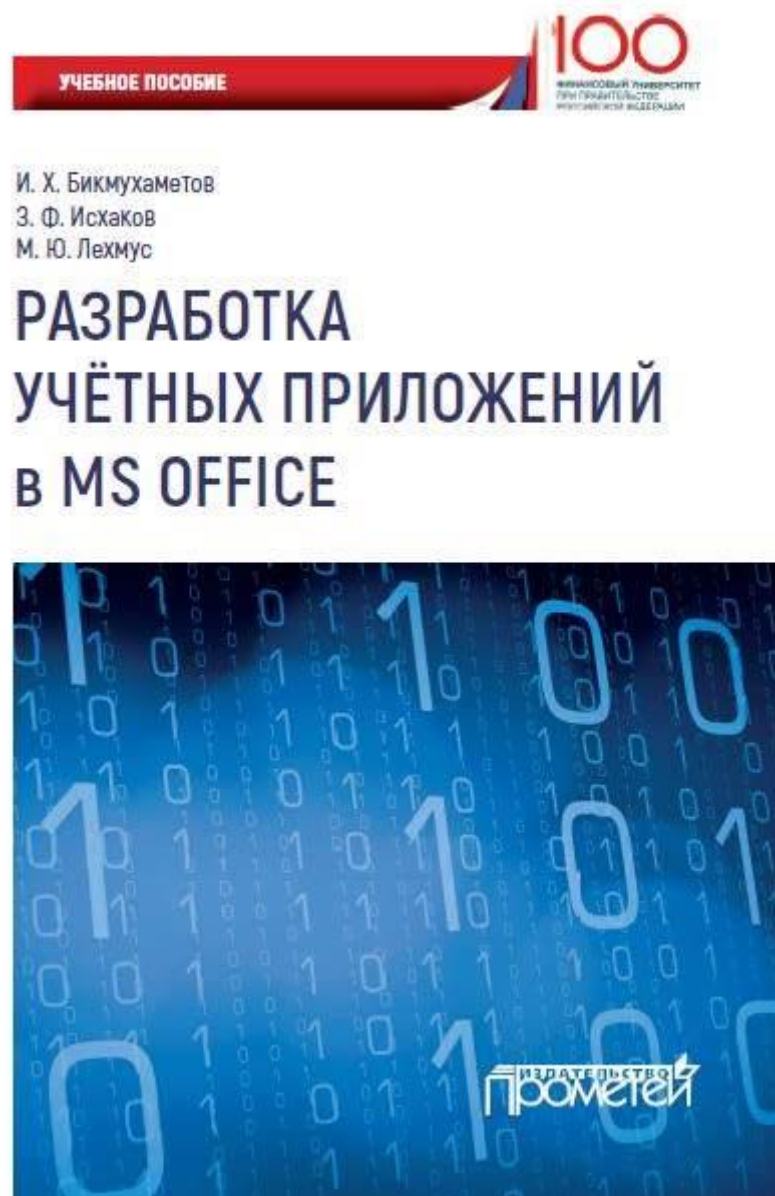


**Михаил Юрьевич Лехмус Ильдар Хайдарович Бикмухаметов Зуфар Фаргатович
Исхаков
Разработка учетных приложений в MS Office**



«Разработка учетных приложений в MS Office»: Прометей, Москва; 2018
ISBN 978-5-907003-16-3

Аннотация

Рассматриваются практические вопросы создания учетных приложений, подходы и инструменты разработки приложений, приводятся готовые фрагменты программного кода для автоматизации отдельных процессов.

Издание предназначено для студентов, обучающихся по направлению 38. 03. 01 «Экономика», при изучении дисциплины «Разработка учетных приложений в среде MS Office», а также может быть рекомендовано всем желающим получить практические навыки работы в офисном пакете с элементами автоматизации своих приложений с использованием макросов и VBA.

