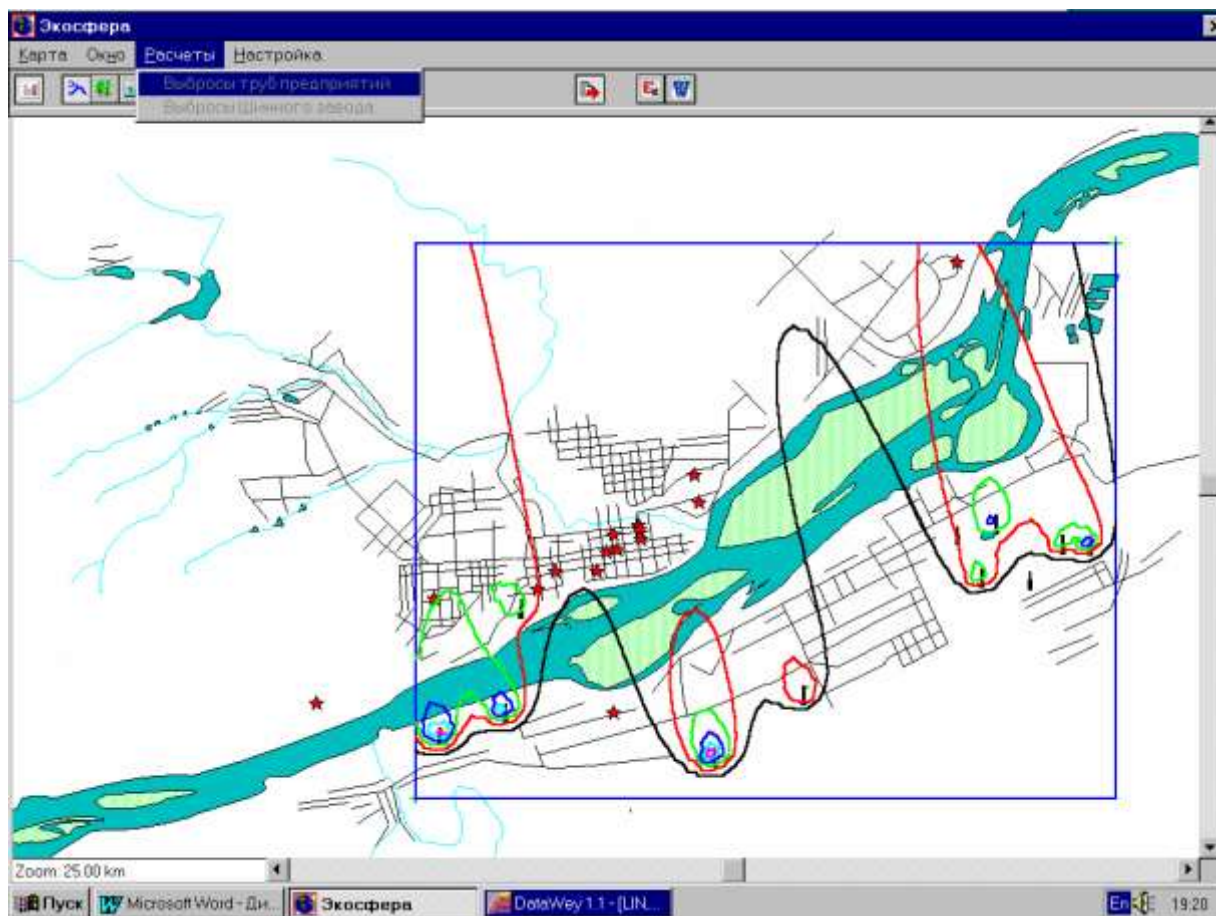


Рис. 1.3. Моделирование распространения выбросов загрязняющих веществ промышленных предприятий в атмосфере.

Прикладная программа, разработанная на Borland Delphi, использует электронные таблицы Microsoft Excel для хранения информации по источникам выбросов и ГИС MapInfo для визуализации результатов. Проект

студентов СибГТУ, выполненный в Межвузовском центре информационных технологий в экологическом образовании

Здравоохранение

В дополнение к обычным задачам управления основными средствами аналитические возможности ГИС используют в приложениях охраны здоровья, например, для определения кратчайшего пути от станции скорой помощи до пациента с учетом текущей ситуации на дорогах, а также при анализе эпидемиологических ситуаций: характера распространения различных заболеваний и причин их возникновения (рис. 1.4.).



Рис. 1.4. Карта распространения раковых заболеваний - по материалам комплексных исследований ряда научных и медицинских учреждений города. Проект студентов КГТУ, выполненный в Межвузовском центре информационных технологий в экологическом образовании

Транспорт

ГИС имеют огромный потенциал для приложений на транспорте. Планирование и поддержка транспортной инфраструктуры - это очевидная область применения. В настоящее время увеличивается интерес к использованию новых технологий, например навигационных, для контроля

за движением большегрузных автомобилей. Отображение их места нахождения на цифровой карте на дисплеях в кабине водителя и в центре управления перевозками требует поддержки со стороны ГИС.

Розничная торговля

Крупные западные коммерческие фирмы используют ГИС для выбора места расположения большинства новых супермаркетов за пределами центра города, для хранения социально-экономических деталей обстановки и потенциальных заказчиков в заданной области. Расположение склада и зона обслуживания могут быть разработаны с помощью вычислений времени доставки и моделирования влияния конкурирующих складов. ГИС используют также и для управления поставками.

Финансовые услуги

В секторе финансовыми услуг ГИС используются так же, как и в приложениях для розничной торговли: для определения расположения филиалов банков и зданий обществ; в качестве инструмента для оценки риска вложений средств в недвижимость и страхования, для определения областей высшего/низшего риска. Это требует баз данных о криминальной обстановке, ресурсах территории, характеристиках недвижимости.

Внедрение ЭВМ, современных средств переработки и передачи информации в различные сферы деятельности послужило началом нового этапа развития человеческого общества, называемого *информатизацией*. *Информатизация общества* – организованный социально экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей граждан, органов государственной власти, местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

На первый план выходит новая отрасль – *информационная индустрия*, связанная с производством технических средств, методов, технологий для производства новых знаний. Важнейшими составляющими информационной

индустрии становятся все виды информационных технологий, особенно телекоммуникации. Современная информационная технология опирается на достижения в области компьютерной техники и средств связи.

Усложнение индустриального производства, социальной, экономической и политической жизни привело, с одной стороны, к росту потребности в знаниях, а с другой – к созданию новых средств и способов удовлетворения этой потребности. Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества.

Информатизация общества – совокупность взаимосвязанных политических, социально-экономических, научных факторов, которые обеспечивают каждому члену общества свободный доступ к любым источникам информации, кроме законодательно засекреченных факторов. Фактически сейчас человечество постепенно переходит из постиндустриальной стадии развития в информационную стадию, которая характеризуется высокой (более 70%) занятостью трудоспособного населения в сфере обработки, передачи и хранения информации. Лидирующие позиции в этом процессе занимает Япония, принявшая долгосрочную программу перехода общества в информационную стадию развития.

Цель проекта – связать те услуги, которые раньше предлагались по отдельности. Для этого все виды информации – от телефонных посланий и телепрограмм до собственно компьютерной продукции – должны передаваться по одному общему кабелю. В перспективе каждый абонент кабельной сети сможет получить несколько услуг одновременно. Большое внимание в проекте уделяется созданию терминалов для неопытных пользователей – с интеллектуальным интерфейсом, где ввод информации

осуществляется голосом. Кроме того, принята программа разработки новых типов компьютеров, основанных:

- на принципе высокоскоростной параллельной обработки информации, когда одновременно десятки и сотни процессоров выполняют сложные операции;
- на нейронных сетях, работа в которых аналогична функционированию мозга;
- на принципе фотонной передачи информации.
- Другие страны имеют свои, национальные программы информатизации с учётом местных особенностей и условий. Однако в каждой из них есть общие черты:
- отказ от стремления в первую очередь обеспечить экономический рост страны;
- понимание необходимости замены экономической структуры, основанной на тяжёлой промышленности, структурой, базирующейся на наукоёмких отраслях;
- признание приоритетного характера информационного сектора, при котором основой успешного экономического развития становится создание новой инфраструктуры и широкое использование достижений мировой науки и техники;
- вложение значительных финансовых средств в информатизацию;
- провозглашение главной целью информатизации рост благосостояния страны и её граждан за счёт облегчения условий коммуникации и обработки информации.

Результатом процесса информатизации является создание информационного общества, где манипулируют не материальными объектами, а символами, идеями, образами, интеллектом, знаниями. Традиционные источники материальных ресурсов отходят на второй план, на первый выходит новый ресурс – информация.

Ключевые слова

Информация, данные, документированная информация, экономическая информация, синтаксический анализ, семантический анализ, информационные процессы, информационная система, информационные ресурсы, конфиденциальная информация, информатизация общества, глобализация информационной сферы, информационные технологии(ИТ), информационный сервис, качество информации, геоинформационные технологии(ГИТ), геоинформационная система (ГИС), этапы развития ИТ, автоматизированные информационные системы.

Вопросы для самопроверки:

1. Расскажите об информационных революциях в истории развития цивилизации.
2. Сопоставьте процессы, происходящие в истории развития ИТ, с последней информационной революцией.
3. Определите суть информационных технологий и телекоммуникаций.
4. Как вы себе представляете информационное общество?
5. В чем проявляется информационный кризис?
6. В чем состоит процесс информатизации?
7. В чем отличие процессов компьютеризации и информатизации?
8. Дайте определение информационной культуре. Как она проявляется?
9. Чем определяется информационный потенциал общества?
10. Расскажите о видах ресурсов.
11. Охарактеризуйте информационный ресурс, информационный продукт, информационную услугу. Приведите примеры.

Лекция 2. ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

2.1. Основные технические средства информационных технологий

Компьютером называется электронная машина, способная выполнять арифметические и логические действия. Раньше в нашей стране для наименования таких машин применялся термин ЭВМ (электронные вычислительные машины). Сейчас повсеместно применяется термин *компьютер*, пришедший к нам из английского языка (слово «computer» с английского переводится как «вычислитель»).

Компьютеры повсюду вокруг нас. Ваш мобильный телефон – это маленький компьютер, ваша стиральная машина (если она современная) управляется компьютером, ваш DVD-проигрыватель – это тоже компьютер. Просто компьютеры выглядят по-разному и имеют различное назначение. Они могут быть очень маленькими, а могут быть величиной с платяной шкаф. Фантастические фильмы рисуют компьютер как непостижимую вещь, которая мгновенно производит большой объем вычислений, может дать ответ на любой вопрос и иногда сама собой оживает. Прежде всего следует понять, что компьютер – это просто инструмент в руках человека. Достаточно сложный, довольно мощный, но всего лишь инструмент. Как отвертка, гаечный ключ или дрель. Просто задачи, которые с его помощью решает человек, требуют быстрой обработки большого числа однотипных данных. Все, что делает компьютер, заранее кем-то продумано и определено. Он ничего не может сделать сам, по «своему желанию».

Мы будем рассматривать *персональные компьютеры*. Они разрабатывались еще в 70-х годах прошлого столетия, однако широкое распространение стали получать с начала 80-х годов, когда американская компания IBM выпустила в 1981 году модель IBM PC. Модели IBM-совместимых компьютеров стали наиболее массовыми. Секрет их популярности в том, что компоненты для таких компьютеров выпускает большое количество различных производителей и в результате конкурентной

борьбы и постоянной разработки новых технологий цены на устройства уменьшаются. Хотя бытует такое мнение, что персональный компьютер, полностью состоящий из современных компонентов, во все времена стоит 1000 долларов США.

Параллельно по сей день разрабатываются и продаются персональные компьютеры фирмы Apple. Они отличаются оригинальным дизайном. Однако распространены они не так широко, как IBM-совместимые. Компьютеры фирмы Apple в основном используют в издательствах для верстки и других работ по подготовке печатной продукции. Современные модели компьютеров от фирмы Apple имеют название Macintosh.

Еще одно уточнение: мы будем рассматривать стационарный настольный вариант персонального компьютера (англоязычный термин – *desktop*). Дело в том, что есть еще *карманные персональные компьютеры* (КПК) и переносные персональные компьютеры — *ноутбуки* (английский вариант – *notebook*). А с недавних пор появились маленькие ноутбуки – субноутбуки и нетбуки, которые при малом энергопотреблении позволяют путешествовать по Интернету, питаясь автономно от батарей. Их мощности даже хватает для просмотра фильмов.

Современный компьютер представляет собой сложную систему, состоящую из многих компонентов, заключенных в блоки по функциональному назначению. Назначение всей этой системы – обработка информации, которая может включать поиск данных, их сортировку, редактирование, выполнение вычислений, печать, передачу по каналам связи и т. д.

В конструкции персонального компьютера (рис. 2.1) существуют блоки, без которых работа с ним затруднительна. Они называются основными. Это:

- системный блок, включающий в себя много необходимых устройств, сделанных в виде печатных плат и монтируемых на большой печатной плате, называемой «материнской» или системной платой;

- монитор, позволяющий наглядно отображать процесс обработки данных;
- клавиатура, с помощью которой вводятся команды и данные;
- манипулятор «мышь» для управления в системах с графическим интерфейсом (о том, что такое «интерфейс», поговорим позже).



Рис. 2.1. Основные блоки персонального компьютера

Устройства обработки, анализа и хранения информации

Рассмотрим каждый из основных блоков технических средств подробнее.

Системный блок

Собственно в системном блоке (рис. 2.2) и содержится тот самый «вычислитель» – *центральный процессор* (англоязычная аббревиатура – CPU, Central Processing Unit). Центральный процессор координирует движение потоков информации и управляет компьютером. Управление осуществляется с помощью инструкций, шаги которых называют командами. Центральный процессор понимает определенную систему команд, то есть кодов, которые предписывают ему выполнение каких-либо элементарных операций. Последовательности таких кодов называются программами для компьютера.



Рис. 2.2. Системный блок персонального компьютера

То есть управляет всеми делами в вашем персональном компьютере именно центральный процессор, но при этом он делает не то, что ему «вздумается» (потому что он просто железяка), а четко выполняет распоряжения работающих в данный момент программ.

Процессор

Классическим производителем процессоров для персональных компьютеров является корпорация Intel. Например, именно она выпускает такие модели процессоров, как Pentium IV, Pentium D, Atom, Core Duo, Core Quad, Core i5, Core i7 (рис. 2.3). С некоторых пор ей составляет довольно сильную конкуренцию компания AMD. Она выпускает, например, линейки процессоров Athlon, Sempron, Phenom.



Рис. 2.3. Процессор

Важной характеристикой процессоров, которая приводится в любом рекламном проспекте, является так называемая *тактовая частота*, измеряемая в мегагерцах (МГц) или гигагерцах (ГГц). Этот параметр показывает, насколько быстро работает процессор компьютера.

Последние модели процессоров для персональных компьютеров (например, Core Duo) являются двухъядерными, а некоторые (Core Quad) – четырехъядерные. Это означает, что в одном центральном процессоре содержатся как бы два (или четыре) подпроцессора. Поэтому такие процессоры при той же частоте, что и одноядерные (Pentium IV), дают большую производительность. Процессор расположен на *материнской плате* (mother board), которая обеспечивает его связь с другими устройствами. Если открыть системный блок и заглянуть в него, мы увидим, что самое большое место по площади занимает как раз материнская плата. В обиходе ее часто называют «мамка» или «мать». Поэтому если вы услышите в разговоре двух подростков фразу вроде «я совсем замучился с мамкой», это еще не значит, что у парня проблемы в семье.

Оперативная память. Выполняемые в данный момент программы и информацию, с которой они работают, надо где-то хранить. Для этого придумали специальное устройство — *оперативная память* (англоязычная аббревиатура – RAM, Random Access Memory).

Оперативная память компьютера (ОЗУ) – это устройство хранения информации, обрабатываемой процессором. Оперативной она называется потому, что обеспечивает очень быстрый доступ к данным, однако хранит данные только тогда, когда компьютер включен.



Рис. 2.4. Оперативная память компьютера

Это связано с тем, что такая память создана на микросхемах (рис. 2.4), требующих для своей работы электропитание. На материнской плате есть специальные разъемы, куда вставляются планки с микросхемами оперативной памяти.

Чтобы как-то измерять объемы памяти и данных, записанных в нее, придумали единицу измерения – байт (есть и более мелкие единицы – биты; 1 байт = 8 бит). Чтобы сориентироваться, один байт оперативной памяти может хранить одну букву электронной книги, которую вы можете прочитать на своем персональном компьютере. Сейчас объемы оперативной памяти довольно большие. Чтобы не измерять ее такими мелкими единицами, как байты, решили использовать и более крупные единицы:

1 килобайт (Кбайт) = 1024 байт;

1 мегабайт (Мбайт) = 1024 килобайт;

1 гигабайт (Гбайт) = 1024 мегабайтам;

1 терабайт (Тбайт) = 1024 гигабайтам.

Жесткий диск

Мы обсудили, что при выключении электропитания компьютера оперативная память полностью очищается. Где же хранятся программы и огромный объем полезной информации, когда компьютер выключен?



Рис. 2.5. Жесткий диск компьютера

Для этой цели в персональных компьютерах используются *жесткие диски* (рис. 2.5) (англоязычная аббревиатура – HDD, Hard Drive Disk). Их еще называют *винчестеры*. Они также расположены внутри системного блока и соединены с материнской платой наборами проводов — *шлейфами*. Разъемы для подключения жестких дисков на материнской плате могут быть различными. Еще недавно использовался стандарт разъемов IDE (Integrated Drive Electronics. – встроенный интерфейс накопителей), а сейчас уже большинство жестких дисков подключается к разъему SATA (Serial Advanced Technology Attachment – последовательный интерфейс подключения устройств к материнской плате). По этой причине, если вы вдруг решите заменить свой жесткий диск диском большего объема, нужно покупать его, зная возможности вашей материнской платы. А лучше просто отвезти системный блок в компьютерный магазин и попросить заменить винчестер купленным у них же. Объем жестких дисков, как и оперативной памяти, измеряют в байтах (килобайтах, мегабайтах, гигабайтах, терабайтах). Жесткие диски могут хранить на порядок больше информации, чем оперативная память. Оно и понятно: оперативная память хранит только то, с чем сейчас работает процессор, а на жестком диске лежит все, что вы туда записали. Это могут быть и программы (в том числе игры), и электронные книги, и музыка, и фильмы, и деловые документы. Кроме жестких дисков информацию можно хранить и на съемных носителях.

Съемные носители

Традиционными съемными носителями информации являются *гибкие магнитные диски* (ГМД) размером 3,5" (рис. 2.6). Устройство для чтения и записи таких носителей (Floppy Disk Drive, FDD) в современные системные блоки устанавливать практически прекратили. Использование дискет с появлением носителей типа *Flash-Drive* (флэш-память) (рис. 2.7) сошло на нет. Объем одной дискеты формата 3,5" всего 1,44 Мбайт! В то время как устройства Flash-Drive (называемые «флэшки») могут вмещать себя гигабайты информации, работать с ними просто и надежно, чем дискеты.

Вставляется дискета шторкой вперед, так чтобы центральный металлический диск был обращен в сторону кнопки дисководов.

Для изъятия дискеты из накопителя нужно нажать кнопку, расположенную на устройстве рядом со щелью для дискет. Кроме магнитных дисков и «флэшек», для длительного хранения информации применяются оптические диски. Можно хранить данные на компакт-дисках (CD-дисках). Максимальный объем данных в этом случае – 700 Мбайт (стандартный – 650 Мбайт). А можно и на DVD-дисках (Digital Versatile Disk – цифровой многофункциональный диск). Их емкость намного больше – 4,7 Гбайт. Для работы с CD-дисками и DVD-дисками нужны соответствующие устройства – *приводы CD-дисков или DVD-дисков* (рис. 2.8). И они сразу установлены далеко не во всех новых компьютерах.



Рис. 2.6. Дискета формата 3,5"



Рис. 2.7. Носитель типа Flash-Drive

Поэтому наличием необходимых вам приводов следует интересоваться еще при покупке компьютера. Хотя установить новый привод можно в любой момент, например, слегка поднакопив денег после покупки самого компьютера.



Рис. 2.8. Компакт-диски

Приводы могут иметь возможность только читать CD-диски (CD-ROM) или DVD-диски (DVD-ROM), а могут еще и иметь возможность записи (CD-RW, Compact Disk Rewritable, или DVD-RW). Второй тип устройств, конечно, дороже. Зато можно с их помощью переписывать диски с фильмами и музыкой у друзей. Кстати, DVD-приводы поддерживают и работу с CD-дисками. Есть в продаже и приводы для работы с дисками нового формата большой емкости Blu-ray, но они пока несоизмеримо дороги.

Чистые оптические диски, предназначенные для записи информации, могут иметь возможность только однократной записи (их часто называют «болванками») или поддерживать многократную перезапись данных. Обозначения для CD-дисков однократной записи – CDR (Compact Disk Recordable). Для многократной перезаписи – CD-RW.

Для DVD-дисков обозначение болванок однократной записи – DVD-R (или DVD +R). Для многократной перезаписи – DVD-RW (DVD+RW). В этих обозначениях плюсы и минусы означают несколько различных формат записи дисков. Изначально форматы с минусом в обозначении (DVD-R) разрабатывались для бытовой видеотехники (DVD-проигрывателей), а

форматы с плюсом в обозначении – для применения в компьютерной области. Однако современные приводы работают с дисками обоих форматов одинаково хорошо (чего не скажешь о старых моделях DVD-проигрывателей, которые отказываются читать диски с фильмами формата DVD+R, записанные на компьютере).

Для вставки диска в привод CD-дисков или DVD-дисков надо нажать кнопку на приводе. Из привода выдвинется лоток, куда надо положить диск изображением кверху, а блестящей стороной по направлению внутрь дисковод.



Рис. 2.9. Дисковод

Наравне с устройствами Flash-Drive используются и *внешние жесткие диски* (рис. 2.10). Они имеют большие габариты, нежели «флешки», но и объем хранимой на них информации значительно больше. Объемы современных съемных жестких дисков исчисляются сотнями гигабайт и достигают нескольких терабайт. Часто жесткие диски обозначают сокращенно HDD – от англ. Hard Drive Disk (жесткий диск). В разговорной речи вы можете услышать еще название «винчестер», «хард» или «винт». Внешние жесткие диски подключаются к компьютеру через уже нам знакомый разъем USB.



Рис. 2.10. Внешний жесткий диск

Как обращаться к дискам

Чтобы при работе на компьютере как-то обращаться к дискам с информацией, для всех дисков придумали использовать специальные имена. Для единообразия имена дисков обозначаются буквами латинского алфавита (a, b, c...) с двоеточием после них. Например: a:, b:, c: и т. д.

Имена a: и b: закреплены за гибкими магнитными дисками. При отсутствии соответствующего дисководов имя пропускается. Имена, обозначаемые буквами c:, d: и т. д., закреплены за *логическими дисками* винчестера.

Логические диски – это условные фрагменты одного физического диска (железный ящик жесткого диска установлен внутри системного блока), существующие в некотором смысле автономно. У вашего компьютера может быть один большой логический диск, совпадающий по размерам с физическим диском. А может быть несколько более мелких логических дисков. Количество и размер логических дисков определяет человек, который устанавливает на ваш новый компьютер операционную систему (об этом ниже). Если, компьютер для вас собирают на заказ, и перед его передачей вам будут устанавливать операционную систему, вы можете указать специалистам желаемое количество логических дисков и их размер.

Имена, следующие за именем последнего логического диска винчестера, присваиваются устройствам чтения или записи компакт-дисков или DVD-дисков. Далее идут имена дисков локальной сети, подключенных к данному компьютеру. Если вы подключили устройство Flash-Drive, буква

имени его диска будет следующей за именем последнего устройства чтения или записи компакт-дисков или DVD-дисков.

Графические адаптеры

С хранением данных мы более-менее разобрались. Теперь о том, как увидеть информацию на мониторе. Для перевода малопонятных нам и самому монитору сигналов центрального процессора, а также для снятия части забот по прорисовке изображения на экране монитора в специальный разъем на материнской плате вставлен *графический адаптер* (рис. 2.11). Все это естественно скрыто в корпусе системного блока.



Рис. 2.11. Графический адаптер

Как и в случае с процессорами, основных производителей графических адаптеров – два. Это компания ATI (серия адаптеров Radeon) и компания nVidia (серия адаптеров GeForce). Между ними разгорелась нешуточная конкурентная борьба, в результате чего мы можем приобретать дорогие адаптеры обоих производителей для суперсовременных супер реалистичных игр или покупать стремительно дешевеющие, но весьма хорошие адаптеры их же производства.

На основе микросхем этих ведущих производителей изготовлением графических адаптеров занимаются и другие фирмы. Однако в названиях их продуктов обычно фигурируют слова Radeon или GeForce, в зависимости от микросхем, на основании которых они сделаны.

Адаптеры различают по размеру памяти адаптера (он имеет свою встроенную оперативную память) и возможностям поддержки различных графических режимов. Если вы приобретаете компьютер для игр, важнее выбрать не мощный центральный процессор, а хороший (дорогой) графический адаптер и большой объем оперативной памяти.

Для офисных задач вполне достаточно процессора Pentium Celeron или Atom, графического адаптера за 50 долларов (или встроенного в материнскую плату) и 1 Гбайт оперативной памяти.

Звуковая карта

Чтобы вы могли наслаждаться музыкой в процессе работы на компьютере или реалистичными звуками во время игры, в системном блоке должна быть установлена специальная плата — *звуковая карта* (рис. 2.12). Конечно, кроме нее вам потребуются музыкальные колонки или наушники, которые подключаются к выходному гнезду звуковой карты.



Рис. 2.12. Звуковая карта

Звуковая карта может быть встроена сразу в ваш новый компьютер. Можно приобрести это устройство и отдельно, а затем установить его (с помощью специалиста).

Во многих современных компьютерах имеется встроенная звуковая карта. Это означает, что она не вставлена в виде печатной платы в материнскую плату, а сделана прямо в составе материнской платы.

Различаются звуковые платы по качеству воспроизводимого звука (количество аудиоканалов и отсутствие помех) и качеству оцифровки входящего звука. Обычная звуковая карта вполне справится с

воспроизведением музыки на уровне бытового музыкального центра и записи ваших комментариев с микрофона (который вы подключили к ней). Многие современные встроенные звуковые карты поддерживают семиканальный звук и сабвуфер.

Если ваш компьютер обладает звуковой картой и колонками, его с полным правом можно называть *мультимедийным*. Мультимедиа – это использование средств одновременного воздействия на разные органы чувств. Обычно под этим подразумевается видео– и аудиоинформация.

Почему одни системные блоки не похожи на другие.

По внешнему виду системные блоки могут сильно различаться оформлением и расцветкой. Но глобально их разделяют в зависимости от так называемого *форм-фактора*. Системные блоки персональных компьютеров могут располагаться горизонтально – формфактор «десктоп» (рис. 2.13) и вертикально – форм-факторы «тауэр» (рис. 2.14). В зависимости от высоты корпуса «тауэры» подразделяются на «бигтауэры», «мини-тауэры» и «мидитауэры».



Рис. 2.13. Системный блок «десктоп»



Рис. 2.14. Системный блок «тауэр»

Выбирая системный блок, исходите из того, где вы будете его располагать (на столе под монитором, на столе рядом с монитором, под столом). Важными характеристиками системного блока (кроме всего, что мы обсудили) являются качество самого корпуса и блока питания. Как правило, известные в мире производители персональных компьютеров (Lenovo, Hewlett Packard, DELL, Acer) используют корпуса системных блоков либо собственного изготовления, либо сторонних производителей, но уже зарекомендовавших себя. Таковыми являются в первую очередь компании ASCOT, INWIN, ASUS, THERMALTAKE.

Системный блок содержит много полезных вещей, о которых мы говорили выше, но сам по себе – просто железный ящик. Чтобы извлечь из него пользу, нужно подключить к нему устройства для ввода информации (клавиатура, мышь) и устройство для вывода информации (монитор).

На передней стенке корпуса системного блока располагают кнопку включения электропитания и кнопку перезагрузки компьютера. Кнопка включения электропитания помечается надписью **Power** (Энергия) или значком



Кнопка перезагрузки отмечается надписью **Reset**. Часто ее делают меньше по размерам, чем кнопку включения электропитания. Выключать электропитание компьютера кнопкой **Power** во время работы программ не рекомендуется. Современные операционные системы требуют, чтобы вы корректно завершили работу с ними. В этом случае они сами выключают электропитание системного блока (если ваш компьютер поддерживает программное управление электропитанием).

Монитор

Монитор предназначен для отображения информации в ходе обработки данных. С его помощью программы запрашивают у вас условия выполнения

задач и сообщают результаты своей работы. Современные мониторы позволяют программам отображать для вас красочные изображения с весьма высоким качеством. Требования по безопасности, предъявляемые к мониторам, гораздо более строгие, чем требования к телевизорам. Важными характеристиками монитора являются его *разрешение, размер экрана и палитра*. Разрешением монитора называют способность выводить определенное количество точек по вертикали и горизонтали экрана. Чем больше точек можно вывести, тем изображение будет ближе к естественному. Обычно разрешение монитора обозначают так: *количество точек по горизонтали x количество точек по вертикали*. Размер экрана, измеряемый в дюймах по его диагонали, может быть, например, 15", 17", 19", 22", 24" и т. д. Увеличение размера экрана позволяет при том же размере точки вывести на экран изображение большего размера.

Палитра – это количество оттенков цветов, воспроизводимых монитором. Оно прямо пропорционально объему видеопамяти и обратно пропорционально выбранному разрешению. Палитра обычно содержит 64 000 или 16 млн. оттенков. Мониторы, которые сейчас можно приобрести, бывают двух типов: ЭЛТ-мониторы (на базе электронно-лучевой трубки) (рис. 2.15) и жидкокристаллические (ЖК) мониторы (рис. 2.16). Мониторы на жидких кристаллах (LCD-мониторы, Liquid Crystal Display) фактически отвоевали рынок домашних и офисных компьютеров. И это вполне объяснимо: такие мониторы занимают меньше места, чем их собратья на электронно-лучевых трубках, не испускают рентгеновское излучение (в них стоят обычные лампы дневного света), и их экран абсолютно плоский. Появились также ЖК-мониторы с подсветкой не на лампах дневного света, а со светодиодной подсветкой. Картинка на таких мониторах отличается в лучшую сторону, но и стоят они несколько дороже.



Рис. 2.15. ЭЛТ-монитор



Рис. 2.16. Жидкокристаллический монитор

Кроме указанных основных характеристик, жидкокристаллические мониторы могут быть с классическим соотношением сторон 4:3 (см. рис. 2.16) или широкоэкранными с соотношением 16:9 (рис. 2.17).



Рис. 2.17. Широкоэкранный монитор

Иногда для получения стереозвука колонки встраивают прямо в монитор. Такие мониторы отличаются немного большими размерами, а сами встроенные колонки выдают себя множеством маленьких дырочек в том месте монитора, где установлены динамики.

При выборе жидкокристаллических мониторов следует обращать внимание на следующие характеристики: тип матрицы, время отклика матрицы, размер точки экрана, контраст, яркость, максимальные углы обзора экрана, порты подключения монитора к компьютеру. Отличным считается время отклика матрицы менее 5 мс (миллисекунд). Большая скорость принципиальна, если вы будете играть в динамичные игры. Для набора текстов и работы с изображениями этот параметр не принципиален. Большие углы обзора нужны, если вы решили использовать монитор в качестве телевизора.

Клавиатура

Клавиатура является устройством управления компьютером, предназначенным для ввода данных и команд. Клавиатура имеет 5 групп полей, показанных на рис. 2.18.



Рис. 2.18. Клавиатура

В поле *функциональных клавиш* находятся: Esc (произносится по-английски «эскейп») – как правило, используется для выхода из какой-либо ситуации или отмены какого-либо действия;

F1—F12 – функциональные, то есть клавиши, выполняющие определенные, закрепленные за ними действия, которые зависят от работающей в данный момент программы;

Print Screen («принт скрин») – клавиша печати экрана на принтере или копирования образа экрана в Буфер обмена (что это такое, мы узнаем при рассмотрении операционных систем);

Scroll Lock («скрол лок») – используется в некоторых программах, но в большинстве не задействована;


Pause/Break – приостановка работы компьютера (работает в период загрузки компьютера и в режиме ввода команд).

В поле индикации находятся индикаторы режимов: NumLock – если горит, режим NumLock включен и на дополнительной клавиатуре можно вводить цифры;

CapsLock – если горит, режим CapsLock включен и буквы, набираемые на символьной клавиатуре, будут заглавными;

ScrollLock – если горит, режим ScrollLock включен.

В поле *символьных клавиш* находятся клавиши для ввода символов. Однако некоторые клавиши в этой области особенные:

Tab –  табуляция символов, то есть перемещение курсора к определенной позиции строки;

CapsLock– включение/выключение режима прописных букв;

Shift («шифт») – клавиша временного переключения на режим прописных букв или для ввода другого символа, расположенного на той же клавише (существует две таких клавиши – левая и правая);

Ctrl – вспомогательная клавиша для расширения возможностей клавиатуры;

Alt – то же, что и предыдущая;

Enter (↵) – клавиша исполнения команды или завершения ввода;

Backspace (← – в правом верхнем углу символьной клавиатуры) – клавиша удаления символа слева от курсора.

Курсором называется специальная подвижная отметка на экране монитора, которая показывает место появления вводимого символа. Курсор может выглядеть как горизонтальная или вертикальная черта. Курсор часто изображается мигающим.

В поле клавиш управления курсором находятся:

Insert или Ins – клавиша «вставки» (включает или выключает режим вставки);

Home – перемещает курсор в начало строки; PageUp или PgUp – перемещает курсор на экран вверх; Delete или Del – удаляет символ в позиции курсора; End («энд») – перемещает курсор в конец строки;

PageDown или PgDn – перемещает курсор на экран вниз.

Ниже клавиш управления курсором расположены клавиши с изображениями стрелок, перемещающие курсор на одну строку или позицию в направлении стрелки.

В поле *дублирующих клавиш* при включении режима NumLock можно вводить цифровые данные, а в противном случае – управлять курсором.

Как вы уже, наверное, заметили, на некоторых клавишах нарисовано по два или три символа. С буквами просто: при переключении языка ввода на английский язык вводятся латинские буквы, при переключении на национальную раскладку клавиатуры (русский язык) вводятся символы национального языка (русские буквы).

Если на клавише три значения, одно из них вводится при простом нажатии клавиши,

второе – при одновременном нажатии клавиши и удержании клавиши Shift (можно предварительно нажать клавишу Shift, держать ее, при этом нажимая нужную клавишу), третье значение – при переключении раскладки клавиатуры на другой язык и нажатии клавиши Shift.

Способы переключения языка зависят от используемой операционной системы.

В современных клавиатурах часто имеются дополнительные клавиши Sleep и Wake. Первая отправляет компьютер в спящий режим, когда экран гаснет и компьютер переходит в состояние пониженного энергопотребления. Соответственно вторая будит его. Обычно после нажатия на нее проходит какое-то время (несколько секунд), пока компьютер «потягивается» и «протирает глаза».

Мышь

Первые десять лет после появления в широкой продаже персонального компьютера манипулятор типа «мышь» считался дополнительным устройством. Только с появлением и широким распространением графического интерфейса операционных систем наличие мыши стало просто необходимо для полноценной работы, и ее перевели в разряд основных устройств.

Принцип работы мыши заключается в том, что вы перемещаете ее рукой по столу или другой поверхности, а она отслеживает эти перемещения и передает данные в компьютер специальной работающей программе. Эта программа (*драйвер мыши*) перемещает по экрану монитора специальный *указатель мыши*.

Типичная современная мышь (имеется в виду, конечно, компьютерная мышь) показана на рис. 2.19.



Рис. 2.19. Мышь

Мыши бывают разные. В широкой продаже встречаются *механические* и *оптические мыши*. Механические мыши отслеживают свое перемещение по поверхности с помощью шарика на днище мышки. Вы двигаете рукой мышь, шарик крутится в соответствующую сторону, а датчики внутри мыши определяют, в какую сторону и на сколько продвинулся шарик.

Оптические мыши не имеют шарика. В днище такой мышки светится маленькая лампочка. Отраженный от поверхности, по которой едет мышка, свет улавливается специальным датчиком (тоже установленным в днище мыши). Оптические мыши считаются более удобными в работе, чем механические, поскольку датчики у последней часто засоряются, и мышь в результате работает с трудом.

В первоначальном варианте мыши (имеется в виду распространенный вариант) имелось две кнопки. Такие мыши называют *двухкнопочными*. Затем получили широкое распространение *трехкнопочные мыши*. Сейчас во многих моделях мышек вместо средней кнопки устанавливают колесико. Его можно использовать и как кнопку (если надавить на него), и как средство прокрутки изображения на экране (если крутить его).

Кроме того, мыши бывают *проводные* и *беспроводные*. Проводные мыши подключаются к компьютеру проводом. Беспроводные имеют в комплекте специальные устройства, которые подключаются проводом к компьютеру и принимают от мыши радиосигналы. Сама мышь при этом не связана проводом ни с чем.

Экологические процессы, как вы знаете, тесно связаны с геоинформационными системами (ГИС). Любая ГИС включает в себя следующие компоненты:

- техническое обеспечение (hardware),
- программное обеспечение (software),
- информационное обеспечение (data).

Техническое обеспечение в свою очередь состоит из следующих частей:

- компьютеры (рабочие станции, ноутбуки, карманные ПК),
- средства хранения данных (винчестеры, компакт-диски, дискеты, флэш-память),
- устройства ввода информации (дигитайзеры, сканеры, цифровые камеры и фотоаппараты,
- клавиатуры, компьютерные мыши),
- устройства вывода информации (принтеры, плоттеры, проекторы, дисплеи).

Сердцем любой ГИС являются используемые для анализа данные. Устройства ввода позволяют конвертировать существующую географическую информацию в тот формат, который используется в данной ГИС. Географическая информация включает в себя бумажные карты, материалы аэрофотосъемок и дистанционного зондирования, адреса, координаты объектов собранные при помощи систем глобального позиционирования GPS (Global Position System), космических спутников или цифровой географической информации, хранимой в других форматах.

Приведем несколько примеров аппаратных платформ, которые могут быть использованы в ГИС. Самая простая и недорогая конфигурация ГИС-платформы, которая может быть установлена дома либо в небольшом офисе включает в себя компьютер и лазерный либо струйный принтер (черно-белый). Если же ГИС предназначена для создания высококачественных профессиональных цифровых карт, тогда аппаратная платформа может быть представлена следующими компонентами: высокопроизводительный компьютер, мощный сервер, современный дигитайзер, быстродействующие цветные лазерные принтеры и плоттеры.

2.2. Дополнительные технические средства

Кроме рассмотренных нами основных блоков персонального компьютера часто используются дополнительные устройства, расширяющие его возможности. Такие устройства называют *периферийными*.

Принтер

Самое распространенное периферийное устройство — *принтер*. Принтеры, или печатающие устройства, предназначены для вывода на бумагу текстовой информации, рисунков и графиков. Сейчас в эксплуатации вы можете встретить следующие типы принтеров:

- *матричные принтеры*, печать на которых осуществляется выбиванием краски из ленты с помощью игл, заключенных в специальный блок – матрицу (таких принтеров становится все меньше);
- *струйные принтеры* (рис. 2.20), печать на которых осуществляется разбрызгиванием капель жидкой краски. При цветной печати капли разного цвета смешиваются;



Рис. 2.20. Струйный принтер

- *лазерные принтеры* (рис. 2.21): здесь луч лазера применяется для подготовки поверхности, переносящей на бумагу красящий порошок; затем бумага нагревается, и порошок впекается в нее;
- *сублимационные принтеры*: относительно недавно разработанная технология, когда термическим способом краска переносится со специальной пленки на бумагу. Преимущество сублимационной печати – полученное изображение достаточно высокого качества и не боится воды.



Рис. 2.21. Лазерный принтер

Лазерные принтеры дороги, но имеют высокое качество печати и работают быстро; струйные медленнее, но более дешевы; матричные – самые дешевые, медленные и очень шумные; сублимационные принтеры печатают на маленьких форматах бумаги и используются для вывода фотографий.

При выборе принтера стоит обращать внимание не только на цену самого принтера, но и на стоимость *картриджей* к нему. Картридж – это емкость для материала, которым осуществляется печать (например, чернила или порошок). Может получиться, что, купив дешевый принтер, вы впоследствии потратите уйму денег на картриджи и, в конечном счете, поймете, что имело смысл купить более дорогую модель, но с более дешевыми расходными материалами.

Можно считать примерно так: стоимость принтера + (планируемое количество печатаемых страниц в год/емкость картриджа в страницах) × стоимость картриджа. И получившиеся величины у разных принтеров сравнить. Конечно, это далеко не идеальная формула.

Например, надо учитывать, что у струйных принтеров подсыхают чернила, если вы печатаете редко.

Сканер

Сканер – устройство для считывания графической информации с поверхности и размещения ее в памяти компьютера. Например, у вас есть напечатанная фотография, и вы решили немного ее подретушировать с

помощью компьютера. Или послать ее другу по электронной почте. Как поместить ее в компьютер? Для этого сканер и нужен.

Сканеры могут быть планшетные (устанавливаются на столе) и ручные (это устройство перемещают рукой по сканируемой поверхности). Пример планшетного сканера показан на рис. 2.22.



Рис. 2.22. Планшетный сканер

Модем

Модем обеспечивает передачу данных от одного компьютера другому, удаленному на значительное расстояние, по телефонным линиям связи. Существуют также *радиомодемы*, которые передают информацию другому радиомодему с помощью радиоволн.

Модем преобразует цифровой сигнал компьютера в аналоговый сигнал телефонной линии на передающей стороне и в обратную сторону на принимающей. То есть если вы решили связаться с другом или подругой и поиграть в сетевую игру, и у вас и у вашего друга (подруги) должен быть модем.

Модемы различают между собой по формату передаваемых данных и по быстродействию (количество бит в секунду). По варианту исполнения бывают встроенные (плата в системном блоке) и внешние модемы (рис. 2.23).



Рис. 2.23. Внешний модем

Таким образом, если модем внешний, то его надо одним проводом соединить с соответствующим гнездом компьютера, а вторым – подключить к телефонной розетке. Кроме того, внешние модемы должны питаться с помощью прилагаемого блока питания от электрической сети. О подробностях подключения именно вашей модели модема читайте в прилагаемой к модему инструкции (а она обязательно должна присутствовать и быть напечатанной на русском языке).

Сейчас в продаже присутствуют обычные модемы и ADSL-модемы (Asymmetrical Digital Subscriber Line – асимметричная цифровая абонентская линия). Последние могут осуществлять передачу на более высоких скоростях и не занимать при этом телефонную линию (хоть они и будут к ней подключены). Об условиях работы таких модемов, например, при подключении к Интернету, надо узнавать у организации, предоставляющей услуги по подключению (*провайдера услуг*).



Рис. 2.24. Wi-Fi router

Wi-Fi роутер (маршрутизатор) – устройство для беспроводного подключения пользователей к Интернету(рис. 2.24.), а также эта беспроводная базовая станция, предназначенная для обеспечения беспроводного доступа к уже существующей сети (беспроводной или проводной) или создания новой беспроводной сети.



Рис. 2.25. Медиаконвертор

Медиаконвертор - это маленькое устройство с двумя медиа-зависимыми интерфейсами и источником энергии (рис. 2.25.), получающее сигналы с данными от одного устройства, конвертирующее и передающее их другому устройству. Его можно установить практически в любое место в сети. Тип коннектора зависит от среды, которая должна быть преобразована устройством. Наиболее распространенным является использование кабеля UTP (неэкранированная витая пара) для многомодового или одномодового волокна. Медиаконвертеры могут поддерживать скорость сети от 10 Мбит/с до 10 Гбит/с.

Джойстик

Джойстик – в буквальном переводе с английского языка «палочка радости». Так и есть. Он доставляет немало удовольствия любителям полетать в играх-симуляторах. Такие игры имитируют полеты на существующих самолетах или на фантастических космолетах.

Джойстик – это манипулятор-рукоятка с набором кнопок на ней (рис. 2.26).



Рис. 2.26. Джойстик

Джойстики различаются по форме (от этого зависит, как вы будете держать их рукой), оснащению различными кнопками и точности управления при движениях манипулятором.

Графический планшет

Это устройство полезно любителям рисовать. Также полезен планшет и тем, кто ретуширует фотографии.

Планшет представляет собой пластину, лежащую на столе, по которой можно водить прилагаемой электронной ручкой (рис. 2.27). При этом работающий графический редактор (специальная программа для создания и редактирования изображений) понимает сигналы планшета и отображает на экране монитора заданную линию. Можно вообще использовать графический планшет вместо мыши, хотя к этому надо привыкнуть.



Рис. 2.27. Графический планшет

Различаются планшеты по размеру, разрешающей способности (возможности планшета распознать ваше мельчайшее движение ручкой), поддержке силы нажима (способности планшета делать линию толще или тоньше в зависимости от силы нажима на ручку).

Хорошим соотношением цена/качество обладают большинство моделей планшетов компании Wacom.

Инфракрасный порт и порт Bluetooth

И тот и другой используются для беспроводной связи компьютера с устройствами, имеющими аналогичные порты.

Чаще всего с помощью инфракрасного порта (ИК-порта) и порта Bluetooth осуществляют связь с мобильными телефонами и КПК. Через эти порты перекачиваются мелодии и картинки. Для управления загрузкой на компьютере должна работать специальная программа. Такие программы продаются на компакт-дисках или загружаются из Интернета (практически все производители мобильных устройств выкладывают их на своих сайтах для бесплатной загрузки).

Выглядит ИК-порт, как показано на рис. 2.28. Порт Bluetooth имеет практически такой же вид.



Рис. 2.28. ИК-порт

ТВ-тюнер

ТВ-тюнеры предназначены в первую очередь для просмотра телевизионных программ на компьютере. Практически все они обладают антенными входами для подключения телевизионного кабеля.

Кроме того, можно использовать TV-тюнеры для захвата видео с аналоговой видеокамеры (VHS, SVHS, Hi8 и др.) и размещения его в компьютере для монтажа и записи, например, на DVD-диск.

Линейки TV-тюнеров: Pinnacle PCTV, Fly Video, AVer Media, Beholder. Встречаются и внутренние тюнеры, и внешнего исполнения (рис. 2.29). Внутренние тюнеры устанавливаются на материнскую плату внутри системного блока.



Рис. 2.29. Внешний TV-тюнер

Защита от сбоев электропитания

Компьютер – прибор довольно чувствительный. Это касается, в первую очередь, скачков напряжения в электрической сети. Проблемы здесь две: во-первых, из-за скачка напряжения может сгореть блок питания системного блока, а во-вторых, при отключении питания (даже кратковременном) компьютер перезагрузится, и информация, с которой вы работали в этот момент, может безвозвратно потеряться. И от первого, и от второго есть спасение.

Сетевые фильтры

Для сглаживания скачков напряжения и улучшения характеристик электропитания

выпускается устройство под названием *сетевой фильтр*. По внешнему виду они похожи на обычные удлинители на несколько розеток с кнопкой включения.

Сетевые фильтры – наиболее дешевый способ защиты от сбоев электропитания с возможностью обесточивания одной кнопкой всех устройств, подключенных к нему (так и надо поступать, когда компьютер выключен).

На рынке представлены и российские производители, например, фильтры серии Pilot (это название в нашей стране стало для сетевых фильтров нарицательным), и зарубежные фирмы, например, хорошо зарекомендовавшие себя фильтры APC. Но сетевые фильтры не помогут в случае отключения электропитания.

Устройства бесперебойного питания

Чтобы продолжать питать компьютер и другие устройства электрическим током и дать вам возможность дожидаться восстановления внешнего электропитания или хотя бы успеть сохранить на диске документы, с которыми вы работаете, изобрели аппараты под названием *источники бесперебойного питания* (ИБП). Англоязычная аббревиатура, обозначающая устройства бесперебойного питания, – UPS (Uninterruptible Power Supply).

Такие устройства и выполняют работу сетевого фильтра, и имеют встроенный аккумулятор, благодаря которому при отключении внешнего электропитания, подключенные к ИБП устройства, продолжают работать. При работе в нормальных условиях аккумулятор автоматически подзаряжается.

Если внешнее электропитание пропало, ИБП сигнализирует об этом световым или звуковым сигналом (или и тем и другим). Обычно они начинают противно пищать. Многие модели ИБП при подключении к компьютеру специальным дополнительным кабелем (об этом подробнее читайте в руководстве к ИБП) могут сообщать ему о выключении внешнего электропитания и информировать о времени работы, на которое хватит заряда аккумулятора. Чтобы вы увидели эту информацию на экране компьютера, у вас должно быть установлено специальное программное обеспечение.

Отличаются ИБП, в первую очередь, временем, на которое хватает заряда аккумулятора (оно может быть от нескольких минут до нескольких часов), количеством разъемов для подключения питающихся устройств и способностями сообщать компьютеру о своем состоянии. Стоят ИБП заметно дороже сетевых фильтров. Среди производителей надежных ИБП выделяется компания APC.

Техника безопасности при работе с компьютером

Рассмотрим вопросы безопасности и условия безотказной работы компьютера. Можно привести ряд простых правил, которые позволят вам уберечь себя и свой компьютер от повреждений.

1. Системный блок устанавливается в месте, где он не будет подвергаться толчкам и вибрациям, а также интенсивному запылению.

2. Монитор устанавливается на уровне глаз так, чтобы на экран не попадали прямые солнечные лучи и, если вы пользуетесь ЭЛТ-монитором, позади монитора не работали люди. Уровень определяется так: сидя за компьютером и глядя горизонтально, вы должны видеть верхний край экрана. Расстояние от ваших глаз до экрана должно быть в пределах 50–70 см. Что касается запрета работать позади ЭЛТ-монитора, то следует обратить внимание на то, что ЭЛТ-монитор основан на электронно-лучевой трубке, которая при своей работе создает электромагнитное поле (к жидкокристаллическим мониторам это не относится). Со стороны экрана существует хорошая защита, а с обратной стороны такая защита слабее. Нахождение человека в электромагнитном поле монитора может вредить его здоровью.

3. Клавиатура устанавливается в месте, удобном для работы, так, чтобы она не загрязнялась. Мелкие частицы могут заклинивать клавиши, мешая им работать, жидкость, попав внутрь корпуса клавиатуры, вызывает выход из строя микросхем. Кстати, ремонт клавиатуры стоит дороже, чем приобретение новой. Существуют защищенные клавиатуры, но они довольно дорогие.

4. Принтер должен иметь устойчивое основание и достаточный простор для печати документов. Нельзя устанавливать принтер на горизонтальный системный блок.

5. Все работы, связанные с переключением кабелей, соединяющих устройства компьютера, следует выполнять при выключенном компьютере!

6. Перед включением компьютера необходимо убедиться, что все токоведущие части надежно изолированы, а розетки заземлены.

7. Не рекомендуется работать на компьютере более 4-х часов подряд. После каждого часа работы рекомендуется устраивать физкультурные паузы, снимающие нагрузку с глаз. Общая продолжительность пребывания за монитором не должна превышать 6 часов.

Мы с вами рассмотрели в отдельности все основные блоки персонального компьютера и большинство периферийных устройств. Но чтобы все это заработало, надо соединить устройства и сделать это правильно.

Обязательно выключайте все устройства компьютера перед подключением кабелей. В противном случае вы рискуете получить удар током или сжечь микросхемы, обрабатывающие сигнал.



Рис. 2.30. Задняя стенка системного блока

Главным управляющим устройством компьютера является центральный процессор. Он расположен на материнской плате внутри системного блока. Поэтому практически все устройства либо подключаются к разъемам самой материнской платы, либо к разъемам печатных плат, уже подключенных к материнской плате. Практически все такие разъемы расположены на задней стенке системного блока. Некоторые для удобства выводятся на переднюю стенку (лицевую панель) системного блока.

Набор разъемов у разных материнских плат различный, набор установленных устройств также отличается. На рис. 2.30 показан примерный вид задней стенки системного блока. Рассмотрим разъемы и их назначение подробнее.

В первую очередь нас интересуют разъемы для подключения основных устройств: монитора, клавиатуры и мыши. Разъем для подключения монитора может выглядеть, как показано на рис. 2.31 (аналоговый разъем) либо как показано на рис. 1.30 (цифровой разъем DVI, Digital Video Interface). Если бы системный блок был прозрачным, мы бы увидели, что разъем для подключения монитора расположен на графическом адаптере (видеокарте), печатной плате, подключенной к материнской плате.



Рис. 2.31. Аналоговый разъем для подключения монитора



Рис. 2.32. Цифровой разъем для подключения монитора

Один конец соединительного кабеля монитора подключается к разъему на мониторе, а другой к разъему на задней стенке системного блока. Часто для более жесткой фиксации на разъеме присутствует два винта, которые нужно закрутить. Для подключения клавиатуры предусмотрен специальный разъем (рис. 2.33).



Рис. 2.33. Специализированные разъемы для подключения клавиатуры и мыши

Для манипулятора «мышь» также предусмотрен специальный разъем, хотя многие модели мыши подключаются к USB-порту. Цвета разъемов соответствуют цветам штекеров мыши и клавиатуры. Некоторые разъемы для подключения основных блоков компьютера мы рассмотрели. Кроме них существуют несколько специализированных разъемов и набор универсальных разъемов.

Наиболее часто из специализированных разъемов встречается порт для цифрового видео. Он имеет стандарт *IEEE1394*, известный также как *FireWire* или *i.Link*. Если вы имеете цифровую видеокамеру, передавать видео в компьютер можно, соединив кабелем цифровой выход камеры с портом *IEEE1394* вашего компьютера. Многие современные компьютеры имеют встроенные порты этого стандарта. Такой порт можно также найти на звуковой карте Audigy 2 от фирмы Creative, если она установлена на вашем компьютере. Если вы не обнаружили порта *IEEE1394* на своем компьютере, можно купить дополнительную плату с такими портами и установить ее. Стоимость таких плат относительно невелика. Порты *IEEE1394* бывают двух

типов: 4-контактные (рис. 2.34) и 6-контактные (рис. 2.35). Может получиться, например, что порт на вашей видеокамере – 4-контактный, а порт в компьютере – 6-контактный. Никакой проблемы в этом нет. Надо просто при покупке кабеля учесть это обстоятельство. Соответственно один штекер такого кабеля должен быть 4-контактный, а другой – 6-контактный (рис. 2.36).



Рис. 2.34. Порт IEEE1394 4-контактный

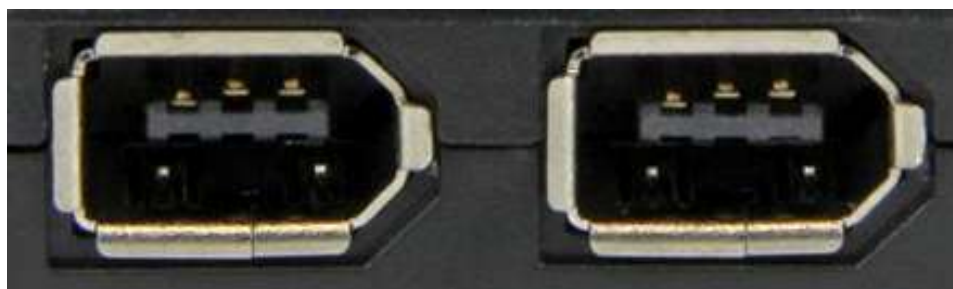


Рис. 2.35. Порт IEEE1394 6-контактный



Рис. 2.36. Кабель для соединения 4-контактного и 6-контактного разъемов

Набор универсальных портов (универсальных в том смысле, что они предназначены не для конкретного устройства, а для любых устройств с

соответствующим разъемом) включает обычно *COM-порт* (Communication port – коммуникационный порт), *LPT-порт* (LinePrinter – построчный принтер) и несколько USB-портов. COM-порт, или, как его еще называют, *последовательный порт* (рис. 2.37), разработан уже давно и до сих пор с успехом используется, например, для подключения модемов. Скорость передачи информации через этот порт невелика, но она все же выше, чем скорость передачи информации при подключении модема к существующим телефонным линиям.

COM-портов в вашем компьютере может быть несколько, но выглядят они совершенно одинаково.



Рис. 2.37. COM-порт

LPT-порт, или, как его еще называют, *параллельный порт* (рис.2.38), исторически используется главным образом для подключения принтеров. Хотя многие современные модели принтеров предусматривают подключение к USB-порту. Чтобы определить, куда подключить ваш принтер, прочитайте инструкцию к нему и посмотрите, какой кабель идет с ним в комплекте.



Рис. 2.38. LPT-порт

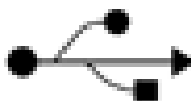
USB-порт (рис. 2.39) был разработан позже COM– и LPT-портов. Сегодня большинство периферийных устройств подключаются именно к портам этого типа. Есть даже маленькие пылесосы для клавиатуры, которые подключаются к USB-порту.



Рис 2.39. USB-порт

В отличие от портов другого типа, некоторые устройства можно вполне безопасно подключать к USB-порту во время работы компьютера. К таким устройствам относятся FlashDrive («флэшки»). Вы подсоединяете этот накопитель к любому из USB-портов, операционная система, скорее всего, выдаст на экран сообщение о том, что она опознала новое устройство, а затем вы сможете увидеть содержимое Flash-Drive как еще один логический диск с соответствующим именем. Далее вы работаете с файлами на Flash-Drive, а по окончании работы, прежде чем вынуть «флэшку», нужно ее отключить с помощью средств операционной системы. Как это сделать, например, при работе с Windows 7, мы рассмотрим позднее.

К портам USB часто подключают цифровые фотоаппараты. Кабель для такого подключения прилагается к фотоаппарату или приобретается отдельно. Обычно USB-порты и штекеры, подсоединяемые к этим портам, помечаются значком



Все, что мы с вами до сих пор рассматривали, все, что состоит из пластмассы и металла и можно потрогать руками (соблюдая технику безопасности), называют *аппаратным обеспечением* компьютера. Но все аппаратное обеспечение, какое бы хорошее и новое оно ни было, «мертво» без компьютерных программ, которые управляют устройствами, выдавая им детальные инструкции. О том, какие программы бывают и для чего они нужны, поговорим в следующем разделе.

Выбор персонального компьютера

В первую очередь, выбирая компьютер, необходимо определиться, для чего он вам нужен и сколько денег вы готовы потратить. Обычно по назначению выделяют три основные группы: офисный компьютер, мультимедийный компьютер и игровой компьютер.

Задачи офисного компьютера – обеспечить работу текстового и табличного редактора, электронной почты, предоставить доступ в Интернет, воспроизводить музыку и видео. Его возможностей будет достаточно для простых игр. Главные преимущества таких компьютеров – малая цена и невысокое энергопотребление.

В мультимедийных компьютерах, как видно из названия, основной упор делается на обработку видео (в том числе и стандарта Full HD), звука и изображений. Поддерживаются и более требовательные к ресурсам компьютера игры. При соответствующих финансовых возможностях этот тип компьютеров наиболее предпочтителен для домашнего использования.

Игровой компьютер дорог, но любители игр с реалистичной и динамичной графикой уверены, что оно того стоит. Львиную долю этой цены занимают затраты на видеокарту и оперативную память.

Самый простой и дешевый вариант офисного компьютера или домашнего неттопа (компьютер для доступа в Интернет, хотя для работы с документами он тоже вполне годится) таков: корпус с блоком питания на 180 Вт (например, INWIN IW-BT553 Micro ATX), процессор Atom 230 (он продается сразу установленным, например, в составе материнской платы INTEL D945GCLF), оперативная память 1 Гбайт, интегрированные в материнскую плату графический адаптер и звуковая карта, интегрированная сетевая карта, жесткий диск на 250 Гбайт стандарта SATA, DVD-привод.

Можно порекомендовать следующую конфигурацию полноценного офисного персонального компьютера: корпус с блоком питания на 350 Вт, процессор Intel Pentium Dual-Core E2180 или AMD Athlon II X2 240, оперативная память 2 Гбайт, интегрированные в материнскую плату

графический адаптер и звуковая карта, интегрированная сетевая карта, жесткий диск на 250 Гбайт стандарта SATA, DVD-привод.

Для мультимедийного типа персональных компьютеров можно ориентироваться на такой набор: корпус с блоком питания на 500 Вт, процессор Intel Core 2 Duo E8200 или AMD Phenom II X3 720 BE, оперативная память 2 Гбайт, графический адаптер ATI Radeon HD 5770 или NVIDIA GeForce GTS 250, интегрированная звуковая карта, интегрированная сетевая карта, жесткий диск на 500 Гбайт стандарта SATA, пишущий DVD-привод (DVD-RW).

Ну, а если, денег на игры и развлечения не жалко, можно присмотреться к такой конфигурации игрового персонального компьютера: корпус с блоком питания на 850 Вт, процессор Intel Core i7 860 или AMD Phenom II X4 955, оперативная память 4 Гбайт, графический адаптер ATI Radeon HD 5870, звуковая карта Sound Blaster X-Fi Xtreme Gamer, интегрированная сетевая карта, жесткий диск на 1 Тбайт стандарта SATA, пишущий DVD-привод (DVD-RW) или привод Blu-ray.

2.3. Понятие о программном обеспечении. Классификация программ.

Как мы уже разобрались, компьютер не будет работать, пока компьютерные программы не начнут выдавать центральному процессору детальные инструкции.

Программ написано великое множество. Назначение их может быть совершенно разным. Многие программы работают так, что вы не замечаете их присутствия на компьютере. А другие, напротив, заявляют о себе живописными изображениями на экране монитора. Так давайте разберемся в этом многообразии.

Классификация программ

Программы для компьютера делят на следующие классы:

- *системные программы*, которые управляют работой компьютера, а также выполняют различные вспомогательные функции: создание копий используемой информации, выдача справочной информации о компьютере, проверка работоспособности устройств компьютера и т. д.;

- *прикладные программы*, обеспечивающие выполнение необходимых пользователям работ: редактирование текстов, рисование изображений, обработка информационных массивов, игры и т. п.;

- *инструментальные системы* или *системы программирования*: предназначены для создания новых программ для компьютеров.

Системные программы

Здесь вам следует обратить внимание на следующие программы:

- *операционная система* (ОС) – это комплекс программ, который осуществляет диалог с пользователем, а также управляет работой компьютера и осуществляет поддержку работы других программ;

- *драйверы* – помогают операционной системе управлять конкретными устройствами, максимально используя их возможности. Для одного устройства может быть несколько драй-веров;

- *утилиты* – выполняют различные вспомогательные функции, например диагностику устройств.

Прикладные программы

Они разделяются по направлениям деятельности пользователя и могут включать в себя, например, такие наиболее широко применяемые программы, как:

- программы подготовки текстов (например, Microsoft Word);
- программы обработки табличных данных (например, Microsoft Excel);
- программы обработки графической информации (например, CorelDRAW или Adobe Photoshop);

- программы для хранения и обработки больших массивов информации, использующие системы управления базами данных (например, Oracle);

- программы обработки звука (например, Sound Forge);
- программы обработки видео (например, Ulead VideoStudio);
- игры (например, Quake или War Craft).

Можно представить себе взаимную работу операционной системы, драйверов и прикладных программ, как показано на рис. 2.40.

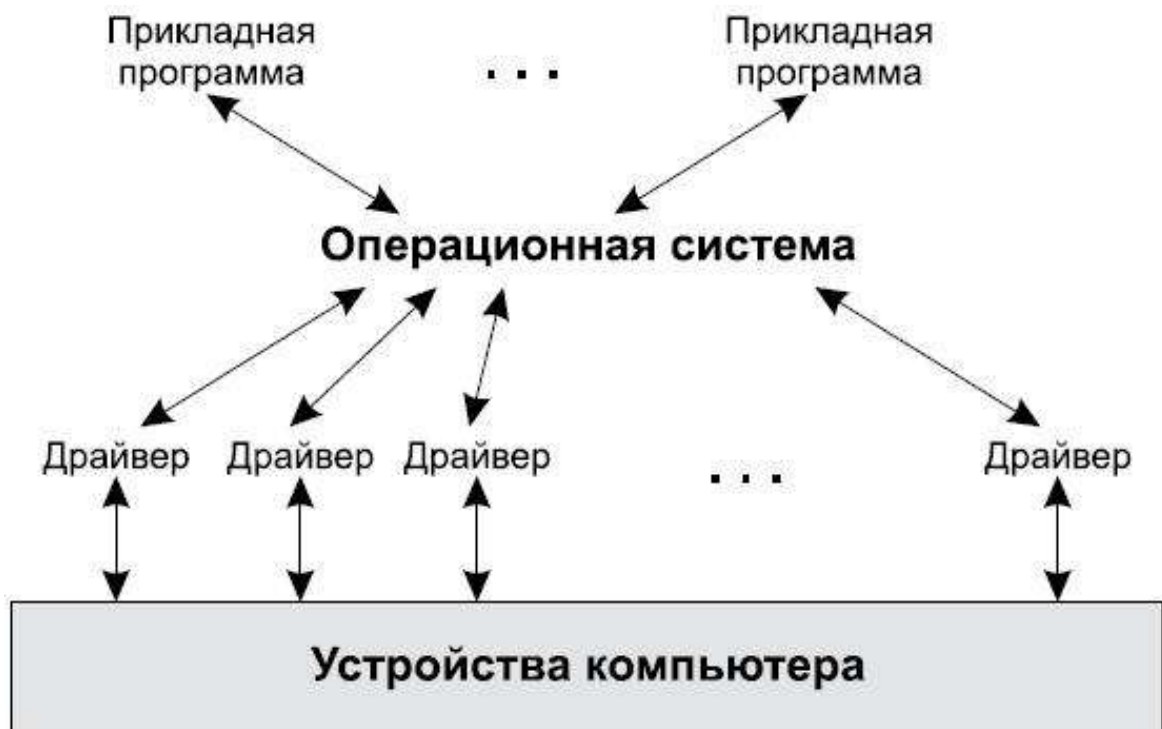


Рис. 2.40. Взаимная работа операционной системы, драйверов и прикладных программ

Инструментальные системы

Они предназначены для создания новых программ для компьютеров и опираются на языки программирования. В этих системах присутствуют: средства ввода текстов программ, средства перевода этих текстов на язык процессора и средства отладки программ.

Для каждого языка программирования существует своя инструментальная система и, как правило, не одна. Примеры таких систем: Borland C++, Borland Delphi, Microsoft Visual C++, Microsoft Visual FoxPro.

С помощью инструментальных систем создаются программы всех других классов и другие инструментальные системы. Надо сказать, что программы создаются специально для работы в определенной операционной системе. Таким образом, программа, написанная для операционной системы MacOS (Macintosh Operating System – ОС для компьютеров Apple Macintosh), в операционной системе Windows работать не будет.

Файлы и папки

Все данные в компьютере хранятся в виде *файлов*. Для простоты можно представить такие ассоциации: вы взяли фотографию на бумаге, подписали ее название – это файл; взяли скрепленную пачку листов бумаги с напечатанным текстом вашего доклада или реферата, где на титульном листе написано его название, – это тоже файл; взяли иллюстрированный журнал, где есть обложка с названием, – и это файл. То есть файлом называется каким-либо образом организованная и поименованная порция информации. Причем эта информация посвящена определенной тематике. И, что важно, у каждого файла есть имя. Например, вы можете файл с письмом другу назвать *Письмо*. Обычно при работе с файлами мы не видим сразу их содержимое (изображение, текст, фильм и т. д.). Нам показывается только список имен файлов, а уж затем мы выбираем, с каким файлом будем работать.

Файлы отличаются не только именами, но и способом хранения данных в них. То, каким способом записывается информация в файл, определяется типом файла. Для того чтобы отличить один тип файла от другого, используют расширения. Расширения располагаются в правой части имени файла и отделяются от самого имени точкой. Например, наш файл с письмом другу – это просто файл с текстом, и тогда его имя может выглядеть так: *Письмо^*. При определенных настройках операционная система скрывает

расширение и тогда тип файла можно определить по значку, который показывается рядом с именем файла (это зависит от операционной системы).

Наименование может быть произвольным, то есть придумывается вами. А расширения уже придуманы и зависят от операционной системы и прикладных программ, с которыми вы работаете. Например:

.exe – исполняемые файлы (программы);

.txt – текстовые файлы;

.doc – файлы документов, создаваемые пользователями в программе Microsoft Word;

.jpg – рисунки и т. д.

Файлов на современных дисках может храниться очень много. Чтобы долго не искать нужную информацию, придумали рассортировывать файлы по отдельным местам – *папкам*. Как мы можем положить в картонную папку листы с докладом, вложить туда же графики или фотографии, так и в электронной папке могут храниться файлы самых разных типов с различным содержанием, но объединенные для вас каким-либо смыслом. Что и куда положить, решаете вы сами. Обычно даже если программы копируют на ваш диск набор файлов автоматическом режиме, они перед этой операцией спрашивают у вас, в какой папке можно все это разместить. В некоторых операционных системах вместо термина *папки* используют термин *каталоги*.

Папки, в свою очередь, можно вкладывать в другие папки. Допустим, вы взяли бумажную папку с фотографиями, сделанными в отпуске, потом взяли другую папку, где лежат авиабилеты (вы оставили их себе на память), и все это сложили в третью большую бумажную папку, а на ней написали «Отпуск 2007 года». А потом добавили в эту папку видео-кассету, снятую во время отдыха. Так и на компьютерном диске можно разместить в одной папке несколько других папок и рядом с ними положить файлы. Для наглядности покажем дерево папок (графическое изображение того, что в какую папку вложено) на рис. 2.41 В самом верху дерева написано имя логического диска, где все эти папки лежат.

Иногда, чтобы указать точное расположение файла, используют специальную конструкцию под названием *путь к файлу*. Если посмотреть на рис. 2.41, можно увидеть там обозначение файла с именем пляж.jpg. Запись полного пути к нему может выглядеть так:

d: \Работа \Документы учебные\ раздаточный материал. doc

В зависимости от установленной операционной системы разделительные косые черты могут быть наклонены в различные стороны. Например, в нашем случае мы использовали запись, стандартную для операционных систем семейства Microsoft Windows.

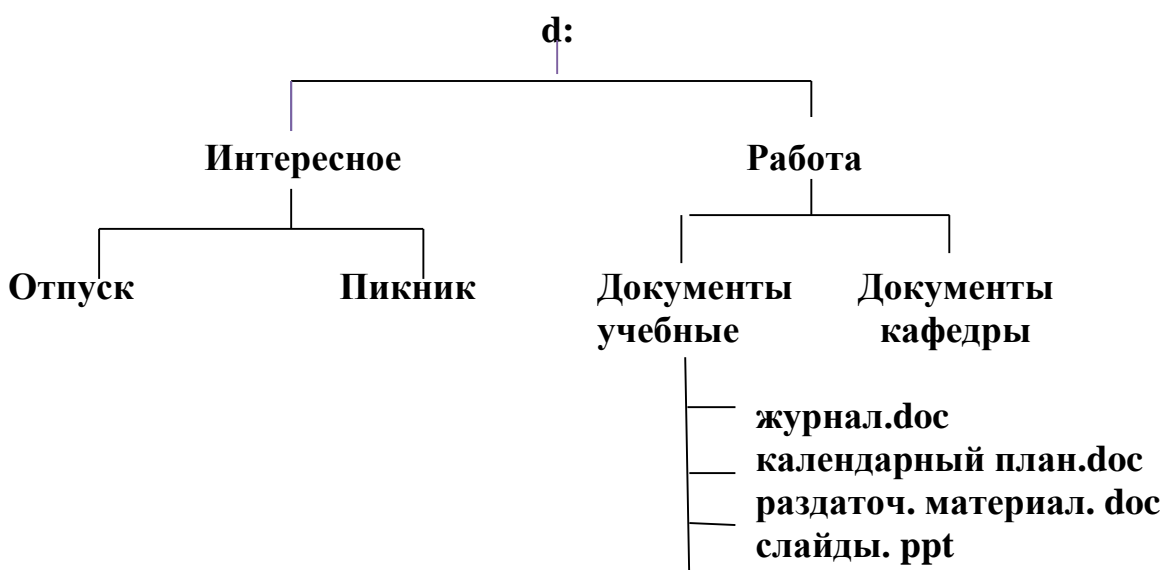


Рис. 2.41. Дерево папок

Как мы уже говорили, при создании новой папки или файла вы сами придумываете для них имя. Требования к именам папок и файлов в различных операционных системах отличаются. В старых операционных системах можно было в именах файлов и папок использовать, например, только латинские буквы, и длина имени не могла превышать восьми символов.

В современных операционных системах это ограничение снято. Теперь можно и использовать русские буквы и называть папки и файлы хоть целыми фразами.

Если говорить о программном обеспечении ГИС, то следует отметить, что большинство программных пакетов обладают схожим набором характеристик, такими как, послойное картографирование, маркирование, кодирование геоинформации, нахождение объектов в заданной области, определение разных величин, но очень сильно различаются в цене и функциональности. Выбор программного обеспечения зависит от конкретных прикладных задач, решаемых пользователем.

Ключевые слова

Персональный компьютер, ноутбук, КПК, основные устройства компьютера, устройство ввода, устройство вывода, устройство обработки, устройства хранения, средства коммуникаций, системный блок, монитор, клавиатура, мышь, процессор, оперативная память, жесткий диск, разъем, шлейф, съемные носители, гибкие магнитные диски, устройства Flash-Drive, DVD-диск, CD-диск, дисковод, внешний жесткий диск, графический адаптер, звуковая карта, hardware, software, дополнительные устройства компьютера, принтер, сканер, модем, Wi-Fi router, медиаконвертер, джойстик, графический планшет, инфракрасный порт, Bluetooth, ИК-порт, ТВ-тюнер, UPS, системные программы, прикладные программы, инструментальные программы, операционная система, драйверы, утилиты, прикладные программы, файл, папка, расширение, путь к файлу.

Вопросы для самопроверки:

1. Назовите основные типы устройств ввода.
2. Назовите основные типы устройств вывода.
3. Что такое арифметическо-логическое устройство?
4. Назовите компоненты блоки ПК.
5. На какие типы делится память?
6. Что такое жесткие и гибкие диски?
7. Что такое программа, программное обеспечение?

8. В чем состоит постановка задачи?
9. Что такое алгоритм решения задачи?
10. Назовите основные свойства алгоритмов.
11. Какие категории специалистов охвачены процессом подготовки программ?
12. Что такое программный продукт, его свойства и классификация?
13. Что входит в системное программное обеспечение?
14. Приведите характеристику основных видов программных продуктов базового программного обеспечения.
15. Расскажите о ППП общего назначения?
16. Что такое пакеты прикладных программ и как их можно классифицировать?

Лекция 3. ТЕКСТОВЫЙ ПРОЦЕССОР MS WORD

3.1. Элементы рабочего окна MS WORD

Рабочий экран MS Word состоит из типичных для среды MS Windows элементов. В верхней части экрана расположена строка заголовка, содержащая название программы Microsoft Word. Рядом с названием программы отображается имя документа. Если документу не было присвоено имя, строка заголовка содержит имя, установленное по умолчанию, например, Документ1, Документ2 и т.д.

Под строкой заголовка расположена строка меню, которая является общей для всех окон документов. В результате выполнения щелчка мышью на имени меню открывается ниспадающее меню, содержащее набор команд. Непосредственно под строкой меню размещается панель инструментов. С каждой пиктограммой этой панели связана некоторая команда. Для вызова команды, как правило, достаточно щелкнуть мышью на соответствующей пиктограмме. Такой способ вызова команд обычно намного эффективнее клавиатурного или обычного пути через меню. Под пиктографическим меню

находится координатная линейка, с помощью которой можно устанавливать размеры полей страниц и абзацных отступов. В нижней части окна находится строка состояния (Status bar). Строка содержит информацию о текущем состоянии документа и режиме работы процессора. Вертикальная и горизонтальная линейки прокрутки размещены вдоль правого и нижнего краев рабочего окна. Линейки прокрутки предназначены для передвижения текста документа по вертикали и горизонтали соответственно. Документ в MS Word открывается следующими способами:

1 способ

Загрузить MS Word.

В главном меню выбрать команду Файл/Открыть (File/Open).

Открыть папку, где находится нужный документ (окно Look in)

Выбрать документ.

Нажать кнопку Открыть (Open).

2 способ (для загрузки файла, с которыми работали недавно)

Загрузить MS Word.

В главном меню команды Файл (File) в нижней части окна найти имя документа и щелкнуть по нему.

3 способ

Если нужная папка или документ находятся на рабочем столе, то дважды щелкните по имени папки и/или документа. В этом случае MS Word загрузится автоматически, а затем загрузится и нужный документ.

4 способ

Открыть окно программы Проводник (Explorer) (например, из меню MS Office), найти нужный документ и щелкнуть по его пиктограмме. При этом MS Word загружается автоматически, а затем загружается и выбранный документ.

5 способ

1. Загрузить MS Word.

2. На панели инструментов найти значок Открыть (Open) и щелкнуть по нему.

3. Открыть папку, где находится нужный документ (окно Look in).

4. Выбрать документ.

5. Нажать кнопку Открыть (Open).

6 способ

1. В меню Пуск (Start) выбрать режим Документы.

2. Найти в списке среди 15 документов свой документ и щелкнуть по его имени.

7 способ

1. Открыть меню Пуск (Start).

2. Выбрать режим Поиск (Find)/ Файл или папка (File or Folder).

3. В появившемся окне набрать имя документа.

4. После окончания процесса поиска, щелкнуть по имени документа.

Программа MS Word и затем выбранный документ будут загружены.

Способы закрытия документа в MS Word

1 способ

В меню Файл (File) нажать команду Закрыть(Close).

2 способ

Нажать кнопку закрытия окна в правом углу.

3 способ

В меню Файл (File) нажать на команду Выход (Exit). При этом появится вопрос записать ли открытый документ. После ответа документ будет закрыт и MS Word завершит работу.

3.2. Создание, открытие и сохранение документа

Документы в MS Word могут иметь длинные имена; именами могут быть даже целые предложения.

Если документ новый и не имеет имени, то сохранение документа выполняется по команде Файл/Сохранить как (File/Save As). При этом необходимо выполнить следующие действия:

1. Выбрать в меню команду Файл/Сохранить как (File/Save As).
2. В окне Сохранить в (Save in) установить нужную папку. В случае ее отсутствия новую папку можно создать, нажав на пиктограмму создания папки и введя имя папки.
3. В окне Имя документа (File name) MS Word предложит имя (по первым нескольким словам документа). Это имя можно изменить на любое другое.
4. Нажать на клавишу Сохранить (Save).

Если документ уже имеет имя - то его следует сохранить либо с помощью пиктограммы Сохранить (Save) Панели инструментов, либо командой Файл/Сохранить (File/Save), либо (если хотите завершить сеанс работы с MS Word) командой Выход(Exit) и на вопрос о желании записать документ - ответ «Да» (Yes).

В случае, если работа происходила с несколькими документами, удобно воспользоваться командой Файл/Сохранить Все (File/Save All).

3.3. Ввод и редактирование текста

Символы – это буквы, пробелы, знаки пунктуации, цифры, специальные символы, такие как @, *, &. Символы можно форматировать для изменения вида символов.

Форматы можно добавлять и удалять, нажав кнопку панели инструментов Форматирование или комбинацию клавиш или выбрав команду Шрифт из меню Формат. Можно отформатировать любую часть текста – от одиночного символа до целого документа; можно также скопировать форматирование символов и затем применить это форматирование в другом тексте.

Когда Вы начинаете набирать новый документ, текст отображается шрифтом и размером, установленными в системе MS Word. Если Вы предпочитаете другой вид, нужно выбрать шрифт, размер и другие форматы символов, которые будут использоваться по умолчанию.

Шрифт – это набор букв, цифр, символов и знаков препинания, разработанных определенным образом. То, какие шрифты и размеры Вы можете использовать, зависит от шрифтов, доступных для принтера и установленных на компьютере.

Три вида шрифтов влияют на работу в MS Word: масштабируемые шрифты, принтерные шрифты и экранные шрифты. Масштабируемые шрифты могут быть отображены любым принтером, который может печатать в графическом режиме.

Чтобы изменить внешний вид текста, нужно выделить его и затем отформатировать. Если Вы выбрали команду без выделения текста, то изменится формат только того текста, который Вы напечатаете, начиная с этого места.

Вы можете быстро применить или удалить форматы наиболее общих символов, используя панель инструментов Форматирование. Чтобы применить или удалить форматы всех символов из выделенной части текста или применить или удалить некоторые форматы одновременно, используйте команду Шрифт из меню Формат.

3.4. Шрифтовое оформление текста

Для шрифтового оформления текста:

1. Выделите необходимый текст, который хотите отформатировать, или установите курсор в том месте, с которого нужно начать текст нового формата.

Чтобы изменить формат одного слова, выделите его.

2. Используя панель инструментов Форматирование, выполните одно из следующих действий:

Чтобы изменить шрифт или размер шрифта, выберите название шрифта или размер шрифта в поле Шрифт или поле Размер или напечатайте имя или размер шрифта и затем нажмите ENTER.

Чтобы добавить или удалить формат полужирного шрифта, курсива или подчеркнутого шрифта, нажмите кнопки Полужирный, Курсив или Подчеркивание.

Использование команды Шрифт:

Вы можете использовать или удалить другие форматы – скрытый текст, цвет, индексы и расстояние между символами, используя команду Шрифт из меню Формат.

Использование клавиатуры:

Вы можете быстро форматировать текст, используя клавиши сокращения.

Если Вы хотите повторить последнее форматирование, выделите текст и в меню Правка выберите Повторить. Если Вы выполняли форматирование командой Шрифт, будет повторено оформление, описанное в диалоговом окне Шрифт. Если Вы использовали панель инструментов Форматирование или клавиши сокращения, чтобы применить несколько форматов, то будет повторено только последнее форматирование.

Форматирование символов:

Вы можете использовать цвет в тексте. Чтобы показывать в цвете, Вы должны иметь цветной монитор; чтобы печатать в цвете, Вы должны иметь цветной графопостроитель или цветной принтер.

Как изменить цвет текста:

1. Выделите текст, подлежащий изменению, или определите место, начиная с которого Вы хотите печатать текст в новом цвете.
2. В меню Формат выберите Шрифт.
3. Выберите вкладку Шрифт.
4. В поле Цвет выберите желаемый цвет и нажмите ОК.

MS Word поддерживает 16 цветов. Опция Авто, установленная с использованием значка Цвет в Панели Управления MS Windows, является установкой по умолчанию.

Можно быстро изменять регистр букв текста без повторного набора, делая текст либо состоящим только из строчных букв, либо только из прописных, либо с обязательным выводом первой буквы каждого слова прописной, либо с выводом только первой буквы выделения прописной, либо меняя все существующие строчные на прописные и наоборот.

Как изменить регистр букв текста

1. Выделите текст, в котором нужно изменить регистр букв.
2. В меню Формат выберите команду Регистр.
3. Выберите нужную опцию, а затем нажмите кнопку ОК.

Работа с клавиатурой:

Можно быстро изменить регистр букв текста, сделав все буквы прописными или строчными или сделав первую букву каждого слова прописной. Выделите текст, который нужно изменить, а затем нажимайте клавиши SHIFT+F3 до тех пор, пока текст не примет желаемый вид.

Набор и исправление текста:

Существует возможность найти каждое вхождение символов, включая прописные и строчные буквы, целых слов или частей слов. Можно найти текст, отформатированный каким-либо образом, определенный формат, стиль или язык. Например, можно найти все вхождения какого-либо слова полужирного формата. Можно также исключить какой-либо формат из поиска текста. Например, можно установить переключатель Полужирный для поиска полужирного текста, а можно очистить его для поиска текста любого формата, кроме полужирного.

Однажды определив, что нужно искать, можно продолжать работать с документом, а потом возобновить поиск.

Как найти текст или форматирование

1. В меню Правка выберите команду Найти.

2. В поле **Что** выполните одно из следующих действий:

Чтобы найти текст выполните следующее:

Текст без форматирования: Наберите текст. Если под полем **Что** появится формат, нажмите кнопку **Без формата**.

Текст с форматированием: Наберите текст. Нажмите кнопку **Формат** и затем выберите **Шрифт**, **Абзац**, **Язык** или **Стиль**. Установите нужный формат и нажмите кнопку **ОК**.

Только форматирование: Удалите весь текст в поле **Что**. Поместите курсор в поле **Что**, нажмите кнопку **Формат** и затем выберите **Шрифт**, **Абзац**, **Язык** или **Стиль**. Установите нужный формат и нажмите кнопку **ОК**.

В поле **Что** можно также воспользоваться предыдущими аргументами поиска, сохраненными MS Word. Для этого нужно щелкнуть мышью на кнопке со стрелкой и выбрать нужный шаблон для поиска из списка четырех последних аргументов поиска.

3. Выберите другие необходимые опции управления поиском.

4. Нажмите кнопку **Следующее**.

Если текст или форматирование найдутся, документ откроется на странице, где этот текст расположен. Можно редактировать документ и при открытом диалоге **Найти**, а затем продолжить поиск, нажав кнопку **Следующее**.

5. Чтобы остановить процесс поиска, нужно нажать клавишу **ESC**. Чтобы закрыть диалог **Найти**, нужно нажать кнопку **Отмена**.

Работа с абзацами

Внешний вид документа зависит от форматирования, которое Вы применили к тексту, и, в большей степени, от форматирования отдельных абзацев. Вы можете изменить такие элементы форматирования абзацев, как выравнивание текста, отступы, расстояние между абзацами, позиции табуляции, обрамление, заполнение, символы бюллетеня и нумерацию.

В MS Word абзацем является любая часть документа, содержащая текст, графику, объекты (например, формулы и графики) или другие элементы, за которой следует маркер абзаца.

Маркер абзаца – это не просто значок, помечающий конец абзаца; в нем содержится информация о форматировании, которое применяется к этому абзацу. Если Вы удаляете маркер абзаца, то удаляется и форматирование абзаца. Текст в этом абзаце получит форматирование следующего за ним абзаца.

Чтобы случайно во время работы не удалить маркеры абзацев, а с ними и все форматирование абзацев, которое они содержат, лучше отобразить маркеры абзацев.

Можно отобразить или скрыть в документе маркеры абзацев, нажав кнопку Специальные символы Стандартной панели инструментов. Чтобы посмотреть форматы отдельного абзаца, нажмите кнопку Справка стандартной панели инструментов или клавиши SHIFT+F1. Когда указатель мыши изменится на знак вопроса, щелкните на интересующем Вас тексте. По окончании просмотра форматов еще раз нажмите кнопку Справка или клавишу ESC. По умолчанию текст сдвигается влево и выравнивается по левому полю. Однако можно задать опции, чтобы отцентрировать текст, выровнять его по правому полю или выровнять текст по формату между левыми и правыми полями. Выбранное выравнивание влияет на весь текст в выделенных абзацах.

Как отцентрировать или выровнять текст:

1. Выделите абзацы, которые требуется отцентрировать или выровнять.
2. Нажмите нужную кнопку выравнивания панели инструментов Форматирование.

Чтобы выровнять текст нажмите:

- По левому полю.
- По центру между полями.
- По правому полю.

- По ширине между левым и правым полями.

Перед тем, как центрировать или выравнивать текст относительно полей, убедитесь, что для абзаца не установлены отступы.

Использование клавиатуры. Можно также быстро выровнять текст с помощью клавиш ускоренного действия.

Если Вы работаете с узкими текстовыми колонками, то рассмотрите вопрос о переносе слов. В случае выравнивания влево или вправо разрешение переноса может уменьшить неровности края текста. В случае выравнивания по формату перенос текста может уменьшить дополнительное расстояние между словами.

Обычно позиции табуляции установлены через каждые 1,27 см от левого поля. В выделенных абзацах можно установить и изменить выравнивание позиций табуляции. Когда Вы устанавливаете новый табулятор, то MS Word отменяет все стоящие слева от него табуляторы, устанавливаемые по умолчанию.

Как установить позиции табуляции:

1. Выделите абзацы, в которых требуется установить или изменить позиции табуляции. При изменении установок позиций табуляции убедитесь, что выделены все абзацы, на которые должны повлиять эти изменения, чтобы за один шаг изменить все установки позиций табуляции.

2. Чтобы установить или изменить выравнивание табулятора, следует нажимать кнопку выравнивания в левой части горизонтальной линейки, пока не установится нужный тип выравнивания.

3. На линейке щелкните на там, где должен стоять этот табулятор.

Использование команды Табуляция: С помощью команды Табуляция из меню Формат можно точно задать позиции табуляции.

Чтобы в документе заполнить пустое пространство между позициями табуляции точками, тире или сплошной линией, воспользуйтесь символами-заполнителями табуляции. Использование заполнителя табуляции предоставляет собой простейший способ для формирования горизонтальной

черты, которая следует за текстом, или для продления текста на определенное расстояние между полями.

Как установить табуляцию с символами-заполнителями:

1. Выделите абзацы, в которых требуется перед табуляторами вставить заполнители.
2. В меню Формат выберите команду Табуляция.
3. В поле Позиции табуляции введите значение новой позиции табуляции или выберите уже существующую, чтобы добавить символы-заполнители.
4. В группе Выравнивание выберите нужный тип выравнивания текста.
5. В группе Заполнитель выберите нужный символ-заполнитель, а затем нажмите кнопку Установить.
6. Нажмите кнопку ОК.

По умолчанию MS Word использует одинарный межстрочный интервал. Когда Вы изменяете межстрочный интервал, это влияет на все строки выделенных абзацев или абзаца, в котором находится курсор.

Как изменить межстрочный интервал:

1. Выделите абзацы, в которых требуется изменить межстрочный интервал.
2. В меню Формат выберите команду Абзац.
3. Выберите вкладку Отступы и интервалы.
4. В поле Межстрочный группы Интервалы выберите нужный тип межстрочного интервала.



Если в поле Межстрочный Вы выбрали значение Минимум или Точно, то в поле Сколько введите или выберите значение межстрочного интервала.

Подробнее об опциях межстрочного интервала см. Команда Абзац (меню Формат).

5. Нажмите кнопку ОК.

Значение межстрочного интервала варьируется в зависимости от шрифта и размера в пунктах текста в каждой строке. Если строка содержит текстовый символ, размер которого больше окружающего текста, графический элемент или формулу, то для этой строки автоматически увеличивается интервал между строками.

Вместо того чтобы добавлять после абзаца пустые строки, нажимая клавишу ENTER, воспользуйтесь командой Абзац из меню Формат, чтобы добавить расстояние перед и/или после абзаца. Можно точно настроить это расстояние, задав, например, 2 пт (пункта) или 0,7 см (сантиметра).

Как настроить расстояние между абзацами

1. Выделите абзацы, расстояние между которыми требуется изменить.
2. В меню Формат выберите команду Абзац.
3. Выберите вкладку Отступы и интервалы.
4. В полях До и После группы Интервалы введите или выберите нужные значения, а затем нажмите кнопку ОК. Вы можете добавить обрамление с любой стороны абзаца. А также можно применить обрамления к ячейкам таблицы, к графике и кадрам.

Как применить или удалить обрамления:

1. Выделите абзацы, ячейки таблицы или кадры, к которым требуется применить обрамления или удалить их.
2. Заполнение применяется для фона абзацев, ячеек таблицы и кадров. Заполнение и обрамление можно использовать одновременно, например, чтобы подсветить абзац на полях документа или создать вынос.

Как применить или удалить заполнение

1. Выделите абзацы, ячейки таблицы или кадры, к которым требуется добавить или удалить заполнение.

2. Нажмите кнопку **Обрамление** панели инструментов **Форматирование**.

3. Выполните одно из следующих действий:

Чтобы применить заполнение, выберите в поле **Заполнение** панели инструментов **Обрамление** нужный узор заполнения.

Чтобы удалить заполнение, на вкладке **Заполнение** диалога **Обрамление** и заполнения (меню **Формат**) выберите опцию **Нет**.

Использование команды **Обрамление** и заполнения: С помощью данной команды из меню **Формат** можно настроить цвет фона и цвет текста. См.

Вкладка **Заполнение**.

Ширину области заполнения можно настроить с помощью маркеров отступов на горизонтальной линейке.

В системе MS Word можно быстро создавать нумерованные списки и списки-бюллетени, облегчающие чтение и понимание текста. Можно создавать иерархические списки -- списки имеющие много уровней, в таких списках допустимы как нумерованные элементы, так и символы бюллетеня. Иерархические списки полезны для структурирования юридических и технических документов. Можно пронумеровать все заголовки в документе, имеющие встроенные стили заголовка. Если элементы нумерованного списка добавляются, удаляются или переупорядочиваются, нумерация элементов обновляется.

Можно быстро упорядочить текст, отсортировав элементы в алфавитном, числовом порядке или по датам.

Система MS Word позволяет быстро создать нумерованные списки и списки-бюллетени, облегчающие восприятие текста. Если добавить, удалить или переопределить элементы в нумерованном списке, нумерация обновляется.

Как добавлять и удалять символы бюллетеня и нумерацию из списков.

1. Выделите элемент, к которому нужно добавить или из которого нужно удалить символы бюллетеня или нумерацию.

Когда выделен список-бюллетень или нумерованный список, непосредственно символы бюллетеня или нумерация не выделяются.

2. В панели инструментов Форматирование выполните одно из следующих действий:

Чтобы добавить или удалить символы бюллетеня, нажмите кнопку Список. Чтобы добавить или удалить нумерацию, нажмите кнопку Нумерация.

Когда для начала нового абзаца нажимается клавиша ENTER, форматирование списка переносится на следующий абзац. Например, если в списке, который содержит элементы, перенумерованные с 1 по 5, после пятого элемента нажать клавишу ENTER, то новый абзац начнется с элемента под номером 6.

3.5. Работа с таблицами

Таблицы следует образовывать для организации информации и создания интересных разметок страницы с расположением текста и графики в примыкающих столбцах. В таблице, как и в электронной таблице, работа выполняется со строками и столбцами ячеек.

Чтобы создать таблицу:

1. Установите курсор в то место, в котором нужно создать таблицу.

2. Нажмите кнопку Вставка таблицы Стандартной панели инструментов. Под кнопкой появятся линии сетки.

3. Нажмите кнопку мыши и перемещайте курсор по линиям сетки, пока не выделится нужное число строк и столбцов.

Точку вставки помещается в начало первой ячейки. Можно сразу же набирать текст.

Если таблица вставляется в документ с помощью команды Вставить таблицу из меню Таблица, то при этом можно точно задать ширину столбцов, также можно поэтапно создавать таблицу с помощью мастера.

В большинстве случаев перемещение по таблице выполняется, как и в других местах документа с помощью мыши или клавиатуры.

Чтобы перемещаться по таблице щелкните на ячейке, на которую нужно переместиться или нажмите одну из следующих клавиш:

Позиция	Клавиши
Следующая ячейка	TAB
Предыдущая ячейка	SHIFT+TAB
Следующий символ	СТРЕЛКА ВПРАВО
Предыдущий символ	СТРЕЛКА ВЛЕВО
Предыдущая или следующая строка	СТРЕЛКА ВВЕРХ или СТРЕЛКА ВНИЗ

Первая строка ALT+HOME или ALT+7 на цифровой клавиатуре

Последняя ячейка строки ALT+END или ALT+1 на цифровой клавиатуре

Первая ячейка столбца ALT+PAGE UP или ALT+9 на цифровой клавиатуре

Последняя ячейка столбца ALT+PAGE DOWN или ALT+3 на цифровой клавиатуре

Если в ячейке имеется текст, нажатие клавиш TAB или SHIFT+TAB выделит его. Если курсор находится в последней ячейке таблицы, нажатие клавиши TAB добавит новую строку.

Когда в документ вставляется пустая таблица, курсор помещается в первую ячейку. Можно тут же начинать вводить текст.

Чтобы набирать текст в таблице выполните одно из следующих действий:

Начать новый абзац -ENTER

Переместиться на следующую ячейку - TAB

Переместиться на предыдущую ячейку - SHIFT+TAB

Добавить новую строку внизу таблицы - TAB в конце последней строки

Переместиться на символ вперед или назад - СТРЕЛКА ВПРАВО или СТРЕЛКА ВЛЕВО

Переместиться на соседнюю строку СТРЕЛКА ВВЕРХ или СТРЕЛКА ВНИЗ

Чтобы добавить строки в таблицу:

1. Выделите строку или строки, над которыми нужно вставить новую строку или строки.

Выделите столько строк, сколько нужно добавить.

2. Выполните одно из действий:

Нажмите кнопку Вставка строк стандартной панели инструментов.

Установите курсор вне таблицы в конце строки, а затем нажмите клавишу ENTER.

В меню Таблица выберите команду Вставить строки.

Над выделенными строками будет вставлена строка (строки).

Чтобы добавить строку в конце таблицы, установите курсор в последней ячейке последней строки, а затем нажмите клавишу TAB. Ячейки в новой строке будут иметь форматирование ячеек предыдущей строки. Столбцы можно добавлять внутри существующей таблицы или в ее конце.

Чтобы добавить столбцы в таблицу:

1. Выделите столбец или группу столбцов, слева от которых нужно вставить новый столбец или столбцы.

Выделите столько столбцов, сколько нужно добавить.

2. Выполните одно из действий:

Нажмите кнопку Вставка столбцов стандартной панели инструментов.

В меню Таблица выберите команду Вставить столбцы.

Слева от выделенного столбца или столбцов будет вставлен столбец (столбцы). В таблице может быть не более 31 столбца.

Чтобы добавить столбец справа от таблицы, установите курсор за последним столбцом. Выберите команду Выделить столбец из меню

Таблица, а затем нажмите кнопку Вставка столбцов стандартной панели инструментов.

Столбцы можно добавлять внутри существующей таблицы или в ее конце.

Чтобы удалить строки или столбцы в таблице:

1. Выделите удаляемые строки или столбцы.
2. В меню Таблица выполните одно из следующего:

Выберите команду Удалить строки. Выберите команду Удалить столбцы.

Другие способы удаления частей таблицы

Строки или столбцы можно также удалять из таблицы с помощью кнопки Вырезать стандартной панели инструментов или команды Вырезать из меню Правка. Подробнее об удалении текста см. Набор и исправление текста.

Если Вы выделили только текст внутри ячейки, этот текст добавляется в новое место без изменения форматирования имеющегося текста. Однако, если в выделение попадает маркер конца ячейки, переносимый текст заменяет существующий текст и форматирование текста переносится на новое место.

Для выполнения простых арифметических операций и вычисления математических выражений можно использовать поле = (Формула).

Для использования формул в таблице необходимо знать, что в таблице строки обозначаются цифрами а столбцы обозначаются буквами

	А	В	С
1			
2			
3			

Содержимые ячеек можно суммировать или производить над ними какие-либо арифметические операции:

1. Выделите ячейку, в которую следует поместить результат расчета.
2. Выберите команду Формула в меню Таблица.
3. Если в поле "Формула" предлагается не та формула, которую следует использовать, удалите ее.
4. Выберите нужную формулу из списка "Вставить функцию". Например, для прибавления числа выберите SUM.

Для ссылки на ячейки таблицы введите их в скобках в формулу. Например, для ссылки на ячейки A1 и B4 введите a1;b4.

Для получения более подробных сведений о ссылках на ячейки таблицы нажмите кнопку.

5. В поле "Формат числа" введите формат для чисел. Например, для вывода чисел в виде процентов введите 0,00%.

Результат вычисления появляется в выбранной ячейке в виде поля. При изменении ссылок на ячейки для обновления результата вычисления выделите нужное поле и нажмите клавишу F9.

3.6. Создание колонтитулов

Колонтитул — это текст и/или рисунок, который печатается внизу иливерху каждой страницы документа. В зависимости от места расположения (на верхнем или на нижнем поле страницы) колонтитулы бывают верхними и нижними.

Допускается создавать уникальный или вообще не создавать колонтитул для первой страницы документа или каждого из разделов. Можно также создавать отличающиеся колонтитулы для четных и нечетных страниц разделов или документа.

Просмотр, правка и форматирование колонтитула

1. Выберите команду Колонтитулы в меню Вид.
2. Внесите в колонтитул необходимые изменения.

Нетрудно изменить расположение колонтитулов по горизонтали путем центрирования колонтитула, выравнивания его по левому или правому краю или вынесения текста колонтитула на поле. Можно также изменить

расположение колонтитулов по вертикали путем изменения расстояния от колонтитула до верхнего или нижнего края страницы.

Удаление колонтитула

1. Выберите раздел, содержащий колонтитул, подлежащий удалению.
2. Выберите команду Колонтитулы в меню Вид.
3. Выделите колонтитул, который следует удалить, после чего нажмите клавишу DEL.

При изменении или удалении колонтитула в любом разделе точно так же изменяются или удаляются колонтитулы в остальных разделах, если связь с предшествующим разделом не была разорвана принудительно.

Ключевые слова

Создание, открытие, сохранение документа, обрамление, редактирование текста, таблицы, строка, столбец, графика, работа с объектами, создание и редактирование формул, редактор формул, раздел, колонтитул, пиктограмма, элементы экрана, основное меню, панель инструментов, шрифт, интервал, вид, вставка, формат, буфер обмена.

Вопросы для самопроверки:

1. Как создается документ?
2. Как сохранить документ?
3. Что собой представляют многоколоновые документы?
4. Как вставляется таблица?
5. Что такое шрифтовое оформление текста?
6. Как вставляется объект?
7. Как редактируется текст и линии столбца и строк?
8. Как вставляется и удаляется столбец и строка?
9. Как выполняются операции вычисления в MS Word?
10. Приложение Word Art, основное назначение и использование.

Лекция 4. ТАБЛИЧНЫЙ ПРОЦЕССОР MS EXCEL

4.1. Основные понятия табличного процессора MS Excel

Microsoft Excel – одно из наиболее популярных приложений Microsoft Office, используемых в работе специалистами различного профиля (экономистами, финансистами, бухгалтерами, статистиками, инженерами, аналитиками и др.). Электронные таблицы являются идеальной средой для выполнения вычислений различной сложности без особых затрат на программирование, обеспечивают хранение больших массивов информации типа реляционных баз данных. В Microsoft Excel включен развитый математический аппарат, специализированные информационные технологии статистического анализа, доступны графические средства представления и анализа данных.

Информационные технологии Microsoft Excel доступны пользователям различной квалификации – от обычных пользователей до профессиональных программистов. Продвинутые пользователи создают комплексные информационные технологии, основанные на применении всего арсенала средств Microsoft Excel, в первую очередь, это - набор встроенных функций, стандартные надстройки, решают задачи моделирования и анализа данных, используют возможности Microsoft Excel по интеграции данных и функций обработки других программных средств, и в первую очередь Microsoft Office.

Microsoft Excel широко применяется для автоматизации бухгалтерского учета. Документы расчетного характера (разработочные и расчетные таблицы, учетные регистры) могут переводиться в электронную форму, сохраняться как шаблоны для многократного использования. Microsoft Excel поддерживает электронный документооборот, обеспечивая создание, описание, поиск и хранение документов сложной структуры, взаимосвязь комплексов расчетных документов.

Окно Microsoft Excel содержит:

- Главное меню команд;
- Панели инструментов;
- Дополнительные элементы экрана (имена столбцов, номера строк, строка формул, окно имен блоков ячеек, статусная строка, полосы прокрутки).

Настройка Microsoft Excel выполняется в диалоговом окне Параметры команды меню Сервис – Параметры, содержащем ряд вкладок:

- *Вид* – управляет формированием элементов экрана (вывод полос прокрутки, ярлыков листов, строки состояния, строки формул, заголовков строк и столбцов, сетки ячеек, примечаний к ячейкам, показ объектов и др.).
- *Вычисления* – выбор способа вычисления выражений в формулах (автоматически, вручную, перед сохранением), ограничение на погрешность и число итераций в расчетах, обновление удаленных ссылок (на внешние источники) и др.
- *Правка* – выбор способа редактирования информации (в ячейке или только в строке формул), задание направления перехода курсора после ввода, контроль перезаписи ячеек, обновления автоматических связей книги с другими источниками и др.
- *Общие* – выбор стиля ссылок в формулах, вывод окна для заполнения свойств файла рабочей книги, установка числа листов в новой рабочей книге, выбор стандартного шрифта (тип и размер символов) для содержимого ячеек, указание рабочего каталога (папки), куда записываются новые файлы, указание каталога (папки) автозагрузки, из которого при запуске Microsoft Excel автоматически открываются все рабочие книги, ввод имени пользователя, являющегося автором новых книг и др.

- *Переход* – выбор формата файла для новых рабочих книг, переход к Microsoft Excel при работе с электронной таблицей Lotus 1-2-3.
- *Списки* – перечень стандартных списков для автозаполнения ячеек, создание и редактирование списков.
- *Диаграмма* – настройка активной диаграммы.
- *Цвет* – настройка цветов для оформления таблиц и диаграмм.

4.2. Рабочая книга табличного процессора MS Excel

Рабочая книга – основной документ приложений Microsoft Excel. Новая книга может содержать постоянный текст, данные, форматы ячеек и областей печати, формулы, диаграммы, рисунки, пользовательские панели инструментов, меню, макросы, элементы автотекста и др. Рабочая книга состоит из листов, максимальное количество листов в одной книге – 255. Новая книга создается с помощью команды *Файл – Создать*, сохраняется новая рабочая книга с помощью команды меню *Файл – Сохранить*. Для автоматического сохранения рабочей книги через указанный интервал времени устанавливается надстройка *Автосохранение*. Для изменения состава надстроек Microsoft Excel служит команда меню *Сервис – Надстройки*. После установки надстройки появляется новая команда меню *Сервис – Автосохранение*, с помощью которой устанавливается интервал времени для сохранения рабочей книги.

Microsoft Excel ориентирован на интеграцию данных с другими программами, поддерживает экспорт и импорт данных, обеспечивая преобразование форматов файлов с использованием внутренних и внешних специальных информационных технологий.

Microsoft Excel обеспечивает обмен данными рабочих книг (экспорт книг) с другими программами, а также он ориентирован на интеграцию данных с другими программами, поддерживает экспорт и импорт данных, обеспечивая преобразование форматов файлов с использованием специальных информационных технологий обмена. Команда меню *Файл –*

Открыть обеспечивает преобразование в формат рабочей книги во внешних форматах.

4.3. Листы электронной таблицы и операции в них

В рабочей книге используются стандартные имена листов - Лист1, Лист2 и т.д. Имена листов можно также изменять с помощью команд меню Формат – Лист - Переименовать.

С листами рабочих книг можно выполнять операции *перемещения* и *копирования* в пределах одной и той же рабочей книги или различных открытых книг с помощью команд меню Правка – Переместить/скопировать лист. В диалоговом окне Переместить или скопировать выбирается лист текущей рабочей, указывается место вставки. Другой способ выполнения копирования/перемещения листов – графический, с использованием мыши.

Последовательность действий:

1. Выделить листы рабочей книги;
2. Установить курсор на ярлыке первого листа;
3. При нажатой левой кнопке мыши выполнить графическое перемещение ярлыков листов.

Лист электронной таблицы состоит из ячеек, в которых хранятся данные различного вида:

- текст – максимальный объем хранимой информации в одной ячейке - 32 Кбайт;
- числа – фиксированные числа, денежный и финансовый форматы с указанием обозначения валюты, проценты, дроби.
- дата/время в полном, кратном формате;
- логические константы типа ИСТИНА, ЛОЖЬ;
- формулы.

Предельный размер листа электронной таблицы содержит 256 столбцов и 65536 строк.

В заполненной электронной таблице можно добавлять или удалять ячейки, строки и столбцы. При этом конфигурация листа остается неизменной - 256 строк, 65536 столбцов. При удалении фрагментов таблицы ее размерность восстанавливается, выполняется автоматический сдвиг и переадресация ячеек, расположенных вправо и вниз от удаляемого фрагмента. При вставке фрагментов таблицы происходит сдвиг вниз и вправо ячеек, автоматическая их переадресация, при сохранении предельных размеров листа. Для вставки /удаления строк или столбцов, блоков ячеек таблицы может осуществляться предварительное выделение фрагмента таблицы.

4.4. Расчеты в электронных таблицах

После того, как вы научились вводить и редактировать информацию в ячейки электронной таблицы, нужно научиться обрабатывать эту информацию. В программе MS Excel обработка информации выполняется с помощью формул. Далее мы рассмотрим, какие бывают формулы, как их вводить и как с ними работать.

Формулами в программе называют выражения, по которым выполняются вычисления на странице. Используя формулы, вы можете сравнивать данные в ячейках, складывать и умножать значения ячеек, объединять содержимое ячеек и выполнять множество других действий. Формулы можно вводить вручную или использовать готовые формулы, предлагаемые программой.

Формула всегда начинается со знака равенства, и если в ячейке первый символ «=», программа считает, что эта формула, и ее нужно вычислить. В противном случае, программа воспримет выражение как текстовую строку. Например, выражение 2*2- это просто строка текста, а выражение =2*2- формула. В этой формуле есть операция умножения, обозначаемая символом «*», а также два числа, являющихся константами, то есть постоянными, невычисляемыми значениями.

Для создания формул в программе предусмотрено несколько математических и логических операций, каждая из которых описывается своим оператором. Если вы хотите сложить и вычесть, нужно использовать операторы «+» и «-». Для умножения и деления применяют операторы «*» и «/», а если нужно возведение в степень, следует использовать оператор «^». Для сравнения величин в формулах используются отдельная группа операторов. Для проверки равенства или неравенства используют операторы «=» и «<>», для сравнения нужно использовать оператора «>», «<», «>=», «<=».

Вид выражения в формуле может быть довольно сложным. При вычислении формулы сначала выполняется возведение в степень, затем умножение и деление, а потом сложение и вычитание. Сравнение выполняется только после выполнения всех остальных операций. Если встречаются несколько однотипных операторов, вычисления идут слева направо. Однако вы можете изменить порядок выполнения оператора с помощью скобок. Вначале будут выполняться операции в скобках.

Кроме констант, в формулах можно использовать адреса и имена ячеек электронной таблицы. В этом случае в формуле участвует не сам адрес, а значения в ячейке с данным адресом, то есть в формуле используется ссылка на ячейку. Например, формула $=A1+A2*3$ берет содержимое ячейки A2, умножает его на три, после чего прибавляет содержимое ячейки A1. Результат записывается в ячейку, где расположена формула.

При помощи ссылок можно использовать в одной формуле информацию, находящуюся в разных частях листа, а также использовать в нескольких формулах значения одной ячейки. Кроме того, можно задавать ссылки на ячейки других листов той же книги и на другие книги. В ссылках можно использовать не только одинарные, но неодинарные диапазоны ячеек. Например, ссылка A1 указывает на ячейку с данным адресом, а ссылка A1:A3 указывает на диапазон ячеек между этими адресами. Ссылка A1;A3 будет

указывать на две ячейки A1 и A3. Ссылка A1:A3;A7 указывает на ячейки A1, A2, A3 и A7.

Если нужно указать ссылку на ячейку из другого листа рабочей книги, следует указать имя листа, символ «!» и адрес ячейки, например, Баланс!A1. Иногда формулы используют ячейки из разных рабочих книг. Чтобы ввести ссылку на ячейку из другой рабочей книги, следует записать путь к файлу с книгой, а затем имя файла в квадратных скобках, например: «C:\Мои документы\». Если в пути к файлу есть пробел, он должен быть заключен в кавычки.

Кроме констант и ссылок на содержимое ячеек в формулах могут участвовать функции, то есть стандартные формулы, которые возвращают результат различных действий над значениями, входящими в функцию, как аргументы. Для вызова функции нужно написать ее имя и в скобках указать список аргументов, разделенных запятыми, например ЦЕЛОЕ (A1). В качестве аргументов могут выступать числа, адреса ячеек, формулы и др. функции. Например, выражение СУММА (A1,МАКС(A2,A3)) содержит функцию суммирования содержимого ячейки A1 или максимального значения из ячеек A2 и A3.

Если вы введете, например, в ячейку A4 следующую формулу: =СУММ(A1:A3), а затем скопируете содержимое ячейки A4 в ячейку B4, вы увидите, что в ячейке формула будет выглядеть по-другому: СУММ(B1:B3). Это происходит потому, что по умолчанию в MS Excel принято относительная адресация ячеек в формулах. На самом деле, запись ссылок A1:A3 в формуле означает, что используется не сами ячейки, а их координаты по отношению к той ячейке, где расположена формула. При переносе формулы на новое место относительные координаты будут другими, и ссылки в формуле изменятся. Это очень удобно, так как вы можете легко копировать ячейки с формулами в любое место таблицы, и при этом их значение будет автоматически скорректировано.

Иногда в формуле требуется указать адрес конкретной ячейки, который не должен меняться при перемещении ячейки с формулой. Для этого используется абсолютная адресация ячеек. Чтобы адрес ячейки не менялся при копировании формулы, перед адресом строки или столбца нужно поставить символ «\$». Если мы запишем нашу формулу так: =СУММ(\$A\$1:\$A\$3), при переносе этой формулы любую другую ячейку, ссылки на ячейки не изменятся. Естественно можно сделать абсолютным только часть адреса, например, \$A1. В этом случае при переносе будет меняться только адрес строки, а адрес столбца будет неизменным.

Для удобства работы с таблицей вы можете присвоить ячейке или группе ячеек собственное имя, которое можно использовать в адресации ячейки в дальнейшем. Это имя может быть произвольным, но с некоторыми ограничениями. Если ячейке присвоено имя, то адресовать эту ячейку можно как обычным способом, так и по этому имени. Имена можно присваивать ячейкам, диапазонам, формулам и константам. Имена могут заменять собою ссылки, что существенно облегчает задачу адресации в электронных таблицах. Преимущество от использования имен заключается в том, что формулы благодаря именам становятся более наглядными и понятными, например, формула =СУММ(доходы_за_1_квартал) лучше воспринимается, чем =СУММ(A1:B1).

Имя обладает некоторыми ограничениями. Оно должно начинаться с буквы или с символа подчеркивания. После первой буквы можно использовать любые символы, кроме дефисов и пробела, то есть два слова в имени не должны разделяться пробелом. Длина имени не должна превышать 255 символов, но предпочтительнее короткие имена, которые проще набираются и легко читаются в раскрывающихся списках. В имени могут присутствовать как строчные, так и прописные буквы, но программа их не различает. Имя не должно напоминать адреса ячеек, например, AB1, B\$20 и т.д.

Чтобы присвоить имя ячейке или группе ячеек, выделите их и щелкните мышью левее строки формул – там, где показан адрес выделенной ячейки. Введите вместо адреса произвольное имя, и оно будет присвоено ячейке. После того, как вы присвоили имя выбранной ячейке, вы можете использовать это имя для адресации ячейки при работе с Excel, например, вставлять его в формулы. Также можно присвоить имя ячейке с помощью команды меню. Выберите команду Вставка - Имя - Присвоить. Появится диалог ввода имени ячейки.

В поле ввода Имя следует ввести имя ячейки. В списке, расположенном ниже, отображаются имена, уже присвоенные ячейкам. После ввода имени в поле нажмите кнопку Добавить, и имя появится в списке имен. Добавление имени с помощью команды меню менее удобно, чем первый из рассмотренных способов, а вот для удаления имени ячейки или группы ячеек удобнее воспользоваться этим диалогом. Чтобы удалить имя из списка, выберите его и нажмите кнопку Удалить. После ввода имени или его удаления нажмите кнопку ОК, чтобы закрыть диалог.

4.5. Работа с формулами

Простейшие формулы лучше вводить вручную, а для ввода формул, содержащих функции, лучше воспользоваться мастером, который мы рассмотрим ниже. Сложные формулы, содержащие множество функций, также удобнее вводить вручную. Для ручного ввода формулы введите в строке формул знак равенства =, обозначающее начало формулы, после чего введите собственно формулу, в которой могут быть константы, ссылки и операторы. Отметим, что вместо ввода адреса ячейки можно просто щелкнуть на нужной ячейке мышью, и соответствующий адрес будет добавлен в формулу. Например, вы можете ввести формулу = A1+A2, а можете ввести знак равенства, щелкнуть мышью на ячейки A1, ввести знак «плюс», после чего щелкнуть мышью на ячейке A2. По окончании ввода выражений

нажмите клавишу Enter на клавиатуре. В активной ячейке появится результат. Так вы можете вводить любые формулы в разные ячейки.

Часто используемые функции можно вставлять в формулы простым нажатием кнопки на панели инструмента. Выделите ячейку и щелкните мышью на строке формул. Нажмите кнопку - *Автосуммирование* на панели инструмента, и в строку формул будет вставлена функция суммирования СУММ(). Далее нужно щелкнуть мышью между скобок и указать адреса или диапазон ячеек, которые следует суммировать. Можно ввести адрес прямо в строке формул, а можно щелкать мышью на нужных ячейках, и ссылки на них будут подставляться в формулу.

Открывающийся список десяти последних использовавшихся функций появляется в левой части строки формул в тот момент, когда вы вводите символ =. Откройте список, нажав кнопку в его правой части, и вы увидите перечень используемых функций. Выбрав функцию из списка, Вы подставите ее в строку формул, и на экране появится диалог для ввода аргументов функции. В полях ввода можете указать адреса ячеек или константы, участвующие в формуле. После ввода всех аргументов нажмите кнопку ОК, диалог будет закрыт, а функция добавлена в формулу.

Большинство функций вводятся в ячейки с помощью специального мастера. Сначала выбирается ячейка, в которую мы будем заносить формулу. После появления курсора для ввода информации в строке формул нажмите кнопку f_x в строке формул программы или на стандартной панели инструментов в зависимости от версии программы. В строку формул автоматически будет подставлен символ =, и на экране появится первый диалог мастера вставки функций.

Выбор функции осуществляется в несколько этапов. Сначала в списке Категория нужно выбрать группу функций, объединенных по их назначению, к которой принадлежит искомая функция. После того, как категория выбрана, в списке Выберите функцию, появится список функций, относящийся к выбранной категории.

Программа предлагает вам 12 категорий функций. Если вам нужны математические расчеты, выбирайте категорию Математические. Среди функций этой группы вы найдете тригонометрические функции, округление, вычисление степеней и логарифмов, и так далее. Часто при работе с таблицами используют текстовые функции. Чтобы выбрать функцию из этой группы, выберите категорию Текстовые. Среди текстовых функций есть функции поиска строк, изменения регистра символов, и так далее. Для сравнения различных величин используются логические функции. Выберите категорию Логические, и вы можете выбрать функции логического сложения, умножения и функции проверки условий. Если вы хотите использовать функции работы с датами, выбирайте категорию Дата и время. С помощью функций этой категории вы можете узнать текущую дату, время, день недели и так далее. Много полезных функций находится в категории Финансовые. Если вы не представляете себе, к какой области относится нужная вам функция, выберите категорию Полный алфавитный перечень.

После того, как функция найдена, щелкните на ней мышью в списке. Выберите функцию. Обратите внимание, что ниже списка отображается синтаксис выбранной функции, то есть правило ее написания и набор аргументов. Еще ниже находится описание функции, помогающее определить, для чего она предназначена. Нажмите кнопку ОК, и первый диалог мастера будет закрыт. Появится следующий диалог, который предназначен для конкретизации параметров функции, определенной ранее. Количество и вид параметров зависят от типа выбранной функции. Может быть один параметр, а может быть несколько.

Параметры можно задать вручную, введя константу, или ссылку в поле ввода, а можно с помощью мыши указать ячейки таблицы, значения которых используются как параметры. Нажмите кнопку в правой части поля ввода параметров функции, диалог мастера будет свернут, освобождая тем самым рабочую область окна программы. Теперь можно щелкнуть мышью на требуемой ячейке, и ее адрес будет автоматически подставлен в функцию.

Чтобы вернуть диалог, нажмите кнопку в правой части свернутого диалога. Аналогично можно заполнить остальные поля для ввода параметров, если они есть. После того, как параметры функции установлены, нажмите кнопку ОК, чтобы закрыть второй диалог мастера. На этом работа мастера будет закончена, и в активную ячейку будет вставлена нужная формула.

Чтобы изменить формулу в ячейке, следует сделать ячейку с формулой активной, и текст формулы появится в строке формул. Если щелкнуть мышью на строке формул, ячейки, участвующие в формуле, будут выделены разными цветами. Таким же цветом выделяются аргументы функции в строке формул. Это сделано специально, чтобы вы могли наглядно увидеть, какие ячейки участвуют в редактируемой формуле. Вы можете внести исправления в формулу, отредактировав ее в строке формул. По окончании нажмите кнопку в строке формул или нажмите клавишу.

Если вы допустили ошибку при вводе или редактировании формулы, программа сообщит вам об этом. Если в ячейке с формулой отображается текст, начинающийся с символа «#», значит, в формуле есть ошибка. Если в ячейке вы увидите # # # # # , ширина ячейки недостаточна для отображения результата. Чтобы устранить ошибку, нужно расширить ячейку или изменить формат числа.

Когда в ячейке появляется сообщение #ИМЯ?, значит, программа не распознала имя функции или имя ячейки, используемое в формуле. Сообщение #ЗНАЧ! говорит о попытке некорректного использования функции. Обычной ошибкой является несоответствие данных установленному формату, например, вместо числа или даты в аргументе стоит текст. Это же значение ошибки будет появляться, когда для функции или оператора, которые требуют одного значения аргумента, задается диапазон.

Сообщение #ЧИСЛО! появляется, если возникли проблемы с представлением или с использованием чисел. Не исключено, что в функции с числовым аргументом используется аргумент нечислового формата.

Возможно также, что в ячейку введена формула, возвращающая слишком большое значение. Если в ячейке появилось сообщение #ССЫЛКА!, значит есть проблема с адресацией ячеек, участвующих в формуле. Возможно, что формула содержит ссылку на ячейку, которая уже удалена, или ссылку на ячейку, в которую скопировано содержимое других ячеек.

Сообщение #ДЕЛ/0! возникает при попытке деления на ноль. Такая ситуация чаще возникает не из-за того, что в ячейку записано явное деление на ноль, а как следствие использования в качестве делителя ссылки пустого множества ячеек. Возможно, что вы ошиблись в определении пересечения двух диапазонов. Сообщение #Н/Д говорит о наличии неопределенных данных.

Если после ввода формулы программа не сообщила об ошибке, это еще не значит, что ошибки нет. В составе программы есть ряд средств проверки правильности выделенных формул. Например, вы можете наглядно посмотреть, как значения в одних ячейках влияют на результат формул в других ячейках. Выберите команду меню Сервис – Зависимости – Панель зависимостей, чтобы отобразить на экране панель инструментов проверки формул.

Рассмотрим назначение инструментов данной панели. Чтобы увидеть, какие ячейки участвуют в формуле, выделите ячейку с формулой и нажмите кнопку на панели инструментов зависимостей. Будут автоматически нарисованы связи от используемых ячеек к ячейке с формулой.

Чтобы убрать связи от ячеек к формуле, нажмите кнопку на панели инструментов зависимостей. Если результат формулы в ячейке используется в других формулах, вы легко можете это проверить. Выделите ячейку с формулой и нажмите кнопку. Будут нарисованы стрелки к другим ячейкам, в которых используется результат данной ячейки. Если таких ячеек нет, будет открыт диалог с сообщением о том, что данных связей нет.

Ключевые слова

Табличный процессор, вертикальная линейка, горизонтальная линейка, координаты ячейки, строка наименования, строка основного меню, строка формул, строка состояния активной ячейки, категория, функция, категории функций, вид функций, формула, диаграмма, редактирование таблицы, редактирование текста, виды диаграмм, шаги, лист, переименование листа, диалоговое окно, контекстное меню, редактирование, логические выражения.

Вопросы для самопроверки:

1. Как вставляются, переименовываются листы?
2. Перечислите элементы экрана табличного процессора MS Excel?
3. Как создаются, редактируются сложные таблицы?
4. Как производятся вычисления в табличном процессоре MS Excel?
5. Что такое редактор формул и как им пользоваться?
6. Сколько типов категорий функции вы знаете?
7. Как тип категории функции используется при расчете задач с условиями?
8. Как решаются задачи фильтрации, группировки, сортировки?
9. Какие команды используются при решении оптимизационных задач?
10. Какие дополнительные подпрограммы табличного процессора MS Excel вы знаете?

Лекция 5. БАЗА ДАННЫХ. СИСТЕМЫ УПРАВЛЕНИЯ БАЗЫ ДАННЫХ (СУБД)

5.1. Основные понятия базы данных. Проектирование БД

База данных (БД) – хранилище данных, относящихся к определенной предметной области, которое обеспечивает реализацию приложений (задач и запросов). БД находится под управлением специализированного программного средства – системы управления базами данных (СУБД).

Применение БД в информационных системах позволяет:

- ✓ легко наращивать функции информационной системы, используя ядро – данные БД;
- ✓ моделировать функции управления предметной области с помощью информационных моделей;
- ✓ выполнять анализ информации для принятия управленческих решений;
- ✓ повысить качество хранимых данных за счет полноты, согласованности и целостности, контроля достоверности вводимой информации, обеспечения санкционированного доступа к БД;
- ✓ разделить «логический» и «физический» уровни представления данных и сделать относительно независимыми данные от обрабатывающих их программ;
- ✓ исключить в прикладных программах функции логики обработки данных;
- ✓ стандартизировать основанные процедуры работы с данными (ввод и редактирование данных в БД, поиск и публикация данных, формирование отчетов, графическое представление и публикация данных БД на сетевом сервере или в сети Интернет и т. п.);
- ✓ снизить трудозатраты на поддержку БД, уменьшить расход вычислительных ресурсов, в первую очередь внешней памяти, для хранения и эксплуатации БД и т.п.

Для БД характерно многоуровневое представление и описание так называемых моделей данных, для создания которых используется теория БД и методики проектирования.

БД содержит исходные данные для реализации приложений (задач и запросов) предметной области. В БД могут накапливаться результаты решения задач, которые используются в качестве входной информации для решения других задач. Основной проблемой проектирования БД является

определение состава и структуры данных предметной области, достаточных для реализации перечня приложений.

БД рассматривается как совокупность моделей данных, описывающих логическую структуру и физическую организацию данных на машинных носителях, находящихся под управлением СУБД. При проектировании БД используется «многоуровневый» подход к представлению данных.

Типовые проектные работы по созданию БД под управлением СУБД включают:

1. Построение информационно-логической модели (ИЛМ) предметной области;
2. Выбор СУБД;
3. Проектирование логической структуры БД;
4. Проектирование организации БД на машинных носителях;
5. Разработку процедуры администрирования БД;
6. Разработку информационных технологий работы с БД приложений.

Формализованное описание данных предметной области, являющееся основой для проектирования логической структуры БД, носит название информационно-логической модели предметной области (ИЛМ). Роль ИЛМ в проектировании структуры БД трудно переоценить. Неправильное понимание данных и информационных потребностей приложений предметной области приводит к ошибкам в структуре БД. В современных методиках проектирования БД делается существенный акцент на информационном анализе предметной области.

На эффективность реализации БД и информационной системы оказывает решающее влияние выбор СУБД. На основании ИЛМ для выбранной СУБД строится комплекс взаимосвязанных моделей БД:

1. *Концептуальная модель*, соответствующая интегрированному логическому представлению БД;
2. *Внутренняя модель*, описывающая хранение БД на машинных носителях с использованием методов организации и доступа данных;

3. *Внешние модели*, поддерживаемые для защиты и санкционированного доступа к БД различных приложений.

БД является важнейшим стратегическим ресурсом информационной системы. На его создание расходуются значительные средства. Наличие БД, отвечающей требованиям полноты, актуальности, достоверности, целостности и т. п., является основой для эффективной реализации приложений. В информационных системах осуществляется системное администрирование БД с помощью сервисных средств СУБД (обеспечение санкционированного доступа к БД, страховое копирование БД, восстановление БД и др.)