И. Х. Бикмухаметов, З. Ф. Исхаков, М. Ю. Лехмус

Разработка учетных приложений в MS Office

© Бикмухаметов И. Х., Исхаков З. Ф., Лехмус М. Ю., 2018

© Издательство «Прометей», 2018

* * *

Введение

Деятельность предприятий и организаций различного профиля на современном этапе развития общества невозможно представить без широкого использования информационных технологий. Применение информационных технологий дает предприятиям существенные конкурентные преимущества в таких вопросах, как оперативный учет информации, подготовка отчетов и учетной документации, анализ рыночной среды, обмен информацией с партнерами и контрольными органами и т. д. В настоящее время компьютерные технологии внедряются во все этапы основной деятельности: от планирования и организации деятельности, текущего учета, отчетности на различных календарных этапах до управления основным производством и становятся одной из важнейших составляющих успешной работы предприятий.

Для реализации различных запросов заинтересованных структур по вопросам автоматизации своей деятельности, многими IT-компаниями и IT-сообществами разработаны и активно предлагаются на рынке предметные решения. Это, прежде всего, комплексные информационные системы проблемно-ориентированного профиля.

Эффективность внедрения таких систем всегда носит спорный характер и зависит от множества факторов: высокой начальной стоимости, правильности поставленного технического задания на внедрение информационной системы, наличия IT-структуры организации для автоматизации различных процессов, готовности сотрудников к использованию соответствующего аппаратного и программного обеспечения и т. д.

Но вместе с тем, остаются некоторые задачи разработки и применения информационных технологий, которые можно эффективно решить без больших финансовых и временных затрат. Это, прежде всего, вопросы разработки учетных приложений для текущих оперативных задач организации.

В данном учебном пособии рассматриваются практические вопросы создания учетных приложений. Рассматриваются полезные, с точки зрения авторов, подходы и инструменты разработки приложений, приводятся готовые фрагменты программного кода для автоматизации отдельных процессов. Это обусловило наличие в пособии большого количества скриншотов процесса создания учетных приложений в различных офисных программах.

Учебное пособие может быть использовано при изучении дисциплины «Разработка учетных приложений в среде MS Office» и рекомендовано всем желающим получить практические навыки работы в офисном пакете с элементами автоматизации своих приложений с использованием макросов и VBA.

1. Учетные приложения и автоматизация хозяйственной деятельности

Учетные приложения очень широко используются в различных структурных подразделениях предприятий и организаций. На любом компьютере можно найти с десяток электронных таблиц с массивом данных на различную тематику. Обычно пользователи создают такие таблицы, чтобы вести учет в какой-либо предметной области. При этом всегда есть возможность быстро провести над этими данными различные процедуры обработки или использовать часть из них для различных расчетов и построения различных диаграмм.

Под «приложением» принято называть программный продукт, разработанный для выполнения конкретных задач в выбранной программной среде, содержащий конкретный алгоритм реализации и программный код. Часто можно услышать Windows-приложение, DOS-приложение, офисное приложение и т. д. Обыкновенная электронная таблица еще не является приложением. Необходимо **действие**, которое должно выполнять приложение, характеризующее приложение как приложение – иначе это просто документ. Другое дело, когда электронная таблица содержит в своих ячейках формулы или функции листа. В этом случае можно говорить об активной электронной таблице, которая производит различные вычисления, да к тому же автоматически корректирует красочную диаграмму на листе в зависимости от изменившихся исходных данных в ячейках таблицы. А если еще с данными электронной таблицей работают с помощью кода VBA и используют элементы управления, тогда свойства электронной таблицы приближаются к исполняемому файлу приложения.

Приведем отличия учетного приложения от простой электронной таблицы:

1. Учет – это динамический процесс, поэтому таблица постоянно меняет свой размер в зависимости от занесенных в нее

данных.

2. Если необходимым условием для приложения является проведение каких-либо действий с данными таблицы, то для электронной таблицы это только возможное действие.

3. Данных действий должно быть более одного, т. е. отличительное свойство приложения это различная, контекстная работа с набором данных. Пользователь сам выбирает, какое действие необходимо выполнить с табличным набором данных в данный момент времени.

4. Наличие программного кода для автоматизации интерфейса программного приложения, работы с переменным табличным набором данных и автоматизации взаимодействия с другими приложениями является необходимым условием для приложения.