Современный подход к разработке информационных систем основан на рассмотрении предметной области как совокупности бизнес-процессов, для реализации которых необходима информация определенного состава, структуры и уровня качества. Уровень качества информации – это ее полнота, достоверность, актуальность, точность и своевременность. Эффективность бизнес-процессов зависит от технологий сбора, регистрации, передачи, хранения, обработки и выдачи информации.

Информационный анализ предметной области для проектирования БД предполагает создание информационно-логической модели (ИЛМ) для проектирования структуры БД.

5.2. Этапы разработки информационно-логической модели (ИЛМ)

Для различных подходов выделяются этапы разработки ИЛМ предметной области:

1. Определение предметной области.

Предметная область может быть ограничена отдельным приложением или комплексом взаимосвязанных приложений. Часто предметная область ограничена рамками функциональной подсистемы управления (бухгалтерский учет, производственный менеджмент, сбыт и реализация продукции и т. п.). В больших информационных системах корпоративного типа функции управления тесно интегрированы, и в первую очередь это

касается учетных функций, являющихся основой бухгалтерского учета. Учетные функции представлены во всех функциональных подсистемах.

Для разработки ИЛМ выполняется формализованная постановка приложения (задач, запросов). Традиционно постановка комплекса задач конкретизирует:

- ✓ организационно-экономическую сущность приложения, его место в функциональной системе управления, взаимодействие с другими приложениями;
- ✓ цели, которые должны быть достигнуты в связи с переходом к использованию БД;
- ✓ требования к информационным технологиям сбора, передачи, обработки и выдачи информации;
- ✓ алгоритм обработки информации приложений;
- ✓ временной регламент (периодичность, момент времени) реализации приложения;
- ✓ условия и место выполнения приложения – структурное подразделение, пользователь, особенности информационных технологий (диалоговый режим, удаленный доступ, выход в Интернет и т.п.)

2. Анализ состава и содержание входной информации для приложений.

Для каждого приложения определяется состав входной информации в виде перечня форм входных документов, массивов информации на машинном носителе. Ключевым моментом является обеспечение полноты и комплектности входной информации, установление требуемого уровня достоверности, точности, оперативности получения, актуальности входной информации применительно к каждому источнику.

Входная информация обычно классифицируется на нормативно-справочную и оперативную (учетную) информацию, а также на информацию, представленную в виде первичных документов, в виде массивов информации на машинном носителе (БД или файлы обмена).

3. Анализ выходной информации приложений.

Для каждого приложения устанавливается состав и структура данных формируемых выходных сообщений, форма представления, требования к уровню достоверности и точности значений.

Выходная информация обычно классифицируется на исходящую (выдаваемую во вне информационной системы) и внутреннюю (остающуюся в информационной системе). С другой стороны, выходная информация классифицируется на хранимую в БД и выводимую за ее пределы - на экран, печатные документы, файлы обмена. В связи с внедрением сетевых технологий выделяют выходную информацию, публикуемую в сети Интернет.

4. Изучение системы документации предметной области.

Изучение системы документации предполагает:

- ✓ создание альбома форм документов;
- ✓ анализ форм документов и систем документации (плановые, учетные, организационно-распорядительные и др.), сокращения многообразия форм документов;
- ✓ унификацию структуры данных документов и свойств реквизитов;
- ✓ оценку объема обрабатываемой входной и формируемой выходной информации за определенный интервал времени (месяц, год);
- ✓ построение схем документооборота по каждому документу, структурному подразделению и функции системы управления.

5. Формирование ИО.

Структура данных ИО должна отвечать требованиям третьей нормальной формы реляционной модели данных (выше).

6. Формирование структурных связей ИО.

Структурные связи ИО должны основываться на одно- многозначном отношении их экземпляров и учитывать функциональные связи для реализации приложений (выше).

7. Построение графа ИЛМ.

Для визуального представления структуры данных ИЛМ предметной области строится граф ИЛМ, вершины которого – ИО, а дуги – структурные связи ИО. Все ИО расположены по уровням иерархии согласно одно-многозначному типу отношений экземпляров ИО. Для расположения ИО по иерархии используется матрица смежности (см. далее). При значительном числе ИО ИЛМ подобное графическое представление облегчает понимание структуры данных, обеспечивает плавный переход к логической структуре базы данных.

5.3. Основные понятия СУБД. Создание базы данных с помощью СУБД MS Access

Система управления базами данных Microsoft Access- версия 1.0 была разработана в 1992 году и явилась первой реляционной СУБД для операционной системы Microsoft Windows. В 1995 году версия Access 95 стала первой в мире 32- разрядной реляционной СУБД, а Access 97- первой СУБД, которая использовала информационные сетевые технологии для обмена данными по корпоративной интрасети.

БД под управлением СУБД Access – это файл с расширением MDB, включающий в себя классы объектов:

- ✓ таблицы для хранения данных;
- ✓ формы для ввода и редактирования БД в интерактивном режиме;
- ✓ запросы для обработки таблиц и других запросов;
- ✓ страницы доступа для работы с БД с помощью программ-обозревателей Интернета;
- ✓ макросы для автоматизации выполнения рутинных действий;
- ✓ отчеты для вывода результатов обработки данных .

СУБД MS Access позволяет создавать БД различного объема, с которыми работают в монопольном режиме или режиме коллективного доступа.

Принципиально новыми функциями СУБД MS Access являются;

- ✓ преобразование БД (файлов .MDB) в форматы предыдущих версий;
- ✓ автоматическое изменение имен объектов БД во всех приложениях;
- ✓ условное форматирование, проверка вводимых значений в таблицы БД;
- ✓ вложенные таблицы по типу главная – подчинения в одном окне;
- ✓ изменение макета форм и отчетов (группирование элементов управления);
- ✓ создание статических копий отчетов для их распространения по электронной почте;
- ✓ управление сжатием БД СУБД MS Access при ее сохранении на магнитном диске и др.

СУБД MS Access содержит много новых функций для работы в сети Интернет, это: страницы доступа к данным, в СУБД MS Access вывода таблиц, запросов, отчетов в формате HTML, добавление полей-ссылок на объекты БД и Web-компоненты Microsoft Office 2000, в том числе электронные таблицы, диаграммы, динамические сводные таблицы. В СУБД MS Access 2000 усовершенствована функция создания и перехода по гиперссылкам. СУБД MS Access 2000 поддерживает новый стандарт доступа к данным OLE DB, благодаря которому можно создавать полноценные клиент-серверные приложения.

Клиентская часть приложения. Клиентская часть приложений разрабатывается с использованием средств MS Access, серверная часть – с помощью Microsoft SQL Server.

Создан новый формат хранимых данных – Microsoft Access Project (файлы с расширением.adp) для серверных БД различных СУБД;

- ✓ Microsoft Data Engine (MSDE), включенный в комплект Office ;
- ✓ SQL Server 6.5;
- ✓ SQL Server 7.0;

На сервере находятся данные (таблицы), интерфейс с ними поддерживает клиент на рабочей станции. При этом возможно

преобразование данных и объектов БД (файл формата .MDB) в формат SQL Server (новую или существующую базу данных Microsoft SQL Server версий 6.5 или 7.0) либо в проект Microsoft Access (файл формата.ADP).

Настройка режимов работы СУБД MS Access выполняется с помощью команды меню Сервис- Параметры. Диалоговое окно Параметры содержит вкладки:

- Вид – выполняется настройка внешнего вида экрана, вывод скрытых и системных объектов в окне БД, выбор способа открытия объектов в окне БД;
- Общие – установка границ печатных полей для отчетов, форм и страниц доступа, поддержка авто замены имен и журнала авто замен; указание рабочего каталога для БД , настройка параметров Web-страниц и др.;
- Правка и поиск – выбор варианта поиска и замены по умолчанию, требование подтверждения операций, связанных с изменениями данных БД, и др.;
- Клавиатура – настройка перемещения курсора при работе в полях;
- Режим таблиц – установка форматов символов, сетки, столбцов и др. для конструктора таблиц и запросов;
- Формы и отчеты – выбор шаблона формы и отчета;
- Другие – порядок обновления связей с внешними источниками данных (операции DDE), тип доступа у БД при открытии (монопольный, общий), способ блокировки записей при работе в форме или таблице (все записи или только редактируемая);
- Таблицы и запросы – настройки конструктора таблиц и конструктора запросов, права доступа к таблицам и запросам.

5.4. Таблицы базы данных

СУБД MS Access позволяет создавать БД различного типа:

- новый файл («пустая» БД);

- новый файл на основе другой БД (БД «прототип»);
- страница доступа для публикации данных в Интернете;
- проект для крупномасштабной БД под управлением СУБД Microsoft SQL Server;
- конвертирование внешней базы данных (текстового формата, формата электронной таблицы, реляционной базы данных и т.п.).

Команда меню Файл > Создать выводит диалоговое окно Создание, содержащее вкладки:

- Общие – выбор формата файла БД;
- База данных, файл формата .MDB;
- Страница доступа к данным, файл формата .HTML;
- Проект (существующая база данных), файл формата .ADP для подключения к существующей базе данных на сервере MS SQL Server;
- Проект (новая база данных), файл формата .ADP для создания новой базы данных на сервере MS SQL Server.
- БД –выбор прототипа для создания приложения БД в виде файла формата .MDB.

Прототипы приложений БД имеют разнообразную направленность, например: Основные фонды, Главная книга, Склад, MS SQL Server Мероприятия, Ресурсы, Прием заказов и др.

Команда меню Файл – Открыть позволяет конвертировать данные внешних форматов в формат СУБД MS Access: d BASE, Paradox, Microsoft Exchange, Microsoft Excel и др., доступ к которым основан на драйверах ODBC. Автоматически создается новый файл БД MS Access (.MDB) в той же папке и с тем же именем, что и источник данных. В окне БД указывается связь с каждой таблицей внешней базы данных.

Этот способ соответствует информационной технологии OLE, режим связывания с объектом. В этом случае обеспечена возможность одновременной работы с объектом как в исходной программе, так и в БД MS

Access. Эта же информационная технология реализована с помощью команды меню Файл- Внешние данные – Связь с таблицами.

В открытую БД MS Access можно импортировать внешние данные по команде меню Файл- Внешние данные- Импорт (рис.7.7). Для каждого класса объектов БД указывается состав импортируемых объектов, задаются общие параметры настройки импорта таблиц, запросов, панелей инструментов, схемы данных, спецификаций. Импорт данных в БД MS Access соответствует информационной технологии OLE –внедрение объекта, связь с источником объекта не поддерживается.

Обратное действие – передача объектов БД MS Access во внешние приложения выполняется с помощью команды меню Файл –Экспорт. Выделенный объект (таблица, запрос, форма и т. п.) сохранить под другим именем в текущей БД или во внешнем файле определенного формата.

Команда меню Сервис>Служебные программы>Преобразовать БД к предыдущей версии позволяет преобразовать файл БД формата СУБД MS Access 2000 в формат предыдущей версии.

Удаление объектов БД выполняется в окне БД стандартным образом:

1. Выделить объект, щелкнув на нем курсором.
2. Выбрать команду Правка>Удалить (или нажать клавишу Del).

Удаление БД выполняется как удаление файла в программе операционной системы MS Windows для работы с файлами: Проводник, Диспетчер файлов, Мой компьютер.

Основой БД являются таблицы, создаваемые с помощью команды меню Вставка>Таблица или при нажатии кнопки Создать на вкладке Таблицы. На экран выводится окно, в котором указывается способ создания таблицы:

- Режим таблицы -таблица стандартной структуры, поля которой имеют стандартные имена –Поле1-Поле10. Стандартные заголовки полей можно изменить в любой момент времени.

- Конструктор -создание структуры таблицы, описание свойств полей таблицы.

- Мастер таблиц – выбор таблицы-прототипа для создания новой таблицы, отбор включаемых полей.

- Импорт таблиц – заимствование таблицы из внешнего источника данных (информационная технология внедрения с объектом).

Связь с таблицами – установление связи с таблицей внешнего источника данных (информационная технология связывания с объектом).

Структуру записи таблицы можно редактировать в конструкторе таблиц, который вызывается для выделенной таблицы при нажатии кнопки Конструктор на вкладке Таблицы. Конструктор таблиц выводит панель инструментов Конструктор таблиц и изменяет состав команд главного меню. Команда меню Вид > Свойства выводит окно для задания свойств таблицы:

- Условие на значение –проверка вводимых записей таблицы;
- Фильтр –условие отбора записей таблицы для отображения;
- Порядок сортировки –выбор поля для упорядочивания записей таблицы;
- Имя подтаблицы и соответствующие связи, подчиненные и основные поля и.т.д.

Поля таблицы и их свойства

Состав полей записи таблицы и их свойства определяются в конструкторе таблицы на специальном бланке, содержащем столбцы: Имя поля, тип данных, описание. Существуют ограничения для имен полей и элементов управления (поле со списком, кнопка, флажок и т.п.):

- длина имени –максимум 64 символа;
- любая комбинация букв, цифр, пробелов и специальных символов за исключением точки (.), восклицательного знака (!), надстрочного символа (^) и квадратных скобок ([]);
- не начинается с символа пробел;
- не включает управляющие символы (с кодами ASCII от 0 до 31);
- не включает прямые кавычки (“);

- не совпадает с именами свойств или элементов управления.

Для поля выбирается тип данных (табл. 5.1). Описание поля содержит произвольный текст, поясняющий структуру таблицы.

Таблица 5.1.

Типы полей таблиц СУБД MS Access

Тип данных	Содержит	Размер
Текстовый	Произвольные символы	До 255 символов, пустые значения не хранятся
Поле МЕМО	Неструктурированный текст	До 65 535 (64K) символов
Числовой	Любые числовые данные для математических вычислений, за исключением денежных операций	1, 2, 4 или 8 байт (16 байт для реплики)
Дата/время	Даты и время (100–9999 годы)	8 байт
Денежный	Значения валют, предотвращает округления во время вычислений, до 15 символов в целой части числа и 4 в дробной	8 байт
Счетчик	Последовательные с шагом 1 или случайные числа при добавлении записи (не может обновляться, повторно номер не используется)	4 байт
Логический	Значения «да/нет», «Истина /Ложь», «Включено/ Выключено»	1 байт
Объекты OLE	Объекты, использующие протокол OLE и технологию связывания и внедрения	До 1 Гбайт (ограничено объемом диска)
Гиперссылки	Содержит как минимум 3 части : Текст	Каждая часть

	для показа ссылки; Адрес –путь к файлу (UNC путь) или к странице (URL); Под адрес –для ориентации внутри файла или страницы	гиперссылки содержит до 2048 символов
Мастер подстановок	Выбор значения из другой таблицы или из списка значений, используя поле со списком	Размер ключевого поля, обычно 4 байта

Если тип данных – текстовый, размер поля устанавливается в диапазоне 1-255 символов. Для типа данных – счетчик размер поля определяется как Длинное целое или Код реплики. Для числового типа данных допустимы различные размеры поля. Размер поля должен быть минимально необходимым. Для числовых полей можно указать формат внешнего представления данных.

Для полей типа текстовый, поле MEMO и гиперссылки используется кодировка Unicode (Юникод), в которой каждый символ представлен 2 байтами (в ASCII-1 символ представляется 1 байтом). С помощью новой кодировки поддерживается до 65 536 символов (вместо 256 в ASCII). Свойство Сжатие Юникод –Да для этих типов полей означает, что все символы, первый байт которых равен 0, будут сжиматься при сохранении и автоматически восстанавливаться при выборке.

Для полей типа дата/время возможен любой формат представления значения (краткий, полный), а также использование маски ввода для представления даты и времени – выбор разделителей.

Для денежного типа данных указывается валюта и количество десятичных знаков для внешнего представления.

Для поля таблицы может задаваться подпись, которая автоматически выводится как название поля таблицы в формах и отчетах. Для полей определенных типов данных можно указать значение по умолчанию. В этом

случае происходит автоматическое заполнение полей в новых записях значениями по умолчанию.

5.5. Формы базы данных

Формы предназначены для ввода и редактирования данных таблиц, просмотра результатов запросов выборки, создания пользовательского меню, диаграмм.

С формой для ввода и редактирования данных таблиц можно работать в следующих режимах:

- в конструкторе форм – макет экранной формы;
- в режиме формы – экранная форма для ввода и редактирования данных;
- в режиме таблицы – обычная таблица.

Новые формы могут создаваться на вкладке с помощью кнопки Создать или по команде меню Вставка>Форма несколькими способами:

- с использованием Конструктора форм – наиболее трудоемкий способ, обеспечивает создание и редактирование форм различной сложности;
- с помощью Мастера форм – специальной программы для автоматизации создания форм:
 - с помощью Авто формы (в столбец, ленточная, табличная) – быстрое создания простейших однотабличных форм;
 - используя Диаграмму – настраивание диаграмм для числовых данных таблиц;
 - с помощью Сводной таблицы – построение итоговой экранной формы, обеспечивающей формирование и просмотр итогов в электронной таблице Microsoft Excel.

Макет формы.

Макет формы строится из элементов управления, расположенных на Панели элементов. Панель выводится на экран с помощью команды меню Вид>Панель элементов.

Первый ряд кнопок:

- Выбор элемента;
- Мастера (для определенных элементов управления).

Второй ряд кнопок:

- Надпись (произвольный текст);
- Поле (таблицы, запроса или вычисляемое поле);
- Группа переключателей (флажков, кнопок, выключателей);
- Выключатель, Переключатель, Флажок (имеют два состояния 0/1, Да/Нет, Вкл/Выкл);
- Поле со списком (выбор значения из списка, ввод нового значения);
- Список (выбор значения из списка);
- Кнопка (связь с макросом, процедурой обработки событий);
- Рисунок;
- Свободная рамка объекта (внедрения объекта);
- Присоединенная рамка объекта (связь с объектом);
- Разрыв страницы;
- Набор вкладок (независимые подчиненные элементы);
- Подчиненная форма;
- Линия;
- Прямоугольник;
- Дополнительные элементы;

Макет формы состоит из разделов. Состав разделов формируется с помощью команд меню Вид>Заголовок/Примечание формы и Вид>Колонтитулы. Основные разделы макета формы;

- Заголовок формы.
- Область данных.
- Примечание формы.
- Колонтитулы.

В Заголовке формы размещается формы, постоянные элементы управления (поля, надписи, графика, рисунки, командные кнопки), присутствующие на экране. В области Примечание формы выводится

итоговая информация (количество записей области данных, итоги по столбцам), кнопки, надписи и т. п. Колонтитулы содержат постоянную информацию для печатной страницы. Формы могут выводиться на печать с сохранением форматных характеристик.

Положение, габариты и внешнее оформление элементов управления формы можно легко изменить. Команда меню Формат конструктора форм позволяет выполнить следующие операции:

- Авто формат –стандартные стили для формы с дополнительной настройкой элементов стиля (шрифт, цвет, границы).

- Условное форматирование –задать условия изменения формата (от 1 до 3 условий) для выделенного элемента управления.

- Задать стандартные свойства.

- Преобразовать элемент в –изменение типа элемента управления, например, поле можно преобразовать в поле со списком, и наоборот, выключатель –в переключатель, флажок и т. д.

- Привязать к сетке –разметка в макете для удобства позиционирования элементов управления.

- Выровнять –выделенную группу элементов управления можно выравнивать относительно друг друга: по левому/правому/верхнему/нижнему краю, по узлам сетки.

- Размер –для выделенной группы элементов можно задать размер по указанному элементу или по размеру данных.

- Интервал по горизонтали/Интервал по вертикали –для группы выделенных элементов управления можно сделать равным, увеличить или уменьшить интервал.

- Группировать, Разгруппировать –выделенные элементы управления можно объединять в одну группу, разбить группу на составляющие.

- На передний план, На задний план –элементы управления формы могут размещаться в нескольких слоях (например, рисунки, диаграммы, надписи, поля и т. д. могут накладываться друг на друга).

5.6. Запросы базы данных

Запросы СУБД MS Access являются основной для алгоритмической обработки данных БД, используются для формирования подсхемы данных, обеспечивающих создание многотабличных форм и отчетов. Запросы используются и для задания условий фильтрации записей таблиц, формирования страниц доступа. Исходными данными для запроса являются таблицы и/или другие запросы. Имена запросов не должны совпадать с именами таблиц БД.

Запросы классифицируются различным образом:

1. По числу обрабатываемых таблиц –запросы делятся на однотобличные и многотабличные (в последнем случае –все таблицы/запросы должны быть связанными).

2. По типу алгоритмов обработки различают запросы:

- выборки –результат запроса отображается только на экране, может использоваться для создания форм, отчетов или страниц доступа, а также других запросов;

- перекрестный –создание свободной таблицы, содержащей групповые итоги;

- на создание таблицы –автоматическое формирование структуры записи новой таблицы и загрузка;

- на добавление ввод новых записей, являющихся результатом выполнения запроса;

- на удаление –удаление записей из таблицы.

3. По типу языка запросов.

Используются два типа языка запросов:

- QBE (Query By Example) –запрос по примеру, построенный на основе реляционного языка запросов графического типа;

- SQL (Structured Query Language) –реляционно- полный язык запросов.

Как правило, запросы создаются с использованием средств языка QBE. Запросы QBE автоматически транслируется на язык SQL.

4. По стабильности условий фильтрации записей в запросах различают:

- статические запросы с неизменными условиями;
- динамические (гибкие) запросы с изменяемыми условиями.

Запросы создаются при выполнении команды меню Вставка>Запрос или при нажатии кнопки Создать на вкладке Запросы. Построение запроса выполняется в конструкторе запросов, который использует панель инструментов Конструктор запросов. Для вывода панели инструментов конструктора запросов выполняется команда меню Вид>Панели инструментов, панель –Конструктор запросов.

Для запроса выбираются источники информации –таблицы и/или другие запросы, устанавливаются связи между ними. Для каждого запроса определяется тип, уточняются условия выполнения запроса, состав полей результирующей таблицы –бланк запроса. Для отображения данных в результирующей таблице используется свойство Вывод на экран. Для выбора таблиц или запросов, используемых в текущем запросе, выполняется команда меню Вид>Отобразить таблицу или Добавить таблицу. Для удаления ненужной таблицы/запроса после выделения таблицы/запроса выполняется команда меню Запрос>Удалить таблицу или нажимается клавиша Del. В многотабличных запросах все исходные таблицы/запросы должны быть связанными между собой по общим ключам связи. С помощью команды меню Вид>Параметры объединения можно изменить тип объединения.

В бланк запроса с помощью команды меню Вид>Имена таблиц можно включить строку Имя таблицы, которая идентифицирует источник полей. С помощью команды меню Вставка>Столбцы добавляются новые в результирующую таблицу по месту установки курсора.

СУБД MS Access позволяет использовать построитель выражения для создания вычисляемых полей в результирующей таблице в условий отбора записей. Условия отбора состоят из произвольного числа строк. С помощью команды меню Вставка>Строки добавляются строки условий отбора. Запросы могут использовать параметры, которые требуют определения типа

данных с помощью команды меню Запрос>Параметры. По умолчанию в конструкторе создается запрос выборки. Для изменения типа запроса выполняется соответствующая команда меню: Запрос>Выборки/Перекрестный/Создание таблиц/Обновление/Добавление/ Удаление.

Для формирования групповых итогов служит команда меню Вид>Групповые операции, которая обеспечивает выбор статической функции для вычисления итогов. Функция указывается для каждого столбца в строке Групповая операция бланка запроса. В начале результирующей таблицы размещаются поля группировки, их состав и порядок следования (слева направо) определяют виды итогов. Для запуска запроса на выполнения из режима конструктора запроса выполняется команда меню Запрос>Запуск или нажимается кнопка со знаком!. Для запуска закрытого запроса требуются на вкладке Запросы выделить объект БД и нажать кнопку Открыть.

Для формирования специальных запросов на языке SQL выполняется команда меню Запрос>Запрос SQL. На языке SQL разрабатываются запросы следующих типов:

- к серверу –передача команд на сервер БД, который поддерживает драйверы ODBC для обмена данными;
- управляющие – запросы на создание и реструктуризацию таблиц, создание/удаление индексов таблиц;
- объединение таблиц – слияние данных нескольких таблиц;
- подчиненный запрос – вложенный запрос в макет запроса.

Для изменения свойств запроса служит команда меню Вид>Свойства, в диалоговом окне Свойства запроса указывается:

- вывод всех полей результирующей таблицы (игнорирование свойства Вывод на экран);
- ограничение на число или процент выводимых записей;
- проверка уникальности выводимых записей;
- связь с внешней БД и др.

С запросом можно работать в трех режимах:

- Конструктор – работа с макетом запроса;
- Режим SQL – запрос на языке SQL;
- Режим таблицы – табличное представление.

Запросы на выборку обеспечивает формирование результирующей информации на экране в момент выполнения запроса. В БД сохраняется лишь схема (макет) запроса, а не сам результат. Результирующая таблица содержит поля таблиц или запросов, вычисляемые поля.

В запросе участвуют записи, удовлетворяющие условиям фильтрации, которые задаются для полей или вычисляемых полей в строках условий в виде:

Оператор сравнения Значение.

Несколько условий для одного и того же поля можно записать как одно условие с помощью логических связок AND, OR, NOT. Условия фильтрации можно задавать непосредственно в момент выполнения запроса – так называемый «гибкий» запрос или параметрический запрос. Результат запроса можно сгруппировать по указанным полям для вычисления статических итоговых функций: Sum, Count, AVG, Min, Max и др.

5.7. Отчеты базы данных

Отчеты – традиционная форма представления информации для управления. Отчеты выводятся на экран, принтер или в файл для передачи по каналам связи, размещаются на Web-серверах. Отчеты создаются на основании базовых таблиц, запросов на выборку данных и перекрестных запросов.

При построении многотабличных отчетов учитывается схема данных. По аналогии с формами, в многотабличных отчетах определяется иерархия подчинения таблиц друг-другу, выделяется главный отчет для одной таблицы и подчиненные (связанные) с ним отчеты для других таблиц.

Отчеты, как и формы, содержат различные элементы управления, которые создаются с помощью панели элементов – команда меню

Вид>Панель элементов в конструкторе отчета. Макет отчета содержит стандартные части:

- верхний и нижний колонтитулы – постоянная информация, выводимая на каждой печатной странице отчета («шапка» таблицы, постоянный текст, служебные поля типа номер страницы, дата и т. п.);
- область данных основная информация для детальной строки отчета, содержит поля источника отчета и другие элементы управления.

Ключевые слова

База данных (БД), предметная область, система управления базой данных (СУБД), информационно-логическая модель (ИЛМ), проектирование базы данных, выбор СУБД, структура базы данных, информационный объект (ИО), анализ, запрос базы данных, таблицы хранения базы данных, форма БД, запросы, страницы доступа, отчеты для вывода результатов, электронные таблицы, команды, импортировать внешние данные, поля таблицы.

Вопросы для самопроверки:

1. Что такое база данных?
2. Для чего применяют базы данных?
3. Какие модели базы данных существуют?
4. Какие этапы разработки ИЛМ существуют?
5. Что такое определение предметной области?
6. Что такое анализ входной и выходной области?
7. Перечислите функции СУБД MS Access.
8. Перечислите элементы экрана MS Access.
9. С помощью каких команд создается, удаляется база данных в MS Access ?
10. Как создается табличный вид БД?

11. Какие формы БД существуют и как они создаются?
12. Что такое запрос, какие виды запросов существуют?
13. С помощью каких команд создается запрос?
14. Что такое отчеты БД?

Лекция 6. УНИФИЦИРОВАННАЯ ПРОГРАММА РАСЧЕТА ЗАГРЯЗНЕНИЯ АТМОСФЕРЫ «ЭКО ЦЕНТР». ПРИНЦИП РАБОТЫ

6.1. Общие понятия о программных средствах «ЭКО центр»

В наше время очень популярна в мире высоких технологий экологического проектирования унифицированная программа расчета загрязнения атмосферы «Эколог», предназначенная для расчета концентраций загрязняющих веществ в атмосфере. Нами рассмотрена бесплатная программа общедоступная по ссылке <http://www.eco-s.ru/products/emission>, аналогичная «Экологу», УПРЗА «ЭКО центр».

Опишем основные функциональные возможности данной программы и приведем стандартные подходы к решению базовых экологических задач.

Программные средства «ЭКО центр» имея в своем арсенале широчайший набор средств для редактирования и обработки данных, программа ориентирована на создание профессионально оформленных отчетов. Для решения простых задач интерфейс программы сделан контекстно-зависимым, т.е. часть инструментов становятся доступными пользователю по мере усложнения задачи. В результате простые задачи решаются максимально быстро. Использование цветовых схем, отражающих состояние каждой ячейки с данными, позволяет сделать логичным и предсказуемым поведение программы, которая стремится, опираясь на логику нормативных документов, автоматически заполнить и рассчитать все, что только возможно. На рисунке 6.1 представлено главное окно УПРЗА "ЭКО центр".

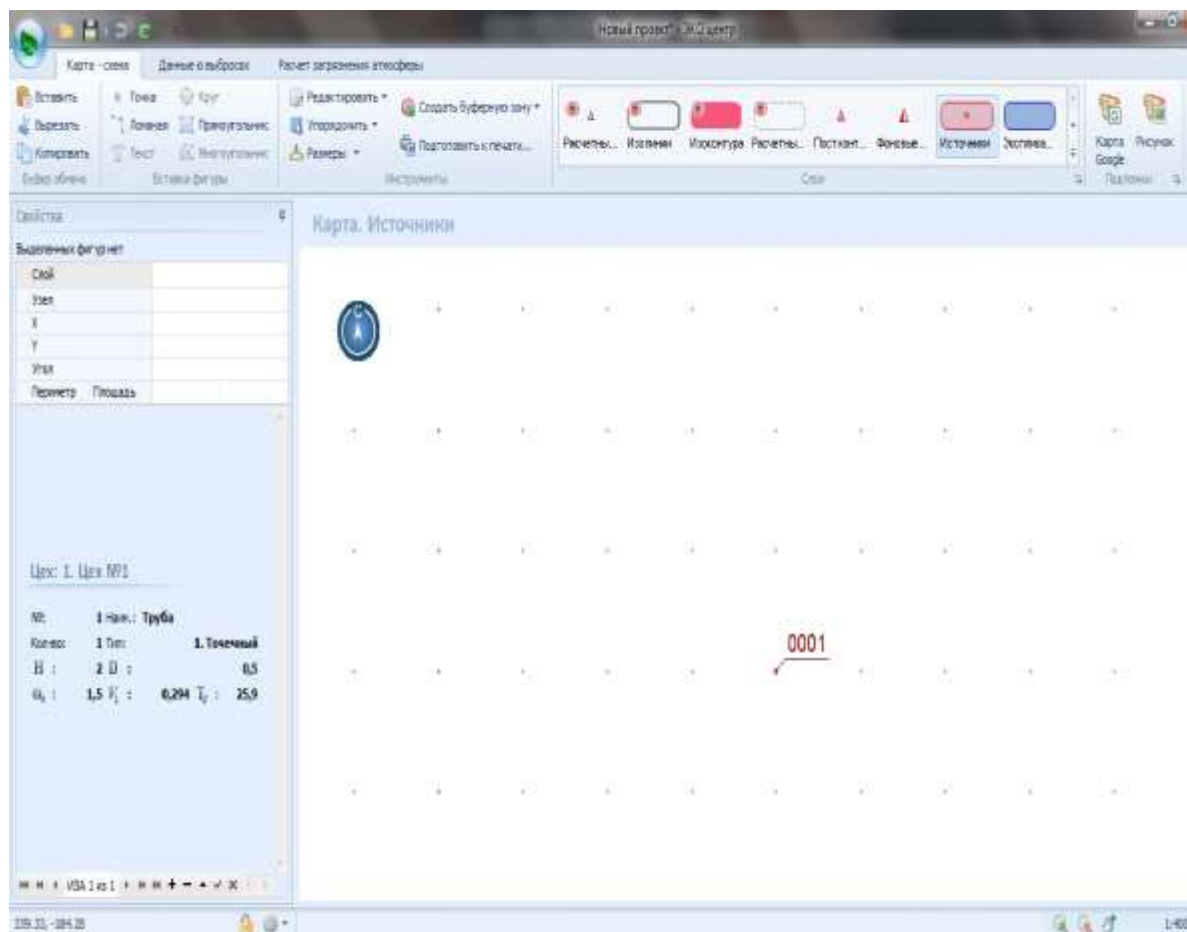


Рис. 6.1. Главное окно УПРЗА "ЭКО центр"

При работе с картой - схемой мы можем визуальнo расставлять все точки, рисовать источники загрязнения и т.д. Кроме того, можно связать с картой - схемой фрагмент карты Google. Как и в случае со многими другими УПРЗА, обычные рисунки принимаются в формате Bmp, Jpg, Jpeg, Gif или Png.

Безусловно, у нас есть возможность также передать в "ЭКО центр" уже имеющиеся карты-схемы, т.к. УПРЗА "ЭКО центр" поддерживает импорт и экспорт карт - схем следующих форматов: DXF, Mif, Shp, Wlg, Int. А все параметры визуальнo расставленных точек и объектов можно отредактировать. На рисунке 6.2 показано редактирование расчетной точки.

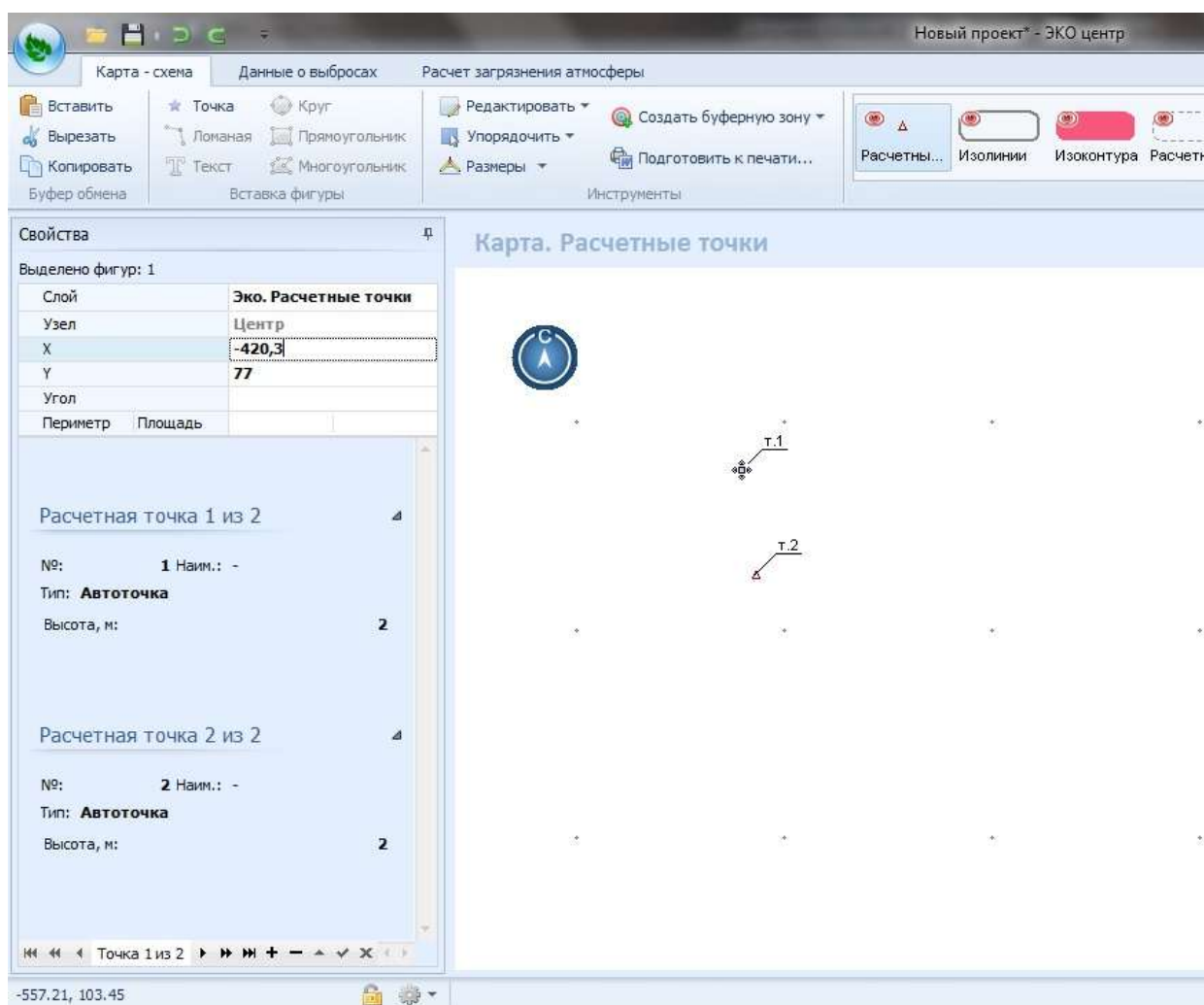


Рис. 6.2. Редактирование расчетной точки

Следующий момент работы с программой - ввод данных об источниках загрязнения атмосферы. Здесь (на вкладке "Данные о выбросах") приходится разбираться с взаимосвязями таблиц, в чем помогает руководство пользователя видео-урок на видеохостинге <http://www.youtube.com/>.

Довольно интересно и практично сделан в УПРЗА "ЭКО центр" диалог создания нового объекта (рис. 6.3).

Объект (предприятие)

Данные по объекту

Номер: 1 Наименование: Объект №1

Адрес:

ОГРН: ИНН:

Примечание:

Город (населенный пункт):

Метеоданные

Коэффициент стратификации	180
Коэффициент рельефа	1
Максимальная температура	25,9
Минимальная температура	-9,8
Скорость ветра (u*)	8
Среднегодовая роза ветров, %	
С	12
СВ	13
В	11
ЮВ	14
Ю	10
ЮЗ	13
З	15
СЗ	12

Объект 1 из 2

Заккрыть

Рис. 6.3. Создание нового объекта

Удобство также добавляет и встроенный справочник по розам ветров и метеорологическим параметрам (рис. 6.4).

Выбор города из справочника метеоданных

Код	Наименование	A	η	T	T	u _ж	Роза Ветров
4401	Красноярск	200	1	24,3	-18,2	-	2, 4, 4, 2, 17, 51, 16, 4
42401	Липецк	160	1	25,9	-10,3	-	12, 10, 8, 13, 11, 15, 15, 16
45	Москва	140	1	23,6	-10,2	8	12, 8, 9, 10, 14, 18, 15, 14
22401	Нижний Новгород	140	1	23,5	-11,8	8	9, 7, 10, 11, 13, 23, 15, 12
32431	Новокузнецк	200	1	27,7	-19,1	-	15, 22, 6, 2, 12, 26, 8, 9
50401	Новосибирск	200	1	24,6	-18,8	-	7, 9, 8, 11, 19, 26, 13, 7
52401	Омск	200	1	25	-19	10	9, 7, 8, 7, 16, 24, 16, 13
53401	Оренбург	180	1	28,5	-14,8	9	14, 11, 20, 8, 6, 16, 15, 10
56401	Пенза	160	1	25,3	-12,2	-	13, 5, 5, 15, 18, 13, 10, 21
57401	Пермь	160	1	23,4	-15,3	-	8, 8, 8, 12, 12, 23, 18, 11
60401	Ростов-на-Дону	200	1	29,1	-5,7	-	8, 14, 31, 8, 4, 10, 17, 8
61401	Рязань	140	1	24,1	-11	-	10, 7, 8, 12, 14, 19, 17, 13
36401	Самара	160	1	25,9	-13,5	-	12, 8, 16, 11, 9, 16, 16, 12

Выбрать Отмена

Рис. 6.4. Справочник по розам ветров и метеорологическим параметрам

Для расчёта приземных концентраций загрязняющих веществ вводится фон, определяется перебор параметров и т.д. (рис. 6.5).

В целом программа достаточно функциональна и имеет современный и удобный интерфейс, даже не смотря на некоторые недочеты в комфорте при

работе с источниками выбросов. Главная ее особенность – это единственная бесплатная УПРЗА, а, следовательно, любая экологическая фирма, если ей будет достаточно функционала программы, может воспользоваться именно УПРЗА «ЭКО центр», таким образом, сэкономя на собственных средствах.

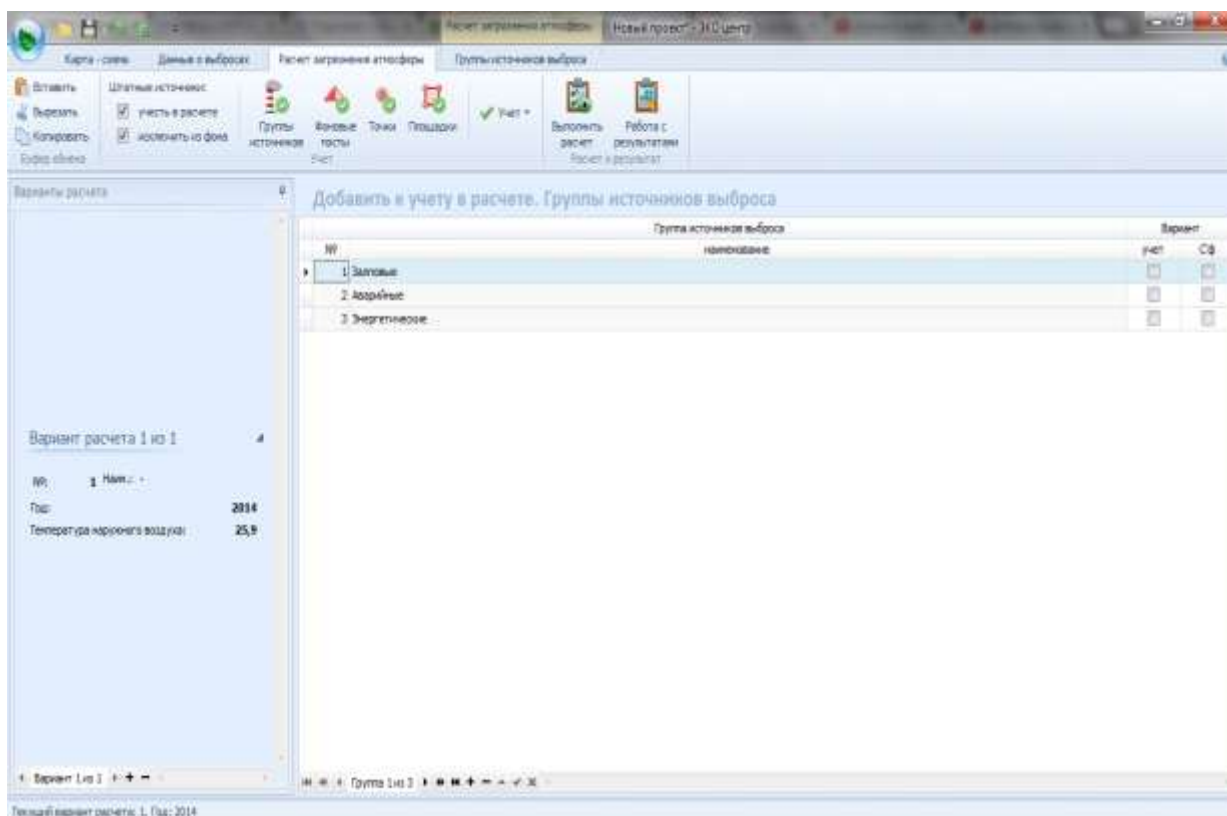


Рис. 6.5. Расчет загрязнения атмосферы

6.2. Создание карты – схемы района размещения предприятия.

Теперь я подробно расскажу о том, как же можно связать карту – схему с фрагментом карты Google.

Создание карты - схемы, в целом, может стать одним из самых трудоемких этапов работы эколога. Однако для успешного выполнения экологических проектов вопрос создания проработанной до мельчайших деталей векторной карты не стоит остро. УПРЗА «ЭКО центр» содержит набор профессиональных инструментов, с помощью которых можно создавать чертеж топоосновы с нуля, либо импортировать его из популярных форматов DXF, Mif/Mid, Shp. Достаточным для достижения оптимального

результата может быть использование наглядного растрового изображения, полученного, например, в результате сканирования бумажного варианта, представляемого предприятием в качестве приложения к техническому заданию или использования изображений различных интернет ресурсов, таких как <http://www.yandex.ru/>,
<https://www.google.ru/>
<http://kosmosnimki.ru/> (рисунки 6.6, 6.7 и 6.8 соответственно).

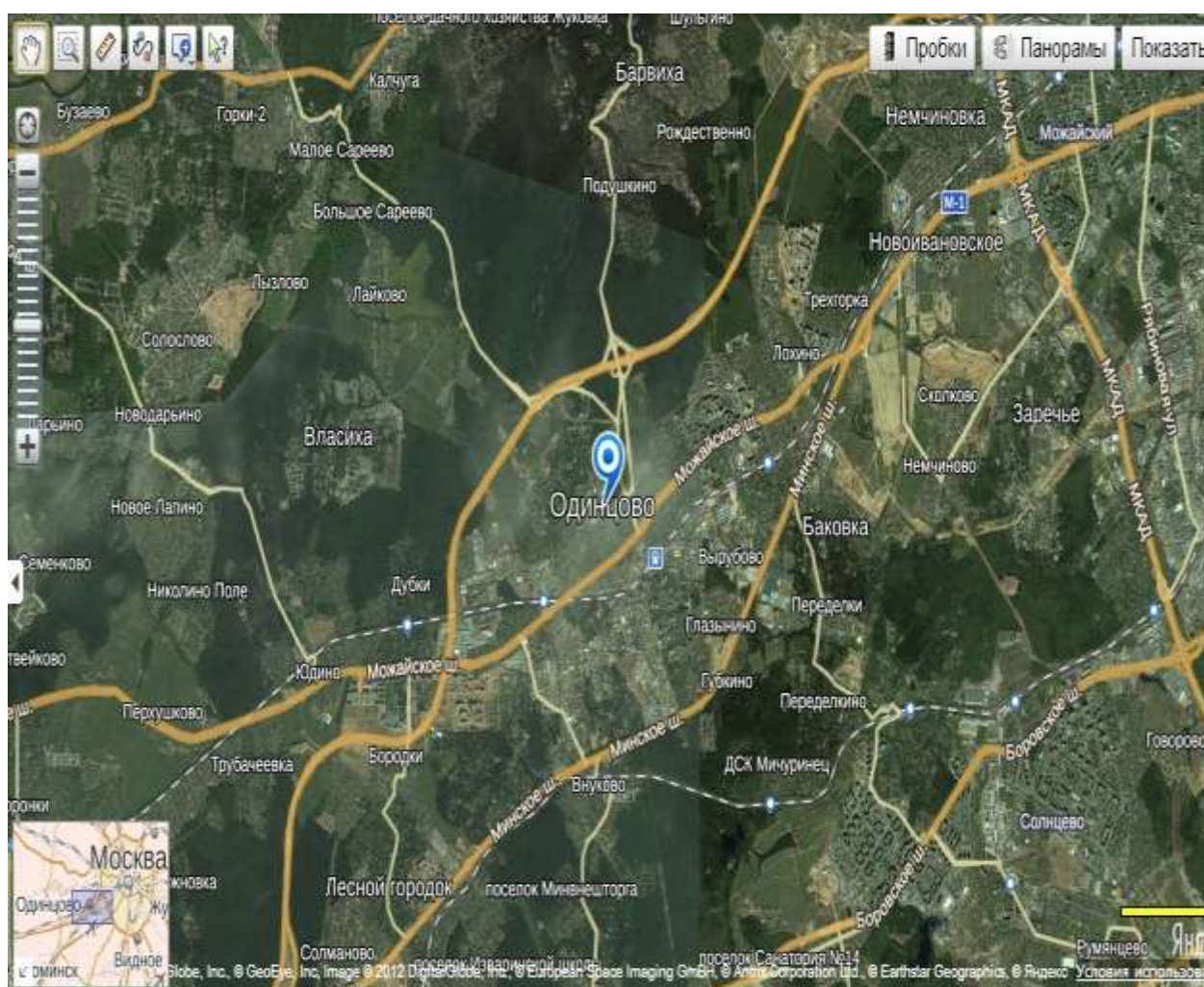


Рис. 6.6. Яндекс карты

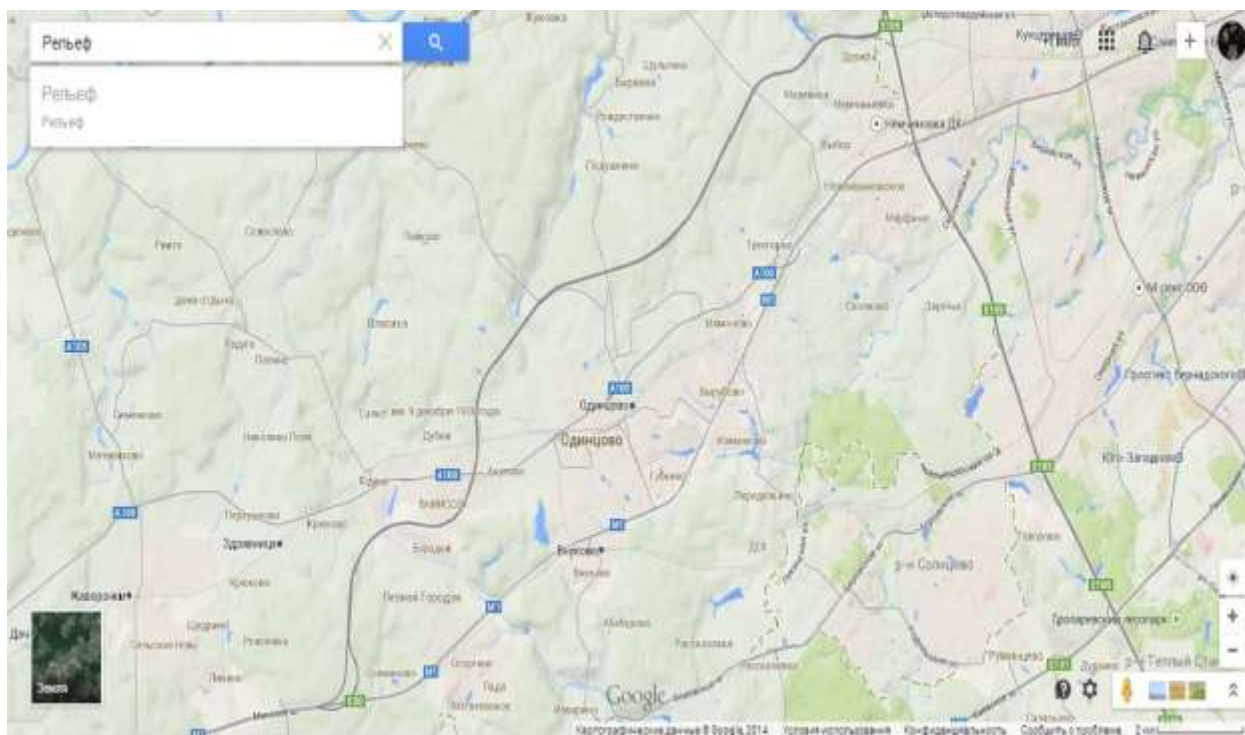


Рис. 6.7. Google карты

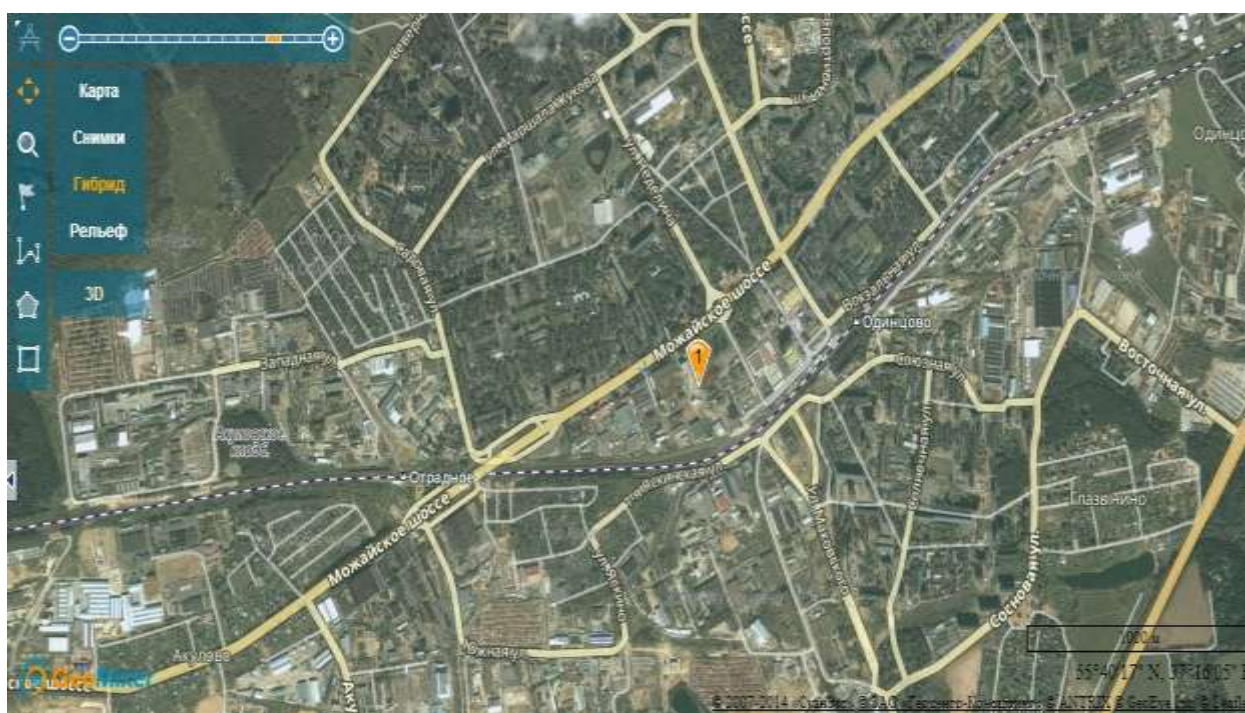


Рис. 6.8. Космоснимки.ру

Для получения изображения с указанных сайтов самым простым способом может быть возможность копирования изображения монитора в буфер обмена с помощью клавиши клавиатуры «prt sc» (print screen) и вставки его в самую простую программу с возможностью редактирования

графики «Paint». Лично я для вставки и кадрирования использовал программу «Adobe Photoshop».

Но наиболее эффективным вариантом может быть использование программы Google Earth Pro (рисунок 6.9), которая своим лицензионным соглашением позволяет использовать детальные снимки поверхности Земли в своих проектах и презентациях. В нашем случае, так как эта программа платная, я оформил заявку на скачивание и использование пробной версии в течение недели по ссылке https://geoauth.google.com/gev0/free_trial.html.



Рис. 6.9. Программа Google Earth Pro

В программе Google Earth Pro мы находим необходимое предприятие. Устанавливаем высоту таким образом, чтобы в рабочей области экрана можно было увидеть исследуемое предприятие и прилегающие к нему окрестности, включающие район санитарно-защитной зоны и ближайшей территории (зоны жилой застройки). Для примера, ввели в поиске ООО «Штарком», компанию по производству и продаже бетона и ЖБИ в Москве и Московской области, расположенную в городском поселении Одинцово (рисунок 6.10).

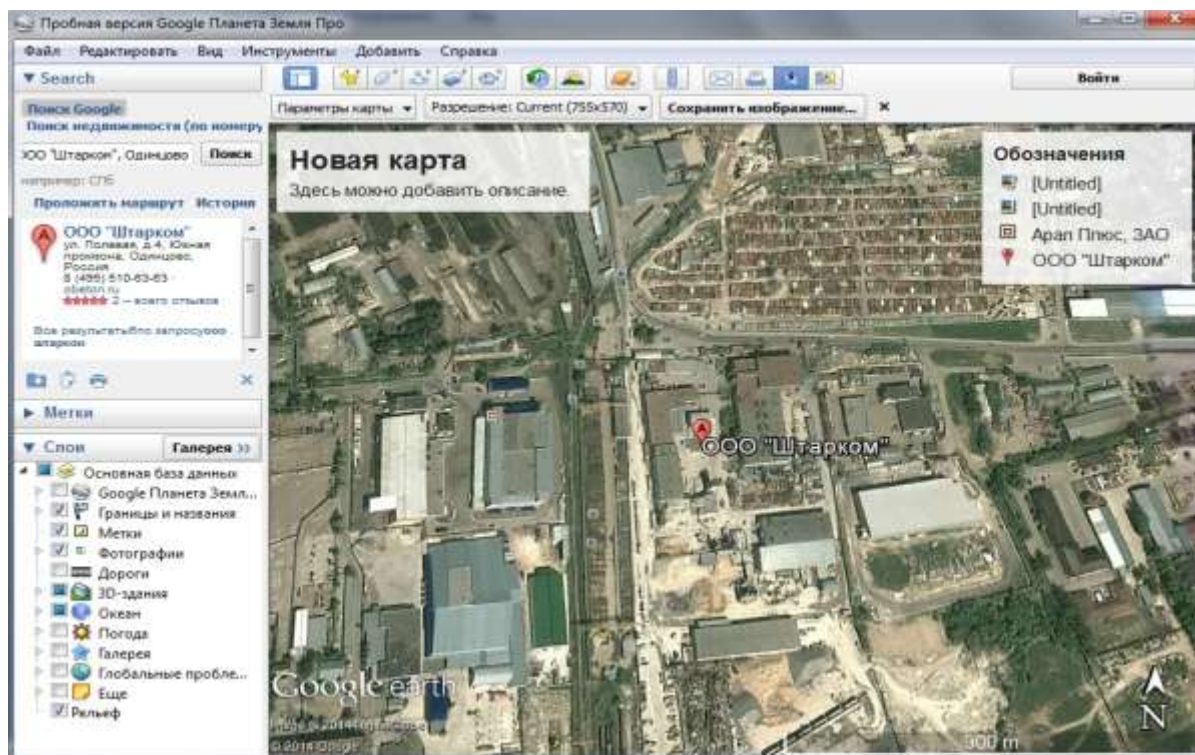


Рис. 6.10. Создание подложки

Сохранив растр, воспользовавшись меню «Файл -> Сохранить -> Сохранить изображение» (Ctrl + Alt + S), командой меню «Добавить -> Метка» (Ctrl + Shift + P), получаю необходимые для привязки растра параметры вида: широту, долготу, диапазон, курс (рисунок 6.11).

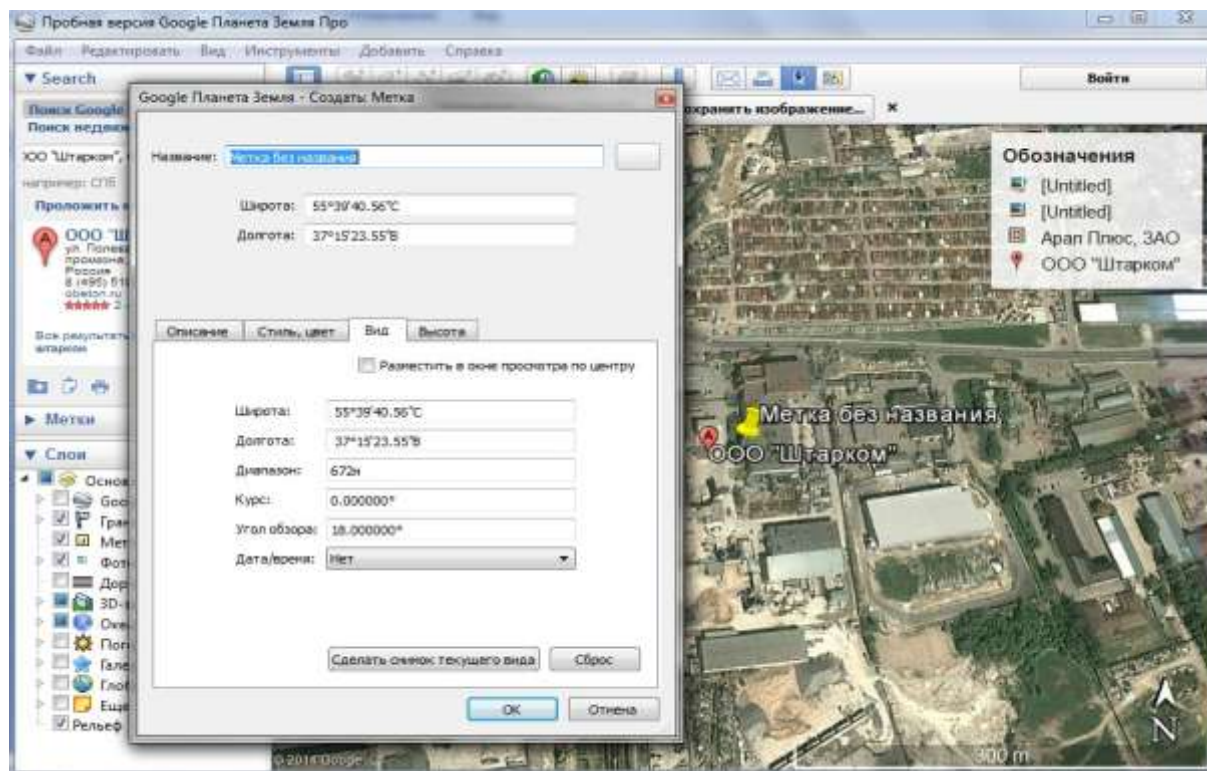


Рис. 6.11. Получение параметров вида

Далее командой «Карта Google» в программе УПРЗА «ЭКО центр» вставляем и привязываем растр с изображением предприятия и прилегающих к нему территорий (рисунок 6.12).

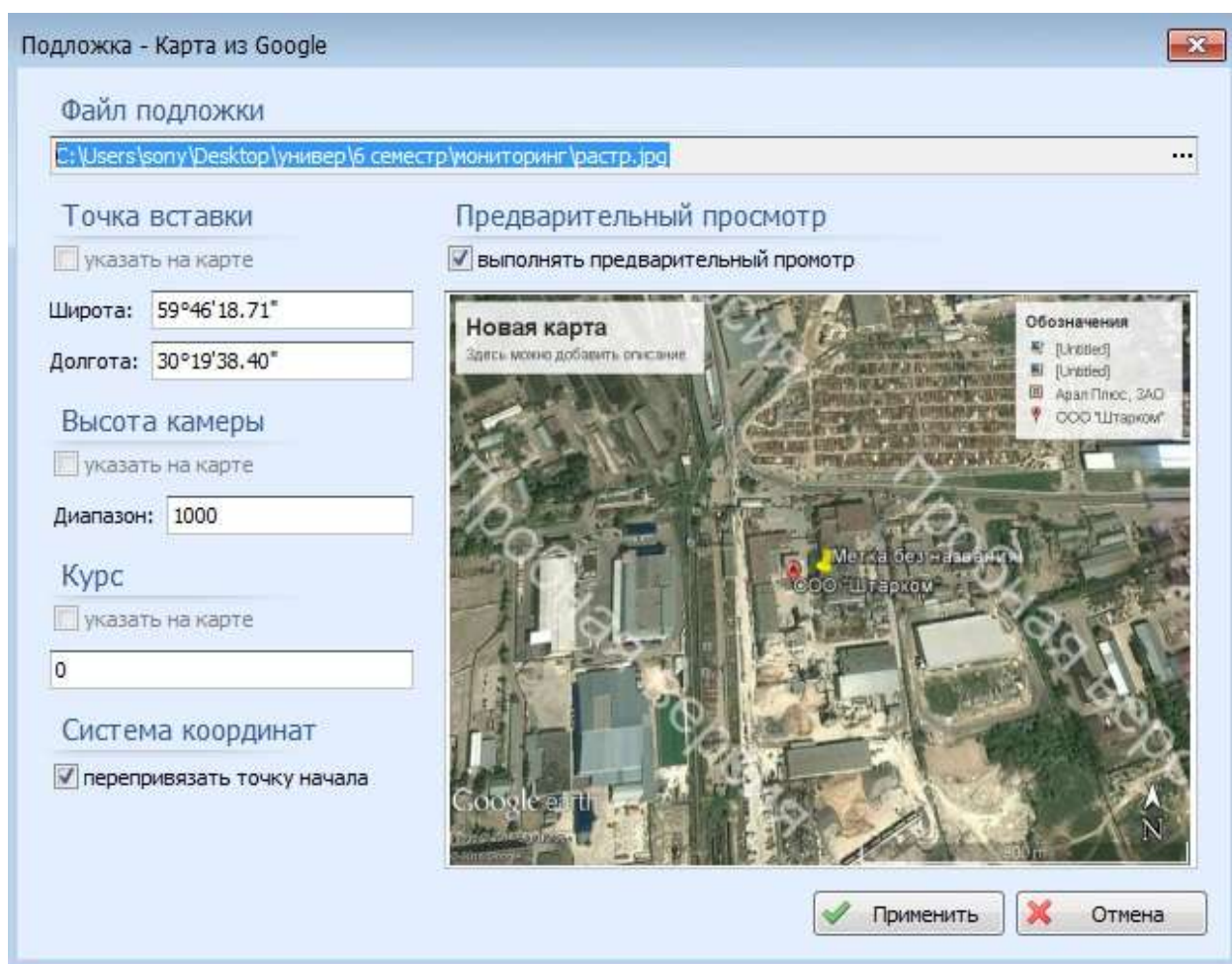


Рис. 6.12. Загрузка подложки в УПРЗА «ЭКО центр»

В галерее слоев выбираем слой «Граница предприятия». Добавляем многоугольник по границе территории предприятия. Командой «Создать буферную зону...» создаем ориентировочную санитарно-защитную зону в соответствии с санитарной классификацией предприятия.