Основные теоретические вопросы разработки учетных, а точнее учетно-расчетных приложений рассмотрены в учебном пособии Лукьянова П. Б. [1]. Каждое учетно-расчетное приложение имеет большую расчетную составляющую в виде различных блоков анализа данных в базе, прогнозирования, и возможно элементов какого-то подхода к оптимизации. Его уже нельзя назвать просто учетным приложением. Это скорее расчетное приложение с таблицей исходных данных. Расчетный блок учетного приложения, прежде всего, зависит от предметной области использования данного приложения: экономика, менеджмент, образование, наука и др. Следовательно, у каждого учетного приложения он будет свой.

Поэтому граница перехода между учетным, учетно-расчетным и расчетным приложением достаточно расплывчата. Элементы расчета есть в любой базе данных в зависимости от ее назначения. В то же время расчетное приложение не может работать без массива данных: в виде таблицы данных или же в виде файла с исходными данными.

Для реализации учетного приложения «родной», инструментальной средой разработки, безусловно, является табличный процессор, хотя учетные и расчетные функции можно реализовать и в других офисных приложениях. При этом продукт, созданный в среде СУБД, правильно называть базой данных, хотя его также можно назвать и учетным приложением.

При разработке учетного приложения может возникнуть вопрос: какую среду использовать – среду электронных таблиц или базы данных, а может даже документа MS Word или комбинацию приложений? Все зависит от решаемой задачи и от объема исходных данных, с которыми предстоит работать. При этом не следует исключать из рассмотрения навыки пользователя по владению каким-либо офисным приложением. Опыт общения с различными структурами организации показывает, что выбор инструментария определяет, прежде всего, знания исходного приложения – инструментария. И часто это приложение бывает не лучшим для поставленной задачи. Поэтому можно сделать вывод, что для учетных приложений, когда объем информации небольшой (не более 200 записей), ограничены время и средства на разработку – лучше всего подходит среда электронных таблиц. При больших значениях перечисленных показателей следует обратить на базы данных.

2. Базовые составные части учетного приложения

Независимо от выбранной среды работы учетные приложения могут иметь явно выраженные составные части:

- 1) Интерфейсная часть для работы с приложением в целом.
- 2) Интерфейсная, диалоговая форма (часть) для сбора исходных данных. Необязательная, но желательная часть учетного приложения. Она позволяет безошибочно, легко, наглядно вводить данные для последующего переноса их в учетную таблицу-реестр и в бланк формуляра – образца.
- 3) Справочник исходных данных для использования в диалоговой форме. Представляется в виде дополнительной электронной таблицы (таблиц) с данными, которые должны быть введены в часть ячеек таблицы – регистра. Это название принято потому, что бухгалтеры используют их, например, в программном комплексе 1С, в его различных конфигурациях.
- 4) База (регистр) для хранения введенных данных в диалоговую форму.
- 5) Бланк формуляра-образца, шаблон текстового документа или другой стандартный, заранее подготовленный шаблон документа, в которые должны быть занесены данные в нужных местах. Обычно для поиска таких документов в сети Интернет в поисковую систему вбивают «Бланки документов» или обращаются к справочно-правовым системам, например, «КонсультантПлюс» и «Гарант».
- 6) Справочная (инструкция) система для работы с приложением.

Бланк формуляра-образца – это то, с чего начинается построение учетного приложения. Таблицы для хранения информации должны как минимум иметь такие же поля, как в бланке формуляра-образца. В качестве примера бланка формуляра-образца можно привести форму счета-фактуры.

3. Особенности использования MS Office для разработки учетных приложений

Использование программного комплекса MS Office для разработки учетных приложений имеет большую привлекательность и возможности. Это, прежде всего, связано с тем, что данный пакет имеется практически на всех персональных компьютерах организаций и граждан России.