Также выбираем слой «Жилая зона». Добавляем многоугольники, характеризующие местонахождение селитебных территорий (рисунок 6.13).



Рис. 6.13. Карта – схема. Приблизительная карта схема в учебных целях готова.

6.3. Характеристика источников выброса загрязняющих веществ.

Для того чтобы охарактеризовать источники выброса загрязняющих веществ, нужно перейти на вкладку «Данные о выбросах», о чем упоминалось во втором пункте работы. Туда нужно занести данные о структурной схеме предприятия, включающей в себя иерархический набор данных об объекте (предприятии), промышленной площадке (производстве), цехе (участке) (рисунк 6.14).

Объект (предприятие)

Данные по объекту

Номер: 1 Наименование: ООО "Штарком"

Адрес: Южная промзона, ул. Южная д.4

ОГРН: ИНН:

Примечание:

Город (населенный пункт): Одинцово

Метеоданные

Коэффициент стратификации	180
Коэффициент рельефа	1
Максимальная температура	25,9
Минимальная температура	-9,8
Скорость ветра (u^*)	8
Среднегодовая роза ветров, %	
С	12
СВ	13
В	11
ЮВ	14
Ю	10
ЮЗ	13
З	15
СЗ	12

Объект 1 из 1

Заккрыть

Рис. 6.14. Данные об объекте

На дополнительной вкладке «Источники выброса на карте» перемещаем координаты существующего источника (один источник у каждого цеха добавляется программой автоматически) и щелчком мыши указываем место вставки нового точечного источника. Вернувшись на вкладку Данные о выбросах, указываем номер, наименование, высоту и диаметр источников загрязнения атмосферы, а так же параметры газовой воздушной смеси (рисунок 6.15).

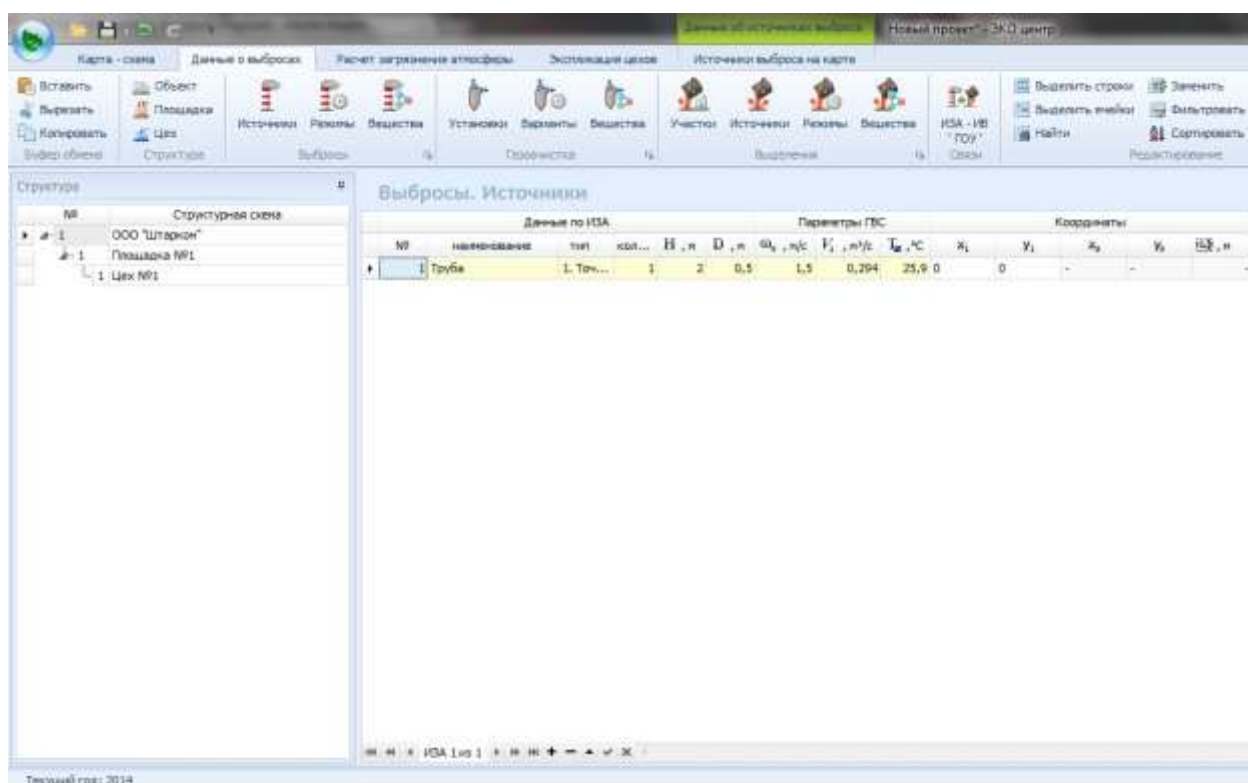


Рис. 6.15. Источники загрязнения

Выбираем работу с таблицей «Выбросы. Вещества». Нужно ввести данные по количественной и качественной характеристике выброса загрязняющих веществ в атмосферу для источника загрязнения или воспользоваться вызовом расчётного окна программы-методики серии «ЭКО центр» (асфальтобетонный завод), скачанной по ссылке <http://www.eco-s.ru/products/emission> вместе с основной программой, что я в учебных целях и делаю (рисунок 6.16).



Рис. 6.16. Характеристика выброса загрязняющих веществ в атмосферу

6.4. Исходные данные для расчёта загрязнения атмосферы.

Далее переходим на вкладку «Расчёт загрязнения атмосферы». Выбираем работу с фоновыми постами. Нужно добавить (можно это сделать по карте, воспользовавшись дополнительной вкладкой) фоновый пост и указать в выпадающей подтаблице вещества, по которым в Справке Росгидромета указаны фоновые концентрации для исследуемого предприятия. Но, к сожалению, справки по фоновым концентрациям примесей выдаются заказчику только *территориальными управлениями Федеральной службы по гидрометеорологии и мониторингу окружающей среды (УГМС)*. В ознакомительных и учебных целях мы приведем изображение из «2006 - 2010 ООО «ЭКОцентр» 5 УПРЗА «ЭКО центр». Быстрый старт. Руководство пользователя» (рисунок 6.17).

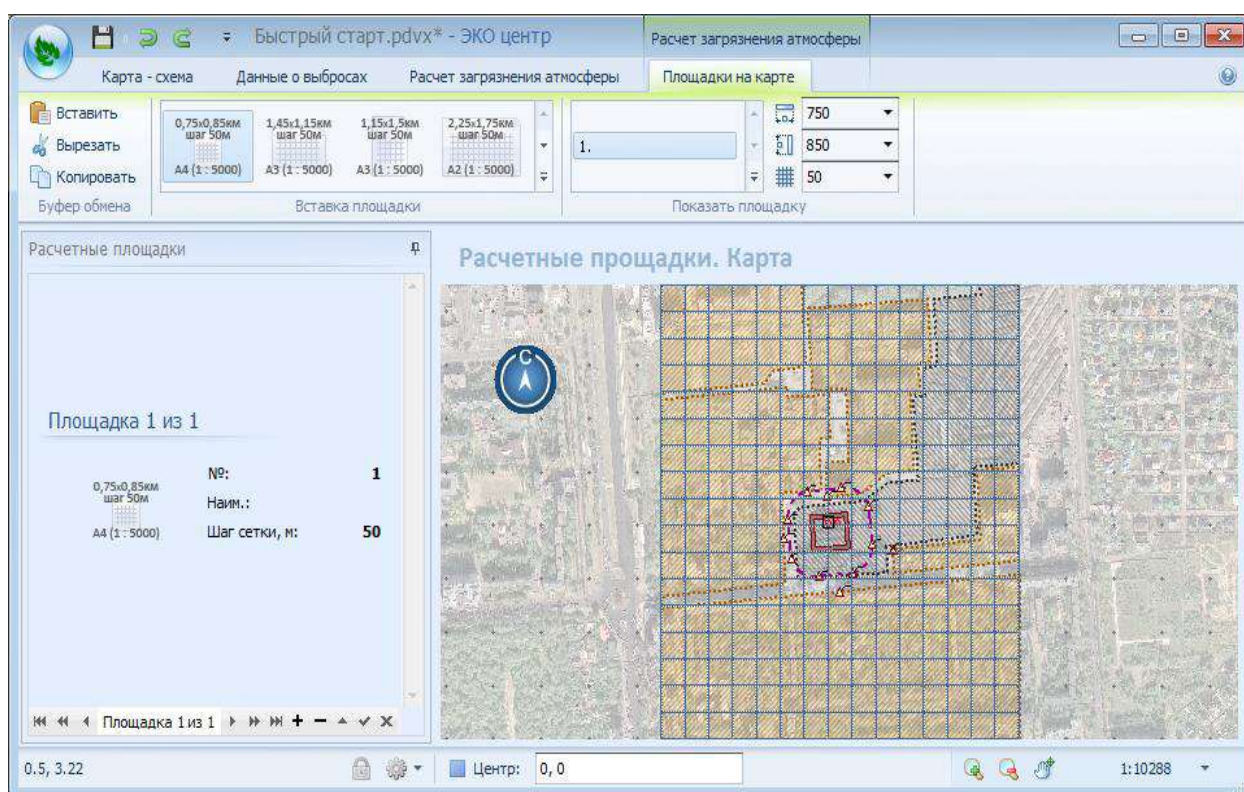


Рис. 6.17. Расчетная площадка

После этого должен быть произведен расчет и анализ всех результатов (команда «Выполнить расчет»). По окончании расчёта мы увидим реальную по всем целесообразным для расчёта веществам карту с изолиниями наибольших концентраций (рисунок 6.18).

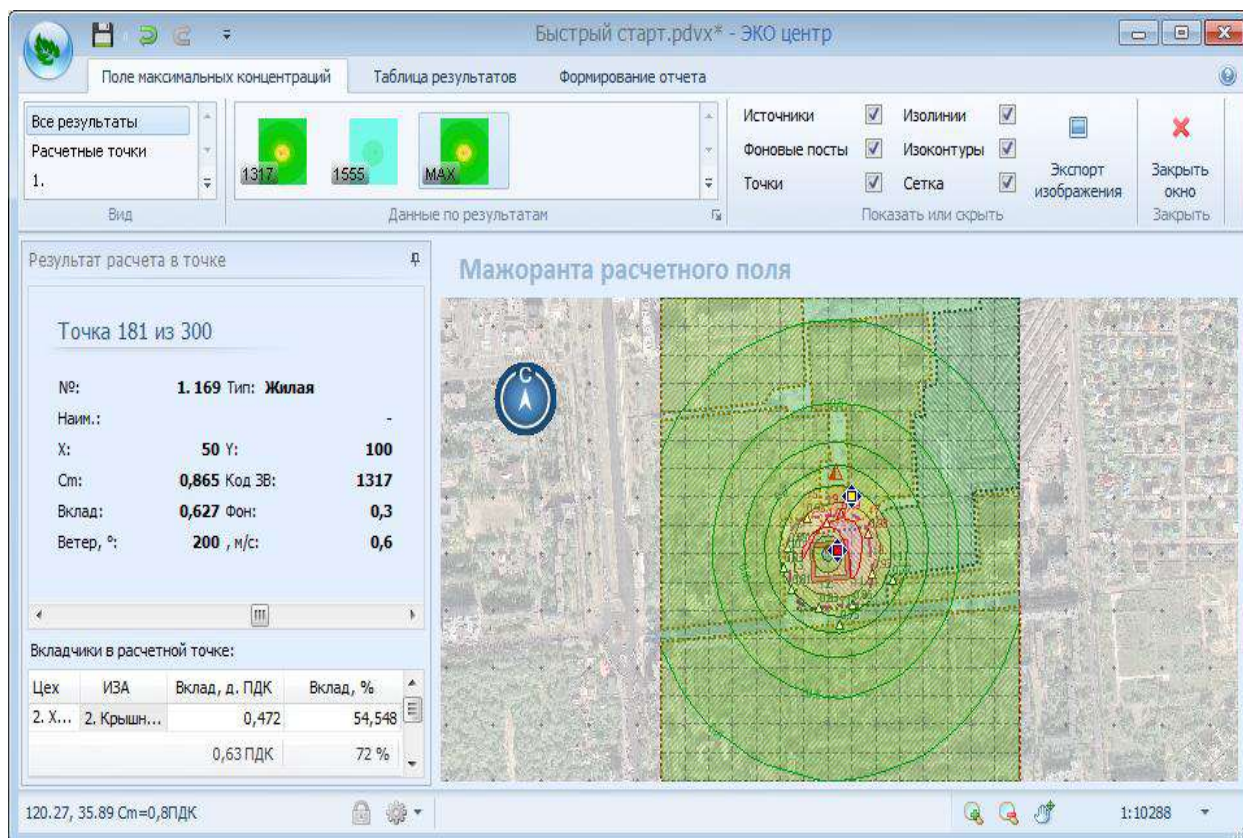


Рис. 6.18. Итоговая карта

На этом этапе мое изучение работы программы было закончено. Но в руководстве также указано о возможности вывода отчета на закладке «Формирование отчёта», где можно детально настроить перечень исходных данных и результатов, включаемых в отчёт, а также указать формат карт - схем с изолиниями наибольших концентраций. В дополнение есть возможность оформить отчёт в соответствии с ГОСТ 21.1101-2009 СПДС, воспользовавшись бесплатным шаблоном, изменив его для реквизитов необходимой организации.

В учебных целях были изучены и опробованы только основные и простейшие функции УПРЗА «ЭКО центр», так как функционал этой программы очень велик. Непременно, он будет полезен любому экологу, которому для получения максимально эффективного результата, будет необходимо изучить данную программу тщательно и глубоко.

Ключевые слова

Унифицированная программа, программа расчета загрязнения атмосферы «Эколог», расчёт концентраций загрязняющих веществ в атмосфере, УПРЗА «ЭКО центр», экологическая задача, интерфейс программы, карта – схема, источник загрязнения, данные о выбросах, формирование отчёта, функции УПРЗА.

Вопросы для самопроверки:

1. Что такое унифицированная программа расчета загрязнения атмосферы?
2. Для чего применяются программы «Эколог» и УПРЗА «ЭКО центр»?
3. Какие форматы поддерживает импорт и экспорт карт - схем программа УПРЗА "ЭКО центр"?
4. Как производится ввод данных об источниках загрязнения атмосферы?
5. На что нужно обратить внимание при расчете загрязнения атмосферы?
6. С помощью каких программ и сайтов получается наглядное растровое изображение?
7. После чего производится расчет и анализ всех результатов?
8. Как формируется отчёт?
9. Перечислите какие функции имеет УПРЗА «ЭКО центр» ?

ГЛАВА II. ОСНОВЫ АЛГОРИТМИЗАЦИИ. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++

Лекция 1. АЛГОРИТМЫ. СВОЙСТВА И ВИДЫ АЛГОРИТМОВ

1.1. Понятие алгоритма и его свойства

Под алгоритмом понимается точное предписание, задающее последовательность действий, которая ведет от произвольного исходного данного (или от некоторой совокупности возможных для данного алгоритма исходных данных) к достижению полностью определяемого этим исходным данным результата. Алгоритм должен обладать определенными свойствами, наличие которых гарантирует получение решения задачи исполнителем.

Дискретность. Решение задачи должно быть разбито на элементарные действия. Запись отдельных действий реализуется в виде упорядоченной последовательности отдельных команд, образующих дискретную структуру алгоритма. Это свойство непосредственно отражено в определении алгоритма.

Понятность. На практике любой алгоритм предназначен для определенного исполнителя, и любую команду алгоритма исполнитель должен уметь выполнить.

Определенность (детерминированность). Каждая команда алгоритма должна определять однозначные действия исполнителя. Результат их исполнения не должен зависеть от факторов, не учтенных в алгоритме явно. При одних и тех же исходных данных алгоритм должен давать стабильный результат.

Массовость. Разработанный алгоритм должен давать возможность получения результата при различных исходных данных для однотипных задач. Например, пользуясь алгоритмом решения квадратного уравнения, можно находить его корни при любых значениях коэффициентов. Свойство массовости полезное, но не

обязательное свойство алгоритма, так как интерес представляют и алгоритмы, пригодные для решения единственной задачи.

Результативность (конечность). Это свойство предполагает обязательное получение результата решения задачи за конечное число шагов. Под решением задачи понимается и сообщение о том, что при заданных значениях исходных данных задача решения не имеет. Если решить задачу при заданных исходных данных за конечное число шагов не удастся, то говорят, что алгоритм «зацикливается». Смысл условий дискретности, понятности и определенности ясен: их нарушение ведет к невозможности выполнения алгоритма.

Можно сформулировать общие правила, руководствуясь которыми следует записывать алгоритм решения задачи.

1. Выделить величины, являющиеся исходными данными для задачи.
2. Разбить решение задачи на такие команды, каждую из которых исполнитель может выполнить однозначно.
3. Указать порядок выполнения команд.
4. Задать условие окончания процесса решения задачи.
5. Определить, что является результатом решения задачи в каждом из возможных случаев.

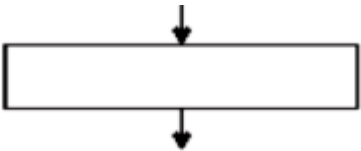
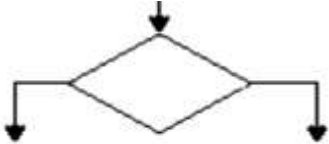

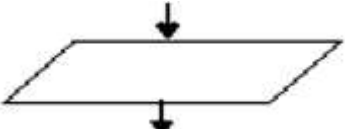
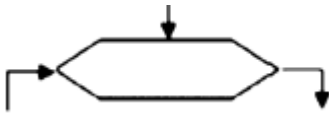
Хотя алгоритмы обычно предназначены для автоматического выполнения, они создаются и разрабатываются людьми. Поэтому первоначальная запись алгоритма обычно производится в форме, доступной для восприятия человеком. Самой простой является *словесная* форма записи алгоритмов на естественном языке. В этом виде алгоритм представляет собой описание последовательности этапов обработки данных, изложенное в произвольной форме. Словесная форма удобна для человеческого восприятия, но страдает многословностью и неоднозначностью.

Графическая форма представления алгоритма является более компактной. Алгоритм изображается в виде последовательности, связанных между собой блоков, каждый из которых соответствует выполнению одного или нескольких действий. Графическое представление алгоритма называется *блок-схемой*. Блок-схема определяет структуру алгоритма.

Графические обозначения блоков стандартизованы. Некоторые из часто используемых блоков представлены в таблице 1.1.

Таблица 1.1

Изображение основных блоков на блок-схеме

Обозначение блока	Пояснение
	Процесс (вычислительное действие, реализованное операцией присваивания)
	Решение (проверка условия, реализующая условный переход)
	Начало, конец алгоритма
	Ввод-вывод в общем виде
	Модификация (начало цикла с параметром)

Отдельные блоки соединяются линиями переходов, которые определяют очередность выполнения действий. Направление линий сверху вниз или слева направо принимается за основное.

Алгоритм, записанный на языке программирования, называется *программой*. При использовании этих языков запись алгоритма абсолютно формальна и пригодна для выполнения на ЭВМ.

Отдельная конструкция языка программирования называется *оператором*. Программа — это упорядоченная последовательность операторов.

1.2. Создание блок-схем различных видов алгоритмов

Число реализованных конструкций конечно в любом языке программирования. Структурной элементарной единицей алгоритма является команда, обозначающая один элементарный шаг обработки или отображения информации. Простая команда на языке блок-схем изображается в виде функционального блока «процесс», который имеет один вход и один выход. Из команд проверки условий и простых команд образуются составные команды, имеющие более сложную структуру, но тоже один вход и один выход.

Алгоритм любой сложности может быть представлен комбинацией трех базовых структур:

- следование;
- ветвление (в полной и сокращенной форме);
- цикл (с предусловием или постусловием).

Характерной особенностью этих структур является наличие у них одного входа и одного выхода.

Линейные алгоритмы

Базовая структура «следование» означает, что несколько операторов выполняются последовательно друг за другом, и только один раз за время выполнения программы. Структура «следование» используется для реализации задач, имеющих *линейный* алгоритм решения. Это означает, что такой алгоритм не содержит проверок условий и повторений, действия в нем выполняются последовательно, одно за другим.

Пример 2.1. Построить блок-схему алгоритма вычисления высот треугольника со сторонами a, b, c по формулам:

$$h_a = \left(\frac{2}{a}\right) \cdot \sqrt{p(p-a)(p-b)(p-c)};$$

$$h_b = \left(\frac{2}{b}\right) \cdot \sqrt{p(p-a)(p-b)(p-c)};$$

$$h_c = \left(\frac{2}{c}\right) \cdot \sqrt{p(p-a)(p-b)(p-c)},$$

где $p = \frac{(a+b+c)}{2}$ полупериметр.

Для того чтобы не вычислять три раза одно и то же значение, введем вспомогательную величину:

$$t = 2\sqrt{p(p-a)(p-b)(p-c)}.$$

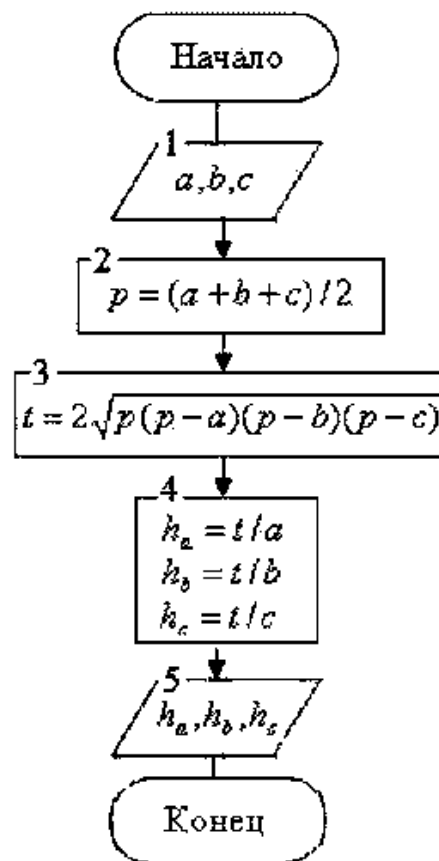
Блок 1. Ввод значений сторон треугольника.

Блок 2. Вычисление полупериметра.

Блок 3. Вычисление вспомогательной величины t .

Блок 4. Вычисление высот, опущенных на стороны a, b, c .

Блок 5. Вывод результатов.



Разветвляющиеся алгоритмы

Второй базовой структурой является «ветвление». Эта структура обеспечивает, в зависимости от результата проверки условия, выбор одного из альтернативных путей работы алгоритма, причем каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Существует структура с полным и неполным ветвлением.

Структура с полным ветвлением (если — то — иначе) записывается так:

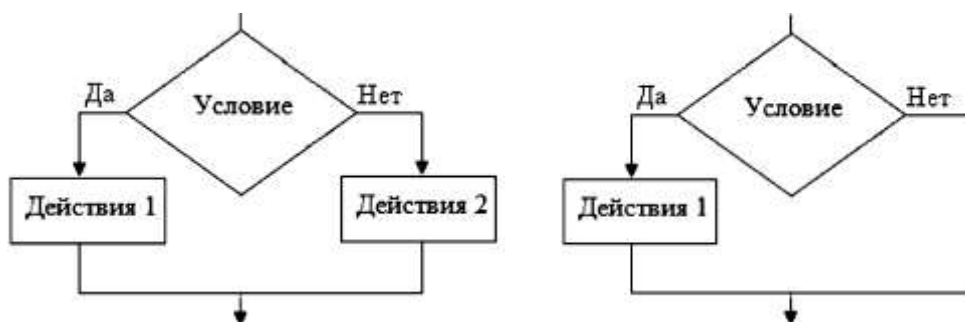
Если <условие> *то* <действия 1> *иначе* <действия 2> *Все если*

Команда выполняется так: если <условие> является истинным, то выполняются <действия 1>, записанные после ключевого слова *то*, если <условие> является ложным, то выполняются <действия 2>, записанные после слова *иначе*.

Структура с неполным ветвлением (*если — то*) не содержит части, начинающейся со слова *иначе*: *Если* <условие> *то* <действия 1> *Все если*

Команда выполняется так: если <условие> является истинным, то выполняются <действия 1>, записанные после ключевого слова *то*.

Блок-схема алгоритма с ветвлением выглядит так:



Полное ветвление.

Неполное ветвление.

Структура *Если — То — Иначе*

Структура *Если — То*

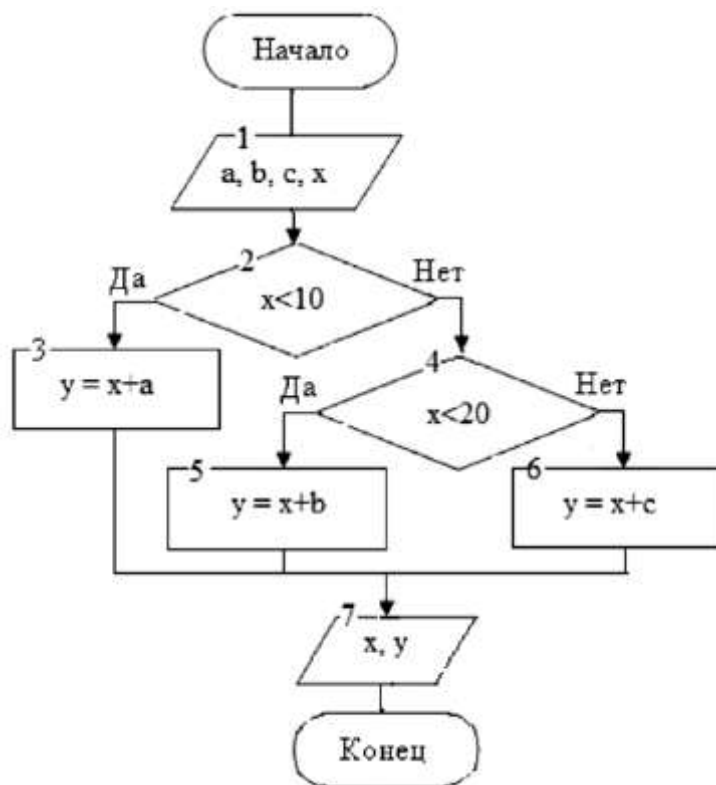
Пример 2.2. Вычислить значение функции

$$y = \begin{cases} x + a, & \text{при } x < 10; \\ x + b, & \text{при } 10 \leq x \leq 20; \\ x + c, & \text{при } x > 20. \end{cases}$$

Представим задачу графически на числовой оси

$x < 10$	$10 < x < 20$	$x > 20$
10		20
$y = x + a$	$y = x + b$	$y = x + c$
1 интервал	2 интервал	3 интервал

Так как значение переменной x вводится произвольно, то оно может оказаться в любом из трех интервалов. Приведем блок-схему.



Блок 1. Ввод исходных данных.

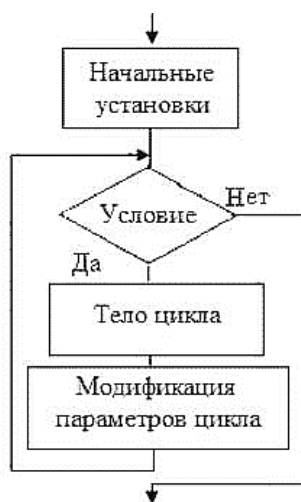
Блок 2. Проверка введенного значения. Если $x < 10$ (выход «Да»), то точка находится в первом интервале. В противном случае $x > 10$ (выход «Нет»), и точка может оказаться во втором или третьем интервале.

Блок 4. Проверка ограничения значения x справа ($x < 20$). Если условие выполняется (выход «Да»), то x находится во втором интервале, иначе $x > 20$, и точка находится в третьем интервале.

Блоки 3, 5 и 6. Вычисление значения y .

Циклические алгоритмы

При составлении алгоритмов решения большинства задач возникает необходимость в неоднократном повторении одних и тех же команд. Алгоритм, составленный с использованием многократных повторений одних и тех же действий (циклов), называется *циклическим*. Однако слово «неоднократно» не означает «до бесконечности». Организация циклов, никогда не приводящая к остановке в выполнении алгоритма («заикливание» алгоритма), нарушает требование его результативности — получения результата за конечное число шагов. Блок, для выполнения которого организуется цикл, называется *телом цикла*. Остальные операторы служат для управления процессом повторения вычислений: это начальные установки, проверка условия продолжения цикла и модификация параметра цикла. Один проход цикла называется *итерацией*.



Цикл с предусловием



Цикл с постусловием

Начальные установки служат для того, чтобы до входа в цикл задать значения переменных, которые в нем используются. Проверка условия продолжения цикла выполняется на каждой итерации либо до тела цикла (*цикл с предусловием*), либо после тела цикла (*цикл с постусловием*). Тело цикла с постусловием всегда выполняется хотя бы один раз. Проверка необходимости выполнения цикла с пред-

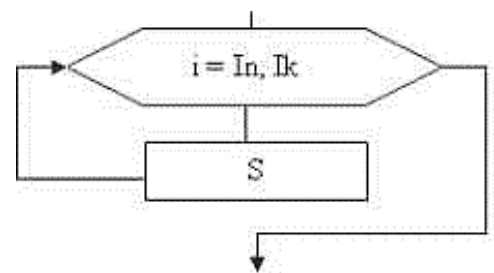
условием делается до начала цикла, поэтому возможно, что он не выполнится ни разу.

При конструировании циклов следует соблюдать обязательное условие результативности алгоритма (т. е. его окончания за конечное число шагов). Практически это означает, что в условии должна быть переменная, значение которой изменяется в теле цикла. Причем, изменяется таким образом, чтобы условие в конечном итоге перестало выполняться. Такая переменная называется управляющей переменной цикла или *параметром* цикла. Еще один вид циклов — цикл с *параметром*, или *арифметический цикл*. Тело цикла выполняется, пока параметр цикла i пробегает множество значений от начального (I_n) до конечного (I_k). Переменная i определяет количество повторений тела цикла S . Если шаг изменения значения параметра цикла обозначить через ΔI , то количество повторений тела цикла n можно вычислить по формуле:

$$n = \frac{I_k - I_n}{\Delta I} + 1.$$

Если параметр цикла i изменяется с шагом 1, то шаг может не указываться.

Цикл выполняется так: начальное значение параметра цикла i равно I_n . Если $i < I_k$, выполняется тело цикла S , после чего параметр цикла увеличивается на 1 с помощью оператора присваивания $i = i + 1$, и снова проверяется условие $i < I_k$.



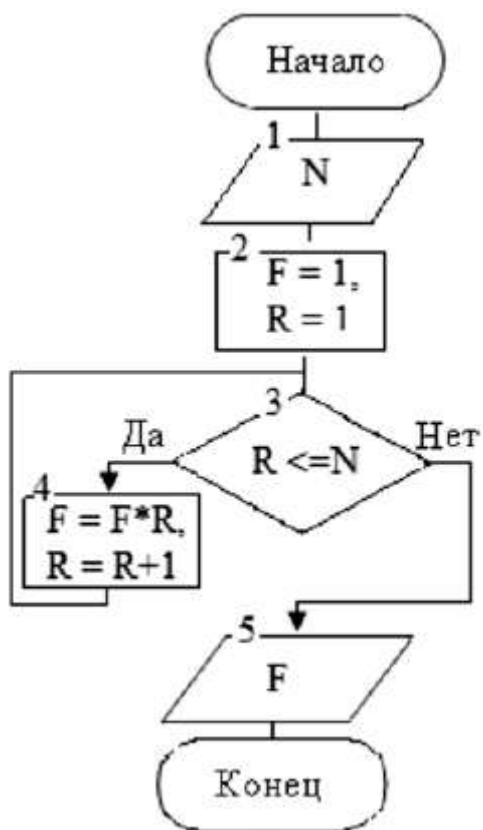
Пример 2.3. Дано целое положительное число n . Вычислить факториал этого числа. Известно, что факториал любого целого положительного числа n определяется как произведение чисел от 1 до заданного числа n :

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n.$$

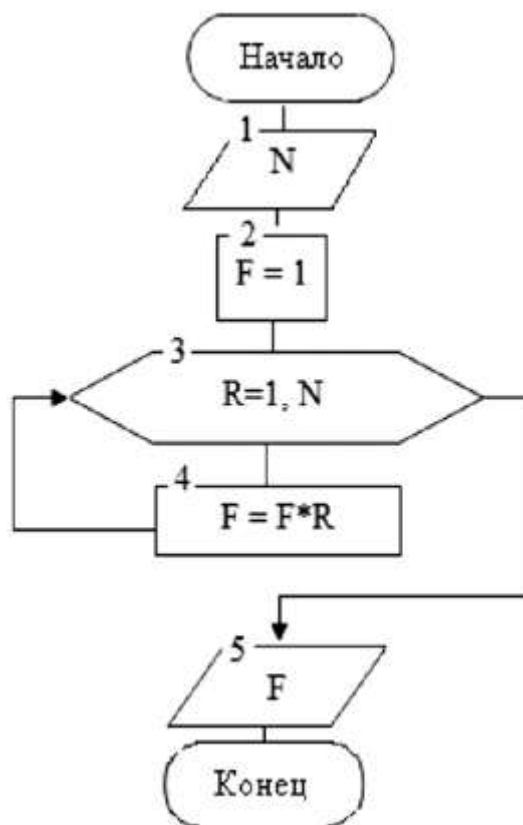
По определению $0! = 1$ и $1! = 1$.

Задача решается с помощью циклического алгоритма. Введем следующие обозначения: N — заданное число, F — факториал числа, R — параметр цикла. Составим два варианта алгоритма: с использованием цикла с предусловием и цикла с параметром.

Правильность алгоритма можно проверить, если выполнить его формально «вручную». Выполним алгоритм при $n = 4$.



Цикл с предусловием



Цикл с параметром

Цикл с предусловием

R	$R \leq N$	F
1	Да	$1 \cdot 1 = 1$
2	Да	$1 \cdot 2 = 2$

R	$R \leq N$	F
3	Да	$2 \cdot 3 = 6$
4	Да	$6 \cdot 4 = 24$
5	Нет	— (кц)

Цикл с параметром

R	$R \leq N$	F
1	Да	$1 \cdot 1 = 1$
2	Да	$1 \cdot 2 = 2$
3	Да	$2 \cdot 3 = 6$

R	$R \leq N$	F
4	Да	$6 \cdot 4 = 24$
5	Нет	— (кц)

Итак, при решении данной задачи выполнение цикла с предусловием ничем не отличается от выполнения цикла с параметром. При $R = 5$ произойдет выход из цикла и окончательное значение $4! = 24$.

В чем же отличие? Посмотрим на структуру этих циклов. В цикле *пока* начальное значение параметра цикла R задается перед входом в цикл, в теле цикла организовано изменение параметра цикла командой $R = R + 1$, условие $R < N$ определяет условие продолжение цикла. В цикле с заданным числом повторений эти же команды неявно заданы в операторе заголовка цикла.

Пример 2.4. Даны натуральное число n , действительное число x . Вычислить сумму:

$$S = \sum_{i=1}^n \frac{x^i}{i!}$$

Математическая модель. Обозначим s — слагаемое. Начальное значение суммы $S = 0$.

Количество слагаемых известно и равно n . Следовательно, можно использовать арифметический цикл.

Выведем рекуррентное соотношение, определяющее формулу, по которой вычисляется слагаемое на очередном шаге выполнения цикла.

Вспомним, что факториал числа n определяется как произведение чисел от 1 до n включительно. Следовательно, $i! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot i$, по определению $0! = 1$, $1! = 1$. Для вывода рекуррентного соотношения вычислим несколько значений слагаемых.

$$i = 1, c = x^1/1! = x;$$

$$i = 2, c = x^2/2! = x^2/(1 \cdot 2) = c \cdot x/2 = c \cdot x/i;$$

$$i = 3, c = x^3/3! = x^3/(1 \cdot 2 \cdot 3) = c \cdot x/3 = c \cdot x/i;$$

$$i = 4, c = x^4/4! = x^4/(1 \cdot 2 \cdot 3 \cdot 4) = c \cdot x/4 = c \cdot x/i.$$

В правой части всех формул значение c равно значению, вычисленному на предыдущем шаге. Итак, получаем следующее соотношение, которое будем использовать в алгоритме $c = c \cdot x/i$, где начальное значение $c = 1$, $i = 1, 2, \dots, n$.

Сумму накапливаем, используя формулу $S = S + c$. Записанное соотношение определяет, что значение суммы увеличивается на величину слагаемого при каждом повторении цикла. Начальное значение суммы равно 0.

Переменные i , n — целого типа, переменные c , x , S — вещественного типа.

Приведем словесное описание и блок-схему алгоритма.

Блок 1. Ввод x , n .

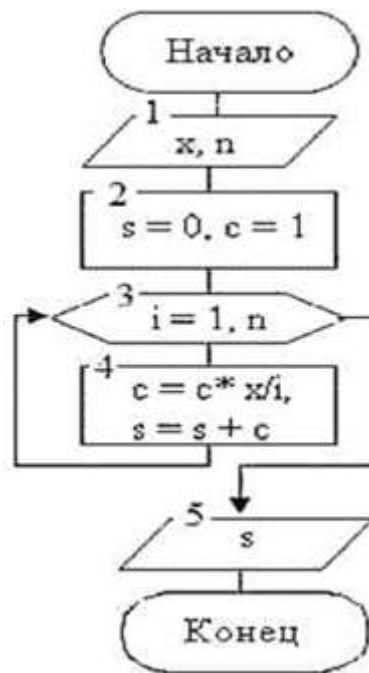
Блок 2. Присваивание начальных значений $S = 0$, $c = 1$.

Блок 3. В цикле для всех значений параметра цикла i вычисляем тело цикла (блок 4):

$$c = c * x/i;$$

$$S = S + c.$$

Блок 5. Вывод суммы S .



1.3. Алгоритмы, использующие одномерные и двумерные массивы

Массив — это упорядоченный набор однотипных значений (элементов массива). Каждый массив имеет имя, что дает возможность различать массивы между собой и обращаться к ним по имени.

Каждый элемент массива имеет три характеристики:

- 1) *имя*, совпадающее с именем массива;
- 2) *индекс* — целое число или множество целых чисел, однозначно определяющее местоположение элемента в массиве. В качестве индекса может использоваться также переменная или арифметическое выражение целого типа. Примеры индексов: 3, 15, i , j , $i - 1$, $j + 2$;
- 3) *значение* — фактическое значение элемента, определенное его типом.

Элементы массива могут использоваться произвольно и являются одинаково доступными. Доступ к элементам массива производится по его индексу.

Массивы могут быть одномерными и многомерными. В этом подпараграфе рассмотрим некоторые алгоритмы на одномерных массивах.

Ввод и вывод элементов массива

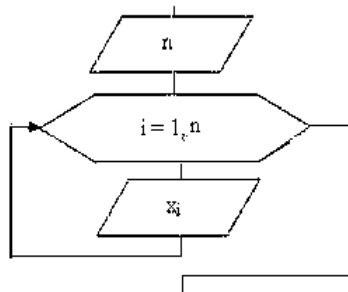
Одномерный массив определяется именем и числом элементов (размером), и мы обозначим его $a[n]$, где a — имя массива; n — число элементов массива. Например, $a[10]$, где a — имя массива; 10 — число элементов в массиве.

Каждый элемент одномерного массива имеет один индекс, равный порядковому номеру элемента в массиве. Например, массив из 10 элементов выглядит так:

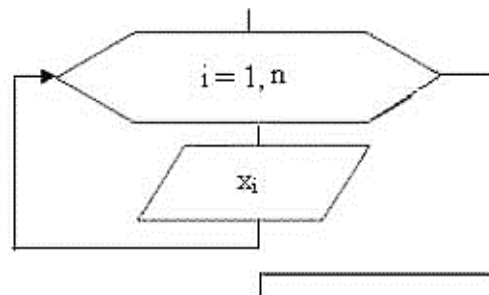
Индекс	1	2	3	4	5	6	7	8	9	10
Значение	3	0	15	4	6	-2	11	0	-9	7

Мы рассматриваем массив, состоящий из произвольного числа элементов. Поэтому, прежде чем задать значения элементов массива, требуется ввести количество элементов массива n .

Вывод элементов также производится с использованием цикла с заданным числом повторений.



Ввод элементов массива



Вывод элементов массива

Здесь и далее будем обозначать через i — текущий индекс элемента массива. Он же будет являться параметром цикла, так как количество повторений цикла зависит от количества элементов в массиве.

Далее во всех задачах будем считать, что элементы массива заданы тем или иным способом, и показывать только реализацию алгоритма решения задачи, опуская команды ввода данных и вывода результата.

Также не будем обозначать блоки начала и конца выполнения алгоритма. Подразумевается, что все пере численные блоки должны всегда присутствовать в алгоритме.

Алгоритмы, использующие двумерные массивы

Понятие «двумерный массив» определим на примере.

Пусть имеются данные о продажах некоторого товара по месяцам:

Месяц	Объем продаж, пар	Цена продажи, сум.	Себестоимость, сум.
1	4500	1000	500
2	3900	1100	550
3	3100	1200	600

Таблица представляет собой множество из двенадцати однородных величин — это массив. Ее элементы расположены в 3 строки по 4 столбца в каждой.

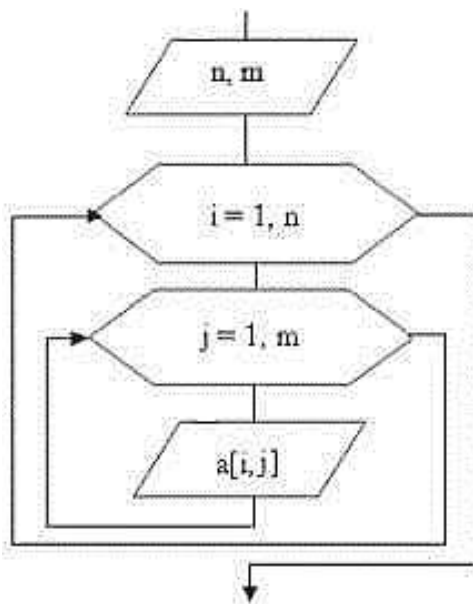
Подобного рода таблицы из нескольких строк с равным числом элементов в каждой называют двумерными массивами или матрицей. Двумерный массив определяется именем, числом строк и столбцов и обозначается, например, так: $A[n, m]$, где A — произвольное имя массива; n — число строк; m — число столбцов. Обратите внимание на то, что сначала всегда указывается количество строк, а потом — количество столбцов массива. Если имеются данные о продажах за 3 мес., то нашу таблицу можно обозначить так: $A[3, 4]$, т. е. массив состоит из 3 строк и 4 столбцов. Строки двумерных массивов нумеруются по порядку сверху вниз, а столбцы — слева направо.

Элементы таблицы, представленной в примере, получают такие обозначения:

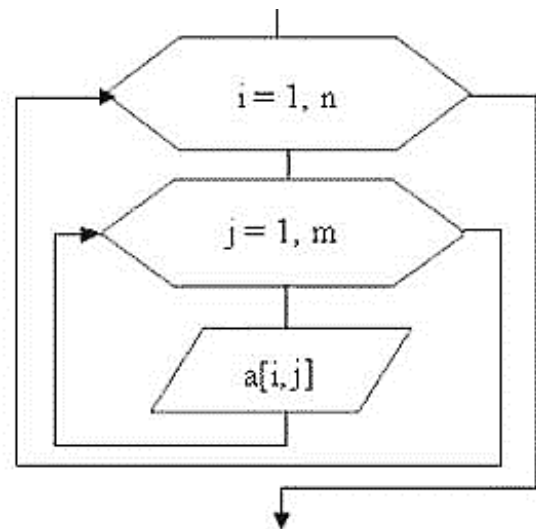
$$\begin{aligned}
 A[3, 4] &= \begin{pmatrix} A[1, 1] & A[1, 2] & A[1, 3] & A[1, 4] \\ A[2, 1] & A[2, 2] & A[2, 3] & A[2, 4] \\ A[3, 1] & A[3, 2] & A[3, 3] & A[3, 4] \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 4500 & 100 & 50 \\ 2 & 3900 & 110 & 55 \\ 3 & 3100 & 120 & 60 \end{pmatrix}.
 \end{aligned}$$

Каждый элемент двумерного массива определяется номерами строки и столбца, на пересечении которых он находится, и в соответствии с этим обозначается именем массива с двумя индексами: первый — номер строки, второй — номер столбца. Например, $A[1, 3]$ — элемент находится в первой строке и третьем столбце.

Таким образом, $A[1, 1] = 1$, $A[2, 3] = 110$ и т. д. Произвольный элемент двумерного массива мы будем обозначать $A[i, j]$, где i — номер строки, j — номер столбца. Поскольку положение элемента в двумерном массиве описывается двумя индексами, алгоритмы для решения большинства задач с их использованием строятся на основе вложенных циклов. Обычно внешний цикл организуется по строкам массива, т. е. в нем выбирается требуемая строка, а внутренний — по столбцам, в котором выбирается элемент внутри строки.



Ввод двумерного массива



Вывод двумерного массива

В отличие от одномерных массивов для ввода и вывода данных в двумерные массивы необходимо использовать вложенные циклы. Циклы являются арифметическими, так как количество строк и столбцов известно.

Мы ввели и вывели элементы произвольного массива, имеющего n строк и m столбцов. Понятно, что при вводе необходимо задать количество строк и столбцов. Напомним, что во всех задачах будем считать, что

элементы массива заданы тем или иным способом, и показывать только реализацию алгоритма решения задачи, опуская команды ввода данных и вывода результата. Также не будем обозначать блоки начала и конца выполнения алгоритма. Подразумевается, что все перечисленные блоки должны всегда присутствовать в алгоритме. Следует понимать, что все алгоритмы, приведенные ранее для одномерных массивов, можно использовать и для двумерных массивов, применив вложенный цикл.

Ключевые слова:

Алгоритм, дискретность, определенность, массовость, результативность, графическая форма представления алгоритма, процесс, следование, ветвление, цикл, линейные алгоритмы, разветвляющие алгоритмы, циклические алгоритмы, тело цикла, параметры цикла, шаг, массивы, одномерные, двумерные массивы, имя массива, индекс массива, значение.

Вопросы для самопроверки:

1. Какие виды алгоритмов существуют?
2. Какие свойства алгоритмов вы знаете?
3. Объясните понятие дискретности, определенности, результативности алгоритма?
4. На какие этапы делятся алгоритмы решения задач?
5. Способы описания алгоритма?
6. Объясните сущность графического способа описания алгоритма?
7. Дайте понятие линейного алгоритма. Приведите пример.
8. Что представляет собой структура разветвляющего алгоритма. Приведите пример.
9. Объясните сущность циклического алгоритма и пути составления блок-схем циклического алгоритма.

Лекция 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ СТРУКТУРЫ.

2.1. Алфавит языка C++

Программа — это алгоритм, записанный на каком-либо языке программирования (формальном языке).

Программа предназначена для обработки данных (управления данными). В процессе работы приложения сама программа (или ее часть) и данные (или их часть) находятся в оперативной памяти компьютера (ОЗУ).

Язык программирования является инструментом для написания программ. Языки программирования, в том числе C++, относятся к *формальным языкам*. Задачей формальных языков является описание алгоритмов и функций, которые выполняет компьютер. Формальные языки отличает небольшой набор изобразительных средств соответственно целям этих языков.

Алгоритм решения задачи записывается программистом согласно синтаксическим правилам языка, и это программа. Семантика реализована в компиляторе, который переводит код во внутреннее представление. Компилятор обнаруживает синтаксические ошибки, допущенные в коде программы.

Обязательными этапами обработки кода программы на языке C++ являются:

- ✓ препроцессорное преобразование;
- ✓ компиляция;
- ✓ компоновка.

Задачей *препроцессора* является преобразование текста программы до ее компиляции. Препроцессор сканирует текст, находит в нем команды, называемые директивами препроцессора, и выполняет их. В результате происходят изменения в исходном тексте: вставка фрагментов текста и замена некоторых его фрагментов. Полученный текст называется полным текстом программы.

На этапе *компиляции* полный код программы преобразуется во внутреннее машинное представление, некоторую последовательность команд, которая понятна компьютеру. Компилятор находит ошибки несоответствия кода программы синтаксическим правилам языка.

На этапе *компоновки* происходит редактирование связей и сборка исполнимого кода программы. Компоновщик обрабатывает все вызовы библиотечных функций и выполняет их подключение. Таким образом, к скомпилированному исходному коду добавляются необходимые функции стандартных библиотек. Готовый код является исполнимым и может быть выполнен компьютером.

Алфавит — набор символов, разрешенных для построения синтаксических конструкций. Алфавит языка C++ содержит четыре группы символов.

1. *Буквы*. Разрешается использовать буквы латинского алфавита, прописные (A-Z) и строчные (a-z). Русские буквы не входят в алфавит, но используются в комментариях и текстовых константах, там, где они не влияют на смысл программы.

2. *Цифры*. Используются арабские цифры 0, 1, ..., 9.

3. *Специальные символы*. Они могут быть разделены на подгруппы:

- ✓ знаки препинания (разделители): , ; . : ;
- ✓ знаки операций: +, -, *, /, %, &, |, ?, !, <, =, >;
- ✓ парные скобки: [], { }, (), " ", ' ';
- ✓ прочие символы: _, #, ".

4. *Невидимые символы*. Они могут считаться разделителями, их особенность в том, что символы существуют (каждый имеет код), но в редакторе не видны. Это такие символы, как пробел, табуляция, разделитель строк. Их общее название — обобщенные пробельные символы.

Из символов алфавита строятся все конструкции языка.

Лексема — это единица текста (конструкция, слово), воспринимаемая компилятором как единое неделимое целое. Можно выделить пять классов лексем.

1. *Имена* (идентификаторы) для именования произвольных объектов программы, например, x1, Alpha, My_file.

2. *Служебные* (ключевые) слова, обозначающие конструкции языка (имена операторов), например, for, while, do.

3. *Константы*, например, 1, 12.5, “Василий”.

4. *Операции* (знаки операций), например, ++, >=, !=, >.

5. *Разделители* (знаки пунктуации), например, [], (), { }.

Концепция данных в языке C++

Данные — все, что подлежит обработке с помощью программы. Данные, используемые приложением, должны быть описаны в программном коде. Классификацию данных можно выполнять по нескольким категориям.

1. *По типу*. Каждое данное имеет тип. Типы данных можно разделить на *базовые*, т. е. такие, правила организации которых predetermined реализацией языка, и *конструируемые*, т. е., которые пользователь строит по определенным правилам для конкретной задачи.

2. *По способу организации*. Для каждого из простых (базовых) типов каждое данное может быть неизменяемым или изменяемым. По способу организации данные делятся на два класса:

- *константа* — данное, которое не меняет своего значения при выполнении программы и присутствует в тексте программы явным образом. Тип константы определен ее записью;
- *переменная* — данное, которое изменяется при выполнении программы и в тексте присутствует своим именем (идентификатор). Тип переменной величины должен быть объявлен в тексте программы.

Каждое данное, независимо от способа организации, обладает типом. Тип данного очень важен, так как он определяет:

- *механизм* выделения памяти для записи значений переменной;

- *Диапазон* значений, которые может принять переменная;
- *список* операций, которые разрешены над данной переменной.

Механизм выделения памяти и внутреннего представления данного важнее всего, диапазон значений и операции являются следствием из первого.

Тип любого данного программы должен быть объявлен обязательно. Для констант тип определен явно записью константы в тексте программы. Для переменных тип должен быть задан в объявлении объекта. При объявлении переменной происходит выделение области памяти, размер которой определен типом переменной и в которой хранится значение переменной.

Классификация типов данных:

- *простые типы* (базовые, скалярные, внутренние, предопределенные);
- *конструируемые типы* (массивы, строки, структуры, функции и др.).

Основные типы данных определены ключевыми словами, список которых приведен в таблице 2.1.

Таблица 2.1

Ключевые слова, определяющие основные типы данных

Имя типа	Значение типа	Размер	Диапазон значений
char	Символьный (целое)	1 байт	-128-127
int	Целый	4 байта	$\pm 2\ 147\ 483\ 649$
float	С плавающей точкой	4 байта	$3.4e-38-3.4e38$
double	Двойной точности (действительный)	8 байт	$1.7e-308-1.7e308$
void	Любой или никакой		Не ограничен

Константы на языке C++

Синтаксис языка выделяет пять типов констант: целые, действительные (вещественные), символьные, перечислимые, нулевой указатель.

Целые константы

Синтаксис языка позволяет использовать константы трех систем счисления: десятичные, восьмеричные, шестнадцатеричные. Основание

определяется префиксом в записи константы. По умолчанию основание 10, префикс 0 предваряет восьмеричную константу, префикс 0x или 0X предваряет шестнадцатеричную константу. В остальном запись целых констант соответствует общепринятой.

Примеры записи целых констант приведены в таблице 2.2.

Таблица 2.2

Примеры записи целых констант

Десятичные	Восьмеричные	Шестнадцатеричные
127	012	0xA
-256	-014	-0x10

Целые числа в памяти компьютера представлены в форме с фиксированной точкой. Эта форма позволяет представить значение с абсолютной точностью. На рисунке 2.1 показано представление целого числа в двухбайтовой ячейке памяти.

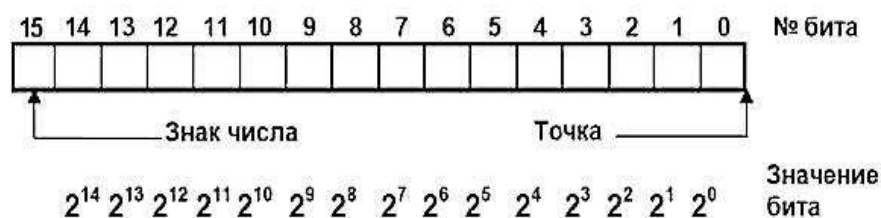


Рис. 2.1. Представление целых чисел

Действительные константы

Вещественные (действительные) числа представлены в памяти компьютера в форме с плавающей точкой в виде $M \cdot 10^p$, где M — мантисса вещественного числа; 10 — основание системы счисления; p — показатель десятичной степени, целое число со знаком. Основание системы счисления в языке C++ заменяется буквой e или E . Некоторые из составляющих могут быть опущены.

Примеры записи действительных констант:

44. 3.1415926 44.e0 .31459E1 0.0

На рисунке 2.2 показано представление вещественного числа в четырехбайтовой ячейке памяти.

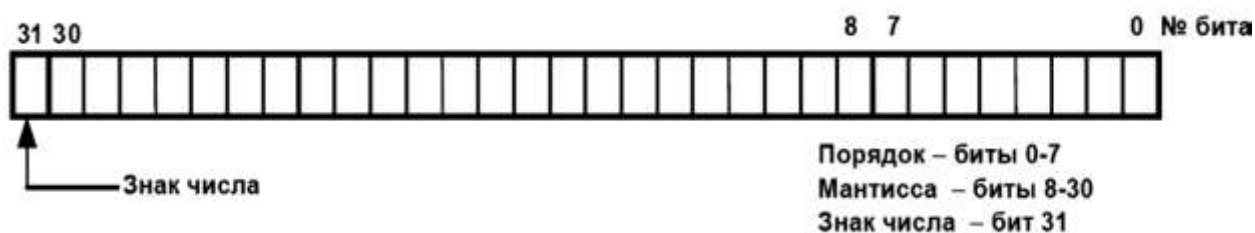


Рис. 2.2. Представление вещественных чисел

Символьные константы

Символьная константа — это лексема, которая содержит произвольный символ, заключенный в одинарные кавычки (апостроф). Хранится в одном байте памяти в виде целого числа, определяющего целочисленный код символа. Символьной константой может быть любой символ, изображаемый на экране в текстовом режиме, или не имеющий отображения (пробельный символ).

Примеры символьных констант: '!', '.', 's', 'A', '+', '2'.

Существуют также управляющие символы (*ESC*-последовательности), которые используются в строковых константах и переменных. Они имеют признак '\' (backslash или обратный слэш), который используется для указания управляющего воздействия. Вот их неполный список:

'\0' — нулевой символ;

'\n' — перевод строки;

'\r' — возврат каретки;

'\t' — табуляция;

'\'' — апостроф;

'\\' — обратный слэш;

'\ddd' — восьмеричное представление символьной константы, где ddd — ее восьмеричный числовой код, например, '\017' или '\233';

'\xhh' или '\Xhh' — шестнадцатеричное представление символьной константы, где hh — ее числовой код, например, '\x0b' или '\x1F'.

Символьные константы имеют целый тип и могут использоваться в выражениях.

Строковые константы

Формально строки не относятся к константам языка C++, а представляют отдельный тип его лексем. Строковая константа определяется как набор символов, заключенных в двойные кавычки " ".

Например,

```
#define STR "программа"
```

В конце строковой константы добавляется нулевой символ '\0'. В тексте строки могут встретиться *ESC*-последовательности, которые будут выполнять управляющее воздействие.

Например,

```
"\nПри выводе \ntекст может быть \nразбит на строки."
```

Здесь "\n" выполняет управляющее воздействие, и вывод будет таким:

при выводе

текст может быть

разбит на строки.

При хранении строки все символы, в том числе управляющие, хранятся последовательно, и каждый занимает ровно один байт.

Типы числовых констант и модификаторы типов

Для модификации типа числовых констант используются префиксы и суффиксы, добавляемые к значению константы. Суффикс l(L) от слова long увеличивает длину данного в два раза, суффикс u(U) от слова unsigned используется для чисел, не содержащих знак, т. е. положительных, при этом увеличивается число разрядов, отводимых под запись числа, а значит, и его возможное значение.

Например, длинные константы 123451, -54321L; константы без знака 123u, 123U; шестнадцатеричная длинная константа 0xb8000000l; восьмеричная длинная константа без знака 012345LU.

Переменные в языке C++

Одним из основных понятий в языках программирования является объект. Будем использовать это понятие как объект программного кода.

Объект — это некоторая сущность, обладающая определенным набором свойств и способная хранить свое состояние.

Объектами программного кода являются константы, переменные, функции, типы и т. д.

Переменная — это объект, для использования которого необходимо определить три характеристики:

- имя;
- тип;
- значение (не обязательно).

Имя переменной (идентификатор) обозначает в тексте программы величину, изменяющую свое значение. Для каждой переменной выделяется некоторая область памяти, способная хранить ее значение. Имя позволяет осуществить доступ к области памяти, хранящей значение переменной.

Тип переменной определяет размер выделяемой памяти и способ хранения значения (см. рис. 2.1, 2.2).

Значение переменной не определено вначале, изменяется при присваивании переменной значения.

Имя переменной (идентификатор) включает в себя латинские буквы, цифры, знак подчеркивания "_". Первым символом должна быть буква.

Примеры идентификаторов: x1, Price, My_file1, alpha, PI. В качестве имен рабочих переменных используются простые имена i, j, k1, k2 и им подобные.

Ограничения:

- длина имени содержит не более 32-х символов;
- в качестве имен нельзя использовать ключевые слова языка C++, например, названия операторов for, if, do;
- в качестве имен не рекомендуется использовать имена объектов стандартных библиотек, например, имена функций sin(), sqrt(), pow();

- имена, которые начинаются со знака подчеркивания, зарезервированы для использования в библиотеках и компиляторах, поэтому их не следует выбирать в качестве прикладных имен, например, `_asm`, `_AX`, `_BX`.

Перечень основных, не всех служебных слов приведен в таблице 2.3.

Таблица 2.3

Служебные слова в языке С++

Типы данных	char, double, enum, float, int, long, short, struct, signed, union, unsigned, void, typedef
Квалификаторы типа	const, volatile
Классы памяти	auto, extern, register, static
Названия операторов	break, continue, do, for, goto, if...else, return, switch, while, case, default, sizeof
Модификаторы	asm, far, interrupt, pascal и др.
Псевдопеременные	Названия регистров AH DS DX и др.

Объявление переменных

Каждая переменная должна быть объявлена в тексте программы перед первым ее использованием. Начинающим программистам рекомендуется объявлять переменные в начале тела программы перед первым исполнимым оператором. Синтаксис объявления переменной требует указать ее тип и имя:

Имя типа Имя переменной;

Можно одновременно объявить несколько переменных, тогда в списке имен они отделяются друг от друга запятой, например:

```
int a, b, c;           // целые со знаком
```

```
char ch, sh;           // однобайтовые символьные
long l, m, k;          // длинные целые
float x, y, z;         // вещественные
long double u, v       // длинные двойные
```

На этапе компиляции для объявленных переменных в ОЗУ будет выделена память для записи их значений. Имя переменной определяет адрес выделенной для нее памяти, тип переменной определяет способ ее хранения в памяти. Как следствие, тип определяет диапазон ее возможных значений и операции, разрешенные над этой величиной.

Переменные, объявленные в программе, не имеют значения, точнее имеют неопределенные значения. Так как память, выделенная под запись переменных, не очищается, то значением переменной будет «мусор» из того, что находилось в этой памяти ранее.

Инициализация — это прием, который позволяет присвоить значения переменным при их объявлении. Синтаксис инициализации:

```
Имя_типа Имя_переменной = начальное_значение;
```

Например,

```
float pi=3.1415;
```

```
unsigned Year=2014;
```

В языке C++, кроме переменных, можно использовать *именованные* константы, т. е. константы, имеющие фиксированные названия (имена). Имена могут быть произвольными идентификаторами, не совпадающими с другими именами объектов. Принято использовать в качестве имени константы большие буквы и знаки подчеркивания, что визуально отличает имена переменных от имен констант. Способов определения именованных констант три.

Первый способ — это константы перечисляемого типа, имеющие синтаксис:

```
enum тип_перечисления {список_именованных_констант};
```

Здесь «тип_перечисления» — это его название, необязательный произвольный идентификатор; «список_именованных_констант» — это разделенная запятыми последовательность вида:

```
имя_константы = значение_константы
```

Примеры:

```
enum BOOL {FALSE, TRUE};
```

```
enum DAY {SUNDAY, MONDAY, TUESDAY, WEDNESDAY,  
THURSDAY, FRIDAY, SATURDAY};
```

```
enum {ONE=1, TWO, THREE, FOUR, FIVE};
```

Если в списке нет элементов со знаком «=», то значения констант начинаются с 0 и увеличиваются на 1 слева направо. Так, FALSE равно 0, TRUE равно 1. SUNDAY равно 0 и так далее, SATURDAY равно 6. Если в списке есть элементы со знаком «=», то каждый такой элемент получает соответствующее значение, а следующие за ним увеличиваются на 1 слева направо, так, значение ONE = 1, TWO = 2, THREE = 3, FOUR = 4, FIVE = 5.