Особенность использования MS Office для построения учетных приложений заключается в том, что большинство приложений, о которых мы расскажем далее, имеют возможность табличного хранения и отображения данных. А это одно из основных условий для создания учетных приложений. Кроме того, в этих таблицах можно не только хранить данные, но и производить вычисления и анализ данных, даже в таблицах MS Word. Важным следует считать также возможность строить на основе табличных данных различные диаграммы, что позволяет наглядно отображать информацию.

В составе программного комплекса MS Office присутствуют программные приложения, специально ориентированные для работы с массивами данных в табличной форме. Это табличный процессор MS Excel и система управления базой данных (СУБД) MS Access. Данные приложения обладают достаточным функционалом для создания учетных приложений любой сложности. Более того, эти приложения специально созданы для учета и обработки больших массивов данных, что характерно для учетных приложений.

Другой важной чертой пакета MS Office является организация удобного обмена данных между объектами приложения и возможность управления ими средствами VBA.

Положительной особенностью продуктов компании Microsoft является преемственный интерфейс у различных приложений. Пользователь, знакомый с интерфейсом одного приложения, без особых затруднений освоит работу в другом. Разумеется, при этом приложения имеют различия интерфейса в зависимости от своего функционального назначения.

Небольшие трудности возникли у поклонников продуктов фирмы Microsoft при появлении очередной версии пакета MS Office 2007. Внешний вид органов управления приложением и документами существенно изменились по сравнению с предыдущими версиями пакета. Измененный интерфейс программ пакета первоначально вызывает определенные трудности у пользователя. Это связано с привыканием пользователей к традиционному интерфейсу приложений, которое сформировалось в результате их многолетнего использования. Выход в данной ситуации только один: привыкнуть к новому веянию разработчика пакета.

4. Использование средств контроля данных при создании учетных приложений

Для разработки удобного в пользовании учетного приложения следует использовать различные элементы автоматизации процессов. Автоматизация, прежде всего, необходима для безошибочного ввода и переноса данных. Часто для этого используют контроль введенных данных. При работе в среде MS Excel имеются инструменты для выполнения данных функций (рис. 1, рис. 2, рис. 3):

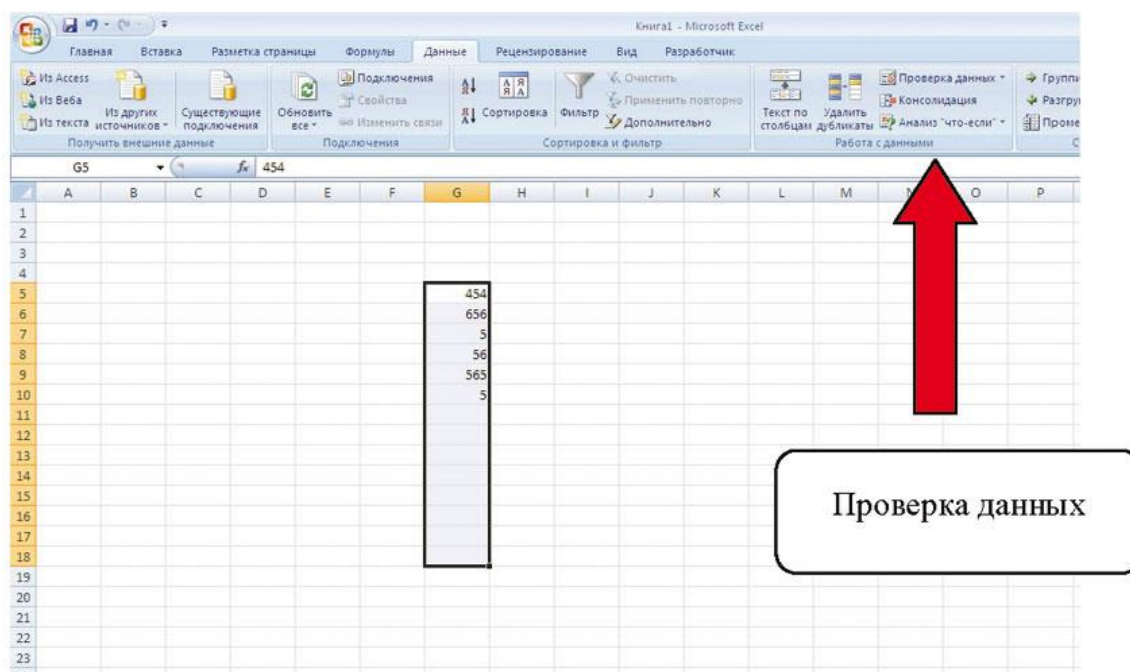


Рис. 1. Позиция меню для контроля ввода данных

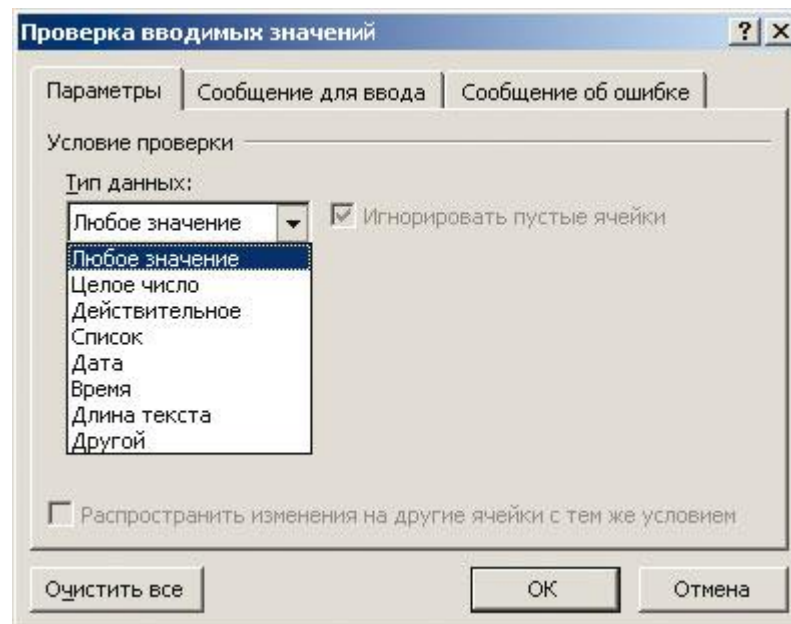


Рис. 2. Окно диалога для настройки параметров контроля введенных данных

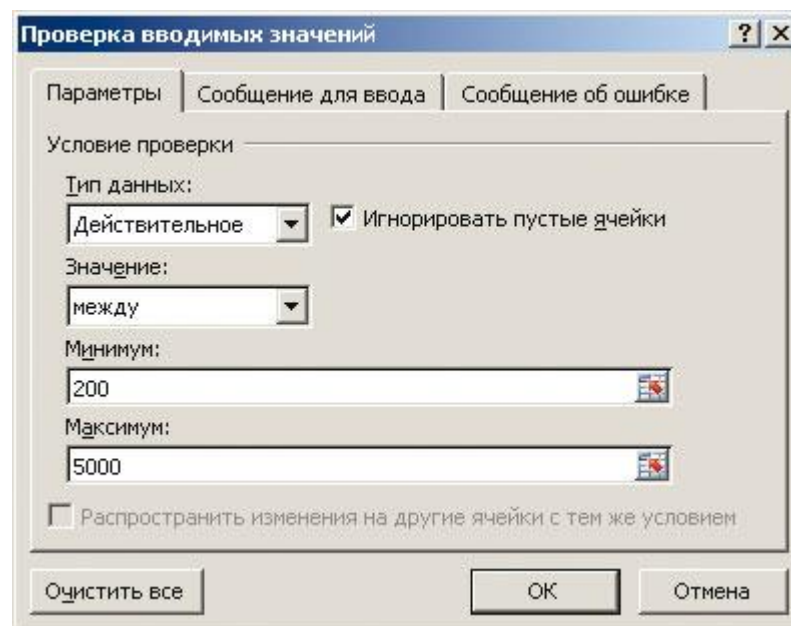


Рис. 3. Параметры для ввода действительных чисел в заданном интервале

При вводе данных, выходящих за пределы допустимых значений, выводится сообщение (рис. 4):