Второй способ — это именованные константы, заданные с ключевым словом `const` в объявлении типа:

```
const имя_типа имя_константы = значение_константы;
```

Ключевое слово `const` имеет более широкий смысл и используется также в других целях, но здесь его суть в том, что значение константы не может быть изменено.

Примеры.

```
const float Pi=3.1415926;
```

```
const long M=999999999L;
```

Попытка присвоить значение любой из констант Pi, M будет неудачной. Визуализация имен констант с помощью использования больших букв необязательна.

Третий способ ввести именованную константу дает препроцессорная директива `#define`, подробный синтаксис и механизм выполнения которой будет рассмотрен далее.

Операции и выражения языка C++

Операция — это символ или лексема вычисления значения. Операции используются для формирования и вычисления значений выражений и для изменения значений данных.

Классификация операций

Классифицировать операции можно по нескольким признакам. Приведем два варианта: по типу возвращаемого значения и по числу операндов.

1. *Тип возвращаемого значения* является важным свойством любой операции. В таблице 2.4 приведена классификация операций по типу возвращаемого значения.

Таблица 2.4

Классификация операций по типу возвращаемого значения

Арифметические	Логические
+ сложение	> больше
- вычитание	< меньше
* умножение	== равно
/ деление (для целых операндов — целая часть от деления)	!= не равно
% остаток от деления (только для целых)	>= больше или равно
	<= меньше или равно
	&& логическое «И»
	логическое «ИЛИ»
	! логическое «НЕ»

Арифметические операции имеют обычный математический смысл. Для целого типа данных существуют две операции деления: целочисленное деление / и остаток от деления %. Каждая из этих операций, примененная к данным целого типа, возвращает целое значение.

Пример 2.1.

```
int x=5, y=3;
```

```
x/y          // результат = 1
x%y          // результат = 2
y/x          // результат = 0
y%x          // результат =3
x=-5         // знак сохраняется
x/y          // результат=-1
x%y          // результат= -2
y/x          // результат=0
y%x          // результат=3
```

Логические операции предназначены для проверки условий и для вычисления логических значений.

Логическими являются операции, возвращающие логическое значение. Их можно, в свою очередь, разбить на две группы.

Операции отношения. Связывают данные числовых или символьных типов и возвращают логическое значение, например:

```
3>1          // истина, не равно 0
'a'>'b'       // ложь равна 0
x>=0         // зависит от x
y!=x         // зависит от x и от y
```

Значение операции отношения может быть присвоено целой переменной или переменной логического типа, например:

```
int a=5, b=2;
bool c;
c=a>b;       // c = 1
c=a<b;       // c = 0
c=a==b;      // c = 0
c=a!=b;      // c = 1
```

Операции отношения могут участвовать в записи выражения наряду с

арифметическими операциями, тогда при вычислении значения выражения важен порядок вычисления, например:

$c = 10 * (a > b);$ // $a > b = 1, 10 * 1 = 10$, значит, $c = 10$

$c = 10 * (10 * a < b);$ // $10 * a = 50, 50 < b = 0, 10 * 0 = 0$, $c = 0$

$c = 1 < a \&\& a < 10;$ // $1 < a = 1, a < 10 = 1$, значит, $c = 1$

Собственно логические операции. Связывают данные логического типа (целые) и возвращают логическое значение. Это логические операции конъюнкция $\&\&$, дизъюнкция $\|$ и отрицание $!$, которые применяются к значениям операндов, имеющих логическое значение. Если операндом является отношение, то предварительно вычисляется его значение, например:

// оба операнда ($x > 0$ и $y > 0$) одновременно истинны $x > 0 \&\& y > 0;$

// хотя бы один операнд ($x \% 2 == 0$ или $y \% 2 == 0$) истинен $x \% 2 == 0 \| y \% 2 == 0;$

// значение, обратное значению операнда

// $(x * x + y * y \leq r * r)$

$!(x * x + y * y \leq r * r);$

2. Второй вид классификации операций языка C++ — классификация по числу операндов.

1) Унарные операции имеют только один операнд, например: $+10$, $-a$, $!(x > 0)$, $\sim a$.

2) Бинарные операции имеют два операнда, например: $x + 5$, $x - y$, $b \&\& c$.

3) *Тернарная* операция имеет три операнда, это операция *условия*, приведем ее синтаксис здесь: логическое_выражение ? Выражение1 : Выражение2

Операция *условия* выполняется в два этапа. Сначала вычисляется значение логического выражения. Затем, если оно истинно, вычисляется значение второго операнда, который назван «Выражение1», оно и будет результатом, иначе вычисляется значение третьего операнда, который назван «Выражение2», и результатом будет его значение.

Например, вычисление абсолютного значения некоторого i :

```
int abs;
abs=(i<=0)?-i:i;
```

Операции изменения данных

Операции изменения данных имеют особое значение, позволяя выполнить присваивание значения переменной.

Список операций изменения данных приведен в таблице 2.5.

Таблица 2.5

Операции изменения данных

Основные операции	Арифметические операции с присваиванием
= операция присваивания	+= сложение с присваиванием
++ увеличение на единицу	-= вычитание с присваиванием
— уменьшение на единицу	*= умножение с присваиванием
	/= деление с присваиванием
	% = остаток от деления с присваиванием

Операция присваивания в C++ — это обычная операция, которая является частью выражения, но имеет очень низкий приоритет и поэтому выполняется в последнюю очередь. Семантика этой операции заключается в том, что вычисляется выражение правой части и присваивается левому операнду. Это означает, что слева может быть только переменная величина, которая способна хранить и изменять свое значение (левостороннее выражение). В правой части, как правило, записывается вычисляемое выражение.

```
x=2;           // x принимает значение константы
cond=(x<=2);   // cond принимает логическое
               // значение
3=5;           // ошибка, слева константа
x=x+1;         // x принимает значение
               // увеличенное на 1
```

Расширением операции присваивания являются операции вычисления с присваиванием, которые еще называются «присваивание после» и позволяют сократить запись выражений, например:

$x+=5$; // аналог записи $x = x + 5$

$y*=3$ // аналог записи $y = y * 3$

Пример 2.2.

Пусть $c = 5$, $a = 3$; тогда

$c+=a$; // $c = 8$

$c-=a$; // $c = 2$

$c*=2$; // $c = 10$

$c/=4$; // $c = 1$

Необычными операциями являются операции *увеличения* и *уменьшения* на *единицу*, имеющие самостоятельные названия *инкремент* и *декремент*, например:

$x++$; // аналог записи $x = x + 1$ или $x += 1$

$--x$; // аналог записи $x = x - 1$ или $x -= 1$

Каждая из этих операций имеет две формы.

1. Префиксная $++x$ или $--x$. При ее выполнении x изменяется прежде, чем выполняются другие операции, входящие в выражение.
2. Постфиксная $x++$ или $x--$. При ее выполнении сначала выполняются все операции выражения, и только потом изменяется x .

Пример 2.3.

int q, a; a=2;

$q=2*++a$; // сначала a увеличивается на 1, т. е. $a = 3$,

 // затем a умножается на 2 и результат

 // присваивается q

$q=2*a++$; // сначала a умножается на 2, результат

 // присваивается

 // q , затем a увеличивается на 1,

 // после чего $a = 5$

Выражения в языке C++

Выражение — это правило вычисления одного значения.

Формально выражение — это несколько операндов, объединенных знаками операций. В качестве операндов выступают константы, имена переменных, вызовы операций-функций. Кроме того, в запись выражения могут входить технические символы, такие как скобки (). Как правило, смыслом выражения является не только вычисление значения, но и присваивание его переменной, поэтому в записи выражения присутствует операция присваивания «=» или ее клоны.

Поскольку все данные в C++ (переменные, константы и другие объекты) имеют тип, то значение, вычисленное выражением, также будет иметь тип, который определяется типом операндов, например, как в таблице 2.6.

Таблица 2.6

Примеры определения типа выражения

Операнды	Операции	Результат
Целые	+ - * / %	Целый
Вещественные	+ - * /	Вещественный
Числовые или символьные	> < != == >= <=	Логический (int, bool)
Логические	&& !	Логический

Если тип вычисленного выражением значения не совпадает с типом переменной левой части, то происходит *преобразование (приведение)* типов.

Порядок вычисления выражений определяется *рангом (приоритетом)* входящих в него операций и правилами ассоциативности, которые применяются при вычислении значений выражений, если несколько операций одного приоритета стоят подряд. Большинство операций левоассоциативны (^), т. е. вычисляются слева направо естественным образом, например, бинарные операции. Однако есть правоассоциативные операции (^), которые вычисляются справа налево.

Например, операция присваивания позволяет выполнить цепочку присваиваний: `x = y = z = 10;`

Здесь цепочка раскрывается по порядку $z = 10$, затем $y = z$, затем $x = y$.

В таблице 2.7 приведены приоритеты операций C++.

Из таблицы 2.7 видно, что принятый ранг операций наиболее близок к математическому, также как и принятый порядок их вычисления. Так, умножение и деление (мультипликативные операции) старше сложения и вычитания (аддитивные операции). Унарные операции $+$ и $-$ старше бинарных, стало быть, знак операнда вычисляется в первую очередь.

Таблица 2.7

Полная таблица приоритетов операций C++

Ранг(приоритет)	Операции	Ассоциативность
1	() [] -> (операции разыменования)	→
2	! ~ + - ++ — & * (тип) sizeof	←
3	* / % (бинарные)	→
4	+ - (бинарные)	→
5	<< >> (сдвиг)	→
6	< <= > >= (отношения)	→
7	== != (отношения)	→
8	&	→
9	" (поразрядное исключающее или)	→
10		→
11	&&	→
12		→
13	? : (условная операция)	←
14	= *= /= %= += -= &= " = = <= >=	←
15	, (операция запятая)	→

Операция присваивания и ее клоны младше прочих, что позволяет выполнить присваивание только после того, как значение выражения вычислено полностью. Операции отношения младше арифметических операций, что позволяет использовать естественную запись логических выражений, например, $x > 0 \ \&\& \ y > 0$. Здесь в первую очередь вычисляются значения отношений, которые затем являются операндами конъюнкции.

Пример 2.4.

```
int c, a=5, b=3;
```

```
c=a++;           // c=5, a=6
```

```
c=a++*3;         // c=a*3=15, потом a=7
```

```
c=-a/2;          // -a, a=6, потом c=a/2=3
```

```
c=a+++b;         // сначала c=a+b=9, потом a++, a=7
```

```
c=a++ + ++b;     // ++b=4, потом c=a+b=11, потом a++8
```

2.2. Структура программы на языке C++

Программа может состоять из одной или нескольких функций. Функция — это самостоятельный именованный алгоритм решения некоторой законченной задачи. Одна из функций должна иметь имя `main()`. С нее всегда начинается выполнение программы. Любая функция, кроме `main()`, может быть вызвана из другой функции. При вызове функции могут быть переданы параметры (данные). По окончании выполнения функции в вызывающую функцию может быть возвращено значение (результат), а может не быть возвращено ничего. Примером обращения к функциям являются библиотечные функции, например, `sin(x)`, `fabs(a)`.

Код программы, которая состоит из одной функции `main()`, обычно содержит следующие составляющие, хотя почти все они могут отсутствовать:

#Директивы препроцессора	// начинаются с символа # и
	// записываются в одну строку
Тип_функции тат(параметры)	// заголовок функции

```

{ // блок тела функции;
    определения объектов;
    исполняемые операторы;
    return выражение;           // если функция возвращает
                                значение
}

```

В C++ объявление переменной (объекта) возможно не только в начале программы, но и в любом месте текста до первого обращения к ней. Областью действия объекта является только непосредственно охватывающий его блок. Как правило, так объявляют рабочие переменные.

Блок (операторов) — это произвольная последовательность определений и операторов, заключенная в фигурные скобки:

```

{
// блок;
}

```

Блок используется для укрупнения структуры программы. Точка с запятой в конце блока не ставится.

Текст программы на языке C++ обладает структурой. Существует система правил корректной записи текста программ, которые просты, но улучшают зрительное восприятие и понимание кода.

1. Каждый оператор заканчивается знаком «;». Обычная ошибка начинающего — это знак «;», завершающий заголовки функций или операторов цикла. В первом случае синтаксическая ошибка распознается как отсутствие тела функции, во втором случае телом цикла является пустой оператор, что синтаксической ошибкой не является, и программа выполняется.

2. Каждый оператор записывается в одну строку. Такая запись позволяет структурировать текст программы, наглядно видеть ее алгоритм, и облегчает отладку при пошаговом исполнении.

3. Блок, т. е. произвольный фрагмент текста, заключенный в фигурные

скобки {}, размещается в любом месте программы. Использование блоков позволяет укрупнить структуру алгоритма.

4. Структура программы подчеркивается отступами. Это позволяет визуально показать блоки, составляющие структуру алгоритма.

5. Необходимо использовать разумное количество комментариев.

6. Имена объектов программы выбираются осмысленно. Каждое имя подчеркивает назначение и логику объекта, например, имена библиотечных функций `sin()`, `abs()`, `printf()` и прочих говорят сами за себя. Имена объектов, введенные программистом, подчеркивают их абстрактный смысл, например, `Count`, `Square`, `Point.x`, `Point.y` и т. д.

7. Пробелы в тексте являются значащими только в составе текстовых констант. В коде программы пробелы отделяют друг от друга элементы текста. В остальных случаях их использование произвольно, например, лишние пробелы улучшают читабельность программы.

Директивы начинаются со знака # и записываются в одной строке. Являются командами (директивами), выполняемыми препроцессором на стадии предварительной обработки текста программы, т. е. до ее компиляции. Директив препроцессора достаточно много, на начальном этапе достаточно ознакомиться с двумя из них.

Директива #define

Используется для задания именованных констант и для задания строк подстановки.

Синтаксис.

`#define имя Выражение`

Механизм действия директивы — макроподстановки, т. е. препроцессор сканирует весь текст программы и выполняет замены в тексте программы, везде вместо «Имени» подставляя «Выражение».

Имена `define` определенных констант записываются большими буквами, чтобы визуально отличить их от имен переменных и других объектов.

Пример 2.5.

```
#define N 10          // по всему тексту вместо N число 10
#define PI 3.1416926  // вместо PI его числовое значение
#define STR "Строковая константа, подставляется в текст\n"
```

Директива может не только определить именованную константу, но и выполнить макроподстановки.

Пример 2.6.

```
#define N1 N+1        // вместо N1 текст N+1
#define int long       // в тексте все описания int
                      // заменятся на long
```

Если в замещаемом имени есть скобки, следующие за именем без пробела, то это макроопределение с параметрами:

```
#define Имя(список_параметров) Выражение
```

Например,

```
#define Cube(x) x*x*x
```

Имя здесь играет роль имени макроопределения, а параметров может быть несколько, тогда они отделяются запятыми. Между именем и списком параметров не должно быть пробела. Такое макроопределение может использоваться как функция, хотя макроподстановки не заменяют функции, и иногда могут привести к ошибкам.

Пример 2.7.

```
#define Cube(x) x*x*x
// макроопределение возведения в степень
#include <stdio.h>
void main(void)
{
    int a=2;
    printf("%d %d", a, Cube(a)); // выведено 2 8
}
```

Директива #include

Используется для замены в тексте путем добавления текста из других файлов в точку нахождения `#include`.

Синтаксис.

`#include "filename"`

`#include <filename>`

Механизм действия — включение текста указанного файла в текущее место в программе. Включаемые файлы называются заголовочными и содержат информацию, которая для программы глобальна.

Директива `#include "filename"` осуществляет поиск файла сначала в текущем каталоге, а затем в системных каталогах. Так подключаются личные файлы программиста, содержащие произвольные тексты, например, определения констант, объявления или описания функций.

Директива `#include <filename>` осуществляет поиск файла только в системных каталогах. Так подключаются стандартные заголовочные файлы, поставляемые в комплекте со стандартными библиотеками функций.

Каждая библиотечная функция имеет свое описание (прототип). Кроме того, в заголовочных файлах описаны многие константы, определения типов и макроподстановки. Например, стандартный заголовочный файл `<stdio.h>` содержит описания библиотечных функций ввода и вывода, `<math.h>` подключается при использовании математических функций. Их описание содержится в справочной системе. Следует понимать, что использование `#include` не подключает к программе соответствующую библиотеку, а только включает в текст программы на глобальном уровне все нужные описания и объявления. Сами библиотечные функции подключаются к программе в виде объектного кода на этапе компоновки, когда компоновщик обрабатывает все вызовы функций, определяет, какие потребуются программе, и собирает исполнимый файл, включая в него объектные (компилированные) коды тех функций, к которым выполняется обращение.

Ввести данное — означает присвоить произвольное значение переменной во время выполнения программы.

Вывести данное — означает напечатать на экране значение переменной при выполнении программы.

Простейший из способов обмена данных — это форматированный ввод-вывод с определением правил размещения данных во входном/выходном потоке. Для реализации такого обмена к программе директивой `#include <stdio.h>` подключается библиотека `stdio.h`.

Для ввода значения данного с клавиатуры с эхо повтором на экране используется функция `scanf()`, синтаксис которой:

```
scanf("форматная строка", список_ввода);
```

Здесь «список_ввода» — имена переменных, значения которых будут введены с клавиатуры при выполнении функции `scanf()`. Имена переменных предваряются символом «&», который является признаком адресной операции и означает, что введенное значение пересылается по адресу, определенному именем переменной. При вводе данные отделяются пробелами или нажатием клавиши [Enter].

Для вывода значения данного на экран используется функция `printf()`, синтаксис которой:

```
printf("форматная строка", список_вывода);
```

Здесь «список_вывода» — список имен переменных и выражений (в том числе констант), значения которых появятся на экране при выполнении функции `printf()`.

Форматная (управляющая) строка — это строка символов внутри двойных кавычек, содержащая управляющие символы и текст. При вводе данных функция `scanf()` читает посимвольно текст из входного потока, распознает лексемы и преобразует их в машинное представление в соответствии с признаком формата, сопоставленного переменной, ожидающей данное. При выводе функция `printf()` берет машинное представление значения переменной, соответственно признаку формата преобразует в текстовое представление и выводит на экран.

Число управляющих символов равно числу объектов в списке ввода-вывода. Управляющий символ имеет признак % и может иметь одно из следующих значений: %d — ввод-вывод целого десятичного числа (int),

%u — ввод-вывод целого без знака (unsigned),

%f — ввод-вывод числа с плавающей точкой (float),

%lf — ввод-вывод числа с плавающей точкой (double),

%e — ввод-вывод числа в экспоненциальной форме (double и float), %c — ввод-вывод символа (char),

%l — ввод-вывод длинного значения (long),

и др.

При вводе и выводе необходимо строгое соответствие типа данного управляющему символу формата.

Пример 2.8.

Приведем пример форматированного ввода и вывода:

```
#include <stdio.h>

void main(void)
{
    int my_int;
    float my_float;
    printf("\nBeeguTe целое и дробное число\n");
    scanf("%d", &my_int); scanf("%f", &my_float);
    printf("%d %f", my_int, my_float);
}
```

При запуске программы функция printf() выведет на экран строку — приглашение к вводу данных, затем при выполнении каждой функции scanf() будет ожидать ввода данных. Пользователь должен ввести требуемое количество данных, отделяя их пробелами или нажатием клавиши [Enter]. При завершении ввода данные тут же будут выведены на экран самым примитивным образом. Так, если ввести целое 5 и дробное 9.9, то строка вывода будет иметь вид: 59.900000

Поскольку при вводе данного функция `scanf()` находится в состоянии ожидания ввода, рекомендуется каждый ввод предварять строкой, выводящей на экран приглашение к вводу данных, в котором пользователю подробно объясняют, что и как он должен сделать, чтобы правильно ввести данные, например:

```
printf("Введите координаты центра и радиус круга\n");  
scanf("%f%f%f", &x, &y, &R);
```

При выводе данных для улучшения вывода рекомендуется использовать некоторые приемы.

1. Управляющие символы, например:

"\n" для перевода строки при выводе;

"\t" для выполнения табуляции.

2. Произвольный текст в форматной строке для приглашения на ввод данного и для пояснений при выводе, например, функция вывода может быть записана так: `printf("Целое = %d, Вещественное = %f\n", my_int, my_float);`

Пробелы в строке текста являются значащими. Теперь, если ввести целое 5 и дробное 9.9, то строка вывода будет иметь вид:

целое = 5, Вещественное = 9.900000

3. Модификаторы форматов. Они используются для оформления вывода. По умолчанию (без модификаторов) данные выводятся в поле минимальной ширины с точностью 6 знаков после запятой, число прижимается к правому краю поля. Этим выводом можно управлять:

- можно задать ширину поля как строку цифр, определяющую наименьший размер поля вывода, которое будет игнорировано, если число не входит в поле;
- можно задать точность вывода для вещественных чисел: это две цифры, определяющие общий размер поля вывода и число знаков после запятой.

В следующих примерах обозначим знаком □ пробелы, которые будут значимыми в строке вывода.

Пример 2.9.

```
printf("Целое=%4d, Вещественное=%5.2^п", my_int, my_float);
```

Если ввести значения 10 и 2.3, то строка вывода будет иметь вид:

Целое = □□10, Вещественное = □9.90

Если ввести значения 19951 и 12.9999, то строка вывода будет иметь вид:

целое = 19951, Вещественное = 13.00

Можно сделать вывод, что число округляется.

2.3. Операторы языка C++. Классификация операторов

Оператор — это предложение, описывающее одно действие по обработке данных или действия программы на очередном шаге ее исполнения.

Все операторы языка C++ можно разделить на две группы.

1. Операторы преобразования данных.
2. Операторы управления ходом выполнения программы.

Это разделение соотносится с определением алгоритма.

Операторами *преобразования* данных являются выражения. Оператора присваивания в языке C++ нет, его заменяет выражение, в составе которого есть операция присваивания.

Примеры операторов изменения данных:

`y+4;` // выражение

`x=y+4;` // выражение присваивания

`x++;` // выражение - оператор

Операция присваивания правоассоциативна, поэтому допускается запись цепочек присваиваний, например:

`x=y=z=1;` // каждая переменная будет равна 1

К операторам *преобразования* данных можно отнести оператор обращения к функции (вызов функции). Его операцией являются круглые скобки, а операндами — имя функции и список параметров, например:

```
scanf("%d%f", &a, &b);
```

```
printf("a=%d, b=%f", &a, &b);
```

При выполнении вызова функции `scanf()` значения переменных `a`, `b` будут изменены. При выполнении вызова функции `printf()` выполняются действия по преобразованию данных из внутреннего представления в текстовое, пригодное для вывода на экран.

Операторы *управления* предназначены для управления ходом выполнения программы. Обычно операторы программы выполняются в том порядке, в котором записаны. Поскольку каждый оператор выполняет одно действие по обработке данных, тем самым управляя работой компьютера, то порядок выполнения выражений называют потоком управления. Поток управления редко бывает линейным. Изменить направление потока управления позволяют операторы управления, перечисленные в таблице 2.8.

Таблица 2.8

Операторы управления C++

Название оператора	Ключевое слово
Составной оператор	{ ... }
Условный оператор	if
Оператор цикла с проверкой условия до выполнения	while
Оператор цикла с проверкой условия после выполнения	do. while
Оператор цикла типа «прогрессия»	for
Оператор прерывания	break
Оператор продолжения	continue
Оператор переключатель	switch
Оператор перехода	goto

При описании каждого оператора языка C++ будем придерживаться следующей схемы:

- назначение;
- синтаксис;
- семантика, механизм исполнения;

- пример;
- особенности.

2.4. Оператор присваивания. Программирование алгоритмов линейной структуры

Назначение. Изменение значения данного в соответствии с выражением правой части.

Синтаксис.

имя_переменной = Выражение; // форма 1

имя_переменной = (тип) Выражение; // форма 2

Выполнение. В первую очередь вычисляется выражение правой части, во вторую вычисленное значение присваивается переменной левой части.

Примеры:

x=y=0;

x++;

y=sin(2*PI*x);

y+=0.2;

Особенности. Если в выражении встречаются операнды разных типов, то при вычислении значения выражения происходит неявное *преобразование* типов, как правило, к большему типу. При несоответствии типа выражения правой части типу переменной в левой части происходит *приведение* типа выражения к типу переменной, при этом неизбежна потеря данных при приведении от большего типа к меньшему.

Приведем пример реализации линейного алгоритма вычисления высот треугольника, блок-схема которого приведена в примере 1.1 (глава 1).

Пример 2.10.

// вычислить высоты треугольника со сторонами a, b, c

#include <stdio.h>

#include <conio.h>

#include <math.h> void main(void)


```

{
double a,b,c;
cout<<"Введите стороны треугольника";
scanf("%lf%lf%lf", &a, &b, &c);
double p,t;
// объявить переменные можно там, где они
// понадобились
double Ha,Hb,Hc; p=0.5*(a+b+c);
// нет проверки условия существования треугольника
t=2*sqrt(p*(p-a)*(p-b)*(p-c));
Ha = t/a;
Hb = t/b;
Hc=t/c;
cout<<"Значения высот треугольника\n"
Ha=%6.2lf,Hb=%6.2lf, Hc=%6.2lf\n",Ha, Hb, Hc);
}

```

Ключевые слова:

Алгоритм, компиляция, компоновка, алфавит языка, операции, операторы, структура программы, комментарии, директива define, директива include, синтаксис, оператор, классификация операторов, операторы управления, составной оператор, условный оператор, оператор цикла с проверкой условия до выполнения, оператор цикла с проверкой условия до выполнения, оператор цикла, оператор прерывания, оператор перехода.

Вопросы для самопроверки:

1. Определите назначение и правила записи имен в языке C++.
2. Дайте определение и назначение констант. Как определяется тип константы?

3. Перечислите типы данных. Что такое явное и неявное преобразование типов? Какова роль объявления типов в программе?
4. Поясните роль определения переменной в программе.
5. Почему происходит преобразование типов при присваивании? Поясните механизм преобразования типов.
6. Какова структура программы на языке C++?
7. Что такое директивы препроцессора? Поясните назначение и механизм действия директив `#define` и `#include`.
8. Каково назначение операторов языка программирования? Приведите классификацию операторов языка C++.

Лекция 3. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ И ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

3.1. Условный оператор `if`. Программирование разветвляющихся алгоритмов.

К составным операторам относятся собственно составные операторы и блоки. В обоих случаях это последовательность операторов, заключенная в фигурные скобки. Блок отличается тем, что в его состав входят описания каких-либо объектов программы.

Блоком является тело любой функции, в том числе функция `main()`. Составной оператор используется для объединения нескольких операторов в один. Составной оператор, например, формирует ветвь условного оператора или тело цикла в операторах цикла.

Условный оператор `if`

В языке C++ алгоритм разветвляющейся структуры реализуется условным оператором *`if`*.

Назначение. Выбор одного из двух возможных путей исполнения программы в зависимости от условий, сложившихся при выполнении.

Для проверки условия формулируется некоторое утверждение, которое может быть истинным или ложным. Это логическое выражение, которое может принять одно из значений *True* либо *False*. В языке C++ они имеют форму выражений целого типа, принимая значения «не 0», что соответствует истине, либо «0», что соответствует лжи, либо типа `bool`.

Синтаксис. Синтаксис условного оператора имеет две формы.

Сокращенная форма.

`if (логическое_выражение) оператор;`

Здесь «логическое_выражение» — любое выражение типа `int` или `bool`.

«оператор» — это один или несколько операторов, в общем случае составной оператор. Логическое выражение записывается в скобках.

Полная форма.

`if (Логическое_выражение)`

`оператор1;`

`else`

`оператор2;`

Здесь «логическое_выражение» имеет тот же смысл, что и ранее, а «оператор1" или «оператор2" — это, в общем случае, блок.

Выполнение.

1. Вычисляется значение логического выражения.
2. Если оно не равно 0, выполняется «оператор1", а если оно равно 0, выполняется «оператор2" (или ничего в первой форме).

Пример 3.1. Первая форма условного оператора.

Стоимость товара равна произведению цены на количество. Если есть скидка, то стоимость уменьшается на величину `discount`. Вычислить стоимость.

```
pay=cost*count;           // общая формула
if (discount!=0)           // если есть скидка, то стоимость
                           // уменьшается
pay=pay-(pay*discount)/100;
```

```
// вывод pay в любом случае
cout <<"Стоимость = %6.2f\n"<< pay;
```

Пример 3.2. Вторая форма условного оператора.

Оплата труда работника — это произведение количества отработанных часов на стоимость часа. Если отработано более 40 ч, то за каждый час работодатель платит в полтора раза больше.

```
if(hour<40)
    pay=rate*hour;                // обычная оплата,
                                   // hour < 40
else
    pay=rate*40+(hour-40)*rate*1.5; // повышенная оплата
                                   // hour >= 40

// печать одинакова
printf("К оплате %6.2f рублей.\n ", pay);
```

Пример 3.3. Использование блоков в составе условного оператора.

Если необходимо вывести на экран число оплаченных часов и значение суммы к оплате, тогда для каждой ветви нужна своя собственная печать.

```
if(hour<40)
{
    pay = rate*hour; // обычная оплата
    printf("Оплачено %d часов, к оплате %6.2f руб.\n ", hour, pay);
}
else
{
    pay=rate*40+(hour-40)*rate*1.5; // повышенная оплата
    printf("Оплачено %d часов, %d сверхурочно.\n",
    hour, hour-40);
    printf("К оплате %6.2f руб. \n", pay);
}
```

В состав операторов, формирующих ветви условного оператора, могут входить любые операторы, в том числе условные. В этом случае говорят о вложенном условном операторе. Каждая из ветвей, в свою очередь, может содержать условный оператор или несколько. Число уровней вложения не ограничено, однако чем их больше, тем сложнее восприятие текста.

Пример 3.4. Вычислить значение функции

$$y = \begin{cases} x + a, & \text{при } x < 10; \\ x + b, & \text{при } 10 \leq x \leq 20; \\ x + c, & \text{при } x > 20. \end{cases}$$

```
void main(void)
{
float a,b,c;
float x,y;
printf("Введите параметры функции");
scanf("%f%f%f",&a,&b,&c);
printf("Введите аргумент функции");
scanf("%f",&x);
if(x<10)
y=x+a;
else
if(x<20)
y=x+b;
else
y=x+c;
printf("x = %6.2f,y = %6.2f\n",x,y);
getch();
}
```

Логическое выражение в условном операторе может быть как угодно сложным. Для записи сложных логических выражений используются специальные логические операции:

&& — логическое «и»;

|| — логическое «или»;

! — логическое отрицание «не».

Например, для обозначения категории «подростковый возраст» нужна условная градация, скажем, от 12 до 17 лет. Если переменная Age обозначает возраст некоего круга людей, то подростками из них будут только те, для которых справедливо утверждение $Age \geq 12 \ \&\& \ Age \leq 17$. Логическое выражение будет истинно, только когда оба условия выполняются одновременно. Соответственно, если $Age < 12 || Age > 17$, то человек не подросток (это ребенок, если $Age < 12$, или взрослый человек, если $Age > 17$). Это логическое выражение будет истинно, когда хотя бы одно или оба условия выполнены. Для того чтобы проиллюстрировать применение операции отрицания, запишем условие «не подросток» инверсией выражения «подростковый возраст» $!(Age \geq 12 \ \&\& \ Age \leq 17)$.

Приоритеты операций приведены ранее в таблице 2.7.

Наиболее распространены следующие ошибки записи логических выражений.

1. Знаки логических операций похожи на знаки поразрядных операций сравнения & (и) и | (или). Выполнение этих операций происходит различным образом, поэтому не всегда результат поразрядного сложения равен истинной дизъюнкции операндов, а результат поразрядного умножения равен истинной конъюнкции (хотя иногда равен).

2. Знак операции = отличен от знака ==. Знаки операций = и == похожи внешним образом, но радикально отличаются по механизму выполнения.

Каждый программист хотя бы раз допустил ошибку вида:

```
if(Key=27) // ошибка здесь
{
    printf ("Завершение работы.\n");
    return;
}
```

```
else  
printf ("Продолжение работы.\n");
```

Здесь по смыслу алгоритма значение переменной Key определяет момент завершения или продолжения работы программы. Однако в этом тексте, независимо от первоначального значения Key, будет выполнено присваивание Key=27, и значение выражения станет равно 1. Значит, результат проверки значения выражения равен 1, всегда выполняется первая ветка, и программа благополучно всегда завершит свою работу.

3. Пропуск знаков логических операций. Математическая запись $3 < x < 6$ должна быть записана на языке C++ обязательно с использованием логической операции `&&`: `if(-3<x && x<6)`, хотя компилятор согласится с текстом вида `if(-3 < x < 6)`.

При вычислении выражения $-3 < x < 6$ сначала вычисляется $(-3 < x)$. Это выражение имеет логическое значение (0 или 1). Затем полученное логическое значение участвует в вычислении выражения $(-3 < x) < 6$, результат которого всегда истинен.

4. Операции сравнения для вещественных типов. Операции сравнения могут применяться к данным любого из базовых типов C++. Для данных целого и символьного типов точное соответствие возможно. Для данных вещественных типов всегда существует ошибка представления данных, которая может накапливаться. Поэтому проверки на точное равенство вещественных чисел следует избегать.

Пример 3.5.

```
#include <stdio.h>  
  
void main(void)  
{  
    float x=1./3.;  
    float y;  
    y=x+x+x;  
    if(y == 3*x)
```

```
printf("Равны\n");
else
printf("Не равны\n");
}
```

Казалось бы, y должен быть равен $3 \cdot x$, но в силу ошибки представления данных вещественного типа этого не произойдет.

Проверка вещественных чисел выполняется на приближенное равенство с использованием приближенного значения модуля разности сравниваемых значений:

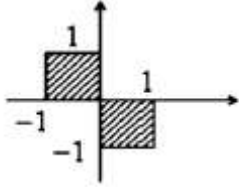
```
(fabs(y-3*x)<0.001);
// вычисление, например с точностью 0.001
```

Примеры записи логических выражений приведены в таблице 3.1

Таблица 3.1

Примеры записи логических выражений

№ п/п	Условие	Математическая формулировка	Синтаксис записи на C++
1	Число X принадлежит отрезку $[-2, +2]$	$-2 \leq x \leq +2$ или $ x \leq 2$	$-2. \leq x \ \&\& \ x \leq 2.$ или $\text{fabs}(x) \leq 2.$
2	Число X принадлежит отрезку $[-2, -4]$ или отрезку $[+2, +4]$	$-4 \leq x \leq -2;$ $+2 \leq x \leq +4$	$-4. \leq x \ \&\& \ x \leq -2. \ $ $2. \leq x \ \&\& \ x \leq 4.$
3	Точка с координатами (x, y) находится в первой четверти	$\begin{cases} x > 0 \\ y > 0 \end{cases}$	$x > 0 \ \&\& \ y > 0$
4	Точка с координатами (x, y) находится в первой или в третьей четверти	$\begin{cases} x > 0 \\ y > 0 \end{cases}$ $\begin{cases} x < 0 \\ y < 0 \end{cases}$	$x > 0 \ \&\& \ y > 0 \ \ x < 0 \ \&\& \ y < 0$
5	Числа a, b, c одновременно четны	Каждое число делится на 2 без остатка	$a \% 2 == 0 \ \&\& \ b \% 2 == 0 \ \&\& \ c \% 2 == 0$
6	Хотя бы одно из чисел a, b, c четно	Одно или два, или все три числа делятся на 2 без остатка	$a \% 2 == 0 \ \ b \% 2 == 0 \ \ c \% 2 == 0$
7	Символ является знаком препинания		<code>char d; // объявление</code> <code>d == '.' d == ','</code> <code> d == '?' d == '!'</code>
8	Символ является буквой		<code>char d; // объявление</code>

	латинского алфавита		$d>='A' \& \& d<='Z' \ $ $d>='a' \& \& d<='z'$
9	Точка с координатами (x, y) принадлежит указанной области 	Во второй четверти $\begin{cases} x > -1 \\ y < 1 \end{cases}$ В четвертой четверти $\begin{cases} x < 1 \\ y > -1 \end{cases}$	$x > -1 \& \& x < 0 \& \& y > 0$ $\& \& y < 1 \ x > 0 \& \&$ $x < 1 \& \& y < 0 \& \& y > -1$
10	Точка с координатами (x, y) находится выше прямой $y = ax + b$	При подстановке в уравнение прямой $y_1 = ax + b$: если $y_1 > y$, то выше, иначе ниже.	// Значения a, b, x, y известны. $a * x + b > y$

3.2. Оператор цикла while

Рассмотрим реализацию циклов в языке C++.

Для реализации цикла с предусловием используется оператор цикла while.

Назначение. Организация многократного повторения фрагмента программы.

Синтаксис.

```
while (логическое_выражение)
{ // Тело цикла;
}
```

Здесь «Тело цикла» — один или несколько операторов, в общем случае составной оператор. «логическое_выражение» — это выражение условия завершения цикла.

Выполнение. Тело цикла выполняется, пока «логическое_выражение» имеет значение «истинно» (отлично от 0). Когда «логическое_выражение» ложно (равно 0), управление передается оператору, стоящему за циклом.

Пример 3.6. Найти сумму n чисел натурального ряда

$$S = \sum_{i=1}^n i$$

Параметр цикла $i \in [1, n]$, шаг изменения параметра $\Delta i = 1$. Тело цикла $S = S + i$.

```
void main(void)
{
    int n, Sum = 0;
    int i;
    printf("\nВведите количество элементов ряда\n");
    scanf("%d", &n);
    i=1;
    while(i <= n)
    {
        Sum+=i;    // очередное сложение
        i++;    // увеличение слагаемого
    }
    printf("Сумма чисел натурального ряда=%d", Sum);
}
```

Поскольку значение суммы не зависит от порядка суммирования, то в стиле языка C++ алгоритм запишется так:

```
i=n;
Sum=0;
while(i)    // пока i отлично от 0
    Sum+=i--;
```

Особенности. Проверка условия происходит до входа в цикл, поэтому для заведомо ложного выражения тело цикла не будет выполнено ни разу.

Оператор цикла do... while

Оператор цикла do ... while используется для реализации цикла с предусловием.

Назначение. Организация многократного повторения фрагмента программы.

Синтаксис.

```
do
{
// тело цикла;
} while(Логическое_выражение);
```

Здесь «тело цикла» — один или несколько операторов, в общем случае составной оператор. «Логическое_выражение» — это выражение условия завершения цикла.

Выполнение. Тело цикла (в общем случае блок) выполняется многократно, пока «логическое_выражение» не равно 0. Как только выражение станет ложно (т. е. равно 0), цикл заканчивается, управление передается следующему по порядку оператору.

Пример 3.7. Решим задачу с использованием оператора цикла do.

```
void main(void)
{
int n, Sum = 0;
int i;
printf("\nвведите количество элементов ряда \n");
scanf("%d", &n);
i= n;
do
Sum+=i—;
while(i);    // пока i отлично от 0
printf("Сумма чисел натурального ряда=%d", Sum);
}
```

Особенности. Проверка условия происходит после выполнения тела цикла. Значит, как бы ни было задано «логическое_выражение», оператор тела цикла выполнится не менее чем один раз.

Оператор цикла for

Назначение. Организация многократного повторения произвольного фрагмента кода программы. Как правило, цикл используется, когда число

повторений тела цикла известно заранее и явно есть переменная величина, которая, изменяясь, составляет прогрессию. Она является параметром цикла. Иногда ее называют счетчик цикла.

Синтаксис.

```
for (Выражение1; Выражение2; Выражение3)  
// тело цикла;
```

Здесь параметр цикла составляет прогрессию и явно присутствует в записи заголовка цикла (так называется строка `for(...)`).

«Выражение1» задает начальное значение параметра цикла, «Выражение2» задает условие завершения выполнения тела цикла и является логическим, «Выражение3» задает приращение параметра цикла. «тело цикла» — это произвольная последовательность операторов, как правило, составной оператор. Первое и третье выражения могут состоять из нескольких операторов, отделенных друг от друга запятой.

Выполнение. Перед входом в цикл, т. е. до первого выполнения тела цикла, однократно выполняется «Выражение1». Тело цикла выполняется многократно, пока «Выражение2» (условие) не равно 0. Как только его значение становится равным 0, управление передается следующему по порядку оператору. Проверка условия завершения цикла происходит до выполнения тела цикла.

Пример 3.8. Решим задачу с использованием цикла `for`.

```
void main(void)  
{  
    int n, Sum = 0;  
    int i;  
    printf("\nВведите количество элементов ряда \n");  
    scanf("%d",&n);  
    for(i=1; i<=n; i++)  
        Sum+= i;  
    printf("Сумма членов натурального ряда=%d", Sum);
```

}

Особенности.

1. Управляющая переменная цикла `for` не обязательно целого типа, она может быть вещественной или символьной.

Приведем пример цикла, вычисляющего сумму геометрической прогрессии.

Пример 3.9. Вычислить сумму геометрической прогрессии по формуле

$$S = \sum_{i=1}^n 1,1 \cdot i$$

где i изменяется по закону $i = i \cdot 1,1$.

```
void main(void)
{
    float Sum = 1, i;
    for(i = 1.; i <= 2.; i* = 1.1)
        Sum+ = i;
    printf("\nСумма геометрической прогрессии %f", Sum);
}
```

2. Любое из трех выражений, стоящих в заголовке цикла, любые два или все могут отсутствовать, но разделяющие их символы «;» опускать нельзя. Если «Выражение2» отсутствует, то считается, что оно истинно, и цикл превращается в бесконечный, для выхода из которого необходимы специальные средства. Бессмысленно использовать несколько «Выражений2», так как управление циклом выполняется по первому условию.

3. Оператор цикла `for` удобен тем, что интегрирует в заголовке описание всего процесса управления. Телом цикла является, в общем случае, блок. Поскольку начальных присваиваний и выражений приращения может быть несколько, весь цикл вместе с телом может быть записан одной строкой. Например, тот же цикл в стиле языка C++ может выглядеть так:

```
for(Sum=0,i=1;i<=n; Sum+=i++);
```

Здесь два начальных присваивания, а тело цикла $Sum+=i++$ вместе с приращением параметра цикла записано в его заголовке.

Приведем код программы вычисления факториала функции с применением циклов `while` и `for`.

Пример 3.10. Дано целое положительное число n . Вычислить факториал этого числа.

// первый вариант

```
void main(void)
{
    int N;
    long int F;
    int R;
    printf("Введите число\n");
    scanf("%d",&N);
    F=R=1; // присваивание начальных значений
    while (R<=N)
    {
        F = F*R;
        R++;
    }
    printf ("Факториал %d равен %ld\n",N,F);
}
```

// второй вариант

```
void main(void)
{
    int N;
    long int F;
    int R;
    printf("Введи число\n");
    scanf("%d",&N);
```

```

F=1;                // начальное значение факториала
for(R=1; R<=N; R++) // здесь R - параметр цикла
F=F*R;
printf("Факториал %d равен %ld\n",N,F);
}

```

При использовании цикла `while` программист должен явно указать начальное значение параметра цикла (в программе это переменная `R`), а в теле цикла изменить его значение. При использовании цикла `for` эти действия выполняются в заголовке цикла.

Пример 3.11. Даны натуральное число n , действительное число x . Вычислить сумму:

$$\sum_{i=1}^n \frac{x^i}{i!}.$$

Прежде чем написать программный код для решения этой задачи, необходимо вывести рекуррентное соотношение, которое определит способ вычисления слагаемого на каждом шаге выполнения цикла.

```

void main(void)
{
    int i,n;
    double c,x,S;
    printf("Введите число слагаемых.\n");
    scanf("%d",&n);
    printf("Введите x\n");
    scanf("%lf",&x);
    S=0;
    c= 1;
    for(i=1; i<=n; i++)
    {
        c=c*x/i;    // неявное приведение типов
    }
}

```

```

S+= c;
}
printf("Сумма равна %8.3lf", S);
}

```

Пример 3.13. Даны действительные числа x , ε ($x \neq 0$, $\varepsilon > 0$). Вычислить сумму с точностью ε :

$$\sum_{k=0}^{\infty} \frac{x^k}{2^k \cdot k!}$$

```

void main(void)
{
double x,eps;
double c,S;
double F; // или long double, значение может быть
           // большим.
int k;
printf("Введите точность вычислений.\n");
scanf("%lf",&eps);
printf("Введите x\n");
scanf("%lf",&x);
F= 1;
S=0;
c=1;
k=1;
while(fabs(c/F)>eps) // итерационный цикл
{
S+=c/F;
k+=1;
c=c*x*0.5;
F*= k;
}
printf("Сумма равна %lf\n",S);
}

```


Ключевые слова:

Алгоритм разветвляющейся структуры, алгоритм циклической структуры, операторы цикла, формат операторов, логическое выражение, блок- схема алгоритма, число повторений, тело цикла, тело программы, фрагмент программы, счетчик цикла, управление программой, проверка, условие цикла, начальное значение параметра цикла, шаг изменения.

Вопросы для самопроверки:

1. Что такое составной оператор? Для чего он используется?
2. Формат условного оператора if.
3. Назначение условного оператора if.
4. В каких случаях используется оператор while?
5. В каких случаях используется оператор do...while?
6. Формат оператора for.
7. Назначение оператора цикла for.
8. Оператор перехода GOTO, его назначение.
9. Приведите пример разветвляющихся и циклических операторов.

Лекция 4. МАССИВЫ. ПРОГРАММИРОВАНИЕ ОДНОМЕРНЫХ МАССИВОВ

4.1. Понятие массива. Виды массивов. Одномерные массивы

Массив — это упорядоченное множество данных одного типа, объединенных общим именем.

Тип элементов массива может быть почти любым типом C++, не обязательно это базовый тип. Обязательные атрибуты массива — это тип элементов, число элементов и размерность. Различают одномерные и многомерные массивы.

Алгоритмы работы с одномерными массивами основаны на использовании циклических алгоритмов, в которых выполняется

последовательное обращение к элементам массива. Управляющей переменной в таких алгоритмах должен быть индекс массива.

Массивы большей размерности трактуются как массивы массивов. Все особенности использования статических многомерных массивов можно рассмотреть на примере двумерных массивов (матриц).

Алгоритмы работы с двумерными массивами похожи во многом на алгоритмы работы с одномерными массивами. В большинстве случаев требуется обращение ко всем элементам матрицы, тогда алгоритм просмотра содержит вложенный цикл, у которого во внешнем цикле управляющей переменной служит номер строки матрицы, а во внутреннем — номер столбца. Тогда просмотр элементов происходит построчно.

4.2. Описание массивов в программе

Чтобы ввести в употребление массив, его необходимо *описать* (конструировать). Можно одновременно выполнить *инициализацию* элементов массива. Смысл описания и объявления для массивов идентичен. Описание массива выполняется, как и для обычных переменных, в разделе описаний.

Синтаксис. тип_массива имя_массива [количество_элементов];

Здесь «количество_элементов» — это константа, заданная явно или именованная (константное выражение).

Пример 4.1.

```
#define N 10
```

```
int mas[3];           // одномерный массив mas содержит 3  
                      // элемента целого типа
```

```
int matr[2][10];      // одномерный массив matr содержит два  
                      // одномерных массива по 10 элементов  
                      // в каждом
```

```
float w[N];           // одномерный массив w содержит N  
                      // вещественных элементов
```

```
char c[5][80];        // массив из 5 строк, по 80 символов в строке
```

Примечание. Элементы массива нумеруются с 0, т. е. для массива, описанного как `int mas[3]` имеются элементы `mas[0]`, `mas[1]`, `mas[2]`.

При объявлении массива имя массива сопоставляется всей совокупности значений. Элементы массива размещаются в памяти подряд в соответствии с ростом индекса (номера элемента внутри массива). Размер выделяемой памяти такой, чтобы поместить значения всех элементов. Такие массивы называются *статическими*, место в памяти выделяется на этапе компиляции, именно поэтому размер массива определен константой.

Чтобы определить общий размер памяти, занимаемой массивом, можно использовать операцию `sizeof()`. Как известно, ее аргументом может быть имя объекта или имя типа, в том числе объекта сконструированного типа.

```
sizeof(имя_типа)           // в байтах для любого данного
                           // этого типа

sizeof(имя_переменной)     // в байтах для этой переменной

sizeof(имя_массива)        // в байтах для переменной
                           // с учетом длины
```

4.3. Программирование одномерных массивов

Над массивом как единой структурой никакие операции не определены. Над данными, входящими в массив, набор операций определен их типом. При работе с массивом можно обращаться только к отдельным его элементам. Операция обращения к одному элементу массива называется операцией *разыменования*. Это бинарная операция `[]`, синтаксис которой: `имя_массива[индекс]`.

Первый операнд «`имя_массива`» показывает, что происходит обращение к данному, в составе которого много значений, т. е. ко всем элементам массива.

Второй операнд «`индекс`» дает возможность выделить один элемент из группы, и может быть только целочисленным. В общем случае, индекс — это

выражение целого типа, определяющее номер элемента внутри массива (счет с нуля), например:

```
mas[0];           // обращение к первому элементу массива
                  // (его номер равен 0)

matr[1][3];       // обращение к элементу матрицы, стоящему
                  // на пересечении 1-ой строки и 3-го столбца

char c[i][2];     // обращение ко 2-му символу i-ой строки текста
```

Замечание. Контроль выхода индекса за границы массива не существует.

Пусть имеем массив `int mas[4];`

На рисунке 4.1. показано размещение элементов этого массива в памяти.

	Имя <i>mas</i> сопоставлено всей совокупности данных				
...	<i>mas</i> [0]	<i>mas</i> [1]	<i>mas</i> [2]	<i>mas</i> [3]	...
	Индекс = 0	Индекс = 1	Индекс = 2	Индекс = 3	

Рис. 4.1. Размещение в памяти элементов одномерного массива

Для каждого элемента массива выделено по 4 байта, как для данного типа `int`. В целом для всего массива выделено $4 \times 4 = 16$ байт. Значения элементов массива неизвестны. Приведем программный код некоторых задач.

Пример 4.2. Сформировать новый одномерный массив из положительных элементов заданного массива.

```
int New_Arr(int x[],int n,int y[],int &k)
{
    k= 0;
    for(int i = 0;i<n;i++)
        if(x[i]>0)
        {
```

```

y[k] = x[i];
k++;
}
return k;          // равенство нулю значения функции
                   // свидетельствует о неуспехе
}

```

Пример 4.3. Вычислить сумму элементов одномерного массива.

```

int Sum(int a[],int Len)
{
    int S = 0;
    for(int i = 0;i<Len;i++)
        S+=a[i];
    return S;
}

```

Пример 4.4. Найти количество положительных и отрицательных чисел в данном массиве.

```

void Counts(int a[],int Len, int &K1,int &K2)
{
    K1 = K2 = 0;
    for(int i = 0;i<Len;i++)
    {
        if(a[i]>0)
            K1++;
        else
            if(a[i]<0)
                K2++;
    }
}

```

Пример 4.5. Дан массив произвольной длины. Найти наибольший элемент массива и определить его номер. Практически, если известен номер

элемента, то известно и его значение. Поэтому в этом коде отыскивается не «значение и номер», а только номер.

```
int N_max(int a[],int Len)
{
    int K = 0;    // номер наибольшего элемента
    for(int i = 1;i<Len;i++)
        if(a[i]>a[K])
            K=i;
    return K;
}
```

Пример 4.6. Удалить из массива наибольший элемент.

```
void Del_max(int a[],int &Len)
{
    // номер наибольшего элемента находит функция N_max,
    // к ней нужно обратиться
    int k; k=N_max(a,Len);
    // сдвиг от k-го элемента до конца массива
    for(int i=k;i<Len-1;i++)
        a[i] = a[i+1];
    Len--;    // параметр передается по адресу,
              // длина массива изменится
}
```

Пример 4.7. Определить, является ли заданная последовательность различных чисел a_1, a_2, \dots, a_n убывающей.

```
Int Decrease(int a[],int Len)
{
    int flag=1;    // flag=1 – последовательность убывает
                  // flag=0 – последовательность не убывает
    int i;
    i=1;
```

```

while(i<Len&&flag==1)
if(a[i]>a[i+1])
i++;
else
    flag=0;
return flag; }
else
i++;
return flag;
}

```

Ключевые слова:

Массив, одномерный массив, двумерный массив, имя массива, матрицы, строки, столбцы, внешний цикл, внутренний цикл, элементы массива, размерность, число массива, описание массивов, аргумент, имя массива, операнд, операторы цикла, операторы перехода, условного операторы.

Вопросы для самопроверки:

1. Что такое массив? Какие виды массивов вы знаете?
2. Как массивы описываются в программе?
3. Сущность одномерного массива?
4. Что показывает «имя массива»?
5. Что такое индекс массива?
6. Что такое двумерный массив?
7. Объясните расположение элементов двумерного массива при программировании?

8. Какой тип алгоритма используется при программировании массивов?
9. Как размещаются элементы массива в памяти?
10. Какие операторы используются при программировании одномерных массивов?

Лекция 5. ПРОГРАММИРОВАНИЕ ДВУМЕРНЫХ МАССИВОВ

5.1. Двумерные массивы. Описание двумерных массивов

В синтаксисе C++ определены только одномерные массивы. Массивы большей размерности трактуются как массивы массивов. Все особенности использования статических многомерных массивов можно рассмотреть на примере двумерных массивов.

Описание двумерного массива требует, чтобы были указаны два размера: число строк и число столбцов матрицы, например:

```
int a[2][5];           // массив из 2-х одномерных массивов,  
                       // по 5 элементов в каждом
```

В памяти элементы двумерного массива размещаются построчно подряд по возрастанию индексов. Индексы, как и раньше, нумеруются с 0.

Пример инициализации двумерных массивов:

```
int a[3][4]={  
    {1,2,3,4}, // значения элементов нулевой строки  
    {2,4,6,8}, // значения элементов первой строки  
    {9,8,7,6}  // значения элементов второй строки  
};
```

При размещении в памяти будет выделено место под запись $3 \times 4 = 12$ элементов, и они будут размещены линейно, как на рисунке 5.1, в следующем порядке.

1	2	3	4	2	4	6	8	9	8	7	6
Нулевая				Первая строка				Вторая строка			

Рис. 2.7. Размещение элементов двумерного массива

Операции над двумерным массивом не разрешены, кроме операции доступа к его элементам (разыменования). Обращение к произвольному элементу массива выполняется по индексу: $a[i][j]$ — адресуется элемент массива, стоящий на пересечении i -го столбца и j -й строки.

5.2. Алгоритмы работы с двумерными массивами

Алгоритмы работы с двумерными массивами похожи во многом на алгоритмы работы с одномерными массивами. В большинстве случаев требуется обращение ко всем элементам матрицы, тогда алгоритм просмотра содержит вложенный цикл, у которого во внешнем цикле управляющей переменной служит номер строки матрицы, а во внутреннем — номер столбца. Тогда просмотр элементов происходит построчно.

Пусть есть описание матрицы:

```
#define N 5
```

```
#define M 7
```

```
int a[N][M];
```

Здесь размер матрицы определяют `define`-константы. Для матрицы условно переменной длины следует описать размер по максимуму и ввести переменные, реально определяющие размер матрицы.

```
int a[25][25];
```

```
// максимальное число строк, столбцов
```

```
int n, m; // реальное число строк, столбцов
```

```
printf("Введите размер матрицы (строк, столбцов <25)\n");
```

```
scanf("%d%d",&n,&m); // теперь это наибольшие значения
```

```
// параметров циклов управления
```

Для обращения к элементам матрицы по строкам цикл записывается так:

```
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
// в теле цикла обращение к переменной a[i][j]
}
```

Если поменять эти циклы местами, просмотр будет происходить по столбцам:

```
for(j=0;j<m;j++)
for(i=0;i<n; ++i)
{
// в теле цикла обращение к переменной a[i][j]
}
```

Использование функций при обработке матриц предполагает два кардинально различных подхода. В первом случае матрица существует или рассматривается как самостоятельная структура данных, к которой необходимо применить какой-либо алгоритм обработки. В качестве параметра такой функции будет использована вся матрица целиком. Для решения задачи обработки в общем виде, в функцию следует передать имя матрицы и ее размеры (как параметры функции), при этом функция получает матрицу как двумерный массив. Компилятор C++ должен знать, каков способ разбиения этой структуры на строки, поэтому число строк матрицы можно опустить, а число данных в каждой строке опускать нельзя. Оно должно быть указано обязательно как константное выражение в квадратных скобках при передаче матрицы. Так, прототип функции, получающей матрицу в качестве входного данных, может выглядеть так:

```
int function(int a[][M],int n,int m);
// здесь M - константное выражение
```

В качестве примера рассмотрим функции для ввода и вывода матрицы на экран. Предполагается, что матрица условно переменного размера, поэтому число строк и столбцов матрицы по описанию определено define-константами $N = 5$ и $M = 5$. Реальный размер матрицы определяют переменные n, m , значения которых вводятся.

```
#include <stdio.h>

#define N 5
#define M 5

void input_matr(int a[][M],int &n,int &m);
// важно, что длина строки - это константа

void print_matr(int a[][M],int n,int m);
// n - число строк, m - число столбцов

void main(void)
{
    // главная программа описывает входные данные
    int n, m;                // реальные размеры матрицы
    int matr[N][M];          // описан двумерный массив
    input_matr(matr,n,m);    // передан в функцию ввода
    print_matr(matr,n,m);    // передан в функцию вывода
}

// описание функции ввода
// параметры функции - имя и размеры массива
void input_matr(int a[][M],int &n,int &m)
// n, m возвращаются по ссылке
{
    int i, j;
    printf ("Введите размер матрицы не более %d на %d\n", N, M);
    scanf("%d%d", &n,&m);
    printf("Введите матрицу.\n");
    for(i=0;i<n;i++)
```

```

        for(j=0;j<m;j++)
            scanf("%d", &matr[i][j]);
    }
    // описание функции вывода
    // параметры функции - имя и размеры массива
    void print_matr(int a[][M],int n,int m)
    {
        int i, j;
        for(i=0;i<n;i++)
        {
            for(j=0;j<m;j++)
                printf("%5d",mas[i][j]);
            printf("\n");           // разбиение вывода на строки
        }
    }

```

Во втором случае матрица рассматривается как массив из одномерных массивов, при этом для обработки отдельных строк матрицы можно использовать функции обработки одномерных массивов. Зная, что элементы матрицы — это одномерные массивы, каждый из них можно по очереди передавать в функцию, которая работает с одномерным массивом.

5.3. Примеры программирования двумерных массивов

Приведем примеры программирования двумерных массивов.

Пример 5.1. Вычислить сумму элементов строк заданного двумерного массива.

```

void Sum_Rows(int a[][N],int n,int m,int S[])
{
    int i,j;
    for(i = 0;i<n;i++)
    {

```

```

S[i] = 0;
for(j = 0;j<m;j++)
S[i] = S[i]+a[i][j];
}
}

```

Пример 5.2. Найти наибольший элемент двумерного массива и его индексы.

```

int Maxx(int a[][N],int n,int m,int &Maxi,int &Maxj)
{
int Max = a[0][0];
int i,j;
Maxi = Maxj = 0;
for(i = 0;i<n;i++)
for(j = 0;j<m;j++)
if(a[i][j]>Max)
{
Max=a[i][j];
Maxi=i;
Maxj=j;
}
return Max;
}

```

Пример 5.3. Вставить в двумерный массив строку, состоящую из нулей, после строки с номером k.

```

int Insert_Row(int a[][N],int &n,int m,int k)
{
int i,j;
if(k>m)
return 0; // если неверно задано значение k
for(i = n-1;i>= k;i--) // раздвигаем строки

```

```

for(j = 0;j<m;j++)
a[i+1][j] = a[i][j];
n++;
// заполняем нулями освободившуюся строку
for(j = 0;j<m;j++)
a[k][j] = 0;
return 1;
}

```

Пример 5.4. Удалить из двумерного массива строку с номером k.

```

int Del_Row(int a[][N],int &n,int m,int k)
{
int i,j;
if(k>m)
return 0; // если неверно задано значение k
// сдвигаем строки
for(i = k;i<n-1;i++)
for(j = 0;j<m;j++)
a[i][j] = a[i+1][j];
n--;
return 1;
}

```

Пример 5.5. Дана матрица A[n, m]. Найти количество строк, содержащих хотя бы один нуль.

```

int Cou_Rows(int a[][N],int n,int m)
{
int i,j;
int k = 0;
int flag;
for(i = 0;i<n;i++)
{

```

```

j= 0;
flag = 0;
while(j<m && flag == 0)
{
if(a[i][j] == 0)
{
flag = 1;
k++;
}
else j++;
}
}
return k;
}

```

Ключевые слова:

Массив, многомерные массивы, двумерные массивы, строки, столбцы, описание массивов, индекс массива, описание массивов, элементы массива, вложенный цикл, внешний цикл, внутренний цикл, параметры цикла, размер массива, управляющая переменная, директива, операнд, синтаксис, оператор, размерность, число элементов, аргумент, имя массива.

Вопросы для самопроверки:

1. Как описываются двумерные массивы?
2. Что такое элементы матрицы?
3. Как описываются элементы двумерного массива?
4. Как размещаются элементы массива в памяти?
5. Какие операторы используются при программировании двумерных массивов?
6. Что такое вложенный цикл?

7. Что такое внешний и внутренний цикл?
8. Как описываются строки двумерных массивов?
9. Как описываются столбцы двумерных массивов?
10. Как организовать печать результатов?

ГЛАВА III. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ

Лекция 1. МОДЕЛИРОВАНИЕ И СОВРЕМЕННЫЕ МЕТОДЫ ВЫЧИСЛЕНИЙ В ЭКОЛОГИЧЕСКИХ ПРОЦЕССАХ

1.1. Понятие модели и моделирования

Термин «модель» происходит от латинского *modulus* – образец, норма, мера. Модель является частным случаем аналогии – важного метода научного познания. В любых отраслях знания при объяснении сложных явлений или процессов исследователь чаще всего ищет сходства с тем, что уже известно науке. Таким образом, люди стремятся к объяснению неизвестного, непонятного через известное и уже понятное.

Сходство или аналогию в жизни можно встретить повсеместно. Например, макет (модель) здания воспроизводит его архитектуру, топографо-геодезическая карта местности говорит о характере ландшафта, модель корабля или самолета свидетельствует об их внешнем виде, возможностях, пропорциях. Наиболее известны три типа моделей: геометрические, физические, математические.

Геометрические модели представляют некоторый объект, геометрически подобный своему прототипу (оригиналу). Они дают внешнее представление об оригинале и большей частью служат для демонстрационных целей. К этому виду моделей можно отнести репродукции или копии картин, написанных одинаковыми красками по определенной технологии, других живописных изделий (икон, фресок); слепки, выполненные в натуральную величину из того же материала; демонстрационные модели деталей машины, муляжи плодов и др. Чаще, однако, модели выполняются в другом масштабе (макет здания, модель корабля, топографо-геодезический макет местности, модель почвенного разреза).

При построении данного типа моделей основную роль играет геометрическое подобие объектам, а не процессам, протекающим в них.

Например, модель тела человека в гипсе или бронзе ничего не говорит о физиологических процессах, протекающих в его организме, тапографо-геодезический макет местности – о кругообороте воды в природе, а модель почвенного разреза – о физико-химических процессах, протекающих в данном типе почв.

Физические модели отражают подобие между оригиналом и моделью не только с точки зрения их формы и геометрических пропорций, но и точки зрения происходящих в них основных физических процессов. По своей природе они могут быть механическими, гидравлическими, электрическими.

При физическом моделировании модель и ее прототип всегда являются объектами, имеющими одинаковую физическую природу. Типичные примеры – определение аэродинамических свойств летательных аппаратов путем «продувки» их моделей в аэродинамической трубе, исследование предполагаемого «поведения» гидротехнических сооружений (плотин, дамб и т. д.) путем проведения испытаний аналогичных объектов значительно меньших размеров, сконструированных специально для этих целей, и т. д. В данном случае изменяются не только геометрические размеры модели, но соответственно им и другие физические свойства объекта. Например, при построении модели плотины 1/10 натуральной величины в 10 раз уменьшается и давление на нее воды, что должно учитываться в дальнейшем при строительстве.

Геометрические и физические модели относятся к классу вещественных моделей. Они являются или материальными копиями, или физически действующими устройствами (например, модель трактора или ирригационной системы), точно копируя объект или заметно отличаясь от него, сохраняя общность лишь в принципе строения или функционирования.

Математические модели представляют собой абстрактные описания объектов, явлений или процессов с помощью знаков (символов), поэтому их называют также абстрактными или знаковыми. Обычно они имеют вид некой совокупности уравнений или неравенств, таблиц, графиков, формул и

других средств математического описания моделируемых объектов, явлений, процессов.

Математические модели применяются, как правило, в тех случаях, когда геометрическое или физическое моделирование объекта затруднено или невозможно вообще. Они имеют особую структуру, отражающую свойства объекта, проявляемые им в конкретных условиях его функционирования. Такие модели широко используются в астрономии, физике, механике, структурной лингвистике.

В экономике и землеустройстве геометрические и физические модели применяются крайне редко. Примером могут служить экспериментальные системы ведения сельского хозяйства, экспериментальные севообороты, системы расселения и организации территории, освоение которых происходит в течение многих лет и эффективность которых проявляется также через многие годы. Как правило, в этих науках пользуются математическими моделями.

Все модели обладают рядом общих свойств:

- ✓ они подобно изучаемому объекту и отражают его наиболее существенные стороны;
- ✓ при исследовании модели способны замещать изучаемый объект, явление или процесс;
- ✓ они дают информацию не только о самом моделируемом объекте, но и о его предполагаемом поведении при изменяющихся условиях.

Таким образом, основное назначение модели – служить средством познания оригинала. При этом нет необходимости, чтобы модель отражала абсолютно все свойства изучаемого объекта (которых может быть бесчисленное множество). Создавая модель, исследователь должен заранее поставить конкретную цель, определяющую ее характер. Для решения практических задач крайне важно обеспечить такое подобие модели оригиналу, при котором в наиболее существенных аспектах достигается цель моделирования.

Под *моделированием* в узком смысле слова мы понимаем построение модели изучаемого объекта, явления или процесса.

Объект – это физическое (материальное) тело, вещь. Для его изучения используются, как правило, геометрические модели (хотя современные компьютерные технологии позволяют создавать и цифровые математические модели материальных объектов).

Явление – это внешние свойства и признаки предмета, постигаемые через ощущения, восприятие, представление. Например, цветок-это объект (предмет), а его свойства проявляются через форму, цвет, запах. В парфюмерной промышленности моделируются запахи, в текстильной – цветовая гамма и формы.

В явлениях обнаруживаются законы: так, упавшее яблоко натолкнуло И. Ньютона на мысль о законе всемирного тяготения.

Особенно важно изучение с помощью моделей экономических явлений. Например, цена отражает объективно действующий экономический закон стоимости. Поэтому моделирование цен может помочь сознательно использовать закон стоимости в экономической политике государства.

Процесс - это ход, развития явления, последовательная смена состояний объекта. Если явление представляет статическое, постоянное качество, то процесс всегда обладает динамическими характеристиками. Например, цепная реакция- это процесс, используемый в атомной энергетике. Моделирование роста и развития растений в биологии – это моделирование процессов.

Термины *модель* и *моделирование* относятся к понятиям *кибернетики*-науки, изучающей общие закономерности строения и функционирования сложных систем управления. Так как любые процессы управления связаны с принятием решений на основе получаемой информации, то кибернетику часто определяют как *науку об общих законах получения, хранения, передачи и преобразования информации в сложных управляющих системах*. Само слово «кибернетика» происходит от греческого *cybernetes*, что означает

«рулевой», «кормчий»; в Древней Греции кибернетикой называли науку о кораблевождении, навигацию. В 1834г., составляя классификацию наук, известный французский ученый А. М. Ампер назвал кибернетикой науку об управлении общественными системами, но вскоре этот термин был забыт.

Появление кибернетики как самостоятельного научного направления относят к 1948г., когда американский ученый, профессор математики Массачусетского технологического института Норберт Винер (1894-1964) опубликовал книгу «Кибернетика, или управление и связь в животном и машине». В этом труде Винер обобщил закономерности, относящиеся к системам управления различной природы – биологическим, техническим и социальным. Вопросы управления в социальных системах были им более подробно рассмотрены в книге «Кибернетика и общество», опубликованной в 1954г.

В конце 60-х – начале 70-х годов Министерством высшего и среднего специального образования СССР и Министерством сельского хозяйства СССР была начата подготовка студентов по специальности «Экономическая кибернетика», а в сельскохозяйственных вузах – по специальности «Экономическая кибернетика в сельском хозяйстве». В ряде сельскохозяйственных вузов были открыты кафедры Экономической кибернетики. Экономическая кибернетика использует наряду с понятием «модель» и «моделирование», ряд других: «система», «информация», «управление».

Системой называется относительно обособленная и упорядоченная совокупность обладающих особенной связанностью и целесообразно взаимодействующих элементов, способных реализовать определенные функции. Более кратко система определяется как *упорядоченная совокупность элементов, рассматриваемых во взаимодействии*.

Информация – это совокупность сведений о состоянии системы, ее подсистем и элементов, а также о происходящих в них процессах.

Управление- это процесс целенаправленного воздействия на управляемую систему на основе имеющейся информации с целью обеспечить ее контролируемое поведение при изменяющихся внешних условиях.

С точки зрения кибернетики объектами моделирования являются системы, а само моделирование предполагает, что имеются две системы:

Система-оригинал, которой мы управляем или должны управлять;

Модель системы, ее аналог, который позволяет открыть свойства системы-оригинала, изучить закономерности ее поведения и получить информацию для воздействия на систему-оригинал в желаемом направлении.

Метод моделирования, сочетающий приемы эмпирического и теоретического познания, эффективно используется в самых различных областях науки. Благодаря этому удается зафиксировать и упорядочить имеющуюся информацию об объектах, объяснить некоторые их свойства и сложные зависимости, получить новую информацию о еще неизвестных свойствах, о возможных изменениях состояния объектов, проверить возникающие при этом гипотезы и теоретические предположения. Еще древние атомисты (Демокрит, Эпикур, Лукреций Кар) строили мысленные модели атома, их движения и соединения между собой, стремясь объяснить при помощи этих моделей физические свойства вещей.

1.2. Математические методы, применяемые в экономических расчетах

Любая математическая модель, описывающая тот или иной объект, явление или процесс, подразумевает наличие определенных числовых показателей, которые их характеризуют. Например, основной характеристикой проекта землеустройства является площадь участка (контура, угодья, строения, севооборота и т. д.) или его длина (при оговоренной ширине). При моделировании эти показатели заранее заданы и являются переменными, так как цель моделирования - это поиск их наилучших значений.

Все переменные в модели обязательно связаны между собой определенными ограничениями (уравнениями или неравенствами). Задача состоит в том, чтобы найти наилучшие значения параметров модели, а для этого нужно решить поставленную задачу. Математические методы как раз и дают эту возможность: с их помощью можно вычислить оптимальные значения переменных. Тем самым решение математической задачи с применением соответствующих методов становится одним из основных этапов моделирования.

Как правило, все землеустроительные, экономико-математические задачи имеют многовариантный, альтернативный характер, и основной вопрос заключается в том, как из множества допустимых вариантов выбрать оптимальный по заданному критерию.

Математически это означает поиск максимума или минимума той или иной функции, то есть решение задачи на экстремум.

При решении таких задач возникает два основных случая, когда:

- задача может быть решена классическими методами дифференциального исчисления;

- классические методы трудно применимы или вообще не могут быть использованы.

Во втором случае применяются так называемые методы *математического программирования*, которые находят широкое применение при решении различных инженерно-экономических задач. Термин «программирование» указывает на использование алгоритма последовательных приближений - программа начинает с произвольного допустимого плана и улучшает его, пока не будет получено наилучшее решение.

Не всегда исходные параметры задачи выражаются определенными числами, иногда это могут быть случайные величины; в этом случае используют методы *стохастического программирования*. Задачи, в которых нет необходимости вычислять экстремум на нескольких этапах, называют

одноэтапными (статистическими); многоэтапные задачи требуют применения методов *динамического программирования*.

В ряде случаев исходные параметры экстремальных задач могут изменяться в определенных пределах; тогда говорят о *параметрическом программировании*. Если же параметры задач по своему реальному смыслу могут принимать лишь ограниченное число значений (например, только целочисленные значения), применяют методы *дискретного программирования*.

Помимо математических программирования в экономических исследованиях широкое применение получили и другие количественные методы - *регрессионного, дисперсионного анализа, межотраслевого баланса* и т. д. К комплексным инженерно-экономическим задачам применимы методы *сетевого планирования*, определяющие пути наилучшего перехода производственной системы из одного состояния в другое. *Теория стратегических решений* рассматривает методы выбора оптимальной стратегии в условиях, когда неизвестные обстоятельства субъективного и объективного характера могут противодействовать поставленной цели и снижать эффективность проводимых мероприятий

Систематическое использование различных разделов математики (линейной алгебры, теории вероятностей, математической статистики, математического программирования, балансовых моделей, теории массового обслуживания, теории графов, теории игр и т. д.) при решении сложных вопросов планирования, проектирования хозяйственной деятельности, по сути, привело к разработке самостоятельной ветви прикладной математики (получившей название операционных исследований). Это стало возможным в первую очередь благодаря широкому использованию новых средств вычислительной техники и соответствующего программного обеспечения.

1.3. Возникновение и развитие средств и методов вычислений.

Среди многочисленных статистических пакетов отметим программы общего назначения STATGRAPHICS, STADIA STATISTICA, обеспечивающие выполнения большинства видов статистических расчетов: проверку статистических гипотез, линейный регрессионный анализ, корреляционный анализ, анализ многомерных данных и др. Для проведения некоторых специфических расчетов (например, анализа временных рядов) лучше использовать специализированные пакеты (SPSS, ЭВРИСТА). Все современные статистические программы обладают дружественным интерфейсом, как правило, совместимым с интерфейсом Windows, и поэтому могут быть достаточно быстро освоены пользователем со средним уровнем подготовки при наличии у него необходимых знаний в области математической статистики.

Реальные возможности *средств специального прикладного программного обеспечения* в основном определяются конкретными прикладными задачами, для решения которых они предназначены. Разработка таких программ ведется практически в каждой научно-исследовательской организации, связанной с землеустройством, но в силу отсутствия адекватных возможностей, да и традиций обмена программными продуктами (отчасти это обусловлено чрезмерно узкой специализацией) достаточно полный обзор указанных средств практически невозможен. Отметим лишь некоторые программные пакеты, разработанные в лаборатории автоматизированного проектирования Государственного университета по землеустройству. Среди них имеются средства для решения:

общих задач линейного программирования симплекс-методом;

задач транспортного типа;

проведение корреляционно-регрессионного анализа и построения производственных функций;

экологического и агроэкономического обоснования проектов внутрихозяйственного землеустройства;

землеустроительного обслуживания сельскохозяйственных предприятий и др.

Первые три пакета из числа названных позволяют решать задачи, рассматриваемые и в математических пакетах общего назначения. Их спецификой является ориентация, во-первых, на учебный процесс, а во-вторых, на обеспечение пользователей полной информацией о результатах решения соответствующей задачи, необходимых для ее всестороннего, в том числе экономического, анализа. Это весьма удобно не только в рамках учебного процесса, но и при решении практических задач Землеустроительного проектирования. В основе алгоритмических модулей названных пакетов лежат современные методы решения математических задач; более подробно их возможности иллюстрируются ниже в соответствующих главах.

Землеустроитель, занимающийся в настоящее время применением методов математического моделирования в своей области, должен быть хорошо знаком с системным программным обеспечением и стандартными программными средствами общего назначения, а также уметь пользоваться прикладными программами общего и специального назначения. Только владея всеми этими средствами, используя возможности современной компьютерной техники, можно быстро получить наилучшие решения по организации рационального использования и охране земель.

1.4. Необходимость и возможность применения математических методов и моделей.

Проект землеустройства для любого предприятия, организации или учреждения, использующего землю, имеет очень большое значение.

Так, например, в проектах *межхозяйственного землеустройства* определяются площади земель, выделяемых предприятиям и гражданам, устанавливаются границы их землевладений и землепользований, состав земельных угодий, целевое назначение, обременения и ограничения в

использовании земель. На основании этих проектов осуществляются отводы земель землевладельцам и землепользователям на местности (в натуре), выдаются документы, устанавливающие права земельной собственности, аренды, долгосрочного или краткосрочного землепользования, без которых нельзя начинать вести хозяйство или проводить операции с землей (куплю, продажу, дарение, наследование и др.).

В проектах *внутрихозяйственного землеустройства* организуется территория сельскохозяйственных предприятий в увязке с требованиями экономики, организацией производства, труда и управления. В них устанавливаются порядок использования земли, состав земельных угодий (площади пашни, многолетних насаждений, кормовых угодий, лесных полос и др.), типы, виды, размеры, количество и размещение севооборотов, сенокос - и пастбищеоборотов, определяются перечень и размеры отдельных хозяйственных отраслей, намечаются объемы производства продукции и перспективы развития хозяйства.

Рабочие проекты на осуществление землеустроительных мероприятий (улучшение кормовых угодий, освоение и культивацию земель, закладку садов и виноградников, строительство дорог, прудов и т. д.) определяют хозяйственное использование каждого конкретного участка земли, направления эффективного инвестирования средств в данные мероприятия.

Таким образом, проекты землеустройства определяют использование земли и хозяйственную деятельность любого предприятия на годы вперед. От того, насколько правильно и качественно они заготовлены, зависит экономическая, экологическая и социальная эффективность хозяйствования на земле.

С формальной точки зрения землеустроительный проект представляет собой совокупность документов (расчетов, чертежей и т. д.) по созданию новых форм организации территории (устройства земли), их экологическому, экономическому, техническому и юридическому обоснованию, обеспечивающих рациональное использование и охрану земель.

В состав землеустроительного проекта входит графическая и текстовая документация.

Графическая документация включает проектный план, рабочие чертежи по перенесению объекта в натуру, карты (почвенные, геоботанические, земельно-оценочные, агроэкологические и др.), схемы, графики, рисунки, диаграммы, учитывающие фактическое состояние объекта и используемые при проектировании.

Текстовая документация состоит из задания на проектирование, расчетно-пояснительной записки, материалов технико-экономического (агроэкономического) обоснования проекта, ведомости площадей угодий (проектной экспликации), сметно-финансовых расчетов и т. д.

Графическая и текстовая документация проектов опирается на числовые данные. Например, границы хозяйства-это линии, связывающие точки с определенными координатами. Подготовка проектной экспликации предполагает вычисление площадей различных угодий, расчет структуры посевных площадей и севооборотов, определение количества и численности населения населенных пунктов, размещение производственных центров, животноводческих ферм и т. д. Все эти данные носят числовой характер, и им можно оперировать, используя законы, правила и методы математики.

Чтобы землеустроительный проект лучше отвечал своему назначению и определял наиболее эффективные пути использования земли, при его составлении должен быть учтен целый ряд факторов, действующих иногда в противоположных направлениях. Например, с точки зрения эффективного использования сельскохозяйственной техники поля севооборотов должны иметь как можно больший размер и длину гона. Напротив, с точки зрения пригодности почв для возделывания конкретных сельскохозяйственных культур эти поля должны быть небольшими из-за обычно наблюдаемого мозаичного размещения почв.

При малом числе такого рода ограничений рациональное проектное решение найти довольно легко, так как проектировщику приходится

рассматривать немного вариантов. Но с введением в задачу каждого нового условия поиск наиболее эффективного решения все более осложняется, а учет всех необходимых взаимосвязей становится весьма затруднительным.

Кроме того, главная трудность заключается не в том, чтобы найти какое-то допустимое проектное решение. Несмотря на то, что каждое дополнительное условие или ограничение почти всегда сужает область определения задачи, количество допустимых решений или вариантов проекта по-прежнему остается бесконечно большим. Задача проектировщика заключается в том, чтобы выбрать из всех возможных самое рациональное (самое выгодное по тому или иному критерию) решение, что и позволяют сделать математические методы.

При традиционных методах проектирования землеустроители в основном опираются на накопленный опыт и знания; в лучшем случае составляют два-три варианта проекта, из которых по системе показателей выбирают лучший. Но вариант является наиболее предпочтительным только из двух-трех составленных, а не из всего множества возможных решений. Нет никакой гарантии, что полученный таким путем проект действительно оптимален.

Для решения этой проблемы предлагались (и практически использовались) различные методы.

Во-первых, делались попытки увеличить количество анализируемых вариантов с тем, чтобы свести к минимуму возможную потерю лучшего из них. Появился так называемый диалоговый (интерактивный) режим работы с ЭВМ, когда землеустроитель-проектировщик, варьируя различными параметрами задачи, мог ставить перед машиной вопрос «что будет, если..?». На этой основе по определенной системе показателей или по одному интегрированному показателю выбирался лучший вариант. Таким образом, осуществлялась компьютерная имитация реальной проектной ситуации осуществлялся выбор наиболее подходящего решения.

Во-вторых, для решения проектных задач стали использовать математические методы, которые позволяли из бесчисленного множества подходящих решений безошибочно выбирать лучшее (методы математического программирования, или экономико-математические методы; задача решалась на получение максимума какого-то экономического результата).

Оба подхода, как правило, использовались при агроэкономическом обосновании проектов землеустройства. Однако с развитием средств вычислительной техники, созданием мощных микрокомпьютеров, графических станций, средств преобразования графической информации в цифровую и цифровой в графическую (дигитайзеров, плоттеров, лазерных принтеров, сканеров и др.) появилась возможность применять математические методы и при решении графических землеустроительных задач, таких, например, как:

- ✓ составление графического проекта землеустройства (вычисление площадей хозяйственных участков и получение проектной экспликации земель сельскохозяйственного предприятия);
- ✓ проектирование хозяйственных участков (полей севооборотов, гуртовых и отарных участков, загонов очередного стравливания, кварталов и клеток на многолетние насаждениях, сенокосооборотных участков и др.) заданных размеров;
- ✓ оптимальное размещение линейных элементов организации территории (границ, дорог, скотопрогонов, лесополос и т. д.);
- ✓ составление чертежей землеустроительного обследования территории, картограмм агроэкологической классификации земель, оценки потенциальной опасности проявления водной эрозии и дефляции, загрязненности территории тяжелыми металлами и радионуклидами и других графических документов, необходимых в проектах землеустройства.

Применению математических методов для решения указанных задач способствовало также появление новых технических средств картографии и фотограмметрии, специальных программных продуктов и геоинформационных систем типа MapInfo, Arc info, Micro station и др., автоматизированных-кадастровых систем.

Возможность применения экономико-математических методов обусловлена, прежде всего, экономическим характером землеустроительных задач, а также следующими обстоятельствами:

- альтернативным характером землеустроительных решений, то есть наличием множества вариантов развития землевладений и землепользований и привязанного к ним сельскохозяйственного производства;

- возможностью выразить искомые величины в числовой форме;

- наличием системы определенных условий и ограничений.

Так, например, сумма площадей земельных угодий должна равняться общей площади землеводства (землепользования); размеры отдельных отраслей, особенно трудоемких, должны быть такими, чтобы в любой период времени было достаточно трудовых ресурсов и техники; производство кормов должно быть сбалансировано и т. д. Эти условия легко формулируются математически и связаны с переменными задачи. Каждое условие будет тогда представлено некоторым уравнением или неравенством, а их совокупность образует систему ограничений задачи.

Поскольку при решении задач сельскохозяйственных предприятий вопросы организации территории рассматриваются во взаимосвязи с организацией производства и его размещением, различные проектные задачи всегда имеют экстремальный характер, то есть решаться на максимум или минимум какого-то показателя (например, получение максимальной прибыли от ведения хозяйства при ограниченных земельных, трудовых и денежно-материальных ресурсах или на минимум площади землевладения при заданных объемах производства сельскохозяйственной продукции).

Обобщая результаты проведенных исследований по данной теме, можно сделать следующие выводы:

1. Математические методы позволяют находить более целесообразные решения по перераспределению, использованию и охране земельных ресурсов на любом уровне - от отдельных сельскохозяйственных предприятий до народного хозяйства в целом.

2. Оптимальные планы использования производственных ресурсов, связанных с способствуют достижению заданных объемов производства при минимальных затратах труда и средств. В результате этого повышается производительность труда, ускоряются темпы воспроизводства в хозяйствах.

1. Результаты, полученные математическими методами, позволяют создать наилучшие организационно-территориальные условия, способствующие повышению урожайности сельскохозяйственных культур, улучшению плодородия почв, прекращению и предотвращению процессов эрозии, высокопроизводительному использованию техники.

2. Благодаря математическим методам и ЭВМ улучшаются качество подготовки исходной информации и ее использование. Землеустроительная наука получает возможность стать точной не только качественно, но и количественно, поднимаясь тем самым на новую ступень.

3. Применение математических методов способствует не только улучшению экономических показателей, но и экологических, социальных и технических характеристик проекта землеустройства.

4. Математические методы позволяют с большой точностью проверять и оценивать реальную значимость разных теоретических моделей и концепций развития моделей землевладения и землепользования на перспективу, сопоставляя их практическую ценность. Здесь количественный анализ выступает как орудие оценки тех или иных методов экономических расчетов в научных исследованиях, а математическое моделирование заменяет этап длительной экспериментальной проверки.

5. Математические методы, по сути, являются связующим звеном между землеустройством и другими науками, изучающими сельское хозяйство как с природоохранной и технологической, так и с экономической и социальной точек зрения.

6. Внедрение математических методов и вычислительной техники позволяет перестроить всю систему землеустройства.

1.5. Типы, виды и классы математических моделей

В настоящее время для решения экологических задач различных классов используются разнообразные виды экономико-математических моделей, позволяющих давать анализ использования земельных ресурсов, выявлять определенные тенденции и находить оптимальные варианты устройства территории. Классификация моделей, применяемых в землеустройстве, всегда базировалась на аналогичных классификациях, используемых в аграрной экономике.

Так, например, Р. Г. Кравченко разделял экономико-математические модели на три группы:

- ✓ корреляционные модели и производственные функции, позволяющие отразить степень влияния различных факторов на результаты производства, обосновать нормативы, сделать прогнозы состояния и динамики процессов воспроизводства;
- ✓ балансовые модели, обеспечивающие обоснование пропорций воспроизводства, его факторов и результатов;
- ✓ модели оптимизации, дающие возможность выбора наилучших вариантов развития экономических систем на основе использования аппарата математического программирования.

К детерминистическим относились модели, в которых результат полностью и однозначно определялся набором независимых переменных. Эти модели строились на основе правил линейной алгебры и представляли собой системы линейных уравнений, совместно решаемых для получения

совместных результатов. В свою очередь, они подразделялись на балансовые (обычно имеющие форму шахматного баланса и записываемые в виде квадратных матриц) и оптимизационные, предназначенные для отыскания оптимальных решений методами математического программирования.

К стохастическим относились модели, описывающие случайные процессы, подчиняющиеся законам теории вероятностей. Это были модели, основанные на выравнивании статистических рядов. А также позволяющие анализировать эмпирические закономерности, не выражающиеся строго функциональными зависимостями.

Приведенная группировка в основном отражает состав моделей, применяемых и в землеустройстве. Главный ее недостаток - отсутствие единого классификационного признака, который позволил бы произвести четкое деление. Например, оптимизационные модели могут быть не только детерминистическими, но и стохастическими (в этом случае оптимальное решение находится методами стохастического программирования). Большинство экономико-статистических моделей также может иметь стохастический характер, то есть описывать влияние случайных факторов.

Для классификации математических моделей, применяемых в землеустройстве, целесообразно использовать пять основных классификационных признаков (табл.1).

По первому признаку модели делятся на два типа: графические и экономические, что обусловлено рядом причин. Прежде всего, землеустроительные решения всегда выражаются в виде определенной организации территории (система севооборотов, полей, рабочих участков, дорогов, лесополос, границ и т. д.). Далее, с ней тесно связаны система расселения и организация производства. Таким образом, любое землеустроительное решение можно, с одной стороны, «посмотреть» (на плане, на местности, в проекте), а с другой – просчитать через показатели эффективности производства и оценить, насколько оно выгодно.

Таблица 3.1

Классификация математических моделей, применяемых при решении специализированных задач

Классификационный признак	Виды моделей
Вид проектной документации	Графические
Степень определенности информации	Экономические
	Детерминистические
	Стохастические
Вид (форма) землеустройства или землеустроительного действия	Межхозяйственного землеустройства
	Внутрихозяйственного землеустройства
Математические методы, лежащие в основе модели	Рабочего проектирования
	Аналитические (дифференциального исчисления)
	Экономико-статистические (математической статистики)
	Оптимизационные (математического программирования)
	Балансовые (межотраслевого баланса)
Класс проекта землеустройства	Сетевого планирования и управления
	Прочие
	Распределяются по классам проектов землеустройства

Землеустроительная документация всегда делилась на две части – графическую и текстовую, причем основу последней составляли расчеты. С развитием информатики и вычислительной техники появилась возможность иметь самостоятельные математические модели для графической и расчетных частей, что и определяет необходимость соответствующего деления. Графические модели могут считаться экономико-математическими лишь условно, так как они лишь характеризуют в цифровом виде условия производства (например, дают цифровую модель местности). Тем не менее, они тесно связаны с моделями организации производства и территории, которые и являются собственно экономическими.

Графические математические модели характеризуют различные элементы проекта землеустройства или их совокупности, которые показываются на проектном плане; к ним относятся площадные, линейные и точечные объекты.

Площадные объекты - это отдельные землевладения и землепользования, севообороты, их поля и рабочие участки, загоны очередного стравливания, гуртовые (отарные) участки, пастбища - и сенокосообороты, бригадные участки и т. д. Они характеризуются площадью, координатами поворотных точек и центра тяжести, что позволяет определять местоположение этих участков, их форму и другие параметры.

Линейные объекты представляют собой линейные элементы организации территории. К ним относятся полевые и магистральные дороги, лесополосы, инженерные коммуникации (газопроводы, ЛЭП и др.), отдельные границы участков, зон и т. д. Эти объекты могут размещаться на местности в виде прямых и ломаных линий, а также в виде кривых. Они характеризуются протяженностью, шириной, координатами начальных, конечных и промежуточных точек.

Точечные объекты позволяют определять на местности местоположение отдельных инженерных сооружений (таких, например, как колодцы, родники, буровые вышки и т. д.). Их размещение характеризуется местоположением.

Графические модели следует отличать от геометрических, так как первые имеют только математический характер и обязательно числовой или цифровой вид. Вторые могут давать характеристику графической части проекта и в традиционной форме (например, в виде чертежа, оформленного вручную).

Экономические модели, применяемые в землеустройстве, представляют собой выраженную в математической форме различные расчеты по проектам землеустройства. К ним относятся модели агроэкономического обоснования проектов внутрихозяйственного землеустройства, технико-экономических

обоснований (ТЭО) проектов межхозяйственного землеустройства, сметно-финансовые расчеты рабочих проектов и др.

В зависимости от степени определенности информации, используемой в математических моделях, они делятся на два вида: детерминированные и стохастические.

Детерминированные модели основаны либо на абсолютно точной информации, либо сведений, которые условно считаются точными. Полученные с использованием этих моделей данные находятся в функциональной связи с набором независимых переменных, то есть решения задач полностью и однозначно определяется этим набором переменных.

Стохастические модели основаны на информации, имеющей стохастический (вероятный) характер. Например, при планировании урожайности сельскохозяйственных структур (результат) невозможно знать заранее будущие погодные условия (факторы). Эти условия задаются в моделях с определенной степенью вероятности. Соответственно полученные результаты также будут устанавливаться с определенной степенью вероятности.

Исходя из вида (формы) землеустройства или осуществляемого землеустроительного действия, народнохозяйственного значения землеустроительных проблем и охвата объекта землеустроительного проектирования, математические модели подразделяются на четыре основных класса: межотраслевые, межхозяйственного землеустройства, внутрихозяйственного землеустройства и рабочего проектирования.

Межотраслевые модели обеспечивают решение задач по прогнозированию и оптимальному планированию использования земельных ресурсов и их охране по стране в целом, на уровне субъекта района, местной администрации и т. п.

Модели данного класса позволяют оптимизировать распределение земель по категориям земельного фонда страны (земли сельскохозяйственного назначения; промышленности, транспорта, связи,

обороны и иного специального назначения; лесного фонда, запаса и др.), решать задачи по планированию агропромышленного производства в регионах, по планированию и осуществлению природоохранных мероприятий и т. д.

Основным видом землеустроительных работ, включающим модели этого класса, является разработка генеральных схем использования и охраны страны (субъекта Федерации) и схем и землеустройства административных районов и других территорий, находящихся в ведении местных администраций.

Модели межхозяйственного землеустройства позволяют решать задачи по перераспределению земель между хозяйствами; по образованию или упорядочению землевладений и землепользований сельскохозяйственного и не сельскохозяйственного назначения; по установлению границ административно-территориальных образований, черты населенных пунктов и т. п. К данному классу относятся задачи по определению оптимальных размеров землепользований и рациональному размещению производства на территории, по наиболее целесообразной ликвидации недостатков в использовании земельных ресурсов и др.

Модели внутрихозяйственного землеустройства предназначены для решения вопросов наиболее полного, рационального и эффективного использования земель и организации производства в конкретных сельскохозяйственных предприятиях. Основные задачи данного класса - установление оптимального сочетания отраслей, площадей и состава угодий; определение видов, количества и площадей севооборотов и их размещение; рациональная организация кормопроизводства; планирование грузоперевозок; планирование комплекса мелиоративных работ; оптимальная трансформация угодий; оптимизация размеров производственных подразделений и др.

Модели рабочего проектирования обеспечивают решение различных задач, связанных с землеустройством различных земельных участков и

инвестициями в эти земли (создание орошаемых культурных пастбищ, выполаживание оврагов, трансформация и мелиорация земельных участков, строительство прудов, дорог и дорожных сооружений, закладка многолетних насаждений и т. п.).

Сложность математических моделей каждого класса зависит от числа учитываемых факторов и характера взаимосвязи между ними, от наличия, точности и достоверности исходной информации и от особенностей изучаемого процесса или явления. Кроме того, она определяется и конструктивными особенностями модели (количеством переменных, их степенью, количеством условий, видом целевой функции и др.).

В практической работе большое значение имеет классификация экономико-математических моделей в зависимости от лежащих в их основе математических методов. По этому признаку все модели можно разделить на аналитические, экономико-статистические, оптимизационные (нередко именно их называют экономико-математическими, так как в общей системе моделей они занимают главное место), балансовые, сетевого планирования и др. Для использования моделей указанных типов необходимо хорошо знать соответствующие методы (математической статистики, математического программирования и т. д.). Вместе с тем ряд землеустроительных задач требует разработки нестандартного математического аппарата, особых методов моделирования.

Аналитические модели в землеустройстве основаны на применении классического математического аппарата (алгебра, дифференциальное и интегральное исчисление, геометрия, тригонометрия, математический анализ); при этом требуется доказательство различных теорем и необходимых формул.

Как правило, аналитические модели имеют вид формул, отражают функциональные зависимости. Каждому определенному значению фактора (независимой переменной) или множества факторов соответствует строго определенное значение результата. Типичным примером аналитической

модели является формула для расчета условной длины поля (рабочего участка) L:

$$L = \frac{5P}{3H + c + d}$$

Где Р - площадь поля, в квадратных метрах; Н-высота (геометрическая) на фигуре поля, м; с+d – сумма длин линий, не параллельных основному направлению обработки поля (рабочего участка), м.

С помощью аналитических моделей в землеустройстве рассчитывают, как правило, технические показатели проектов: средние расстояния, рабочие уклоны, коэффициенты компактности землепользований и др.

Экономико-статистические модели базируются на использовании теории вероятностей и методов математической статистики (корреляционного, регрессионного, дисперсионного анализа, теории выборок и т. д.). Главное место среди них занимают производственные функции, представляющие собой уравнение статистической (усредненной) связи зависимой переменной (результата) и факторов-аргументов. С помощью этих моделей при землеустройстве рассчитывают прогнозируемую урожайность культур, продуктивность животных, а также некоторые параметры организации территории (распаханность, облесенность, освоенность). Они полезны также при анализе уровня использования земли; с их помощью подготавливается необходимая информация для применения оптимизационных методов, производится обоснование землеустроительных проектных решений.

Экономико-статистические модели могут быть *функциональными и корреляционными*. Первые, по сути, идентичны аналитическим моделям, но основаны на статистической информации. А поскольку строго функциональные связи (когда значению факторов в управлении соответствует точно определенное значение результата) в экономике встречаются очень редко, эти модели практически не используются в землеустройстве. Вторые (корреляционные) модели базируются на

статистических (верных лишь в среднем) связях между факторами. Они могут обладать разной степенью достоверности, так как описывают случайные процессы. Уровень достоверности модели оценивается специальным показателем (так называемым коэффициентом корреляции).

Оптимизационные модели основаны главным образом на методах математического программирования, позволяющие находить экстремальные (максимальные или минимальные) значения целевой функции по искомому перечню переменных при заданных условиях.

Например, в процессе решения задачи необходимо найти такие размеры сельскохозяйственного предприятия (общая земельная площадь, состав земельных угодий и отраслей – искомые переменные), которые, исходя из своей специализации, фондооснащенности и трудообеспеченности (задаваемые условия), давали бы максимальную прибыль (максимальное значение целевой функции).

Оптимизационные модели в землеустройстве делятся на две разновидности: комбинированные и дифференцированные. При комбинированном моделировании все вопросы землеустроительного проекта решаются комплексно в их взаимообусловленности и взаимозависимости. Такой подход в принципе правильнее, однако, он приводит к громоздким задачам, решение которых затруднительно. Суть дифференцированного моделирования заключается в последовательном решении нескольких задач по проекту (например, по его составным частям). Модели получаются значительно меньшего объема, и их решение существенно облегчается. Применение именно дифференцированного моделирования в землеустройстве объясняется сложностью объектов и многообразием решаемых вопросов.

Оптимизационные модели применяются, как правило, для разработки наилучших (с точки зрения выбранного критерия) проектных землеустроительных решений. Основу их составляют оптимальные варианты организации производства и территории сельскохозяйственных предприятий,

а также наиболее эффективные варианты развития землевладения и землепользования (оптимальная специализация хозяйств, необходимые размеры перераспределения земель, оптимальные размеры и т.д.).

Балансовые модели обеспечивают обоснование и определение наилучших пропорций территориальной организации производства с учетом его факторов и результатов. Они имеют форму матриц, систем, таблиц и т. п. В землеустроительных расчетах могут использоваться при обосновании проектных решений (балансы кормов, труда, расчеты населения на перспективу, баланс трансформации и перераспределения земель и т.д.).

Модели сетевого планирования и управления, базирующиеся на одноименных математических методах, применяются при планировании и организации землеустроительных работ, при разработке планов перехода к новому составу угодий и новым севооборотам, при составлении планов реализации проектов землеустройства и авторского надзора.

В настоящее время модели данного класса находятся в стадии практической разработки; часть из них будет рассмотрена ниже

1.6. Требования, предъявляемые при использовании экономико-математических методов и моделей

Практика показывает, что экономико-математические методы в землеустройстве оказываются полезными лишь в том случае, когда выдержаны определенные требования к их применению.

1. Прежде всего, не следует забывать, что в основе экономико-математического моделирования лежат количественные методы анализа. Это предполагает детальное изучение объекта проектирования, выявление различных зависимостей и взаимосвязей, их математическое описание в виде набора переменных величин, уравнений, неравенств и т.д. Вместе с тем никакие математические методы не позволят применить приемлемое решение, если не будут в должной мере учтены выводы, полученные в ходе качественного анализа.

В основе такого анализа лежит здравый смысл, а также знание экономических законов, понятий и категорий, благодаря чему исключаются логические ошибки и заведомо неприемлемые решения. В конце концов, математический аппарат- это лишь вспомогательное средство, орудие количественного анализа, а также техника, позволяющая более обоснованно, быстро и точно находить нужные решения, в основе которых всегда лежат качественные закономерности, изучаемые землеустроительной наукой.

2. Разрабатываемые модели должны учитывать экономические, технологические, землеустроительный, технические и другие условия, в которых находится землеустраиваемый объект.

К экономическим условиям относятся: размеры и сочетания отраслей, виды ресурсов, гарантированные объемы производства, условия реализации и распределения продукции. К технологическим - агротехнические особенности возделывания сельскохозяйственных культур, ветеринарные и зоотехнические требования к выращиванию животных и. т.д.

Землеустроительные условия характеризуют особенности организации территории и производства (размещение населенных пунктов, земельных массивов производственных подразделений, производственных центров, организация угодий и устройство территории севооборотов, качество земель и т.д.); они составляют основу любой модели, которую предполагается использовать при землеустроительном проектировании.

Технические условия - это наличие у разработчиков средств вычислительной техники и программного обеспечения, что диктует требования по выбору типа моделей, размерности задач, степени детализации решений. Другими словами, экономико-математические модели должны быть приведены к виду, позволяющему их решать на имеющейся вычислительной технике.

Учет всех перечисленных условий позволит построить экономико-математическую модель, наилучшим образом соответствующую изучаемому

объекту, и избежать в последующем ее трудоемкой доработки и многочисленных корректировок полученных решений.

3. Возможности моделирования прямо связаны с качеством исходной информации. Ни какое решение не будет приемлемым, даже если оно и получено с использованием самых современных методов, если в его основе лежат недостоверные, неполные или несвоевременно полученные данные. Поэтому необходимо учитывать, какие показатели реально могут быть получены на основе статистических, экспериментальных и нормативных материалов. Кроме того, должно быть обеспечено соответствие между этой информацией и точностью применяемых математических методов в процессе реализации модели.

4. Использование экономико-математических методов и моделей не является самоцелью, поэтому не нужно вводить ничего лишнего в условия задачи, заранее навязывать то или иное решение, попытаться «помочь» машине в выборе оптимума. Нельзя также абсолютизировать полученные на компьютере результаты; их следует тщательно проанализировать, проверить и только потом использовать для дальнейших действий.

Необходимо иметь ввиду, что полученное математическими методами оптимальное решение (математический оптимум.) не обязательно согласуется с экономической целесообразностью (экономическим оптимумом).

Это часто бывает в тех случаях, когда модель не вполне адекватна изучаемому объекту. Тогда, оценивая решение логическим, экспертным или специальным математическим путем, а, также осуществляя определенные корректировки, математический и экономический оптимумы приводят в соответствие.

Это достигается двумя основными способами - корректировкой самой модели с последующим решением новой задачи или же путем непосредственной корректировки решение без изменений.

Ключевые слова

Модель, моделирование, построение модели, объект, система, объект-оригинал, информация, переменные, экзогенные, эндогенные, коэффициенты, параметры, статистическая, динамическая модель, детерминированные математические модели, экологические модели, экономико-математические модели, оптимизация, случайность, стохастические модели, этапы построения модели, математический анализ, адекватность модели, классификация модели.

Вопросы для самопроверки

1. Почему необходимо использование математики в экологических процессах?
2. Что такое математическая модель?
3. Как строиться математическая модель явлений или объекта? Приведите пример построения уточнения модели.
4. Какова связь между математической структурой модели и ее содержательной интерпретацией?
5. Какие переменные модели называются экзогенными?
6. Какие переменные модели называются эндогенными?
7. Какие ресурсы в моделях называются взаимозаменяемыми?
9. Чем отличаются равновесные модели от оптимизационных?
10. В чем отличие статистических моделей от динамических?

Лекция 2. МЕТОДЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

2.1. Общая постановка задачи математического программирования

Математическое программирование — это раздел математики, посвященный теории и методам решения задач, поиска экстремального, то есть наибольшего или наименьшего значения заданной функции при соблюдении некоторых условий. Функцию, экстремальное значение которой нужно

найти, называют *целевой*.

Задача математического программирования формулируется следующим образом.

Устанавливается перечень *переменных* x_1, x_2, \dots, x_n , которые могут принимать различные числовые значения. На эти неизвестные налагаются определенные условия, образующие так называемую *систему ограничений*. Ограничениями служат уравнения или неравенства, построенные в соответствии с логическим содержанием задачи. Как правило, они имеют линейный вид (то есть переменные входят в них в первой степени):

$$\begin{aligned} &a_{l1}x_1 + a_{l2}x_2 + \dots + a_{ln}x_n = b_l, \\ &\dots\dots\dots \\ &a_{r1}x_1 + a_{r2}x_2 + \dots + a_{rn}x_n = b_r, \\ &a_{r+11}x_1 + a_{r+12}x_2 + \dots + a_{r+1n}x_n \leq b_{r+1} \\ &\dots\dots\dots \\ &a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \end{aligned} \tag{2.1}$$

но в принципе могут содержать и нелинейные выражения, которые в общем, виде записываются так:

$$q_i(x_1, x_2, \dots, x_n) \Leftrightarrow 0; \quad (2.2)$$

$$x_j > 0; I = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (2.3)$$

Таким образом, система ограничений может быть и смешанной (включать линейные и нелинейные выражения).

Затем составляется некоторая функция из тех же искомых величин, которая выражает принятый критерий (доход, издержки, себестоимость и т. д.). Ее называют *целевой функцией* или *функционалом* задачи. Чаще всего она бывает линейной:

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max(\min), \quad (2.4)$$

Но может быть и линейной:

$$Z=f(x_1,x_2,\dots,x_n)\rightarrow\max(\min). \quad (2.5)$$

Таким образом, требуется найти такой набор значений переменных, который удовлетворяет системе ограничений и при котором целевая функция принимает наибольшее или наименьшее значение.

Если система ограничений и целевая функция линейны относительно искомых величин x_1, x_2, \dots, x_n , возникает задача линейного программирования; если имеется хотя бы одно нелинейное выражение, мы имеем дело с нелинейным программированием. Существуют методы для решения задач обоих типов.

Любая совокупность численных значений переменных именуется *планом* задачи; план, удовлетворяющий системе ограничений, называется *допустимым*. Допустимый план, максимизирующий (или минимизирующий) целевую функцию, называется *оптимальным*. Допустимых планов в задаче, как правило, бесчисленное множество, и алгоритм решения сводится к выбору из этого множества оптимального плана. Система линейных и нелинейных ограничений, которой не отвечает ни одна совокупность неотрицательных значений переменных, называется *несовместной*; такая задача не имеет решения. Несовместность решения можно обнаружить или путем простого логического анализа, или с помощью специальных математических приемов (например, теории определителей). *Совместной* называется система, имеющая хотя бы одно допустимое решение.

Из условных нелинейных экстремальных задач математического программирования выделяются задачи выпуклого программирования, где требуется определить максимум вогнутой функции на выпуклом множестве. Доказано, что любой локальный максимум вогнутой функции, заданной на выпуклом множестве, является ее глобальным максимумом на том же множестве.

Существует немало различных вариантов формулировок задач линейного программирования, но наиболее употребительны две формы их записи.

Для перехода от задачи минимизации к задаче максимизации, нужно изменить знак критериальной функции, т.е. задача минимизации функции

$f(x) = (c, x)$ эквивалентна задаче на поиск максимума функции

$f(x) = -(c, x)$ или $\min z = \max(-z)$.

Любое ограничение в форме неравенства введением дополнительной неотрицательной переменной может превратиться в ограничение – равенства.

Условие

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n \leq a$$

эквивалентно двум ограничениям

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n + x_{n+1} = a$$

$$x_{n+1} \geq 0.$$

Переменные типа x_{n+1} называют фиктивными или дополнительными. Эти формы представления задач линейного программирования называются стандартными.

Будем говорить, что задача линейного программирования записана в канонической форме, если она формулируется следующим образом:

Найти

$$\max(c_1 x_1 + c_2 x_2 + \dots + c_n x_n) \quad (2.6)$$

при ограничениях

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1 \quad (2.7)$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2$$

.....

$$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \quad (2.8)$$

В матричной форме, эта задача записывается так:

$$\max(c, x)$$

$$Ax = b,$$

$$x \geq 0$$

$$x \geq 0.$$

Почему x_1 удовлетворяющим данным ограничениям, называют допустимым решением.

Допустимые решение на котором линейная форма принимает максимальное значение, называется оптимальным решением или просто решением задачи.

Задача 1. Хозяйство может выделить под посев сахарной свеклы и кормовых бобов не более 200 га. Для выращивания этих культур хозяйство может затратить не более 15 000 чел.-ч механизированного труда. Затраты его на 1 га сахарной свеклы 200 чел.-ч, кормовых бобов - 50 чел.-ч. Чтобы обеспечить баланс по протеину, необходимо, чтобы общее его количество составляло не менее 10 % урожая. Средняя урожайность сахарной свеклы 600 ц/га, количество протеина в 1 кг 0,05 кг. Средняя урожайность кормовых бобов 200 ц/га, количество протеина в 1 кг 0,2 кг.

Определить, какую площадь следует отвести под сахарную свеклу и кормовые бобы, чтобы получить максимум кормов, сбалансированных по протеину.

Обозначим площадь, занятую сахарной свеклой, x_1 га, кормовыми бобами — x_2 га.

Исходя из условия задачи, запишем целевую функцию

$$F = 60 * x_1 + 20 * x_2 \text{ ---- макс.} \quad (2.9)$$

Ограничение на общее количество земли, выделяемой под кормовые культуры,

$$x_1 + x_2 \leq 200. \quad (2.10)$$

Ограничение на использование трудовых ресурсов

$$200 \cdot x_1 + 50 \cdot x_2 \leq 15\,000 \quad (2.11)$$

Ограничение по содержанию протеина

$5 * 60 * x_1 + 20 * 20 * x_2 > 0,1 (6000 * x_1 + 2000 * x_2)$, после преобразования примет вид

$$300x_1 - 200x_2 \leq 0. \quad (2.12)$$

Таким образом, решение приведенной задачи сводится к решению задачи математического программирования с целевой функцией (1.4) и условиями (1.5)—(1.7). Так как площадь угодий и трудовые ресурсы не могут иметь отрицательное значение, появляется еще одно условие

$$x_1, x_2 \geq 0 \quad (2.13)$$

(условие не отрицательности переменных). Оно типично для большинства задач математического программирования.

Рассмотрим аналогичную задачу в общем виде.

Таблица 1.1

Исходная матрица задачи

Вид ресурсов	Запасы ресурсов	Число единиц ресурсов, для производства единицы		
		P_1	P_2	P_3
P_1	30	2	4	3,2
P_2	50	5	2,6	7
P_3	40	3,5	5	2,4
P_4	20	2	1,3	1,5
Прибыль от реализации единицы продукции, руб		30	20	40

Задача 2. Для производства 3 видов сельскохозяйственной продукции требуется 4 вида ресурсов (P_1, P_2, P_3, P_4), например, P_1 -площадь земельных угодий, P_2 — трудовые затраты, P_3 — количество удобрений и т. п. Исходная информация для формулировки условий задачи приведена в табл. 1.1.

Необходимо составить план производства каждого вида продукции, который обеспечил бы получение максимальной прибыли. Исходя из этого, записываем целевую функцию

$$F = 30x_1 + 20x_2 + 40x_3 - \max, \quad (2.14)$$

где x_1, x_2, x_3 - искомое количество продукции видов Π_1, Π_2, Π_3 , производство которых должно дать максимальную прибыль.

Затраты ресурсов на производство каждого вида продукции не должны превышать имеющиеся запасы

$$\left. \begin{aligned} 2x_1 + 4x_2 + 3,2x_3 &\leq 30, \\ 5x_1 + 6,2x_2 + 7x_3 &\leq 50, \\ 3,5x_1 + 5x_2 + 2,4x_3 &\leq 40, \\ 2x_1 + 1,3x_2 + 1,5x_3 &\leq 20, \end{aligned} \right\}$$

Запланированные объемы производства должны быть неотрицательными

$$x_1, x_2 \geq 0$$

2.2. Классификация задач математического программирования

Задачи математического программирования классифицируются в зависимости от вида целевой функции и условий.

Если целевая функция и условия — линейные (входящие в их состав переменные не возводятся в степень, не перемножаются и не делятся друг на друга), то соответствующая задача — линейного программирования. Ситуации, приводящие к задачам линейного программирования, описаны в задачах 1.1 — 1.3.

Если целевая функция или, по крайней мере, одно из условий нелинейные, возникает задача нелинейного программирования. Ниже приведены примеры задач нелинейного программирования.

Пример 1. Найти максимальное значение функции

$$F = (x_1 - 3)^2 + (x_2 - 4)^2, \quad (2.15)$$

при условиях

$$\left. \begin{aligned} 3x_1 + 2x_2 &\geq 7 \\ 10x_1 - x_2 &\leq 8 \\ -18x_1 + 4x_2 &\leq 12, \end{aligned} \right\} \quad (2.16)$$

$$x_1, x_2, \geq 0$$

Здесь нелинейная целевая функция.

Пример 2. Найти минимальное значение функции

$$F = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \quad (2.17)$$

при условиях

$$\left. \begin{aligned} 5,6x_1 + 3,7x_2 &\geq 4,9, \\ 7,4x_1 + 4,3x_2 &\geq 11,4, \\ 6,9x_1 + 13,5x_5 &\geq 10,1, \\ 12,6x_3 + 4,6x_5 &\geq 9,8, \end{aligned} \right\}$$

$$x_1, x_2, x_3, x_4, x_5, \geq 0 \quad (2.18)$$

Здесь нелинейная целевая функция.

Пример 3. Найти максимальное значение функции

$$F = 3x_1 + 4x_2 \quad (2.19)$$

при условиях

$$\left. \begin{aligned} x_1^2 + x_2^2 &\leq 25 \\ x_1 \cdot x_2 &\geq 4 \\ x_1, x_2 &\geq 0. \end{aligned} \right\} \quad (2.20)$$

Нелинейные условия.

Пример 4. Найти максимальное значение функции

$$F = x_1 \cdot x_2 \quad (2.21)$$

при условиях

$$\left. \begin{aligned} x_1^2 + 2x_1 + x_2^2 - 2x_2 &\geq 14 \\ 2x_1 + x_2 &\leq 10 \end{aligned} \right\} \quad (2.21a)$$

$$x_1, x_2 \geq 0 \quad (2.22)$$

Нелинейная целевая функция (2.21) и первое условие (2.21a).

Задачи нелинейного программирования очень разнообразны. Методика решения многих из них не разработана. В большей степени изучены методы решения задач, в которых нелинейна только целевая функция. Среди них наиболее распространены задачи с сепарабельными и квадратичными

целевыми функциями. (Сепарабельной называют такую функцию, которая может быть представлена в виде суммы других функций.)

Большие трудности возникают при решении задач с нелинейными ограничениями. Методы, позволяющие во многих случаях получить оптимальное решение, разработаны только для сепарабельных ограничений.

Таблица 2.3.

Содержание различных веществ в 1 кг корма и себестоимость 1 кг корма

Корм	Содержится в 1 кг корма					Себестоимость 1 кг корма, коп.	Максимальное количество в 1 кг корма при откорме	
	кормовых единиц, г	переваримого протеина, г	кальция, г	фосфора, г	каротина, мг		в начале	в конце
Жом	0,11	7	0,7	0,1		0,1	30	25
Концентраты	1,21	81	1,2	3,3	1	4	0,5	1,0
Сено	0,49	116	17,7	2,2	45	2	—	—
Солома	0,2	8	1,4	0,8	3	1	4	1,5
Патока*	0,77	45	3	0,3	—	2	0,5	1
Кормовые бобы	—	—	260	170	—	5	4	—

Рассмотренные примеры линейного и нелинейного программирования можно назвать статическими или одношаговыми — оптимальное решение принимают только для одного этапа планирования. Кроме таких задач,

математическое программирование рассматривает многошаговые задачи, в которых оптимальные решения находят последовательно для нескольких этапов. Раздел, занимающийся изучением подобных задач и разработкой методов их решения, называют динамическим программированием.

Рассмотрим пример постановки задачи динамического программирования.

Таблица 2.4.

Потребность животных в питательных веществах в начале каждой декады

Декада	Суточная потребность на 1 гол.				
	корм. ед., кг	Переваримого протеина,г	кальция,г	фосфора, г	каротина, мг
1	5,8	520	25	12	35
2	5,9	550	26	13	35
3	6,0	580	27	14	35
4	6,1	593	28	15	40
5	6,2	606	29	16	43
6	6,3	620	30	17	45
7	6,6	640	31	18	48
8	6,8	660	33	19	50
9	7,0	680	35	20	52

Ключевые слова

Модель, моделирование, переменные, экономико-математические модели, оптимизация, этапы построения модели, экономические процессы, математический анализ, адекватность модели, математическое программирование, экстремальное значение, целевая функция,

неотрицательность, задача линейного программирования, динамическое программирование, целочисленным программированием, линейное уравнение, допустимое решение, оптимальное решение, дополнительные переменные, геометрическая интерпретация, базисное решение, координаты точек, открытая область.

Вопросы для самопроверки

1. Что изучает наука «Математическое программирование»?
2. Что выражает целевая функция?
3. Приведите классификацию задач математического программирования.
4. Сформулируйте основную задачу линейного программирования.
5. Что такое допустимое решение задачи линейного программирования?
6. Что такое оптимальное решение задачи линейного программирования?
7. В чем суть графического метода?
8. Суть одношаговых, многошаговых задач линейного программирования?
9. Что такое динамическое программирование?

Лекция 3. ГРАФИЧЕСКИЙ МЕТОД РЕШЕНИЯ ДВУМЕРНЫХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

3.1. Постановка задачи

Задачи линейного программирования с двумя переменными (x_1, x_2) называют двумерными. Неравенство и целевую функцию в них можно изобразить в виде прямых линий, проведенных в декартовой системе координат с осями x_1, x_2 . Задачи с большим количеством переменных не допускают такого наглядного представления. Геометрическая интерпретация двумерных задач позволяет наглядно представить многие понятия линейного программирования и распространить их на задачи с большим количеством переменных. Пример.

$$Z = 6000x_1 + 2000x_2 \rightarrow \max$$

$$x_1 + x_2 \leq 200, \quad (3.1)$$

$$200x_1 + 50x_2 < 15000, \quad (3.2)$$

$$300x_1 - 200x_2 \leq 0, \quad (3.3)$$

$$x_1, \geq 0. x_2, \geq 0$$

3.2. Графический метод решения двумерных задач линейного программирования.

Каждое условие (3.1)-(3.3) можно представить геометрически в виде линии, которая делит плоскость на 2 части. В одной части находятся точки, координаты которых удовлетворяют соответствующему условию. Ее называют *областью допустимых значений*. В другой размещены точки, координаты которых данному условию не удовлетворяют, — это область *недопустимых значений*. На рис. 3.1. показаны области допустимых и недопустимых значений для условий (3.1). Действительно координаты любой точки из области допустимых значений, например точки А (120, 40), удовлетворяют условию (рис.3.1.):

$$120 + 40 < 200,$$

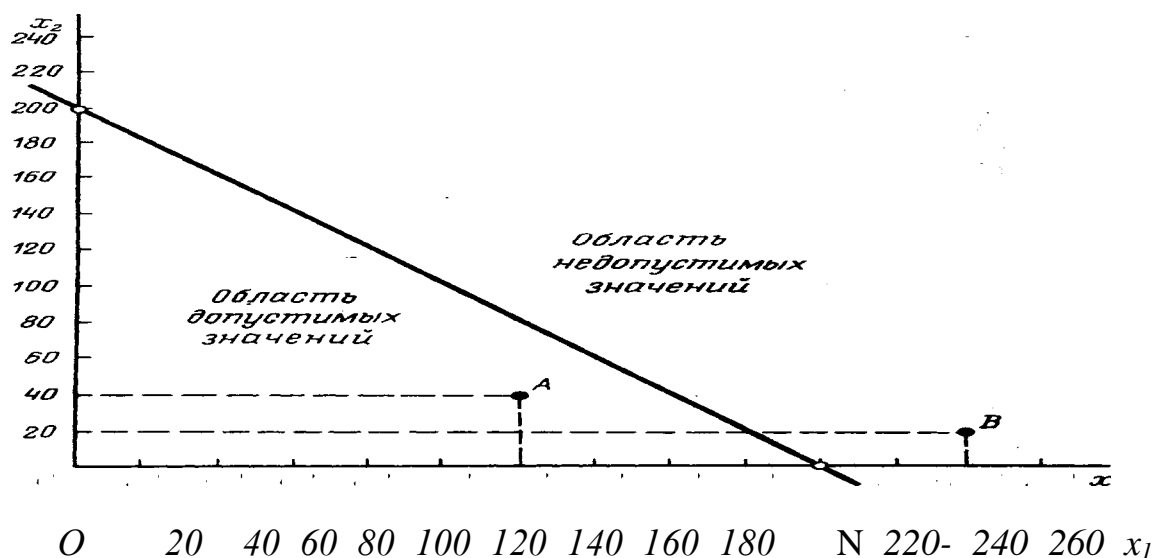


Рис. 3.1. Области допустимых и недопустимых значений для условия $x_1 + x_2 \leq 200$

координаты любой точки из области недопустимых значений, например точки В (240, 20), не удовлетворяют этому условию: $240 + 20 > 200$.

Если провести прямые, соответствующие каждому условию данной

задачи, то можно получить область допустимых значений, которая содержит точки с координатами, удовлетворяющими всем условиям одновременно. На рис. 3.2. каждая прямая снабжена штриховкой со стороны области недопустимых значений. В результате область допустимых значений оказалась не заштрихованной.

Нетрудно убедиться, что координаты любой точки из области допустимых значений (ОДЗ) удовлетворяют всем условиям данной задачи. Координаты точек, лежащих за пределами ОДЗ, не удовлетворяют, по крайней мере, одному из этих условий. Например, координаты точки А не удовлетворяют условию (3.8), координаты точки В - условию (3.5), координаты точки С - условиям (3.6), (3.7) и т. д.

Координаты любой точки внутри области допустимых значений образуют допустимое решение задачи линейного программирования. Очевидно, что если существует область допустимых значений, то допустимых решений бесконечное множество.

Координаты любой вершины области допустимых значений образуют допустимое базисное решение. В рассматриваемой задаче допустимым базисным решениям соответствуют координаты точек 1, 2, 3, 4 (рис.3.1).

Одно из допустимых базисных решений является оптимальным, то есть дающим максимальное или минимальное значение целевой функции. В рассматриваемой задаче оптимальное (максимальное) значение целевая функция достигает в вершине 3. В этом нетрудно убедиться, определив по рис. 2 координаты каждой из вершин и вычислив значения целевой функции в них (табл. 3.1.).

Возможна ситуация, когда оптимальны два допустимых базисных решения (координаты двух смежных вершин).

Таблица 3.1.

Значения допустимых базисных решений.

Номер вершины		1-й	2-й	3-й	4-й
Координаты вершины	x_1	0	0	34	54
	x_2		200	166	80
Значения целевой функции		0	400000	536000	484000

Это происходит тогда, когда коэффициенты целевой функции пропорциональны коэффициентам уравнения прямой, соединяющей эти вершины. Например, если целевая функция в рассматриваемой задаче будет иметь вид

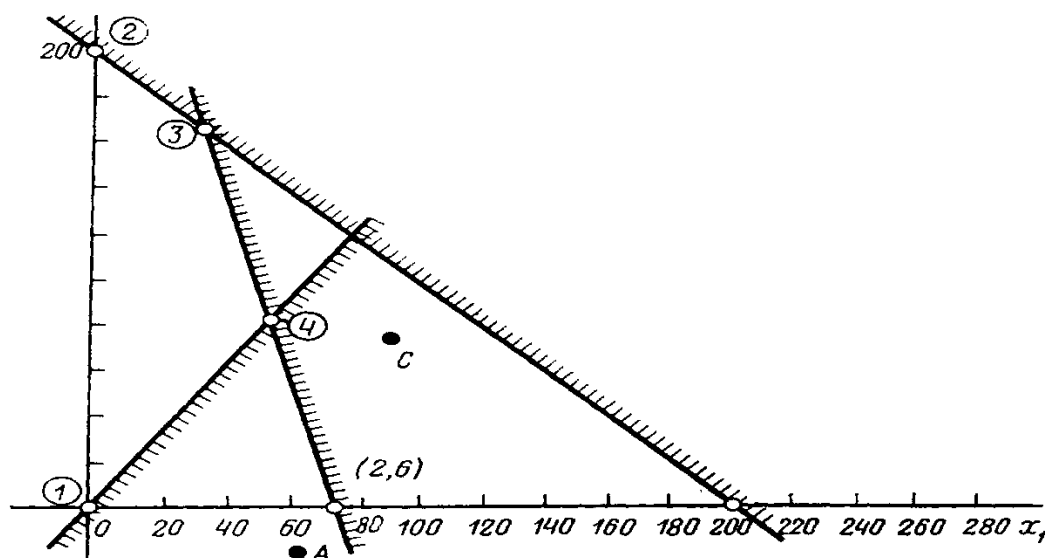


Рис. 3.2. Определение оптимального значения путем перемещения линии целевой функции

$$F = 4000x_1 + 1000x_2, \quad (3.4) \quad (1.4)$$

то есть коэффициенты будут пропорциональны коэффициентам уравнения

$$200x_1 + 50x_2 = 15000, \quad (3.6) \quad (1.5)$$

то значения целевой функции в точках 3 и 4 (рис. 3.2) совпадут. В табл. 3.2. показано, что значения целевой функции в этих точках также и максимальны.

Таблица 3.2.

Значения допустимых базисных решений при наличии множества оптимальных решений

Номер вершины		1-й	2-й	3-й	4-й
Координаты вершины	x_1	0	0	33,3333	54,54545
	x_2	0	200	166,6667	181,81818
Значения целевой функции		0	200000	300000 .	300000

Если значения целевой функции совпадают в двух смежных вершинах, они совпадают во всех точках, принадлежащих отрезку, соединяющему эти вершины. В данном случае это отрезок 3—4. Координаты любой точки этого отрезка, будучи подставленными в целевую функцию (3.4), дадут значение 300 000. Поскольку любой отрезок состоит из бесчисленного множества точек, то и оптимальных решений в описываемой ситуации также бесконечно много.

Геометрическая интерпретация задачи линейного программирования с двумя переменными позволяет сформулировать правила ее графического решения.

1. Каждое условие следует изобразить в виде прямой и построить таким образом область допустимых значений (ОДЗ) (см. рис. 3.2). Для этого нужно:

1. Заменить неравенства уравнениями. В рассмотренном примере уравнения имеют вид

$$X_1 + X_2 = 200, \quad (3.5)$$

$$200x_1 + 50x_2 = 15000 \quad (3.6)$$

$$300x_1 - 200x_2 = 0; \quad (3.7)$$

2. Определить для каждого уравнения координаты двух точек, через которые проходит прямая, описываемая этим уравнением. Для чего нужно произвольно задать значение одной координаты и вычислить вторую. Если прямая не проходит через начало координат, то в качестве произвольной точки удобно выбрать 0. Например, для уравнения (3.8) : 1-я точка $x_1 = 0$, $x_2 =$

200, 2-я точка $x_2 = 0$, $x_1 = 200$, Аналогично для уравнения (3.7): 1-я точка $x_1 = 0$, $x_2 = 15\,000 : 50 = 300$; 2-я точка $x_1 = 0$, $x_2 = 15\,000 : 200 = 75$.

Если прямая проходит через начало координат (первая часть уравнения равна 0), координаты одной точки очевидны ($x_1 = 0$, $x_2 = 0$), а для определения координат второй точки необходимо задать произвольное, но ненулевое значение, x_1 и вычислить x_2 . Например, для уравнения (3.6): 1-я точка $x_1 = 0$, $x_2 = 0$, 2-я точка $x_1 = 100$, $x_2 = 300 \cdot 100 : 200 = 150$; используя вычисленные координаты, наносим на чертеж точки и строим прямые (см. рис. 3.2, 3.3).