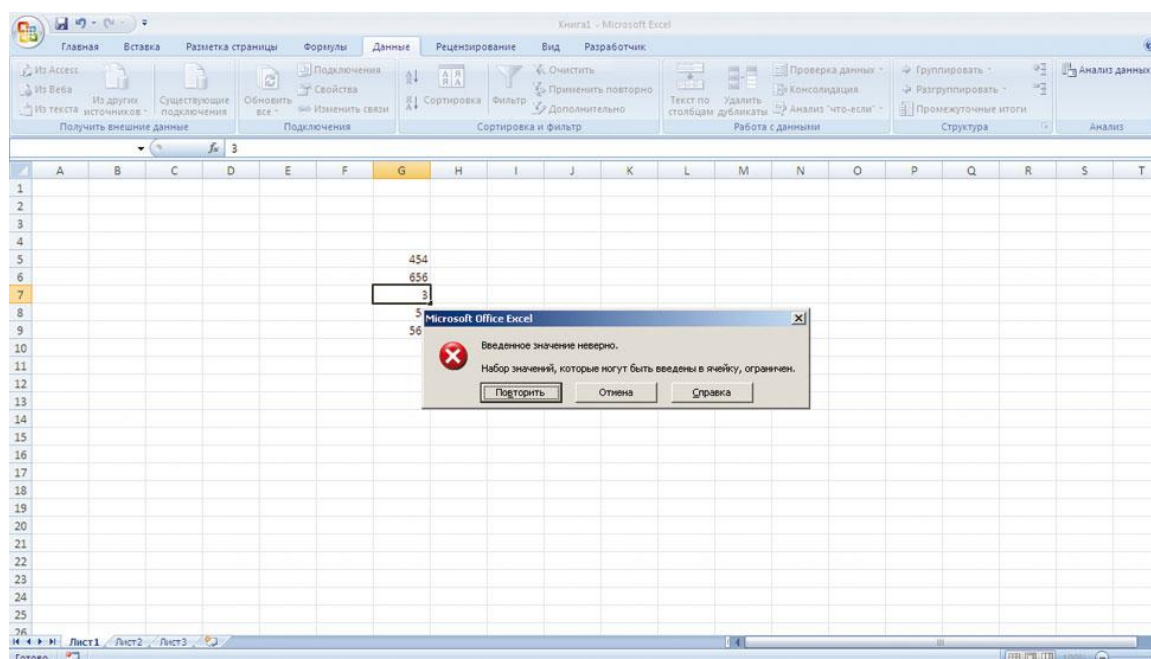


Рис. 4. Окно предупреждения о неправильном значении введенного числа

Для контроля уже введенных данных следует воспользоваться позицией меню «Обвести неверные данные», то результат будет выглядеть вот так (рис. 5):