2. Целевой функции нужно присвоить два произвольных значения. Например, в рассматриваемой задаче

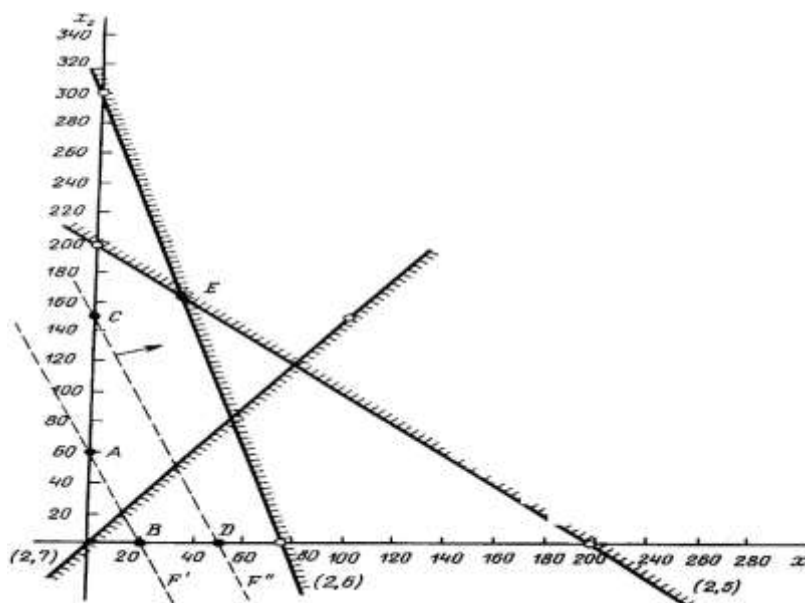


Рис. 3.3. Область допустимых значений (ОДЗ)

$$F' = 6000x_1 + 2000x_2 = 120000,$$

$$F'' = 6000x_1 + 2000x_2 = 300\,000. \quad (3.8)$$

Уравнения (3.8) представляют собой уравнения двух параллельных прямых. Для F' 1-я точка $x_1 = 0$, $x_2 = 120000 : 2000 = 60$, 2-я точка $x_1 = 120\,000 : 6000 = 20$. Для F'' 1-я точка $x_2 = 0$, $x_1 = 300\,000 : 2000 = 150$, 2-я точка $x_2 = 0$, $x_1 = 300\,000 : 6000 = 50$. Проведем прямые по указанным точкам (рис. 3.3.).

3. Определим направление перемещения этих прямых по мере

увеличения (в задачах на максимум) или уменьшения (в задачах на минимум) целевой функции. На рис. 3.3 это направление указано стрелкой (задача на максимум). Координаты точек, принадлежащих проведенным прямым в пределах области допустимых значений (точек и отрезков AB и CD на рис. 3.3, будут удовлетворять условиям (3.1) - (3.3).

Если перемещать линию целевой функции параллельно самой себе в направлении, указанном стрелкой, то в точке, где она покинет область допустимых значений, целевая функция достигнет оптимального значения. В рассматриваемой задаче это будет точка E , координаты которой действительно дают максимальное значение целевой функции.

Если прямая, изображающая целевую функцию, параллельна одной из границ области допустимых значений и перемещается в процессе поиска оптимального значения по направлению к этой границе, то задача имеет бесчисленное множество оптимальных решений. Этим решениям соответствуют координаты точек, принадлежащих указанной границе области допустимых значений.

Подобный случай рассмотрен выше при решении задачи с целевой функцией $F=4000x_1 + 1000x_2$ методом перебора.

Помимо рассмотренных ситуаций, когда область допустимых значений ограничена многоугольником, возможны еще два случая: открытая область допустимых значений и отсутствие области допустимых значений.

Открытой называют такую область, которая ограничена не со всех сторон. При наличии открытой области задача либо не имеет решения, либо имеет решение только на максимум или только на минимум. Например, на рис. 4 с целевой функцией F_1 задача имеет решение только на максимум. При отсутствии ограничения (3.5) задача с целевой функцией F_1 не имеет решения. С целевой функцией F_2 задача имеет решение только на минимум. При отсутствии ограничения (3.6) задача с целевой функцией F_2 не имеет решения.

Отсутствие области допустимых значений означает, что не существует

пар x_1, x_2 , которые удовлетворяли бы условиям (3.2). Подобная ситуация может сложиться, когда условия (3.2) противоречат друг другу. Очевидно, что в области допустимых значений задача не имеет решения.

Ключевые слова

Математический анализ, адекватность модели, математическое программирование, экстремальное значение, целевая функция, неотрицательность, задача линейного программирования, динамические программирование, целочисленным программированием, линейное уравнение, допустимые решение, оптимальное решение, дополнительные переменные, геометрическая интерпретация, базисное решение, координаты точек, открытая область.

Вопросы для самопроверки

1. Что изучает наука «Математическое программирование»?
2. Что выражает целевая функция?
3. Приведите классификацию задач математического программирования.
4. Сформулируйте основную задачу линейного программирования.
5. Что такое допустимое решение задачи линейного программирования?
6. Что такое оптимальное решение задачи линейного программирования?
7. В чем суть графического метода?
8. Какое решение называется допустимым?
9. Как можно построить график ЗЛП?
10. Какие виды ЗЛП решаются графическим методом?
11. Алгоритм решения ЗЛП графическим методом?
12. Как определяется оптимальное решение?
13. Когда ЗЛП не имеет решение?
14. Какую область допустимых значений называется открытым?
15. В каких случаях не существует область допустимых значений?

Лекция 4. СИМПЛЕКСНЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧ

$$F = 30x_1 + 20x_2 + 40x_3 \rightarrow \max, \quad (4.3)$$

условия

$$\begin{aligned} 2x_1 + 4x_2 + 3,2x_3 &\leq 30, \\ 5x_1 + 2,6x_2 + 7x_3 &\leq 50, \end{aligned} \quad (4.4)$$

$$\begin{aligned} 3,5x_1 + 5x_2 + 2,4x_3 &\leq 40, \\ 2x_1 + 1,3x_2 + 1,5x_3 &\leq 20, \\ x_1, &\geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned} \quad (4.5)$$

В начале, задачу необходимо привести в так называемую каноническую форму, преобразовав неравенства (4.4) в равенства. Для этого к левой части неравенства добавляют новую переменную (дополнительную). Если левая часть меньше правой, то дополнительную переменную прибавляют, если левая часть больше правой — вычитают. В данной задаче во всех условиях левые части меньше правых. Поэтому после приведения в каноническую форму они приобретают следующий вид:

$$\begin{aligned} 2x_1 + 4x_2 + 3,2x_3 + x_4 &= 30, \\ 5x_1 + 2,6x_2 + 7x_3 + x_4 &= 50, \\ 3,5x_1 + 5x_2 + 2,4x_3 + x_4 &= 40, \\ 2x_1 + 1,3x_2 + 1,5x_3 + x_7 &= 20, \end{aligned} \quad (4.5)$$

где x_4, x_5, x_6, x_7 — дополнительные переменные.

В результате получена система уравнений (4.5), в которой число неизвестных (7) превосходит число уравнений (4). Как известно, такая система имеет бесчисленное множество решений. Дополнительные переменные так же, как и основные, должны быть неотрицательными.

Дополнительные переменные необходимо включить в состав целевой функции. Но поскольку значения этих переменных не должны влиять на значения целевой функции, их снабжают нулевыми коэффициентами

$$F = 30x_1 + 20x_2 + 40x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7. \quad (4.6) \quad (1.28)$$

После перевода в каноническую форму задачу линейного программирования можно сформулировать следующим образом: из множества решений системы (4.5) выбрать такое неотрицательное решение, которое

наилучшим образом отвечает заданному критерию оптимальности.

Используя элементы канонической формы задачи, заполняют исходную табл. 4.1 по столбцам в следующем порядке: i — номера ограничений;

x_{bi} — базисные переменные (в первой симплексной таблице это дополнительные переменные); c_i — коэффициенты целевой функции при базисных переменных; a_{io} — свободные члены;

a_{ij} — коэффициенты уравнений; c_j — коэффициенты целевой функции, в том числе C_0 — свободный член целевой функции, который в большинстве задач равен нулю.

Таблица 4.1

Симплексная таблица 1

	Базисные переменные	c_j	0	30	20	40	0	0	0	0
I	x_{bi}	c_i	a_{i0}	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	x_4	0	30	2	4	3,2	1	0	0	0
2	x_5	0	50	-5	2,6	7	0	1	0	0
3	x_6	0	40	3,5	5	2,4	0	0	1	0
4	x_7	0	20	2	1,3	1,5	0	0	0	1
Индексная строка			0	-30	-20	-40	0	0	0	0

Индексная строка служит для выяснения ситуации, возникающей в конце очередного шага. Содержимое ее клеток вычисляют по формуле

$$Z_j = \sum_{i=1}^m c_i a_{ij} - c_j \quad (4.6)$$

где m — количество условий.

Симплексным методом задачу решают поэтапно. В начале каждого этапа выбирают базис — набор переменных, число которых равно числу условий задачи. Каждое условие должно содержать одну базисную переменную с коэффициентом, равным единице. В остальных условиях эта

переменная должна отсутствовать, то есть иметь коэффициент, равный нулю. В рассматриваемой задаче на первом этапе решения (табл. 4.1) в качестве базисных приняты дополнительные переменные x_4, x_5, x_6, x_7 .

Индексная строка симплексной таблицы указывает, есть возможность перехода к следующему этапу или нет: если она содержит, по крайней мере, одно отрицательное число (в задачах на максимум) или одно положительное число (в задачах на минимум), то можно переходить к следующему этапу.

1. Найти в индексной строке наибольший по абсолютной величине отрицательный (в задачах на максимум) или наибольший положительный элемент (в задачах на минимум). Столбец, расположенный над выбранным элементом, называют разрешающим (главным) столбцом. В данной задаче на первом этапе главным столбцом будет третий (наибольший по абсолютной величине отрицательный элемент в индексной строке – 40).

2. Поочередно разделить значения элементов нулевого столбца на соответствующие положительные элементы главного столбца

$$\frac{a_{io}}{a_{im}} \quad (4.7)$$

Ту строку, в которой результат такого деления оказался минимальным, называют разрешающей строкой. В нашем примере на первом этапе главной строкой будет вторая, так как из всех величин

$$\frac{a_{10}}{a_{13}} = \frac{30}{3,2} = 9,365,$$

$$\frac{a_{20}}{a_{23}} = \frac{50}{7} = 7,143,$$

$$\frac{a_{30}}{a_{33}} = \frac{40}{2,4} = 16,667,$$

$$\frac{a_{40}}{a_{43}} = \frac{20}{1,5} = 13,333,$$

величина $\frac{a_{20}}{a_{23}} = 7,143$ минимальна. Элемент, находящийся на

пересечении главного столбца и главной строки, называют главным или разрешающим элементом,

3. Провести частичную замену базиса. Для этого нужно исключить базисную переменную, принадлежащую главной строке (в данном случае — переменную x_5), и переместить на ее место новую переменную с номером, равным номеру главного столбца (переменную x_3).

4. Рассчитать новые значения элементов главной строки ($a'_{эл.j}$)

$$a'_{эл.j} = \frac{a_{эл.j}}{a_{эл}}$$

где $a_{эл.j}$ — старые значения элементов главной строки, $a_{эл}$ — значение главного элемента.

В результате главный элемент получит новое значение, равное 1. 5.

Вычислить новые значения (a'_{ij}) в остальных - строках, включая индексную,

$$a'_{ij} = a_{ij} - a_{изг} \cdot a'_{эл.j} \quad (4.8)$$

(для всех $i \neq \text{гл}$).

где $a_{эл.j}$ — значения элементов преобразуемых строк, расположенных в главном столбце, $a'_{эл.j}$ — элементы преобразованной главной строки, расположенные в тех же столбцах, что и преобразуемые элементы.

В итоге все элементы главного столбца, кроме главного элемента, получают нулевые значения.

Результаты указанных выше преобразований приведены в табл. 4.2.

В новой таблице проанализируем ее индексную строку. Если поиск оптимального варианта следует продолжать, то снова выбираем главный элемент, заменяем одну из базисных переменных и преобразуем симплексную таблицу.

Таблица 4.2

Симплексная таблица 2

	Базисные переменные и x_{bi}	c_j	0	30	20	40	0	0	0	0
		c_i	a_{i0}	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
1	x_4	0	7,142	-0,2858	2,812	0	1	-0,4554	0	0
2	x_3	40	7,143	0,7143	0,374	1	0	0,1423	0	0
3	x_6	0	22,86	1,786	4,109	0	0	-0,3415	1	0
4	x_7	0	9,286	0,9286	0,742	0	0	0,2135	0	1
Индексная строка			285,7	-1,428	-5,144	0	0	5,692	0	0

В данном случае главным является второй столбец (минимальное по модулю отрицательное значение в индексной строке — 5,144), главной строкой — первая, где частное от деления $\frac{a_{10}}{a_{12}} = \frac{7,142}{2,812}$. После преобразования табл. 4.2 получаем табл. 4.3.

Таблица 4.3

Симплексная таблица 3

	Базисные переменные	c_j	0	30	20	40	0	0	0	0
		c_i	a_{i0}	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
1	x_4	20	2,540	-0,1016	1	0	0,3555	-0,1619	0	0
2	x_3	40	6,200	0,7520	0	1	-0,1321	0,2024	0	0
3	x_6	0	12,42	2,222	0	0	-1,461	0,3237	1	0
4	x_7	0	7,399	1,004	0	0	-0,2642	-0,0932	0	1
Индексная строка			298,8	-1,951	0	0	1,829	4,859	0	0

Преобразование симплексных таблиц продолжают до тех пор, пока не будет получено оптимальное решение.

В рассматриваемой задаче оптимальное решение будет получено на четвертом этапе. Индексная строка этой таблицы не содержит отрицательных чисел, что свидетельствует об оптимальном варианте (табл. 4.4).

Таблица 4.4

Симплексная таблица 4

	Базисные переменные x_{bi}	c_j	0	0	0	0	0	0	0	
		c_{bi}	a_{i0}	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
1	x_2	20	3,113	0	1	0	0,2888	-0,1471	0,0457	0
2	x_3	40	1,960	0	0	1	+0,3623	0,0928	-0,3384	0
3	x_1	30	5,637	1	0	0	-0,6575	0,1457	0,450	0
4	x_7	0	1,739	0	0	0	0,3959	-0,2395	-0,4518	1
Индексная строка			309,779	0	0	0	0,54	5,143	0,878	0

4.2. Экономический анализ симплексного метода.

Симплексная таблица позволяет провести анализ полученного решения. Значение целевой функции находится в индексной строке столбца a_{i0} . В столбце; x_{bi} перечислены переменные, вошедшие в последний базис, а в столбце a_{i0} — значения этих переменных (оптимальное опорное решение). В данной задаче оптимальное опорное решение $x_1 = 5,637$, $x_2 = 3,113$, $x_3 = 1,960$, $x_7 = 1,739$, целевая функция равна 309,78. Переменные, не попавшие в базис (x_4, x_5, x_6), приравнивают к нулю.

Значения дополнительных переменных позволяют судить о характере выполнения условий при реализации оптимального варианта. В условии вида

$$\sum_{j=1}^n a_{ij} x_j \leq a_{i0}, \quad (4.9)$$

которые, как правило, получают из ограничений на использование ресурсов, равенство дополнительной переменной нулю говорит о том, что при реализации оптимального варианта данный ресурс будет использован полностью. Если дополнительная переменная не равна нулю, значит, что в оптимальном варианте ресурс, расходы которого не должны превышать x_{i0} будет израсходован не полностью, а объем неиспользованного ресурса будет равен значению дополнительной переменной.

В нашем примере дополнительная переменная $x_7 = 1,739$ входит в

условие 4.9. Значит, при реализации оптимального варианта 1 739 единиц ресурса 4 останутся неизрасходованными.

Обобщая сказанное, сформулируем алгоритм решения задачи линейного программирования симплексным методом:

- 1) привести систему ограничений к канонической форме;
- 2) составить исходную симплексную таблицу;
- 3) проанализировать симплексную таблицу: если оптимальный результат не достигнут и нет признаков того, что его невозможно получить (нет области допустимых значений или она открыта в направлении искомого оптимума), переходим к п. 4. В противном случае прекращаем решение;
- 4) найти разрешающий (ключевой) столбец;
- 5) определить разрешающую (ключевую) строку;
- 6) заменить базисную переменную в ключевой строке;
- 7) пересчитать элементы симплексной таблицы по формулам (4.8) и (4.9) и получить новую симплексную таблицу;
- 8) перейти к п. 3.

Ключевые слова

Экономико-математические модели, оптимизация, экстремальное значение, целевая функция, неотрицательность, задача линейного программирования, допустимое решение, оптимальное решение, дополнительные переменные, базисное решение, симплексный метод, каноническая форма, множество решений, разрешающий столбец, симплексное отношение, разрешающая строка, главный элемент, опорное решение, индексная строка.

Вопросы для самопроверки

1. Какие виды задач могут решаться симплексным методом?
2. Как строится каноническая форма задачи линейного программирования?
3. Как строится стандартная форма задачи линейного программирования?

4. Каков алгоритм симплексного метода?
5. Какой столбец и строка называются разрешающими?
6. Что такое симплексное отношение?
7. Какой элемент называется главным?
8. Расскажите правила составления очередной симплексной таблицы.
9. Как проводят экономико-математический анализ решения задачи?

Лекция 5. РАСПРЕДЕЛИТЕЛЬНЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

5.1. Понятие и сущность распределительного метода.

Распределительный метод пригоден для решения широкого класса задач линейного программирования, переменные в которых выражены в одних и тех же единицах измерения. Такие задачи могут решаться также симплексным методом, но распределительный метод процесс решения значительно упрощает. Распределительный метод часто применяют для решения задач, связанных с перевозками грузов, которые получили название транспортные задачи. Сущность транспортной задачи заключается в следующем: имеется m пунктов отправления грузов с запасом a_i ($i = 1, 2, \dots, m$) и n пунктов назначения с потребностями в грузах b_j ($j = 1, 2, \dots, n$). Известна стоимость перевозки единицы груза по каждому возможному маршруту c_{ij} . Требуется составить план перевозок, то есть определить, сколько груза надо перевезти по каждому маршруту $(x_{11}, x_{12}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{m1}, \dots, x_{mn})$. Сведем все заданные и искомые величины в табл.5.1. Сформулируем условия задачи в общем виде:

- 1) количество грузов, перевозимых по маршрутам, должно равняться запасам в пунктах отправления;

$$\sum_{j=1}^n x_{ij} = a_i, i = 1, 2, \dots, m; \quad (5.1)$$

- 2) количество грузов, перевозимых по маршрутам, должно равняться

потребностям в пунктах назначения;

Матрица транспортной задачи

Таблица 5.1

Пункты назначения Пункты отправления	1	2	...	j	...	i	Запасы груза a_i
1	C_{11} x_{11}	C_{12} x_{12}	C_{1j} x_{1n}		C_{1n} x_{1n}		a_1
2	C_{21} x_{21}	C_{22} x_{22}	C_{2j} x_{2n}		C_{2n} x_{2n}		a_2
i	C_{i1} x_{i1}	C_{i2} x_{i2}	C_{ij} x_{ij}		C_{in} x_{in}		a_i
m	C_{m1} x_{m1}	C_{m2} x_{m2}	C_{mj} x_{mj}		C_{mn} x_{mn}		a_m
Потребность в грузах, b_j	1	2	b_j		b_n		$\sum a_i$ $\sum b_j$

$$\sum_{i=1}^m x_{ij} = b_j, j = 1, 2, \dots, n; \quad (5.2)$$

3) количество распределяемых грузов и потребности в них должны быть равны m n

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j; \quad (5.3)$$

4) переменные в задаче должны быть неотрицательными

$$x_{ij} \geq 0.$$

Необходимо составить план перевозок, при котором стоимость перевозимых грузов будет минимальной,

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (5.4)$$

В общей постановке задачи встречаются два случая, требующие различных подходов в алгоритме решения. Первый случай возникает тогда, когда выполняется условие (5.3), то есть запасы грузов в пунктах отправления и потребности грузов в пунктах назначения равны. Такая задача называется *закрытой*. Если условие (5.3) имеет вид

$$\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j, \quad (5.4)$$

то модель задачи называется *открытой* (второй случай). Открытая модель транспортной задачи предполагает, что количество распределяемых грузов и потребность в них неодинаковы: потребность может быть больше или меньше запасов грузов.

Закрытая транспортная задача имеет m , n переменных, $m + n$ условий, из которых $m + n - 1$ независимых, то есть число занятых клеток при решении задачи должно равняться $m + n - 1$.

Модель транспортной задачи имеет следующие особенности:
все ограничения представлены в виде равенств;
коэффициенты при неизвестных в ограничениях равны единице;
каждая неизвестная входит только в два уравнения — по строке и по столбцу.

Любое решение, удовлетворяющее всем условиям транспортной задачи и содержащее не более $m + n - 1$ решений, называется *допустимым*.

Допустимое решение, обращающее в минимум целевую функцию, называется *оптимальным*. Распределительным методом линейного программирования могут решаться следующие землеустроительные задачи:

составление оптимальных планов перевозок продукции с земельных участков в хозяйственные центры (например, кормов на фермы);

устранение чересполосицы и других недостатков землепользования;

рациональное размещение сельскохозяйственных культур на полях различного плодородия (например, чередование культур в севооборотах) ;

оптимальное размещение сельскохозяйственных угодий и севообо-

ротов на различных по качеству почвах и местоположению с уточнением намеченной трансформации земельных угодий.

Порядок решения задач распределительным методом: 1) постановка задачи с формулировкой цели; 2) сбор исходной информации и ее обработка; 3) заполнение исходной матрицы и составление первоначального опорного плана; 4) вычисление промежуточного значения целевой функции и анализ плана на оптимальность; 5) последовательное улучшение плана до получения оптимального решения.

5.2. Методы составления первоначального опорного плана.

В отличие от первого опорного плана, получаемого с помощью симплексных таблиц, распределительный метод предполагает составление реального опорного плана. Поэтому методы составления первого опорного плана в алгоритме транспортной задачи имеют важное значение. Чем ближе опорный план к оптимальному, тем меньше операций требуется для достижения оптимального решения.

Существуют следующие методы построения первоначального плана: минимального элемента, северо-западного угла, аппроксимации и др. Аппроксимация может использоваться также и как самостоятельный метод.

Методы минимального элемента и аппроксимации в решении специализированных задач находят самое широкое применение.

Алгоритм метода минимального элемента сводится к следующему:

- из всех значений c_{ij} в матрице находят наименьшее;
- в клетку с наименьшим значением c_{ij} ставят a_i ,
- если $a_i < b_j$, или b_j , если $b_j < a_i$;
- выбирают следующее по величине наименьшее значение c_{ij} и заносится поставка груза;
- операции повторяют до тех пор, пока весь груз не будет распределен по маршрутам.

При решении задач на минимум методом аппроксимации поступают

следующим образом:

- в каждой строке и столбце находят два минимальных значения c_{ij} ;
- определяют их разность $\mu_i(\mu_j)$;
- из всех разностей выбирают наибольшую R_μ ;
- по строке или столбцу, к которым относится наибольшая разность, в клетку, где размещено, c_{ij}^{\min} , записывают требуемую поставку груза;
- далее операции повторяют с вычислением новых разностей до получения опорного плана.

При решении задачи методом аппроксимации могут возникнуть следующие случаи: - если есть несколько наибольших разностей R_μ , "то предпочтение - отдается той, для которой есть минимальное значение c_{ij}^{\min} , если c_{ij}^{\min} , имеется по нескольким R_μ ; то для решения берут клетку, в которую можно занести большую поставку груза.

Пример 6. В матрицу задачи дополнительно вводят строку и столбец для вычисления разностей. Вначале рассчитывают первые разности, Например, по строке 1 из трех значений c_{11} , c_{12} , c_{13} , которые соответственно равны 1, 5, 8, выбирают два минимальных: $c_{11}=1$ и $c_{12}=5$. Их разность составит: $\mu_i=4(5-1=4)$.

По столбцу 1 разность определится как $6-1=5$. Аналогично проводят расчет по всем строкам и столбцам. Далее из всех разностей, относящихся одновременно к μ_i выбирают наибольшую. Так, из разностей (5, 3, 1, 4, 1, 16) наибольшая (16) относится к третьей строке. В связи с этим в клетку с минимальным значением c_{ij} для этой строки ($c_{32}=2$) заносится возможная поставка груза ($x_{32}=90$). Значения c_{ij} по второму столбцу из решения задачи исключаются и вычисляют новые разности (алгоритм повторяют). Решение закончено, когда все поставки груза распределены (табл. 5.2).

Таблица 5.2.

Решение задачи методом аппроксимации

Пункты назначения Пункты отправления	1	2	3	Запасы груза a_i	Столбец разностей μ_i
1	1 100	5	8	100	4 7 -
2	6 20	15	7 180	200	1 1 1
3	24	2 90	18 210	300	16 6 6
Потребность в грузах, b_j	120	90	390	600	
Строка разностей μ_i	5 5 18	3 - -	1 1 11		

5.3. Методы улучшения опорного плана.

Составив первоначальный опорный план, необходимо проверить его на оптимальность. Если план не оптимален, его последовательно улучшают до получения оптимального решения. Наилучшие результаты при улучшении оптимального плана дает метод потенциалов.

При использовании этого метода вычисляют условные потенциалы α_i, β_j .

Условный потенциал - это специальная оценка единицы груза, расположенного в пунктах назначения и отправления. Потенциалы можно рассчитать по формуле

$$\alpha_i + c_{ij} = \beta_j, \quad (5.5)$$

которая справедлива для занятых клеток.

Такое условие записывают для каждой занятой клетки, где $x_{ij} = 0$.

Причем первому потенциалу присваивают любое значение. Для удобства расчетов и исключения отрицательных потенциалов первому из них присваивают значение, превышающее наибольшее из c_{ij} .

Для задач на минимум план оптимален, если для свободных клеток выполняется следующее условие:

$$\alpha_i + c_{ij} \geq \beta_j, \quad (5.6)$$

В случае неоптимальности плана его улучшают. При этом для клетки, в которой не выполняется условие оптимальности, строят многоугольник.

Правила построения многоугольника и улучшения плана: 1) стороны многоугольника должны пересекаться под прямым углом и располагаться по строкам и столбцам таблицы; 2) вершины многоугольника должны находиться в занятых клетках, кроме одной, начальной, лежащей в исходной клетке (свободной); 3) начальной вершине, лежащей в исходной клетке, присваивают знак (+), далее знаки чередуются; 4) среди всех грузов, расположенных в клетках с отрицательными вершинами, выбирают минимальное значение x_{\min} , далее в соответствии со знаком клетки проводят алгебраическое сложение грузов.

10	2	1
22	3	7
	8	

2	2	1
30	3	7

Улучшенный план

Улучшение обычно начинают с клетки, характеристика которой по модулю наибольшая. Характеристику свободных клеток (\sum_{ij}) находят по

формуле $\sum_{ij} = (\alpha_i + c_{ij}) - \beta_j$.

Это способствует уменьшению количества итераций и быстрейшему достижению оптимального плана.

После каждого улучшения вычисляют значения целевой функции по формулам

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij},$$
$$Z = \sum_{j=1}^n \beta b_j - \sum_{i=1}^m \alpha_i a_i \quad (5.7)$$

Каждое правильное улучшение плана при решении задачи на минимум должно привести к уменьшению значения целевой функции.

5.4. Решение задачи методом потенциалов на максимум.

Первоначальный опорный план находят методом максимального элемента с заполнением в первую очередь клеток с наибольшими значениями c_{ij} ;

план оптимален, если для свободных клеток выполняется следующее условие: $\alpha_i + c_{ij} \leq \beta_j$,

после каждого улучшения значение целевой функции должно возрасть; все остальные операции аналогичны решению задачи на минимум.

Решение задачи методом аппроксимации на максимум:
по каждой строке и столбцу находят два наибольших значения c_{ij} ;
определяют их разность μ_i, μ_j
из всех разностей выбирают наибольшую, R_{μ} ;
по строке или столбцу, к которым относится наибольшая разность, в клетку, где размещается наибольшее значение c_{ij} ; записывают требуемую поставку груза;
если есть несколько наибольших разностей R_{μ} ; то предпочтение отдается той, для которой есть максимальное значение c_{ij}^{\max}

если c_{ij}^{\max} имеется по нескольким R_{μ} ; то для решения берут клетку, в которую можно занести большую поставку груза.

Вырожденные задачи. Если число занятых клеток или положительных решений больше $m + n - 1$, то план составлен неверно и находят новое решение задачи. Если число занятых клеток меньше $m + n - 1$, то план вырожден. В задачах с вырожденным планом невозможно построить замкнутый многоугольник и улучшить план. При использовании метода потенциала в этих случаях нельзя вычислить α_i и β_j по всем клеткам. Вырожденные планы, как правило, появляются тогда, когда сумма нескольких запасов (a_i) равна потребности (b_j) или наоборот, а также в случае равенства отдельных запасов и потребностей, например:

$$a_1 = b_1, a_2 = b_2; a_1 = b_1 + b_2, b_1 = a_1 + a_2 \text{ и т. д.}$$

Преодоление вырожденности осуществляют следующим образом. Одну из свободных клеток условно считают занятой, подставляя в нее перевозки $x_{ij} = (0)$. Количество нулей в сумме с количеством занятых клеток должно быть равно $m + n - 1$. При этом за условно занятую или нулевую клетку надо принять ту, которая не образует с другими занятыми клетками замкнутые многоугольники (табл.5.3). Кроме того, ноль надо ставить в ту клетку, которая при улучшении плана в многоугольнике будет иметь знак (+). Например, если клетка (2-1) неоптимальна и ноль стоит в клетке (3-2), то при построении многоугольника последней присвоят знак (-) и перераспределение груза будет неосуществимо. В этом случае за нулевую надо принять клетку (1-3).

Открытая модель транспортной задачи. Если запасы и потребности груза не сбалансированы, то есть $\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$, то транспортная задача является открытой.

Таблица 5.3.

Преодоление вырожденности задачи

Пункты назначения Пункты отправления	Потенциалы	B_1	B_2	B_3	Запасы груза, a_i
	β_j α_i	12	11	10	
A_1	110	2 100 -	1 200 +	8 (0)	300
A_2	-6	+ 6	5	16 -100	100
A_3	8	7	3 -(0)	2 100 +	100
A_4	0	21	25	10 100	100
Потребность груза, b_j		100	200	300	600

Чтобы решить такую задачу, ее приводят к закрытому типу введением фиктивного пункта отправления или потребления груза с запасом a_{m+1} или b_{n+1}

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i; \quad b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j. \quad (5.8)$$

При этом стоимость перевозок грузов по фиктивному пункту обычно принимают равной нулю (табл. 5.6).

Следует иметь в виду, что при решении задачи на максимум и на минимум клетка фиктивного пункта $c_{i, n+1}$ ($i = 1, 2, \dots, m$) $= 0$ или $c_{j, m+1}$ ($j = 1, 2, \dots, n$) заполняется в последнюю очередь.

Таблица 5.4.

Приведение задачи к закрытому типу

Пункты назначения Пункты отправления	Потенциалы	B_1	B_2	B_3	Запасы груза, a_i
	β_j α_i	12,6	12,9	20	
A_1	10,0	2,6 280	2,9 840	10,0 80	1200
A_2	10,1	2,5 390	2,7	4,1	390
A_3	15,2	2,8	3,0	4,8 280	280
A_4	29,0	0	0 420	0	420
Потребность груза, b_j		670	1260	360	2290 2290

Варианты решения задач с использованием матрицы оптимального плана. При этом можно найти: новое оптимальное решение (значение целевой функции в оптимальном плане при этом не изменяется); новое решение с изменением (отклонением от экстремального) значения целевой функции..

Условие существования других оптимальных решений, не таких, как полученные — выполнение, по крайней мере, для одной из свободных клеток следующего равенства

$$\alpha_i + c_{ij} = \beta_j \text{ или } \sum ij = (\alpha + c_{ij}) - \beta_j = 0 \quad (5.9)$$

Чтобы получить новое альтернативное решение для клетки с выполнением условия (5.9), строят новый цикл с перемещением доли плана в свободную клетку.

Необходимость корректировки оптимального плана может появиться,

если надо учесть дополнительное условие, возникшее после окончания решения задачи. В этом случае новое решение задачи с дополнительным ограничением искать не нужно, а корректировку ведут в матрице оптимального плана.

Допустим, получено следующее оптимальное решение (табл. 5.4),

Значение целевой функции в таблице минимально, $Z = 129\ 100$. Для клетки (2-3) $\sum ij = 10 + 3 - 13 = 0$, что говорит о наличии, как минимум, еще одного оптимального решения (табл. 5.5).

Значение целевой функции в этом оптимальном плане также равно $129\ 100 (Z=121900)$.

Анализ оптимального плана (табл. 5.4) свидетельствует о том, что могут быть получены по крайней мере, еще 5 (не считая табл. 5.3) различных вариантов.

Исследуем свободные клетки:

$$\sum 12 = 10 + 3 - 11 = 2 \quad \sum_{33} = 8 + 9 - 13 = 4,$$

$$\sum 13 = 10 + 5 - 13 = 2 \quad \sum_{41} = 7 + 7 - 12 = 2,$$

$$\sum 32 = 8 + 8 - 11 = 5 \quad \sum_{43} = 7 + 6 - 13 = 0.$$

Если составить новый план с размещением его в клетке (1-2), то транспортные затраты возрастут на 3600 коп. ($1800 \cdot 2$), (1-3)- на 11 000 коп. ($2 \cdot 5500$), (3-2) - на 9000 коп. ($1800 \cdot 5$), (3-3) -на 22 000 коп. ($5500 \cdot 4$), в клетке (4-3) - на 4000 коп. ($2000 \cdot 2$).

Наиболее близки к оптимальному варианты с размещением доли плана в клетках (1-2) или (4-3).

При корректировке результатов решения задач может возникнуть необходимость перераспределения ресурсов между пунктами отправления или потребления.

Таблица 5.5.

Оптимальный план (вариант 1)

Пункты назначения Пункты отправления	Потенциалы	B_1	B_2	B_3	Запасы груза, a_i
	β_j α_i	12,6	12,9	20	
A_1	10	2 13000	3	5	13000
A_2	10	2 2000	1 7300 -	3 +	9300
A_3	8	4 12000	8	9	12000
A_4	7	7	+ 2700	- 5500 6	8200
Потребность, b_j		27000	10000	5500	42500 42500

Например, запасы в пункте A_2 составят 9200, а в пункте A_3 - 12 100 т. 9 этом случае грузы перераспределяют между маршрутами по занятым клеткам. Так» в клетке (2-1) будет 1900 т, а в клетке (3-1) - 12 100 т. Значение целевой функции при решении задачи на минимум возрастет на $[(c_{31} - c_{21}) \cdot 100]$ или $[(\alpha_2 - \alpha_3) \cdot 100]$, что составит 200 коп. (2-100).

При перераспределении груза между занятой и свободной клеткой, то есть при организации нового маршрута, значение целевой функции также увеличится. Если в клетке (2—2) объем груза будет равен 1700, а в клетке (3—2) — 100, то значение целевой функции возрастет на $[(c_{32} - c_{22}) \cdot 100]$, или на 700 коп. $[(8-1) \cdot 100]$.

При одновременном увеличении поставок и потребностей в грузе по занятому маршруту на единицу значение целевой функции изменится на

величину разности $\beta_j - \alpha_i$, то есть на c_{ij} . Например, если запас груза в пункте A_4 8201, а потребность в нем в пункте B_2 - 10 001, то значение целевой функции увеличится на 4 единицы (11-7).

5.5. Решение транспортных задач с дополнительными ограничениями.

При решении землеустроительных' задач распределительным методом могут возникнуть случаи, требующие ограничить объемы перевозимых грузов по различным маршрутам или, наоборот, увеличить грузооборот. В связи с этим алгоритм распределительного метода линейного программирования будет иметь некоторые особенности, которые заключаются в следующем.

1. Если необходимо зафиксировать перевозку груза объемом d_i

или d_j по заданному маршруту ($x_{ij} = d_{ij}$), поступают следующим образом. Из соответствующих запасов груза и, потребностей в нем a_i и b_j , вычитают значение d_{ij} и оценку клетки блокируют. Блокировка оценки заключается в присвоении c_{ij} заведомо невыгодного значения. Например, при решении задач на минимум значение оценки увеличивают на несколько порядков. Так, если оценка в клетке равна 5, то, изменив ее на 5000, добиваемся невыгодности заданного маршрута. Далее решение продолжают до получения оптимального плана, в который после контроля вычислений заносят вычтенную ранее поставку груза $x_{ij} = d_{ij}$. Клетку с заблокированной оценкой в процессе всего решения задачи считают свободной.

2. Если нужно предусмотреть развитие перевозок грузов по заданному маршруту при выполнении ограничения, $x_{ij} \geq d_{ij}$ до решения задачи, значение d_{ij} вычитают из соответствующих запасов груза a_i и потребностей в нем b_j . Этот прием используют для того, чтобы заранее обеспечить выполнение требуемого ограничения по нижней границе $x_{ij} = d_{ij}$. Затем решение продолжают до получения оптимального плана, в котором вычтенное ранее

значение d_{ij} прибавляют к находящемуся в анализируемой клетке.

3. Если требуется ограничить перевозку грузов $x_{ij} \leq d_{ij}$ или зафиксировать объемы транспортировки в определенных пределах $d_{ij} \leq x_{ij} \leq D_{ij}$ то вначале решают задачу до получения оптимального плана, не обращая внимание на поставленные условия. Затем оптимальный план анализируют с точки зрения выполнения поставленных ограничений и при необходимости корректируют его, перераспределяя грузы в соответствии с необходимостью и предъявляемыми требованиями.

Пример решения и оформления задачи по определению оптимальных маршрутов перевозок грузов. Постановка задачи. В хозяйстве есть пункты, производящие продукцию и потребляющие ее. Например, животноводческие фермы, потребляющие корма, и севообороты, их производящие. Нужно найти такие маршруты перевозок продукции, которые обеспечат минимум транспортных затрат.

В качестве оценок критерия оптимальности задачи применяют стоимость перевозки продукции с каждого пункта отправления на каждый пункт назначения, например стоимость 1 ткм в зависимости от расстояний перевозки.

Исходная информация: потребности продукции в пунктах назначения, ее запасы в пунктах потребления; расстояние перевозок грузов.

Расчетные показатели. Стоимость перевозки единицы грузов в зависимости от расстояний.

С помощью указанной постановки можно решать задачи по формированию сырьевых зон промышленных производств, использующих продукцию сельского хозяйства (сахарных заводов, спиртовых заводов, хлопкоочистительных пунктов, заводов по производству травяной муки и др.). В этом случае находят хозяйства, сырье которых может быть доставлено на заводы с минимальными транспортными затратами.

Задача 1.5.

В колхозе 4 севооборота: 2 полевых, 2 кормовых;

3 фермы: молочная, свиноводческая и овцеводческая. Потребность в кормах на фермах соответственно 27 000, 10 000 и 5500 ц корм. ед. Выход кормов с севооборотов соответственно 13 000, 9300, 12 000 и 8200 ц корм. ед. Необходимо составить такой план перевозок кормов на фермы, чтобы стоимость перевозок была минимальной. Исходные данные приведены в табл. 5.6.

Таблица 5.6.

Исходные данные и первая итерация

Севообороты Фермы	Потенциалы	Полевой		Кормовой		Потребность в кормах, ц корм.ед.
	β_j	1	2	1	2	
	α_i	12	14	14	17	
МФ	10	2 13000	+	4 12000 8	7 - 2000	9300
СФ	13	3 х	3,0	9 х	+ 4 700	12000
ОФ	11	5 х	3 х	9 х	6 5500	8200
Выход кормов, ц корм. ед		13000	9300	12000	8200	42500

Этапы решения задачи:

- 1) сформулировать задачу в общем виде;
- 2) составить расширенную экономико-математическую модель;
- 3) решить задачу методом потенциалов (или любым другим методом);
- 4) вычислить значение функции цели.

Формулировка задачи в общем виде:

$$Z = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \rightarrow \min$$

$$\sum_{i=1}^m a_{ij} = \sum_{j=1}^n b_j, \quad i=1, 2, \dots, m,$$

$$\sum_{j=1}^n x_{ij} = a_i$$

$$j=1, 2, \dots, n, \quad x_{ij} \geq 0. \quad \sum_{i=1}^m x_{ij} = b_j$$

Расширенная экономико-математическая модель:

$$Z = 2x_{11} + 2x_{12} + 4x_{13} + 7x_{14} + 3x_{21} + x_{22} + 8x_{23} + 4x_{24} + 5x_{31} + 2x_{32} + 9x_{33} + 6x_{34} \rightarrow \min$$

$$x_{11} + x_{12} + x_{13} + x_{14} = 27000,$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 10000,$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 5500,$$

$$x_{11} + x_{21} + x_{31} = 13000$$

$$x_{12} + x_{22} + x_{32} = 9300,$$

$$x_{13} + x_{23} + x_{33} = 12000,$$

$$x_{14} + x_{24} + x_{34} = 8200,$$

$$x_{11}, x_{21}, \dots, x_{34} \geq 0.$$

После улучшения задачи во второй итерации получаем оптимальный план (табл.5.7.)

После первой итерации значение целевой функции составляло

$$Z_1 = 13\,000 \cdot 2 + 12\,000 \cdot 4 + 200 \cdot 7 + 9300 \cdot 1 + 700 \cdot 4 + 5500 \cdot 6 = 26\,000 + 48\,000 + 9300 + 14\,000 + 2800 + 33000 = 133\,100 \text{ коп.} = 1331 \text{ руб.}$$

После второй итерации

$$Z_2 = 13\,000 \cdot 2 + 2000 \cdot 2 + 12\,000 \cdot 4 + 7300 \cdot 1 + 2700 \cdot 4 + 5500 \cdot 6 = 26\,000 + 4000 + 48\,000 + 7300 + 10800 + 33000 = 129\,100 \text{ коп.} = 1291 \text{ руб.}$$

Таблица 5.7.

Вторая итерация

Севобороты Фермы	Потенциалы	Половой		Кромовой		Потребность в кормах, ц корм. ед.
	β_j	1	2	1	2	
	α_i	12	14	14	17	
МФ	10	2 13000	2 2000	4 12000	7 х	27000
СФ	11	3 х	1 7300	8 х	4 2700	10000
ОФ	9	5 х	3 х	9 х	6 5500	5500
Выход кормов, ц корм.ед		13000	9300	12000	8200	42500

Ключевые слова

Модель, моделирование, переменные, экономико-математические модели, оптимизация, математический анализ, экстремальное значение, целевая функция, неотрицательность, задача линейного программирования, оптимальное решение, поставщик, потребитель, запасы груза, перевозки груза, пунктов отправления, стоимость доставки, потребность, опорный план, наименьший элемент, северо-западный, транспортная задача, стоимость перевозок.

Вопросы для самопроверки

1. В чем суть транспортной задачи?
2. Экономико-математический модель транспортной задачи.
3. Сколько условие поставлено в модели?
4. Что известно что неизвестно в модели?

5. Какая модель называется закрытой?
6. Как находится опорное решение транспортной задачи?
7. Как найти потенциалы?
8. Как строится цикл?
9. Каков критерий оптимальности транспортной задачи?
10. Как интерпретировать оптимальное решение транспортной задачи?

Лекция 6. СТАДИИ МОДЕЛИРОВАНИЯ И ПОСТРОЕНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ

6.1. Стадии математического моделирования.

Для изучения множества связей (экономических, социальных, экологических) в проектах землеустройства применяют экономико-математическое моделирование, которое описывает основные закономерности экономического явления или процесса в виде уравнений и неравенств.

В ходе землеустроительного анализа формируется цель задачи, выявляются главные факторы, достаточно полно отражающие моделируемое явление.

При различных способах математического моделирования принципы построения моделей едины. Моделирование можно разделить на следующие стадии: постановка задачи; анализ качественных и количественных зависимостей; получение и обработка исходной информации; выбор математического метода решения задачи; математическая формулировка экономических, технологических, землеустроительных и других условий задачи; построение экономико-математической модели; решение задачи; экономико-математический анализ оптимального решения.

При постановке задачи следует обосновать круг моделируемых процессов: определить неизвестные переменные величины, коэффициенты при переменных, константы, выбор критерия оптимальности.

6.2. Учет факторов, действующих на построение математической модели.

Переменные отражают состав и размеры отраслей, технологические способы производства продукции, соответствующие типу сельскохозяйственного производства. Они делятся на основные, вспомогательные, дополнительные. Основные переменные — это сельскохозяйственные культуры, размеры угодий, поголовье скота. Они могут быть разделены на переменные растениеводства и животноводства.

Коэффициенты — постоянные величины в данной задаче, выражают нормы затрат ресурса на единицу переменной, продуктивность единицы площади или единицы поголовья скота, пропорциональность между переменными и ресурсами. Коэффициенты при переменных в целевой функции выражают стоимость и натуральное выражение единицы переменной.

Константы — известные величины поставленной задачи, выражают объемы производственных ресурсов, плановые показатели развития сельскохозяйственного предприятия и наличие лимитированных ресурсов.

Переменные, коэффициенты, константы, связанные между собой, образуют ограничения. Основными являются ограничения по использованию земельных и трудовых ресурсов, основных средств, по использованию и производству кормов, гарантированному объему производства и другим условиям, обеспечивающим выполнение технологических, агротехнических, экономических и экологических требований.

На основе знаний, приобретенных по экономике и организации сельскохозяйственного производства, землеустроительному проектированию, основам математического программирования, студент должен приобрести практические навыки построения экономико-математических задач, их решения и анализа полученных результатов.

Задача 1. Построить ограничения по использованию земельных

ресурсов и по формированию структуры использования пашни при следующих условиях: -пашня - 3000 га, пастбища -1750, сенокосы — 630 га.

Озимые зерновые занимают не менее 15 % площади посева, яровые — не более 32 %.

Условия, связанные с использованием пашни, пастбищ и сенокосов, записывают в виде неравенств

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3000x_{10},$$

$$x_8 \leq 1750,$$

$$x_9 \leq 630 - x_{10},$$

где x_1 - площадь под озимыми зерновыми;

x_2 - яровыми; x_3 - сахарной свеклой;

x_4 - кукурузой на силос;

x_5 — многолетними травами на зеленый корм после уборки озимых;

x_8 - площадь используемых пастбищ (не более 1750 га);

x_9 — используемых сенокосов (не более 630 га);

x_{10} — площадь сенокосов, которая может быть трансформирована в пашню.

В задачу вводят ограничения на максимально или минимально необходимый размер посева определенной группы культур

$$x_1 \geq 450 \text{ (не менее 15 \% площади пашни),} \quad (6.1)$$

$$x_2 \leq 960 \text{ (не более 32 \% площади пашни)} \quad (6.2)$$

В задачу также вводят ограничения, регулирующие технологическую связь между культурами,

$x_1 \geq x_6$ после преобразования получаем

$$-x_1 + x_6 \leq 0 \quad (6.3)$$

где x_1 — площадь посева озимых зерновых; x_6 — площадь повторного посева однолетних трав на зеленый корм после уборки озимых зерновых.

$$0,3x_4 - 1,5x_7 \leq 0, \quad (6.4)$$

где x_7 - площадь семенников многолетних трав.

Коэффициент при x_4 означает нормы высева семян, при x_7 - урожайность семян многолетних трав.

Таким образом, группа ограничений по использованию земельных ресурсов может быть представлена следующим образом:

1. По использованию земельных ресурсов

$$\sum_{j=1} x_j \leq b_i \pm x_i, i \in M_1 \quad (6.5)$$

2. По структуре использования пашни а) по структуре посева отдельных

групп культур

$$\sum_{j=1}^l x_j \leq \alpha_{ij} b_i, i \in M_2 \quad (6.6)$$

- б) по структуре посева основной и пожнивной культуры :

$$\sum_{j=l+1}^k x_j - \sum_{j=1}^l x_j \leq 0, i \in M_3; \quad (6.7)$$

- в) по потребности в семенах и их производству :

$$\sum_{j=1}^l t'_{ij} x_j - \sum_{j=1}^l t''_{ij} x_j \leq 0, i \in M_3 \quad (6.8)$$

Обозначения

x_j - площадь угодий и сельскохозяйственных культур j -го вида;

b_i - площадь угодий i -го вида;

x_i - объемы трансформации i -го вида угодья;

α_{ij} - дельный вес i -го вида культуры на пашне i -го вида;

t_{ij}, t''_{ij} , — коэффициенты уровня затрат и уровня производства на единицу площади;

$j=1...l$ — число сельскохозяйственных культур j -го вида;

$j = 1 + 1...k$; — число пожнивных сельскохозяйственных культур j -го вида;

M_1 - множество видов угодий;

M_2 — множество групп сельскохозяйственных культур, гарантирующих определенный объем производства;

M_3 — множество культур повторного сева;

M_4 - множество культур, семена которых производятся в хозяйстве.

Задача 2. Используя данные задачи 1, составить ограничения по производству и потреблению кормов, необходимых для развития животноводства.

Дополнительные исходные данные приведены в табл. 6.1.

Таблица 6.1.

Исходные данные

Показатели	Зерновые яровые	Кукуруза на силос	Многолетние травы на зеленый корм и сено
Выход корма, ц корм. ед/га	25	40	22
Затраты корма на 1 гол. скота, ц корм. ед.	—	—	—

Продолжение

Показатели	Повторный посев однолетних трав	Сенокосы	Пастбища	КРС	Овцы
Выход корма, ед/га	15	9	6		
Затраты корма на 1 гол. -скота, ц корм. ед.	-	-	-	55	10

Ограничения по производству и использованию кормов определяют

рост поголовья скота и его продуктивности, темпы развития животноводства. Ограничения по группам кормов строятся так, чтобы их количество в рационе было заключено между минимально необходимой и максимально допустимой границами в соответствии с принятыми нормами кормления скота

$$-25x_2 - 40x_4 - 22x_5 - 15x_6 - 9x_8 - 6x_9 + 70x_{11} + 10x_{12} \leq 0; \quad (6.9)$$

$$-25x_2 + 14x_{11} + x_{12} \leq 0 \quad (\text{по концентратам}); \quad (6.10)$$

$$-40x_4 + 22x_{12} \leq 0 \quad (\text{по силосу}), \quad (6.11)$$

где x_{11} - поголовье КРС, гол.; x_{12} - поголовье свиней, гол.

Неравенство (6.9) показывает, что объем производства кормов (корм. ед.) должен быть не меньше потребности в них, в ограничениях (6.10) и (6.11) даны балансы по отдельным видам кормов.

Коэффициенты при x_2, x_4, \dots, x_9 означают выход корм. ед. с 1 га посевов кормовой культуры, сенокосов и пастбищ. Коэффициенты при x_{11} и x_{12} — потребность в корм. ед. на 1 гол. скота.

Математическая формулировка ограничений по кормовым ресурсам может быть представлена следующим образом:

$$-\sum_{j=1}^k v_{ij} x_j + \sum_{j=k+1}^n a_{ij} x_j \leq 0, i \in M_5.$$

Обозначения

x_j — посевные площади сельскохозяйственных культур и угодий при $j=1 \dots k$;

x_j — поголовье j -х видов скота при $j=k+1 \dots n$;

x_{ij} — $k+1 \dots n$ - число видов скота и половозрастных групп j -го вида;

v_{ij} - выход корма i -го вида с единицы j -и культуры, угодья;

a_{ij} - потребность в i -м виде корма на 1 гол. j -го вида скота;

M_5 — множество видов скота.

Запись ограничений по гарантированному объему производства в общем виде

$$\sum_{j=1}^n q_{ij}x_j - x_i \geq Q_i, i \in M \quad (6.12)$$

где q_{ij} — выход товарной продукции i -го вида с единицы j -й отрасли;

Q_i — планируемый объем производства товарной продукции i -го вида;

x_i — продукция сверх плана;

M — множество видов товарной продукции.

Условие неотрицательности переменных;

$x_i \geq 0$; $x_i \geq 0$.

Ключевые слова

Модель, моделирование, переменные, экономико-математические модели, оптимизация, математический анализ, экстремальное значение, целевая функция, неотрицательность, задача линейного программирования, оптимальное решение, поставщик, потребитель, запасы груза, перевозки груза, пунктов отправления, стоимость доставки, потребность, опорный план, наименьший элемент, северо-западный, транспортная задача, стоимость перевозок.

Вопросы для самопроверки

1. В чем суть транспортной задачи?
2. Экономико-математический модель транспортной задачи.
3. Сколько условие поставлено в модели?
4. Что известно что неизвестно в модели?
5. Какая модель называется закрытой?
6. Как находится опорное решение транспортной задачи?
7. Как найти потенциалы?
8. Как строится цикл?
9. Каков критерий оптимальности транспортной задачи?
10. Как интерпретировать оптимальное решение транспортной задачи?

Лекция 7. ЭКОНОМИКО-МАТЕМАТИЧЕСКИЙ АНАЛИЗ НА ОСНОВЕ ДВОЙСТВЕННЫХ ОЦЕНОК И КОЭФФИЦИЕНТОВ ЗАМЕЩЕНИЯ

7.1. Понятие двойственных оценок

Постановка задания. В проектах землеустройства необходимо получить высокий экономический эффект при ограниченных земельных, трудовых, материально-денежных ресурсах и производственных фондах. Здесь важное значение приобретает экономико-математический анализ решения оптимизационных задач. Основная роль при этом отводится оценкам, выражающим эффективность использования ресурсов, сравнительную эффективность видов деятельности, вошедших и не вошедших в решение задачи,

В данном разделе студенты ознакомятся с методикой оценки влияния каждого ресурса на оптимальное решение на основе объективно обусловленных (двойственных) оценок. Двойственные оценки показывают, как изменится целевая функция при увеличении (уменьшении) каждого вида ресурса на единицу, кроме того, они характеризуют устойчивость экономических моделей к корректировке исходных данных,

Студентам предлагается изучить две интерпретации задачи линейного программирования: прямую и двойственную; свойства двойственных оценок, способы их использования при корректировке оптимального решения и для получения вариантов оптимального решения с помощью коэффициентов последней симплексной задачи,

Двойственность в линейном программировании заключается в том, что для каждой задачи (прямая задача) путем замены некоторых ее элементов на двойственные можно сформулировать двойственную.

Прямая задача отражает наиболее эффективную комбинацию использования ресурсов, при которой достигается максимум целевой функции. В двойственной задаче получают (оптимальную) комбинацию расчетных

оценок ограниченных ресурсов.

Прямая задача может быть сформулирована следующим образом:

имеются ресурсы b_1, b_2, \dots, b_m , которые надо использовать для выпуска продукции, Определить объем производства продукции вида $x_1, x_2, x_3, \dots, x_n$, при этом стоимость продукции

$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$ должна быть максимальной,

а расход ресурсов не должен превышать их наличия

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + c_{1n}x_n &\leq b_1, \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + c_{1n}x_n &\leq b_2, \\a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mj}x_j + \dots + c_{mn}x_n &\leq b_m \\x_1 \geq 0; x_2 \geq 0; \dots, x_j \geq 0; \dots, x_n &\geq 0,\end{aligned}\tag{7.1}$$

где c_j - стоимость единицы j -й продукции; a_{ij} - нормативный коэффициент потребления i -го ресурса на производство единицы j -й продукции. На базе этих же исходных данных может быть сформулирована двойственная задача, в которой переменными являются оценки имеющихся ресурсов $u_1, \dots, u_2, \dots, u_i, \dots, u_m$. Они должны быть такими, чтобы затраты при заданных ресурсах были минимальные при условии, что сумма оценок ресурсов, расходуемых на единицу продукции, была меньше, чем стоимость единицы этой продукции, Математическая запись двойственной задачи

$Z' = b_1u_1 + b_2u_2 + \dots + b_iu_i + \dots + b_mu_m \rightarrow \min$ при условиях:

$$\begin{aligned}a_{11}u_1 + a_{21}u_2 + \dots + a_{i1}u_i + \dots + a_{m1}u_m &\geq c_1, \\a_{12}u_1 + a_{22}u_2 + \dots + a_{i2}u_i + \dots + a_{m2}u_m &\geq c_2, \\a_{1n}u_1 + a_{2n}u_2 + \dots + a_{in}u_i + \dots + a_{mn}u_m &\geq c_n, \\u_1 \geq 0, u_2 \geq 0, \dots, u_i \geq 0, \dots, u_m &\geq 0.\end{aligned}\tag{7.2}$$

Содержание задания. Построить прямую и двойственные задачи; установить количественную зависимость между их показателями, проанализировать оптимальное решение с помощью двойственных оценок, используя коэффициенты последней симплексной таблицы, получить

вариантные решения экономико-математической задачи.

Цель задания: студенты должны понять свойства и экономическое содержание двойственных оценок и коэффициентов замещения, получить навыки в их использовании при анализе результатов решения, научиться находить узкие места в использовании ресурсов, сдерживающих дальнейшее развитие сельскохозяйственного производства, с помощью коэффициентов последней симплексной таблицы. Рассмотрим примеры.

Задача 3. В хозяйстве нужно улучшить 100 га пастбищ, На это выделено 9000 руб, и 350 чел,-дн. Затраты материально-денежных средств на 1 га при поверхностном улучшении 20 руб,, при коренном — 100 руб,, дополнительные трудовые затраты на 1 га соответственно составляют 0,5 и 4 чел,-дн. В качестве коэффициентов целевой функции выступает выход зеленого корма с 1 га; при поверхностном улучшении — 100 ц, при коренном — 300 ц. Определить максимальный выход зеленого корма при различных способах улучшения пастбищ, если x_1 - площадь пастбищ поверхностного улучшения; x_2 , — площадь пастбищ коренного улучшения.

Построим экономико-математическую модель прямой задачи.

Целевая функция $Z = 100x_1 + 300x_2 \rightarrow \max$

Условия:

1) по площади улучшения угодий

$$X_1 + X_2 \leq 100;$$

2) по использованию материально-денежных средств

$$20x_1 + 100x_2 \leq 9000;$$

3) по дополнительному использованию труда

$$0,5 x_1 + 4x_2 \leq 350;$$

4) неотрицательность переменных

$$X_1 \geq 0, \quad X_2 \geq 0,$$

Условие задачи представим в матричной форме (табл. 7.1).

Таблица 7.1.

Исходная симплексная таблица прямой задачи

№ п/п	Базисные переменные	Оценки, c_i	Объем ре- сурсов, a_{i0}	100	300	0	0	0
				x_1	x_2	x_3	x_4	x_5
1	X_3	0	100	1	1	1	0	0
2	X_4	0	9000	20	100	0	1	0
3	X_5	0	350	0,5	4	0	0	0
m+1	$Z_j - c_j$	0		-100	-300	0	0	0

Таблица 7.1. содержит исходную информацию прямой задачи: базисные переменные (x_i), объемы ресурса (b_i), технико-экономические коэффициенты при переменных (a_{ij}) и коэффициенты целевой функции (c_j). В процессе решения все показатели исходной симплексной таблицы меняют свое числовое значение и экономическое содержание.

Коэффициенты при небазисных переменных из нормативов затрат производственных ресурсов превращаются в коэффициенты замещения, которые в дальнейшем используют как средство экономического анализа и корректировки оптимального решения.

Производственные ресурсы по мере их потребления на производство продукции становятся значениями основных базисных переменных, равных расчетному объему производства, дополнительных переменных, означающих недоиспользование какого-либо ресурса.

Результаты решения содержит последняя симплексная таблица (табл. 7.2).

Таблица 7.2.

Последняя симплексная таблица

№ п/п	Базисные переменные	Оценки, c_i	Объем ресурсов	100	300	0	0	0
				X_1	X_2	X_3	X_4	X_5
1	X_1	100	14,285	1	0	1,143	0	-0,286
2	X_4	0	142,862	0	0	-8,570	1	-22,857
3	X_2	300	85,714	0	1	-0,143	0	0,286
m+1	$Z_j - c_j$		27142,7	0	0	71,4	0	57,151

На пастбищах площадью 100 га целесообразно на 14,3 га (x_1) провести поверхностное и на 85,7 га (x_2) — коренное улучшение. При этом ресурсы труда используются полностью. Остаются недоиспользованными материально-денежные ресурсы на сумму 142, 8 руб. Общий выход зеленой массы с улучшенных пастбищ составит 27 170 ц.

Рассмотрим содержание и формулировку двойственной задачи.

Переменными будут u_1 , u_2 , u_3 - оценки использования пастбищ, материально-денежных средств и трудовых ресурсов.

Целевая функция

$$Z = 100 u_1 + 9000 u_2 + 350 u_3 \rightarrow \min$$

Условия:

1) по поверхностному улучшению пастбищ

$$u_1 + 20 u_2 + 0,5 u_3 \geq 100;$$

2) по коренному улучшению пастбищ

$$u_1 + 100 u_2 + 4 u_3 \geq 300;$$

3) неотрицательность переменных

$$u_1 \geq 0; u_2 \geq 0; u_3 \geq 0,$$

Построим симплексные таблицы двойственной задачи (табл. 7.3, 7.4).

В оптимальном плане получим $u_1 = 71,4$ и $u_3 = 57,151$. Данные значения переменных удовлетворяют условиям двойственной задачи, в этом можно убедиться, подставив их в первоначальные ограничения

$$71,4 + 0,5 \cdot 57,151 \cong 100,$$

$$71,4 + 4 \cdot 57,151 \cong 300,$$

Исходная симплексная таблица

Таблица 7.3.

№ п/п	Базисные переменны е	Оцен ки, C_i	Объем ресурсо в							
				U_1	U_2	U_3	U_4	U_5	Y_1	Y_2
1	Y_1		100	1	20	0,5	-1	0	1	0
2	Y_2		300	1	100	4	0	-1	0	1
m+1	$Z_j - c_j$		0	-100	- 9000	-350	0	0	0	0
m+2			400M	2M	120 M	4,5 M	-M	-M	M	M

Последняя симплексная таблица

Таблица 7.4.

№ п/п	Базисные переменны е	Оценк и, C_i	Объем ресурсо в					
				U_1	U_2	U_3	U_4	U_5
1	U_1	100	71,4	1	8,570	0	1,143	-0,143
2	U_3	350	57,151	0	22,857	1	-0,286	0,286
m+1	$Z_j - c_j$		27142,7	0	142,86	0	14,285	85,714

Целевая функция приняла минимальное значение, которое совпадает с максимумом целевой функции прямой задачи. Последние симплексные таблицы прямой и двойственной задач содержат одни и те же данные, сопоставляя которые видим, что нет необходимости отдельно решать двойственную задачу. Коэффициенты (m+1) строки *при* небазисных переменных x_3 и x_5 совпадают со значениями переменных u_1 и u_3 находящихся в базисе двойственной задачи. Эти величины — двойственные оценки, они показывают, как изменится значение целевой функции при изменении запаса соответствующего вида ресурса на единицу, то есть характеризуют относительный вклад каждого ресурса в достижение оптимума.

Коэффициенты замещения a_{ij} и двойственные оценки последней симплексной таблицы позволяют исследовать чувствительность оптимального решения к изменению первоначальных условий задачи (табл. 2.9). При росте урожайности пастбищ поверхностного улучшения от 100 до 120 ц/га изменится значение целевой функции — максимум производства

зеленого корма составит 27 428 ц, при этом ограничения прямой задачи и оптимальное решение останутся такими же.

В двойственной задаче изменится одно ограничение по оценке поверхностного улучшения пастбищ. Вместо

$$u_1 + 20 u_2 + 0,5 u_3 > 100 \quad \text{будет} \quad u_1 + 20 u_2 + 0,5 u_3 \geq 120,$$

Минимальная стоимость ресурса совпадает с максимумом прибыли, Подставив результаты решения двойственной задачи (табл. 7.4) в преобразованное ограничение, получим новое значение объективно обусловленной оценки использования пастбищ u

$$u_1 + 20,0 + 0,5 \cdot 57,151 = 120, \quad u_1 = 91,4$$

то есть рост урожайности пастбищ поверхностного улучшения приведет к увеличению оценки единицы лимитированного ресурса — 1 га площади пастбищ. При повышении урожайности пастбищ коренного улучшения лимитированным ресурсом будут трудовые ресурсы при росте урожайности до 400 ц/га, Оценка трудовых ресурсов u_3 составит 82,2 единицы

$$71,4 + 100,0 + 4 u_3 = 400, \quad u_3 = 82,2,$$

Отсюда следует, что система двойственных оценок чувствительно реагирует на изменение коэффициентов целевой функции, но в определенных пределах.

Нижняя граница варьирования коэффициентов c_1 при базисных переменных — максимальное частное от деления чистого эффекта $c_j - z_j$ и коэффициента замещения ($a_{ij} \geq 0$) по тем небазисным переменным, которые при введении их в базис уменьшают i -ю базисную переменную.

Верхняя граница увеличения коэффициентов c_j при базисных переменных равна минимальному отношению чистого эффекта $c_j - z_j$ и коэффициентов замещения ($a_{ij} \leq 0$) по небазисным переменным, которые при введении в базис увеличивают i -ю базисную переменную.

Двойственные оценки имеют определенную устойчивость к изменению объемов ресурсов, Например, изменив количество улучшаемых пастбищ до

110 га, получим измененную систему ограничений

$$x_1 + x_2 \leq 110,$$

$$20 x_1 + 100 x_2 \leq 9000,$$

$$0,5 x_1 + 4 x_2 \leq 350,$$

Решим систему уравнений: $x_1 = 25,7$ га, $x_2 = 84,3$ га. Изменив площадь улучшаемых пастбищ, получили новое оптимальное решение, при котором выход зеленой массы с улучшенных пастбищ увеличится до 27 857 ц. Система двойственных оценок осталась прежней.

Пределы возможного изменения объема ресурсов определяют, исходя из условия неотрицательности переменных в оптимальном плане.

Каждая задача двойственной пары формально является самостоятельной задачей линейного программирования и может решаться независимо от другой, При этом решение одной из них ведет к решению другой.

7.2. Использование двойственных оценок при анализе и корректировке оптимального решения

Двойственные оценки — важный инструмент экономико-математического анализа, Задача анализа — отыскание условий, при которых сравнительно небольшая корректировка исходных данных оставляет полученное решение оптимальным, так как двойственные оценки ограничивающих ресурсов более чувствительны к изменениям коэффициентов целевой функции, чем объемов ресурсов.

В процессе экономико-математического анализа можно получить несколько вариантов решения задачи, выявить погрешности исходных данных, изучить дополнительные условия проекта, позволяющие повысить интенсивность использования земли или других ресурсов, исследовать экономическую эффективность полученного решения.

Основной инструмент экономико-математического анализа оптимального решения — двойственные (оптимальные) оценки, выражающие

эффективность использования ресурсов, и коэффициенты небазисных переменных, характеризующих взаимозаменяемость ресурсов относительно заданного критерия (коэффициенты замещения).

В рассмотренном выше примере переменные u_1 и u_3 выражают оценку использования земельных и трудовых ресурсов, которые ограничены. На 14,3 га из 100 га пастбищ целесообразно провести поверхностное улучшение, на 85,7 - коренное улучшение; выделенные для этого трудовые ресурсы будут затрачены целиком. Материально-денежные ресурсы израсходованы не полностью. В оптимальном решении они оказались избыточными, поэтому их оценка равна нулю. Дальнейший рост материально-денежных средств не повлияет на оптимальный план производства зеленой массы. И, наоборот, при увеличении площади улучшенных пастбищ на 1 га (x_3) производство зеленого корма возрастает на 71,4 ц за счет увеличения площади пастбищ поверхностного улучшения x_1 на 1,143 га и уменьшения на 0,143 га пастбищ коренного улучшения (x_2). При уменьшении x_2 на 0,143 га высвобождаются трудовые ресурсы в объеме 0,572 чел.-дн, ($0,143 \cdot 4$), которые могут быть использованы для дополнительных работ по поверхностному улучшению пастбищ на площади 1,143 га ($1,143 \cdot 0,5$) = 0,572 чел.-дн.

Следовательно, двойственная оценка 71,4 — оценка эффективности использования пастбищ. Она показывает, насколько возросло бы значение целевой функции, если бы площадь пастбищ была увеличена на 1 га.

Коэффициенты при небазисной переменной x_3 в табл. 7.2. (+1,143; -8,570; -0,143) представляют собой нормы замещения, которые характеризуют изменение площади пастбищ (x_1 и x_2) различного вида, значение недоиспользованных материально-денежных средств (x_4) при увеличении площади улучшения (помимо 100 га) еще на 1 га.

Увеличив трудовые ресурсы на 1 чел.-дн., оптимальное решение также изменится. Коэффициенты столбца переменной x_5 (табл. 7.2) показывают, что при увеличении фонда трудовых ресурсов станет выгодным проводить коренное улучшение на площади 0,286 га, сократив площади поверхностного

улучшения на эту же площадь. Недоиспользованные материально-денежные средства сократятся на 22,855 руб, $[20 \cdot (-0,286) + 100 \cdot 0,286]$. Прирост значения целевой функции составит 57,151 ц зеленого корма, 57,151 - оценка экономической эффективности трудовых ресурсов в формировании оптимального плана производства зеленых кормов.

Однако двойственные оценки позволяют судить об эффекте не любых, а лишь сравнительно небольших изменений объемов ресурсов. При резких изменениях сами оценки могут стать другими. Поэтому двойственные оценки измеряют, эффективность малых приращений объемов ресурсов в конкретных условиях задачи, при которых пределы изменений не нарушают условия не отрицательности переменных в оптимальном плане. Прирост различных ресурсов дает неодинаковый эффект, и двойственные оценки позволяют выявить узкие места, сдерживающие рост эффективности производства, показывают, какие ресурсы более дефицитные, какие менее и какие недефицитные.

В нашем примере при выборе способа улучшения пастбищ увеличение их площади дает прирост зеленого корма 71,4 ц на 1 га прироста площади, 57,2 ц — при росте трудовых ресурсов на 1 чел.-дн., и нет прироста за счет увеличения материально-денежных средств на улучшение пастбищ.

На основе выполненного анализа можно сделать вывод, что при необходимости роста выхода зеленых кормов предпочтительней увеличение их выхода за счет пастбищ поверхностного улучшения; увеличение фонда трудовых ресурсов для коренного улучшения дает меньший эффект.

Решение экономико-математических задач с помощью коэффициентов последней симплексной таблицы. Двойственные оценки и коэффициенты замещения позволяют вносить также коррективы в полученное решение, если требуется изменить проект. Изменения в оптимальном плане возможны в следующих случаях: увеличивается значение переменной, вошедшей в базисное решение; в базис вводится небазисная переменная, основная или дополнительная.

Вносимые коррективы ведут к изменению значений базисных переменных. Они изменяются на величину, равную произведению коэффициентов замещения на значение вводимой в базис переменной, при этом следует учитывать знак при коэффициенте замещения. Значения базисной переменной уменьшаются, если коэффициенты замены положительные, и увеличиваются, если отрицательные. Величина вводимой поправки не должна быть больше значения переменной, находящейся в базисе, в противном случае будет нарушено правило неотрицательности переменных в задачах линейного программирования ($x_j \geq 0$). Значения основных переменных, вошедших в базисное решение, оптимальные или после его корректировки должны быть положительными. Для дополнительной переменной требование неотрицательности необязательно, Отрицательное значение дополнительной переменной означает недостачу данного вида ресурса и равнозначно величине, на которую предполагается увеличить ресурс. Максимально допустимое значение переменной, вводимой в базис при неизменяющихся ресурсах, равно наименьшему частному от деления значения базисных переменных и соответствующих коэффициентов замещения, введение которых уменьшает значение базисных переменных.

Задача 2.5. На 200 га пашни необходимо разместить две культуры - товарную пшеницу и фуражный ячмень, Урожайность пшеницы 10 ц/га, ячменя - 20 ц/га, Затраты ручного труда на выращивание этих культур 1250 чел,-дн., механизированного труда — 250 чел,-дн, Затраты ручного и механизированного труда соответственно на производство 1 ц пшеницы 0,3 и 0,6 чел,-дн., ячменя — 0,4 и 0,03 чел,-дн, Цена 1 ц пшеницы - 7 руб., ячмень полностью идет на корм скоту. Обозначим валовой сбор пшеницы x_1 , ячменя – x_2 , Требуется найти оптимальное сочетание площадей пшеницы и ячменя, при котором стоимость товарной продукции будет максимальной.

Целевая функция

$$Z = 7x_1 \rightarrow \min$$

Условия:

1) по использованию пашни

$$0,1 x_1 + 0,05 x_2 \leq 200;$$

2) обеспечение потребности в средствах механизации

$$0,6 x_1 + 0,3 x_2 \leq 250$$

3) обеспечение потребности в рабочей силе

$$0,3 x_1 + 0,4 x_2 \leq 1250;$$

4) неотрицательность переменных

$$x_1 \geq 0; x_2 \geq 0,$$

Не останавливаясь на процедуре вычисления, приведем первую и последнюю симплексные таблицы (табл. 7.5, 7.6).

Первая симплексная таблица

Таблица 7.5.

№	Базисные переменные	Оценки c_i	Свободные члены	Небазисные переменные				
				X_1	X_2	X_3	X_4	X_5
1	X_3	0	200	0,1	0,05	1	0	0
2	X_4	0	250	0,06	0,03	0	1	0
3	X_5	0	1250	0,3	0,4	0	0	0
	$Z_j - c_j$		0	7	0	0	0	0

Таблица 7.6.

Последняя симплексная таблица

№	Базисные переменные	Оценки и c_i	Свободные члены	Небазисные переменные				
				X_1	X_2	X_3	X_4	X_5
1	X_1	7	2000	1	0,5	10	0	0
2	X_4	0	130	0	0	-0,6	1	0
3	X_5	0	650	0	0,25	-3	0	1
	$Z_j - c_j$		14000	0	3,5	70	1	0

При анализе табл. 7.6 видим, что 200 га пашни целесообразно

использовать для производства пшеницы (x_1), остаются недоиспользованными средства механизации (x_4) и трудовые ресурсы (x_5).

В решение не вошла переменная x_2 — производство фуражного зерна. Однако в рацион кормления животных было бы желательно включить концентраты, источник поступления которых — фуражное зерно (x_2). Для корректировки полученного решения установим допустимые границы предполагаемого изменения

$$2000 : 0,5 = 4000, 650 : 0,25 = 2600,$$

Наименьшее частное 2600. Примем, что для кормления животных требуется 1000 ц концентратов. Методика расчета дана в табл. 7.7. Если ввести в решение переменную $x_2 = 1000$ ц, производство товарного зерна и стоимость товарной продукции снизятся, недоиспользованный ручной труд возрастет.

Таблица 7.7.

Оптимальное решение, скорректированное с учетом потребности в фуражном зерне ($x_2=1000$ ц)

Базисные переменные	Значения базисных переменных	Коэффициент при переменной X_2	Произведение коэффициента при X_2 на 1000ц	Расчетный вариант оптимального плана при увеличении урожайности ячменя $X_2=1000$ ц
X_1	2000	0,5	500	$2000 - (+500) = 1500$
X_4	130	0	0	130
X_5	650	-0,25	-250	$650 - (-250) = 900$
X_2	0	-1	-1000	$0 - (-1000) = 1000$
$Z_j - c_j$	14000	3,5	3500	$14000 - (13500) = 10500$

Рассмотренный пример носит условный характер, однако методика

проведения экономико-математического анализа останется той же, Корректировка оптимального решения с помощью коэффициентов замещения ограничена базисом оптимального решения. При существенных изменениях задачу следует решать вновь.

Вопросы для самопроверки

Математический анализ, экстремальное значение, целевая функция, неотрицательность, задача линейного программирования, оптимальное решение, дополнительные переменные, базисное решение, каноническая форма, двойственной, исходная задача, первоначальная задача, стоимости ресурсов, стоимостном выражение, матрица-столбец, матрица-строка, симметричность, несимметричность, опорное решение, индексная строка.

Вопросы для самопроверки

1. Понятие двойственной задачи.
2. Постановка двойственной задачи.
3. Что отражает исходная задача?
4. Что отражает двойственная задача?
5. Объясните использование оптимальной решении двойственной задаче.
6. Решение экономической задачи с использованием последней симплексной таблицы.
7. Значение целевой функции в исходной и двойственной задаче.
8. Пути сопоставлении прямой и двойственной задаче.
9. Какова связь между исходной и двойственной задачами?
10. Приведите классификацию двойственных задач (симметричные и несимметричные).
11. Приведите различные виды двойственных задач.

Лекция 8. ПРОИЗВОДСТВЕННЫЕ ФУНКЦИИ И ИХ ЭКОНОМИЧЕСКИЕ ХАРАКТЕРИСТИКИ

8.1. Понятие, виды производственных функций.

Производственные функции - это математически выраженные связи и зависимости результатов производства от производственных факторов, Производственные функции, как правило, описываются одним уравнением, при этом результативный показатель производства (y) является функцией некоторых аргументов (x)

$$y = (x_1, x_2, \dots, x_n), \quad (8.1)$$

где x_1, x_2, \dots, x_n - факторы производства.

Использование производственных функций позволяет проводить анализ различных производственных параметров, прогнозировать уровень результативного признака, находить экономические оптимумы, определять дополнительные продукты факторов, коэффициенты эффективности и взаимозаменяемости ресурсов.

Существуют следующие способы представления производственных функций: табличный, графический, аналитический и номографический.

Для математического выражения производственных функций используют следующие уравнения:

1) линейные,

а) для парной зависимости

$$y = a_0 + a_1x, \quad (8.2)$$

б) для множественной зависимости m

$$y = a_0 + \sum_{i=1}^m a_i x_i; \quad (8.3)$$

2) степенные

а) для парной зависимости

$$y = a_0 x^{a_1}, \quad (8.4)$$

б) для множественной зависимости

$$y = a_0 \prod_{i=1} x_i^{a_i}; \quad (8.5)$$

3) гиперболические

$$y = a_0 + \frac{a_1}{\sqrt[n]{x}}; \quad (8.6)$$

в частном случае уравнение гиперболы

$$y = a_0 + \frac{a_1}{x}; \quad (8.7)$$

4) полиномиальные

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_j x^j, \quad (8.8)$$

в частном случае уравнение квадратичной параболы

$$y = a_0 + a_1 x + a_2 x^2; \quad (8.9)$$

5) кинетические

$$y = a_0 \cdot \prod x_i^{a_i} e^{-J_i x_i}; \quad (8.10)$$

6) уравнения асимптотического роста

$$y = a_0 - a_1 \cdot 10^{-b_1 x}, \quad (8.11)$$

где y — результативный показатель; x_i — факторы производства; a_1, b_i, J_i — коэффициенты уравнений; a_0 — свободные члены уравнений; e — основание натуральных логарифмов.

В экономическом анализе в землеустройстве встречаются два типа связи между результативными показателями и факторами производства: функциональная и корреляционная. Связь функциональная в том случае, если каждому значению одного признака соответствует строго определенное значение результативного показателя. Однако на практике при анализе использования земельных ресурсов и различных параметров производства чаще встречается неполная форма связи между изучаемыми явлениями — корреляционная.

При корреляционной связи одному и тому же значению фактора производства могут соответствовать несколько значений результативных показателей. Такова, например, зависимость урожайности от количества

вносимых удобрений и эрозированности участков, себестоимости продукции от размера производства и территории. Изменение факторов производства при этом не всегда сопровождается строго определенным изменением результативных показателей, но в среднем это влияние проявляется достаточно хорошо.

В ходе экономико-статистического моделирования определяют корреляционные зависимости и параметры производственных функций. Для расчета параметров моделей используют различные методы, но наиболее распространен способ наименьших квадратов. Применение способа наименьших квадратов позволяет находить параметры производственных функций с помощью решения системы нормальных уравнений, различных для связи каждого вида.

8.2. Определение параметров производственных функций.

Рассмотрим применение этого метода для некоторых вышеназванных уравнений.

1. При парной линейной зависимости $y = (a_0 + a_1x)$ система может быть приведена к следующему виду:

$$\begin{cases} a_0 + a_1 \cdot \frac{\sum x}{n} = \frac{\sum y}{n} \\ a_0 \cdot \frac{\sum x}{n} + a_1 \cdot \frac{\sum x^2}{n} = \frac{\sum xy}{n} \end{cases}$$

или

$$\begin{cases} na_0 + a_1 \sum x = \sum y, \\ a_0 \sum x + a_1 \sum x^2 = \sum xy, \end{cases}$$

где n - число эмпирических данных.

2. Для множественной трехфакторной линейной зависимости вида

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 \quad (8.12)$$

$$\begin{cases} na_0 + \sum x_1 + a_1 + \sum x_2 \cdot a_2 + \sum x_3 \cdot a_3 = \sum y \\ \sum x_1 \cdot a_0 + \sum x_1^2 \cdot a_1 + \sum x_1 x_2 \cdot a_2 + \sum x_1 x_3 \cdot a_3 = \sum x_1 y, \\ \sum x_2 \cdot a_0 + \sum x_1 x_2 \cdot a_1 + \sum x_2^2 \cdot a_2 + \sum x_2 x_3 \cdot a_3 = \sum x_2 y, \\ \sum x_3 \cdot a_0 + \sum x_1 x_3 \cdot a_1 + \sum x_1 x_3 \cdot a_2 + \sum x_3^2 \cdot a_3 = \sum x_3 y, \end{cases} \quad (8.13)$$

3. Для гиперболической зависимости $y = a_0 + \frac{a_1}{x}$

$$\begin{cases} na_0 + \sum \frac{1}{x} \cdot a_1 = \sum y \\ \sum \frac{1}{x} \cdot a_0 + \sum \frac{1}{x^2} \cdot a_1 = \sum \frac{y}{x} \end{cases} \quad (8.14)$$

4. Для квадратичной параболы $y = a_0 + a_1 x + a_2 x^2$

$$\begin{cases} na_0 + \sum x \cdot a_1 + \sum x^2 \cdot a_2 = \sum y \\ \sum x \cdot a_0 + \sum x^2 \cdot a_1 + \sum x^3 \cdot a_2 = \sum xy \\ \sum x^2 \cdot a_0 + \sum x^3 \cdot a_1 + \sum x^4 \cdot a_2 = \sum x^2 y \end{cases} \quad (8.15)$$

Примером расчета параметров линейного уравнения парной связи могут служить данные, характеризующие зависимость площади защитных лесонасаждений от крутизны склонов в районах эрозии почв (табл. 8.1).

Таблица 8.1.

Расчет параметров уравнения прямой

Крутизна склонов, град, x:	Среднее значение интервала, град	Площадь лесополос, %, y	x^2	X, y	Расчетное значение y
1-2	1.5	2.85	2.25	4.35	2.71
2-2.5	2.25	2.98	5.06	6.75	3.32
2.5-3	2.75	3.65	7.85	10.18	3.73
3-3.5	3.25	4.30	11.56	13.98	4.14
Σ	9.75	13.78	25.43	35.26	—
	2.44	3.45	6.36	8.81	—

Система уравнений при этом будет выглядеть следующим образом:

$$\begin{cases} a_0 + 2,44a_1 = 3,45, \\ 2,44a_0 + 6,36a_1 = 8,81. \end{cases}$$

Решив систему, получим следующие результаты:

$$a_0 = 1,48; a_1=0,82,$$

уравнение связи при этом примет вид

$$y = 1,48 + 0,82x,$$

где y - площадь лесополос на пашне, %; x - средняя крутизна склонов пахотных земель, град.

Проведенные расчеты позволяют с наибольшей достоверностью установить степень связи между крутизной водосбора и увеличением насаждений пашни. При построении производственных функций необходимо учитывать следующее:

точность производственной функции выше при большем числе эмпирических данных, включаемых в расчет, то есть при больших выборках; производственная функция должна быть удобна для практического использования и анализа. С этой целью в нее должны быть включены только те факторы, которые оказывают наиболее существенное влияние на результативный показатель. Во избежание искажений при построении производственных функций в них не должны включаться факторы, находящиеся между собой в состоянии функциональной или тесной корреляционной связи;

факторы-аргументы, включаемые в производственную функцию, должны быть статистически достоверными и количественно измеримыми.

Производственная функция должна иметь экономический смысл, то есть чтобы результативный и факториальный показатели были логически связаны. Факторы, включаемые в модель, должны быть представлены одним признаком (абсолютным или относительным, натуральным или стоимостным).

О практической ценности построенных производственных функций можно судить только после оценки полученных результатов, которую проводят на основе расчета коэффициентов корреляции, корреляционных отношений и различных статистических величин, характеризующих тесноту связи результативного и факториального показателей.

Тесноту связи между двумя взаимозависимыми рядами характеризует линейный коэффициент корреляции, который показывает, существует ли и насколько велика связь изучаемых явлений.

Есть много формул расчета коэффициента парной корреляции, но наиболее удобна в практике следующая:

$$r = \frac{n \sum x_j J_j - \sum x_j \sum J_j}{\sqrt{\left[n \sum x_j^2 - \left(\sum x_j \right)^2 \right] \cdot \left[n \sum J_j^2 - \left(\sum J_j \right)^2 \right]}} \quad (8.16)$$

Линейный коэффициент корреляции r может принимать любые значения -1 – $+1$. Если величина r отрицательная, то связь между изучаемыми явлениями обратная, если положительная — связь прямая.

Коэффициенты линейной корреляции указывают на следующую степень связи: 0 — $\pm 0,15$ — отсутствие связи; $0,16$ — $\pm 0,20$ - плохая;

$\pm 0,21$ – $\pm 0,30$ - слабая; $\pm 0,31$ – $\pm 0,40$ - умеренная; $\pm 0,41$ – $\pm 0,60$ - средняя; $\pm 0,61$ – $\pm 0,80$ - высокая; $\pm 0,81$ – $\pm 0,90$ - очень высокая; $\pm 0,91$ – $\pm 1,0$ – полная.

Для малых выборок ($n \leq 50$) коэффициент линейной корреляции можно определять по формуле

$$r = \frac{\sum (x - \bar{x}) \cdot (y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}, \quad (8.17)$$

где x, y - значения фактора и результата; \bar{x}, \bar{y} — средние значения фактора и результата.

При анализе тесноты связи между переменной и несколькими факторами определяют коэффициент множественной корреляции, при криволинейных связях корреляционное отношение — по формуле:

$$R = \sqrt{1 - \frac{\sum_j (y_j - \bar{y}_j)^2}{\sum_j (y_j - \bar{y})^2}} \quad (8.18)$$

где y_j - фактическое значение результативного показателя; \bar{y}_j - среднее арифметическое значение; \bar{y} - среднее значение зависимой переменной,

полученной по уравнению производственной функции при подстановке фактических значений.

В случае парной связи коэффициент множественной корреляции вырождается в парное корреляционное отношение. Когда на результативный признак действуют два факториальных показателя, то при линейной зависимости теснота связи измеряется совокупным коэффициентом корреляции

$$R_{yx_1x_2} = \sqrt{\frac{r_{yx_1}^2 + r_{yx_2}^2 - 2r_{yx_1x_2}r_{yx_1}r_{yx_2}}{1 - r_{yx_1}^2 - r_{yx_2}^2}}, \quad (8.19)$$

где r - коэффициенты парной корреляции между соответствующими факторами и результатом.

В связи с ограниченным количеством исходных данных определяют погрешность (существенность) коэффициентов корреляции и корреляционных отношений, используя критерий, который находят по формуле

$$t_r = \frac{r}{m_r}, \quad (8.20)$$

где m_r - ошибка коэффициента корреляции. Для коэффициента парной корреляции

$$m_r = \frac{1 - r^2}{\sqrt{n - 1}}, \quad (8.21)$$

для коэффициента множественной корреляции

$$m_R = \frac{1 - R^2}{\sqrt{n - k - 1}} \quad (8.22)$$

где n - число измерений; k — число факторов.

При малых выборках ($n \leq 50$) используют следующую формулу:

$$m_r = \sqrt{\frac{1 - r^2}{n - 2}}, \quad (8.23)$$

Полученное расчетное значение критерия достоверности сравнивают с фиксированными значениями по таблице Стьюдента (табл. 8.2).

Примечание, P — вероятность ожидаемых выводов, ν - число степеней

свободы ($V = n - 2$).

Если расчетное значение t_r больше табличного, то коэффициент корреляции является достоверным.

Таблица 8.2.

Значения Г-распределения Стьюдента

$V \backslash P$	0,95	0,99	0,999	$v \backslash P$	0,95	0,99	0,999
4	2,78	4,60	8,61	19	2,093		
5	2,57	4,03	6,86	20	2,086		
6	2,45	3,71	5,96	25	2,064		
7	2,37	3,50	5,41	30	2,045		
8	2,31	3,36	5,04	35	2,032		
9	2,26	3,25	4,78	40	2,023		
10	2,23	3,17	4,59	45	2,016		
11	2,20	3,11	4,44	50	2,009		
12	2,18	3,06	4,32	60	2,001		
13	2,16	3,01	4,22	70	1,996		
14	2,15	2,96	4,14	80	1,991		
15	2,13	2,95	4,07	90	1,987		
16	2,12	2,92	4,02	100	1,984		
17	2,11	2,90	3,97	120	1,980		
18	2,10	2,88	3,92	150	1,980		

При анализе производственных функций также вычисляют коэффициент детерминации по формуле

$$D = r^2 \quad 8.24)$$

Коэффициент детерминации показывает, какое влияние оказывают выбранные факторы-аргументы на результативный показатель. Например, если $r = 0,9$, $D = 0,81$, то есть величина результативного показателя y на 81 % ($0,81 \cdot 100$)-зависит от изменения факторов-аргументов, включенных в производственную функцию.

Если необходимо определить возможную величину коэффициента корреляции в генеральной совокупности, рассчитывают его доверительные границы, то есть $r_d = r \pm \Delta r$, где $\Delta r = t \cdot m_r$

При нахождении параметров производственных функций (составлении уравнения регрессии) и небольшом числе наблюдений также рассчитывают ошибку коэффициента регрессии по формуле

$$\sigma_y = \sqrt{(\bar{y^2}) - (\bar{y})^2} \text{ и } \sigma_x = \sqrt{(\bar{x^2}) - (\bar{x})^2} \quad (8.25)$$

где σ_y , σ_x - средние квадратические отклонения; $(\bar{y^2})$ и $(\bar{y})^2$ — средние значения квадратов; $(\bar{x^2})$ и $(\bar{x})^2$ - квадраты средних значений.

Ошибка коэффициента регрессии позволяет определить доверительный интервал для коэффициента регрессии по формуле

$$a_i - t \cdot \frac{\sigma_y}{\sigma_x} \cdot m_r \leq a_{iq} \leq a_i + t \cdot \frac{\sigma_y}{\sigma_x} \cdot m_r. \quad (8.26)$$

где a_i - коэффициент регрессии.

Задача №1. Найти зависимость потерь времени смены на холостые повороты и заезды комбайна СК-6 «Колос» при прямом комбайнировании на уборке зерновых колосовых от различной длины гона.

Исходные данные расположим в таблице и проведем выравнивание ряда по уравнению гиперболы (табл. 8.3).

Таблица 8.3.

Исходные и промежуточные данные для решения задачи

Фактическая величина потерь, %, у	Длина гона, км, х	1/х	1/х ²	у/х
39,2	0,15	6,667	44,444	261,333
24,0	0,25	4,000	16,000	96,000
18,1	0,35	2,857	8,163	51,714
13,8	0,50	2,000	4,000	27,600
10,0	0,75	1,290	1,665	13,333
7,7	1,00	1,00	1,000	7,700
6,6	1,25	0,800	0,640	5,280
5,8	1,50	0,667	0,444	3,867
5,3	1,75	0,571	0,327	3,029
4,9	2,00	0,500	0,250	2,450

Σ 135,4	9,50	20,352	76,933	472,306
----------------	------	--------	--------	---------

Построим систему нормальных уравнений и решим ее

$$\begin{cases} 10a_0 + 20,352a_1 = 135,4 \\ 20,352a_0 + 26,933a_1 = 472,306 \end{cases}$$

$$\begin{cases} a_0 + 2,0352a_1 = 13,540 \\ a_0 + 3,7801a_1 = 23,207 \end{cases}$$

1,7449 a_1 =9,667, a_1 =5,54. Тогда a_0 =2,26. Уравнение гиперболы примет вид

$$y = 2,26 + \frac{5,554}{x}.$$

Для расчета корреляционного отношения воспользуемся табл. 8.4.

Таблица 8.4.

Расчет тесноты связи y и x

Y	$y - \bar{y}$	$(y - \bar{y})^2$	y	$y - \hat{y}$	$(y - \hat{y})^2$
	39,2	658,43	39,19	0,01	0,0001
24,0	10,46	109,41	24,42	-0,42	0,1764
18,1	4,56	20,794	18,09	0,01	0,0001
13,8	0,26	6,068	13,34	0,46	0,2116
10,0	-3,54	12,532	9,65	0,35	0,1225
7,7	-5,84	34,106	7,80	-0,10	0,0100
6,6	-6,94	48,164	6,69	-0,09	0,0081
5,8	-7,74	59,908	5,95	-0,15	0,0225
5,3	-8,24	67,898	5,43	-0,13	0,0169
4,9	-8,64	74,650	5,03	-0,13	0,0169
Σ	—	1085,9	135,59	-	0,5851

$$\bar{y} = \frac{135,4}{10} = 13,54$$

Подставив найденные значения в формулу, получим

$$R = \sqrt{1 - \frac{0,5851}{1085,968}} = \sqrt{0,9995} = 0,9997,$$

что говорит о полной связи между рассматриваемыми признаками.

Значение

$$m_r = \sqrt{\frac{1 - 0,9995}{10 - 2}} = 0,0079, a \quad t = \frac{0,9997}{0,0079} = 126,5$$

Число степеней свободы в данном примере равно 8

$$y = n - 2 = 10 - 2 = 8$$

Сравнивая, полученное значение $t = 126,5$ с нормативными ($t_c = 2,31; 3,36; 5,04$), видим, что корреляционное отношение достоверно при всех трех уровнях вероятности безошибочного суждения.

8.3. Экономические характеристики производственных функций.

Цель задания - изучить экономические характеристики производственных функций; научиться их рассчитывать по заданным значениям производственных функций, а также использовать экономические характеристики производственных функций для анализа использования земли, поиска оптимальных решений, подготовки исходной информации.

Экономические характеристики производственных функций и способы их определения. Основа анализа состояния и использования земель с помощью производственных функций — вычисление их экономических характеристик.

К экономическим характеристикам производственных функций относят дополнительные продукты факторов или предельные производительности; коэффициенты эластичности; средние производительности; предельные нормы заменяемости ресурсов (в случае 2 и более факторов).

Дополнительный продукт фактора (предельная производительность) — прирост продукции (y) за счет увеличения данного фактора на единицу при неизменной величине других факторов. Он определяется как первая частная производная производственной функции по этому фактору-аргументу

$$\frac{dy}{dx}$$

Если модель многофакторная, ее приводят к двухфакторной путем фиксации остальных факторов, например на средней величине.

Для парной линейной зависимости дополнительный продукт фактора равен коэффициенту регрессии

$$\frac{dy}{dx} = \frac{d(a_0 + a_1 x)}{dx} = d_1 \quad (8.27)$$

Коэффициент эластичности характеризует относительное изменение результата производства на единицу относительного изменения затрат i -го ресурса, то есть это величина, характеризующая соотношение темпов прироста продукции (y) и того или иного фактора (x_i). Приблизительно коэффициент эластичности показывает, на сколько процентов изменяется в среднем результат при изменении данного фактора на 1 % (при неизменной величине других факторов).

Коэффициенты эластичности могут быть вычислены следующим образом:

$$E_i = \frac{dy}{y} : \frac{dx}{x} = \frac{dy}{dx} : \frac{y}{x} = \frac{dy}{dx} \cdot \frac{x}{y}, \quad (8.28)$$

для парной линейной зависимости

$$E_i = \frac{dy}{dx} \cdot \frac{x}{y} = \frac{d(d_0 + a_1 x)}{dx} \cdot \frac{x}{y} = \frac{a_1 x}{a_0 + a_1 x}, \quad (8.29)$$

а при изменении x на 1 % y изменится на

$$\frac{a_1}{a_0 + a_1} \%$$

Средние производительности ($\frac{y}{x}$) показывают темпы изменения

результата производства от фактора производства. Для парной линейной зависимости

$$\frac{y}{x} = \frac{a_0 + a_1 x}{x} = a_1 + \frac{a_0}{x} \quad (8.30)$$

эти темпы изменяются по закону гиперболы.

Рассмотрим геометрическую интерпретацию линейного уравнения вида $y = a_0 + a_1x$ и его средней производительности,

где y - производственные затраты, a_0, a_1 - коэффициенты уравнения, x - объем производства.

В приведенном уравнении a_0 характеризует постоянные производственные затраты, не зависящие от объема производства, а a_1x — переменную часть затрат, зависящую от него.

Средняя производительность ресурса в данном случае превращается в себестоимость единицы производимой продукции

$$\frac{y}{x} = \frac{\text{общие производственные затраты}}{\text{объем выпускаемой продукции}} = \text{себестоимость единицы продукции,}$$

При положительных параметрах уравнения средняя производительность ресурса (в случае функции затрат - средняя себестоимость единицы продукции) убывает по закону гиперболы ($\frac{y}{x}$), что означает снижение средней себестоимости единицы продукции *при* росте производства за счет уменьшения доли постоянных расходов.

Характеристика основных производственных функций при парной зависимости их экономических параметров приводится в табл.8.5.

Таблица 8.5.

Основные экономические характеристики однофакторных производственных функций

Функция	Вид уравнения $Y=f(x)$	Дополнительный продукт, или предельная произво-	Средняя производительность $\frac{y}{x}$	Коэффициент $\frac{dy}{dx} \cdot \frac{x}{y}$
---------	---------------------------	---	---	--

Линейная	$y = a_0 + a_1 x$	a_1	$a_1 + \frac{a_0}{x}$	$\frac{a_1 x}{a_0 + a_1 x}$
Степенная	$y = a_0 x^{a_1}$			
Гиперболическая	$y = a_0 + \frac{a_1}{x}$	$-\frac{a_1}{x^2}$	$\frac{a_0}{x} + \frac{a_1}{x^2}$	
Параболическая	$y = a_0 + a_1 x + a_2 x^2$	$a_1 + 2a_2 x$	$\frac{a_0}{x} + a_1 + a_2 x$	$\frac{(a_1 + 2a_2 x)}{a_0 + a_1 x + a_2 x^2}$

Производственные функции позволяют найти различные сочетания факторов-аргументов, которые обеспечивают данный фиксированный выход продукции. Такие сочетания могут быть получены из уравнений изоквант производственной функции. Изоквантами называются линии, соединяющие точки, характеризующие различные сочетания значений факторов аргументов, при которых достигается заданное (фиксированное) значение результативного показателя. Для построения изоквант нужно решить уравнение производственной функции относительно одного из факторов, считая его зависимой переменной.

Для уравнения (4.31) при фиксированном значении $x_2 = 138$ руб./га (средняя фондообеспеченность в районе 138 руб.) уравнение изоквант примет вид

$$x_3 = \left(\frac{y}{28,3x_1^{-0,109}} \right) \frac{1}{0,0062} = 0,0062 y^{1,52} x_1^{0,166}$$

Подставляя сюда значения x_1 и фиксируя производство продукции y , получаем изокванту рассматриваемого процесса. Например, если $y = 100$ руб. уравнение изокванты будет следующего вида:

$$x_3 = 6,28x_1^{0,166}$$

Результаты производства зависят от ряда ресурсов, многие из которых взаимозаменяемы. Норма, по которой один фактор заменяет другой при сохранении выхода продукции на заданном уровне, называется предельной нормой заменяемости $\frac{dx_i}{dx_j}$.

Уравнение предельной нормы заменяемости может быть получено как производная одного фактора по другому - $\frac{dx_i}{dx_j}$.

Для рассматриваемой производственной функции уравнения предельных норм заменяемости имеют следующий вид:

$$\frac{dx_1}{dx_2} = \frac{x_1}{x_2} \cdot 5,10 = 5,10 \cdot \frac{x_1}{x_2},$$

$$\frac{dx_1}{dx_3} = \frac{x_1}{x_3} \cdot 5,10 = 5,10 \cdot \frac{x_1}{x_3},$$

$$\frac{dx_3}{dx_2} = \frac{-x_3}{x_2} \cdot 0,84 = -0,844 \cdot \frac{x_3}{x_2},$$

Предельные нормы заменяемости эродированности участка x_1 основными производственными фондами растениеводства x_2 приведены в табл. 8.6.

Таблица 8.6.

Предельные нормы заменяемости

Фондообеспеченность, руб/га	Эродированность пашни, %				
	5	10	20	30 ,	40
100	0,26	0,51	1,02	1,53	2,04
150	0,17	0,34	0,68	1,02	1,36
200	0,13	0,26	0,51	0,77	1,02

В данном случае, заменяемость эродированности основными производственными фондами растениеводства означает, что один и тот же выход продукции может быть получен в условиях более сильной эрозии за счет роста фондообеспеченности. При $x_1 = 20\%$ и $лсд = 100$ руб. предельная норма заменяемости на 1 га составит 1,02. Следовательно, *при* увеличении эрозии на 5,1 % для сохранения выхода продукции на прежнем уровне необходимо, чтобы фондообеспеченность возросла на 5 руб, на 1 га пашни. С увеличением фондообеспеченности предельная норма заменяемости падает; При $x_1 = 20\%$ и $x_2 = 200$ руб, увеличение эрозии на 5,1 % требует роста фондов в расчете 10 руб, на 1 га.

Уравнения предельной нормы заменяемости позволяют определить пути увеличения производства продукции — *изоклинали*. Они показывают, в каком направлении необходимо изменить сочетание факторов при постоянной предельной норме заменяемости для увеличения выхода продукции.

Уравнение изоклиналей можно получить из уравнений (8.32), приравнявая их к определенным предельным нормам заменяемости.

Изоклинали соединяют точки на изо квантах с равными предельными нормами заменяемости. Изокванты и изоклинали степенной функции по взаимозаменяемым факторам приведены на рис. 8.5, 8.6.

Построенные изокванты свидетельствуют о следующем: чтобы выход продукции растениеводства с 1 га при эродированности пашни 3 % был 100 руб. нужно затратить 7 чел.-дн. на 1 га. Для получения такого же объема продукции при 10 % эродированности требуется 10 чел.-дн., а при 20 % - 11,5 чел.-дн. (рис. 8.5).

Изоклинали на данных графиках показывают, как изменяются предельные нормы заменяемости при различных сочетаниях факторов.

В случае степенной функции коэффициент эластичности показывает, на сколько процентов изменяется зависимая переменная при изменении одного из факторов на 1 % при постоянной величине других.

Для исследуемой функции коэффициенты эластичности означают следующее: *при* увеличении эродированно ста участков x_1 на 1 % стоимость валовой продукции y снижается на 0,109 %, при росте фондообеспеченности на 1 % y увеличивается на 056%. Наибольшее влияние на изменение результативного показателя оказывают трудовые затраты x_3 . Их увеличение на 1 % сопровождается ростом объема валовой продукции на 0,66 %.

Использование экономических характеристик производственных функций, Производственные функции в землеустройстве позволяют анализировать важнейшие экономические зависимости. Цель анализа — подготовка исходных данных для принятия научно обоснованных проектных решений и планирования использования земель.

При этом применение производственных функций идет по следующим основным направлениям:

1) проектные решения, рекомендации и предложения основываются на результатах анализа производственных функций, хотя последние в плановых расчетах могут и не участвовать. В этом случае для планирования и проектирования важны выводы, получению которых способствовали производственные функции. Указанное используется, как правило, в полевом экспериментировании при проведении большого количества опытов, когда логическим путем получить окончательные выводы трудно (например, опыты с удобрениями, на эффективность использования которых влияет множество факторов: погодные условия, качество земель, реакция культур, норма внесения и др.);

2) с помощью производственных функций планируют (прогнозируют) значения результативных показателей. Условие таких расчетов - инерционность анализируемых систем, то есть предполагается, что в перспективе характер сложившихся экономических связей существенно не изменится;

3) производственные функции используют для нахождения экстремальных (оптимальных) значений результативных показателей. Например,

выбрав в качестве результативного показателя размер землепользования хозяйства, можно установить *при* каких сочетаниях факторов он будет оптимальным.

Рассмотрим порядок использования производственных функций для планирования (прогнозирования) результативных показателей и определения их оптимальных значений. Планирование (прогнозирование) осуществляется в следующем порядке: построение производственной функции и его оценка; установление перспективных параметров производственной функции; вычисление прогнозируемых значений переменной.

Ключевые слова

Производственная функция, фактора производства, одноресурсный, однофакторный, независимая переменная, зависимая переменная, многофакторный, мультипликативных, однородная функция, изоквант, маржинальный, средние значения, аддитивный, предельная норма замены, взаимодополняемость, взаимозаменяемость, производительность труда.

Вопросы для самопроверки

1. Как определяется (средняя) производительность труда и капиталовооруженность (фондовооруженность) труда? Какие возможны варианты взаимосвязи между ними в случае производственной функции Кобба-Дугласа?
2. Назовите основные свойства, которыми должна обладать производственная функция. Приведите примеры производственных функций, которые обладают всеми основными свойствами.
3. Что такое изокванта? В чем ее экономический смысл?
4. Как определяется (средняя) производительность капитала (капиталоотдача)?

5. Как определяется (предельная) производительность капитала и (предельная) производительность труда?
6. Какая существует связь между средней и предельной производительностью капитала (труда) в общем случае и в случае производственной функции Кобба-Дугласа?
7. Сформулируйте определение (частной) эластичности выпуска по i -му ресурсу (i -му фактору производства) ($i=1,2$) и определение эластичности производства.
8. Дайте содержательную интерпретацию (частной) эластичности выпуска по i -му ресурсу.
9. Сформулируйте определение (предельной) нормы замены одного ресурса другим. Дайте содержательную интерпретацию этому понятию.
10. Как меняется (предельная) норма замены одного ресурса другим при движении по изокванте? Дайте содержательную интерпретацию характеру изменения предельной нормы замены.

Глоссарий

Адаптер — устройство для согласования параметров входных и выходных сигналов других устройств с целью их сопряжения.

Алгоритм —пошаговое описание решения задачи, ведущее к получению верного однозначного результата, выполненное на одном из алгоритмических языков.

Алгоритмизация — составление алгоритмов для решения поставленных задач.

Алгоритмические ошибки — ошибки в методе, постановке, сценарии и реализации.

Алгоритмический язык — язык описания алгоритмов.

Антивирусная программа — программа для защиты компьютера от компьютерных вирусов.

Аппаратура — технические средства, физическое оборудование — механические, магнитные, электрические, электронные устройства.

Архив — хранилище данных во внешней памяти.

Архивирование — процесс сжатия файлов с целью хранения их в более компактном виде.

Архитектура ЭВМ — совокупность общих принципов организации аппаратно—программных средств и их характеристик, определяющая функциональные возможности ЭВМ при решении соответствующих классов задач.

Атрибуты файла — дополнительные параметры, определяющие свойства файла.

База данных — хранилище интегрированных и коллективно используемых данных, организованное с целью обеспечить независимость структур хранимых данных от обрабатывающих программ, оптимизировать использование памяти и время доступа.

База знаний — совокупность моделей данных, выраженная в понятиях определенной проблемной области, организованная таким образом, чтобы обеспечить независимость языка общения с нею от специфики СУБД.

Брандмауэр (firewall) — программа, которая образует границу между сетями, перекрывая свободный доступ из Интернета в локальную сеть.

Браузер — компьютерная программа, позволяющая находить и просматривать гипертекстовые документы, опубликованные в Сети и на компьютере.

Буфер обмена — область памяти, в которую временно помещается вырезанный или скопированный фрагмент документа или графическое изображение при выполнении команд Копировать (Copy) или Вырезать (Cut).

Винчестер (HDD) — несъемный жёсткий магнитный диск (пакет дисков).

Виртуальная реальность — искусственный мир, созданный программистами.

Вирус — программа или фрагмент программ, причиняющий вред компьютеру и данным.

Внешняя память (ВЗУ) — предназначена для размещения больших объемов информации и обмена ею с оперативной памятью.

Входные данные — данные, вводимые в ЭВМ.

Вычислительная машина — устройство для организации автоматических вычислений и обработки информации.

Вычислительная сеть — совокупность ЭВМ, объединенных сетью связи, позволяющей компьютерам проводить обмен информацией. Вычислительная система — объединение вычислительных машин для организации хранения, обработки и поиска информации.

Вычислительная техника — совокупность устройств, предназначенных для автоматической или автоматизированной обработки данных.

Гиперссылка — текст или графическое изображение на сайте или в письме электронной почты, указывающий на другой файл, который может быть расположен в Интернете, и позволяющий перейти к этому файлу.

Гипертекст (hypertext) —система просмотра текстовой, графической и другой информации на экране монитора при помощи гиперссылок, которые связывают друг с другом страницы гипертекстового документа.

Графический редактор — программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений.

Данные — информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами.

Двоичное кодирование — система представления данных последовательностью двух знаков: 1 и 0.

Диаграмма — графическое представление данных, используемая для анализа и сравнения данных, представления их в наглядном виде.

Дисплей — устройство отображения информации на электронном экране.

Дистанционное обучение — обучение на расстоянии с использованием учебников, персональных компьютеров и сетей ЭВМ.

Драйвер — программа, управляющая каким—либо модулем компьютера или периферийным устройством (мышью, принтером, памятью и т. д.).

Инструментальное ПО — предназначено для использования в ходе проектирования, разработки и сопровождения программ.

Интернет — международная компьютерная телекоммуникационная сеть.

Интерпретатор — система интерпретации (выполнения) программ на ЭВМ.

Информационная система — организация совокупности документов с использованием средств вычислительной техники и телекоммуникационной связи.

Информационная технология — это совокупность методов, производственных процессов и программно—технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, обработку, хранение, распространение и отображение информации с целью снижения трудоемкости процессов использования информационного ресурса, а также повышения их надежности и оперативности.

Информация – сведения об окружающем мире, которые повышают уровень осведомленности человека.

Искусственный интеллект — компьютерные системы, моделирующие или воспроизводящие интеллектуальную деятельность.

Каталог (папка) – определенное место на диске (в области данных диска), где содержится информация о файлах и подкаталогах, привязанных к данному каталогу.

Клавиша — элемент ручного управления, срабатывающий от нажатия.

Ключ — данные, определяющие возможность доступа к другим данным.

Код — набор знаков в совокупности со схемой кодирования для представления информации в виде данных.

Кодирование данных — выражение данных одного типа через данные другого типа.

Количеством информации - числовая характеристика сигнала, отражающую ту степень неопределенности (неполноту знаний), которая исчезает после получения сообщения в виде данного сигнала.

Компилятор — система трансляции программ с одного языка на другой.

Компьютер — электронный прибор, предназначенный для автоматизации создания, хранения, обработки и транспортировки данных.

Компьютерная сеть — совокупность компьютеров и других устройств, объединяемых вместе с помощью сетевых кабелей таким образом, что они могут взаимодействовать друг с другом с целью совместного использования информации и ресурсов.

Компьютерный вирус — программа, способная размножаться самостоятельно, дописывая свой код к другим файлам или в служебные области диска.

Консоль — средство для контроля за работой вычислительной системы и для управления ею вручную.

Контроллер — устройство управления внешними устройствами.

Концентратор (Hub хаб) — устройство повторяют всю информацию,

которую они получают, то есть все устройства, подключенные к портам концентратора, получают одну и ту же информацию.

Курсор — значок для указания места на экране ЭВМ.

Легенда — область, позволяющая сопоставить каждому из рядов данных на диаграмме цвет и стиль оформления.

Логические функции — функции, принимающие логические значения «истина» или «ложь».

Логические элементы — устройства, на входах и выходах, на которые могут появляться сигналы 0 или 1.

Магнитные диски — средство хранения информации, программ и данных в ЭВМ.

Маршрутизатор (router) — специальное устройство для соединения различных локальных сетей.

Мастер диаграмм — Средство для создания в электронных таблицах диаграмм данных.

Межстрочный интервал — суммарная высота каждой строки текста в абзаце.

Микропроцессор — программируемое устройство обработки данных, выполненное на основе одной или нескольких больших интегральных схем.

Моделирование — выявление свойств каких—либо объектов, систем объектов или процессов путем построения и исследования их моделей.

Модем (модулятор—демодулятор) — устройство преобразования данных к виду, принятому для определенного типа линий связи и обратно.

Модуль (библиотека) — четко определенная функциональными свойствами часть какой—либо хорошо структурированной системы.

Носитель — материал либо устройство, на который могут быть занесены данные.

Оперативная память — служит для приема, хранения и выдачи информации.

Операционная система — совокупность программных средств, обеспечивающая управление аппаратным и программным обеспечением.

Отладка программ — процесс поиска и исправления ошибок в программах на ЭВМ.

Персональный компьютер — это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

Пиксель (pixel) — минимальная единица для отображения на экране монитора.

Подпрограмма — программа решения некоторой задачи, синтаксически оформленная по правилам какой—либо определенной системы программирования так, что она может быть использована в качестве конструктивного элемента при решении более общей задачи на компьютере.

Поиск данных — отбор данных по определенной комбинации признаков.

Пользователь — лицо или организация, применяющие средства обработки данных для решения прикладных задач.

Пользовательский интерфейс — методы и средства взаимодействия человека с аппаратными и программными средствами.

Постоянная память — обеспечивает хранение и выдачу информации.

Прикладное программное обеспечение — комплекс программ, с помощью которых на данном рабочем месте выполняются конкретные задания.

Приоритет — ранг задачи, характеризующий ее преимущественное право на получение ресурсов системы по отношению к другим задачам.

Программа — синтаксически оформленная по правилам какого—либо языка программирования последовательность предложений, имеющая в том или ином смысле самостоятельное значение.

Программирование — теоретическая и практическая деятельность, связанная с созданием программ.

Программное обеспечение (software) — совокупность программ для определенной заранее обработки данных и необходимых для их эксплуатации документов.

Протокол — совокупность правил, описывающих то, как программы и компьютеры должны действовать, когда они взаимодействуют друг с другом.

Процессор, или микропроцессор — основное устройство ЭВМ, предназначен для выполнения вычислений по хранящейся в запоминающем устройстве программе и обеспечения общего управления ЭВМ.

Пункт — единица измерения размера шрифта (в том числе в компьютерных системах, например, в MS Word).

Рабочая станция — специализированная высокопроизводительная ЭВМ, ориентированная на профессиональную деятельность в определённой области (обычно САПР, графика), имеющая дополнительное оборудование и специализированное программное обеспечение.

Разрешение — показатель качества графического изображения. Измеряется в точках на дюйм — dpi (dots per inch).

Количество dpi — показатель разрешения, с которым печатает принтер или вводит изображение сканер.

Реляционная база — база данных, в которой информация хранится в форме таблиц.

Ряд данных на диаграмме — набор связанных между собой элементов данных, соответствующий одному столбцу или одной строке таблицы данных.

Сектор — минимальное логическое пространство на жёстком диске. Сервер — ЭВМ, предоставляющая свои ресурсы другим пользователям.

Сигнал — (от латинского signum — знак) представляет собой любой процесс, несущий информацию.

Система глобального поиска — программа, позволяющая искать и находить в интернет материалы по той или иной теме.

Система счисления - согласованный набор приемов, правил для представления числовых данных.

Системное программное обеспечение — это комплекс программ, которые обеспечивают эффективное управление компонентами вычислительной системы.

Системный интерфейс — это конструктивная часть ЭВМ, предназначенная для взаимодействия ее устройств и обмена информацией между ними.

Сканер — устройство ввода, служащий для ввода в компьютер разнообразных изображений.

Служебное программное обеспечение (утилиты) — обеспечивает автоматизацию проверки, наладки и настройки компьютерной системы.

Спам (англ. spam) — Нежелательная корреспонденция рекламного или иного характера, массово рассылаемая людям, не выразившим желание её получать.

СУБД — система управления базами данных, необходимая для определения, создания и использования баз данных и обеспечивающая целостность баз данных, восстановление, проверку правильности и секретности данных.

Терминал — устройство, подключенное к более мощной ЭВМ, не предназначенное для работы в автономном режиме и обеспечивающее ввод—вывод информации и команд пользователя.

Файл — логически связанная совокупность данных или программ, для размещения которой во внешней физической памяти выделяется именованная область.

Файловая система — Часть операционной системы, управляющая размещением и доступом к файлам и каталогам на диске.

Флоппи—диск (дискета) – съемный гибкий магнитный диск.

Флэш—память — это энергонезависимая полупроводниковая перезаписываемая память с произвольным доступом (Random Access Memory, RAM).

Функциональная клавиша — клавиша из группы клавиш с программируемыми функциями.

Шаблон документа — специальный вид документа, например, MS Word, предоставляющий ряд готовых средств для формирования окончательного документа.

Шина — физический канал передачи электрических сигналов в компьютере для связи между устройствами.

Электронная почта (E-mail) — интернет—служба, позволяющая пользователям отправлять сообщения с одного компьютера на другой.

Электронная таблица — электронный документ подобный обычной таблице, в ячейках которой записаны различные данные и формулы для вычислений.

Язык программирования — система обозначений для описания программ (алгоритмов и структур данных).

Ячейка — независимо адресуемая последовательность смежных позиций в оперативной памяти вычислительной машины, содержимое которых выбирается за одно обращение.

Адекватность модели — требование к модели, состоящее в её способности воспроизводить свойства, состояние и поведение исследуемого объекта с достаточной для поставленных целей точностью и в достаточно широком диапазоне изменения её состояния и состояния её среды.

Аналитические модели — математические модели, разрабатываемые для исследования структуры моделируемой системы.

Баланс — в экономико-математическом моделировании — уравнение или неравенство, устанавливающее соответствие между источниками ресурса и направлениями его использования.

Вычислительный эксперимент—метод исследования явления, процесса или машины, для которых разработана компьютерная модель.

Двойственная оценка ограничения — величина, характеризующая прирост значения целевой функции задачи математического программирования при

малом изменении величины свободного члена данного ограничения; частная производная оптимального значения целевой функции, рассматриваемого в качестве функции свободных членов ограничений задачи математического программирования, по величине свободного члена данного ограничения.

Динамическая модель – математическая модель, описывающая развитие процесса во времени.

Имитационная модель – математическая модель, воспроизводящая поведение исследуемого объекта и применяемая для постановки компьютерных экспериментов, выявляющих особенности функционирования объекта при различных внешних условиях и управляющих воздействиях.

Компьютерная имитация–реализованная на ЭВМ математическая модель, оперирующая системой динамических уравнений.

Компьютерная модель – математическая модель, оперирующая численными алгоритмами и реализованная на ЭВМ.

Линейное программирование – раздел математического программирования, исследующий задачи отыскания экстремума линейной функции на множестве допустимых значений переменных, заданном системой линейных уравнений и (или) неравенств;

Макроэкономическая модель – экономико-математическая модель, в которой не выделяются переменные, описывающие отдельных хозяйствующих субъектов (предприятия, отрасли), составляющих моделируемую хозяйственную систему, и которая отражает только связи, присущие этой системе как целому.

Математическая модель – совокупность математических зависимостей, гомоморфная исследуемой системе

Математическое моделирование – метод исследования реальных объектов при помощи постановки экспериментов на их математических моделях.

Математическое программирование – раздел математики, исследующий методы решения задач отыскания экстремума на заданном множестве допустимых значений переменных.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Абдуллаев З.С., Мирзаев С.С., Шодмонова Г., Шамсиддинов Н.Б. Информатика ва ахборот технологиялари. Дарслик, Т.2012.
2. Конова Е. А., Поллак Г. А. Алгоритмы и программы. Язык С++: Учебное пособие.— 2-е изд., стер. — СПб.: Издательство «Лань», 2017. — 384 с.: ил.— (Учебники для вузов. Специальная литература).
3. Компьютерные технологии обработки информации. Под ред. С.В.Назарова. М., Финансы и статистика. 2011.
4. Голицина О.Л., Максимов Н.В., Попов И.И. Информационные системы. Москва. Форум. 2014.
5. Незнанов А. А. Программирование и алгоритмизация: учебник. М. : Академия, 2010. — 303 с.
6. Павловская Т. А. С++. Программирование на языке высокого уровня : учебник. СПб. : Питер, 2013. 238 с.
7. Тимофеева Т. С. Задачи по программированию: учебное пособие / Т. С. Тимофеева О. А. Тихонова, И. Г. Ларионова; под ред. В. И. Васильева. М. : Вузовская книга, 2011. 94 с.
8. Франка, П. С++. СПб. : Питер, 2012. 491 с.
9. Замков О.О. и др. Математические методы в экономике. М. Издательство ДИС. 1998.
10. Кузнецов Ю.Н. и др. Математическое программирование. М. Высшая школа 1980.
11. Беркинов Б.Б. Эконометрика. Ташкент. «Фан ва технология». 2015.
12. Шадманова Г., Мирзаев С.С. Экономико-математические методы и модели. Ташкент. ТИМИ. 2011.
13. 2006 - 2010 ООО «ЭКОцентр» 5 УПРЗА «ЭКО центр». Руководство пользователя.
14. 2006 - 2010 ООО «ЭКОцентр» 5 УПРЗА «ЭКО центр». Быстрый старт. Руководство пользователя.

Оглавление

ВВЕДЕНИЕ	3
ГЛАВА I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	5
Лекция 1. ПОНЯТИЕ ИНФОРМАЦИИ. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ. ГЕОИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.....	5
Лекция 2. ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	46
Лекция 3. ТЕКСТОВЫЙ ПРОЦЕССОР MS WORD.....	93
Лекция 4. ТАБЛИЧНЫЙ ПРОЦЕССОР MS EXCEL	112
Лекция 5. БАЗА ДАННЫХ. СИСТЕМЫ УПРАВЛЕНИЯ БАЗЫ ДАННЫХ (СУБД).....	125
Лекция 6. УНИФИЦИРОВАННАЯ ПРОГРАММА РАСЧЕТА ЗАГРЯЗНЕНИЯ АТМОСФЕРЫ «ЭКО ЦЕНТР». ПРИНЦИП РАБОТЫ. 147	
ГЛАВА II. ОСНОВЫ АЛГОРИТМИЗАЦИИ. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++	163
Лекция 1. АЛГОРИТМЫ. СВОЙСТВА И ВИДЫ АЛГОРИТМОВ	163
Лекция 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ СТРУКТУРЫ. 180	
Лекция 3. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ И ЦИКЛИЧЕСКИХ АЛГОРИТМОВ	210
Лекция 4. МАССИВЫ. ПРОГРАММИРОВАНИЕ ОДНОМЕРНЫХ МАССИВОВ	225
Лекция 5. ПРОГРАММИРОВАНИЕ ДВУМЕРНЫХ МАССИВОВ.....	232
ГЛАВА III. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ.....	241
Лекция 1. МОДЕЛИРОВАНИЕ И СОВРЕМЕННЫЕ МЕТОДЫ ВЫЧИСЛЕНИЙ В ЭКОЛОГИЧЕСКИХ ПРОЦЕССАХ	241
Лекция 2. МЕТОДЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ	269
Лекция 3. ГРАФИЧЕСКИЙ МЕТОД РЕШЕНИЯ ДВУМЕРНЫХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	279
Лекция 4. СИМПЛЕКСНЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	286
Лекция 5. РАСПРЕДЕЛИТЕЛЬНЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	295
Лекция 6. СТАДИИ МОДЕЛИРОВАНИЯ И ПОСТРОЕНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ	313

Лекция 7. ЭКОНОМИКО-МАТЕМАТИЧЕСКИЙ АНАЛИЗ НА ОСНОВЕ ДВОЙСТВЕННЫХ ОЦЕНОК И КОЭФФИЦИЕНТОВ ЗАМЕЩЕНИЯ...	320
Лекция 8. ПРОИЗВОДСТВЕННЫЕ ФУНКЦИИ И ИХ ЭКОНОМИЧЕСКИЕ ХАРАКТЕРИСТИКИ	334
ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА	353

MUNDARIJA

KIRISH

I BOB. AXBOROT TEXNOLOGIJALARINING NAZARIY ASOSLARI . AXBOROT TEXNOLOGIJALARINING DASTURIY VA TEXNIK TA'MINOTI

1-Ma'ruza. Axborot tushunchasi. Axborot texnologijalari va tizimlari.

2- Ma'ruza. Axborot texnologijalarining dasturiy va texnik ta'minoti

3- Ma'ruza. MS WORD matn muxaririda xujatlar yaratish

4- Ma'ruza. MS EXCEL jadval protsessorida ishlash

5- Ma'ruza. MS ACCESS ma'lumotlar bazasini boshqarish tizimi

6- Ma'ruza. Atmosferani ifloslanishini hisoblash unifikatsiyalangan
«EKO Sentr» dasturi. Ishlash tamoyillari.

II BOB. ALGORITMLASH ASOSLARI. C++ TILIDA DASTURLASH

7- Ma'ruza. Algoritmlar. Algoritmlarni xossalari va turlari

8- Ma'ruza. C++ tilida dastur strukturasi

9- Ma'ruza. Tarmoqlanuvchi va takrorlanuvchi algoritmlarni dasturlash.

10- Ma'ruza. Massivlar. Bir va ikki o'lchamli massivlarni dasturlash.

11- Ma'ruza. Qism dasturlar va protseduralar. Grafik imkoniyatlari.

III BOB. JARAYONLARNI MATEMATIK MODELASHTIRISH

12- Ma'ruza. Ekologik jarayonlarni modellashtirish va hisoblash usullari.

13- Ma'ruza. Chiziqli dasturlashning ikki o'lchamli masalani grafik
usuli

14- Ma'ruza. Chiziqli dasturlash masalasini simpleks usulida echish

15- Ma'ruza. Chiziqli dasturlash masalasini taqsimot usulida echish

16- Ma'ruza. Modellashtirish bosqichlari va ekologik modellarni
matematik modelini tuzish

17- Ma'ruza. Ikkilangan baho asosida iqtisodiy-matematik tahlil qilish va
koeffitsientlarni almashtirish

18- Ma'ruza. Ishlab chiqarish funksiyalari va ularning xarakteristikalarini

FOYDALANILGAN ADABIYOTLAR RO'YHATI

TABLE OF CONTENTS

INTRODUCTION

CHAPTER I. THEORETICAL BASICS OF INFORMATION TECHNOLOGIES. SOFTWARE AND HARDWARE OF INFORMATION TECHNOLOGIES

Lecture 1. Concept of information. Information technologies and systems.

Geo-information technologies

Lecture 2. Software and hardware of information technologies

Lecture 3. Creating documents in MS Word text editor

Lecture 4. Working in the table processor MS Excel

Lecture 5. MS Access database management system

Lecture 6. Unified program for the calculation of atmosphere pollution

“EcoTSentr”. Work principles

CHAPTER II. BASICS OF ALGORITHMIZATION. PROGRAMMING IN C++ LANGUAGE

Lecture 7. Algorithms. Properties and types of algorithms

Lecture 8. Program structure in C++ language

Lecture 9. Programming algorithms of ramified and cycling structures

Lecture 10. Massives. Programming one-dimensional and two-dimensional massives

Lecture 11. Functions and procedures. Graphic opportunities

CHAPTER III. MATHEMATICAL MODELING OF PROCESSES

Lecture 12. Modeling and modern methods of calculations in ecological processes. Mathematical programming methods.

Lecture 13. Graphic method for solving two-dimensional problems of linear programming

Lecture 14. Simplex method for solving problems of linear programming

Lecture 15. Distribution method for solving the problem of linear programming

Lecture 16. Stages of modeling and scheme of the mathematical model in ecological processes

Lecture 17. Economico-mathematical analysis based on dual assessments and determination factors

Lecture 18. Production functions and their specialized characteristics

LIST OF USED LITERATURE