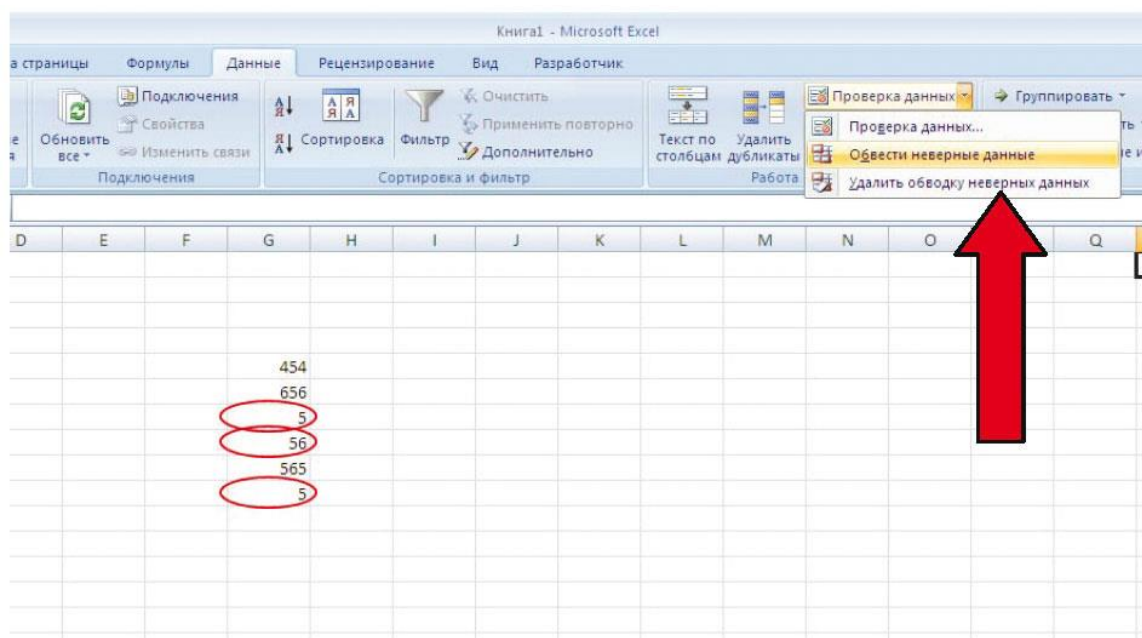


Рис. 5. Неверные данные обведены овалом после применения позиции меню «Обвести неверные данные»

Такой начальный контроль при вводе исходных в электронную таблицу, а также после заполнения электронной таблицы, поможет избежать возможных больших потерь фирме и исключить ошибки на самом раннем этапе работы с данными.

Следующим удобным методом контроля введенных и расчетных значений является условное форматирование ячеек с данными. Когда значения данных в ячейки выходят за пределы отведенного диапазона, то формат ячейки становится отличным от остальных ячеек листа. Для того, чтобы воспользоваться этим инструментом необходимо на вкладке «Главная» выбрать группу «Стили» и позицию меню «Условное форматирование» (рис. 6):

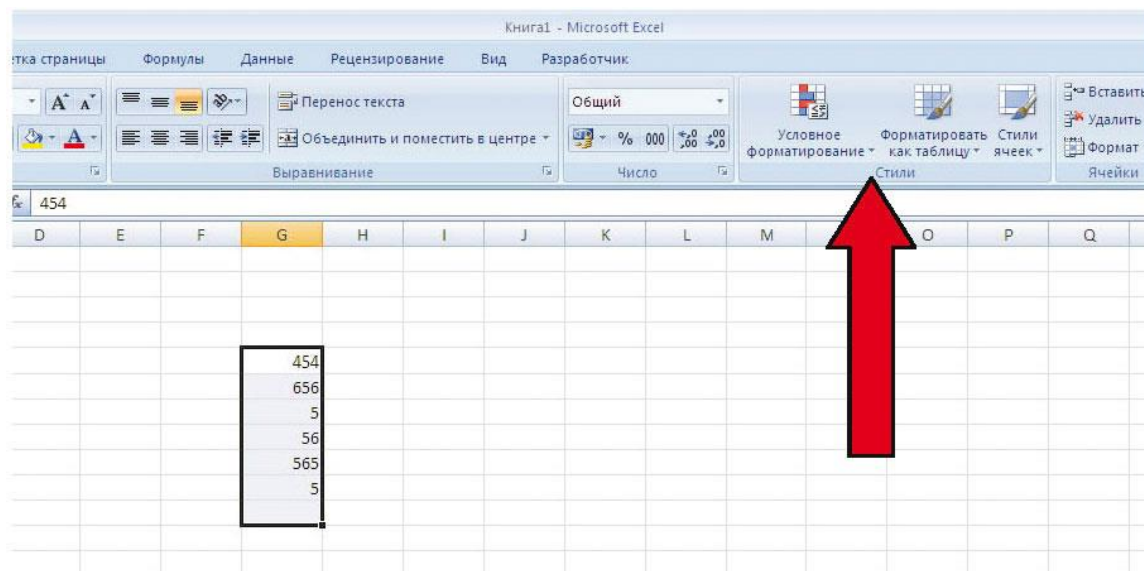


Рис. 6. Позиция меню условного форматирования ячеек листа

Введем правило выделения ячейки при превышении значения 500 (рис. 7, рис. 8):

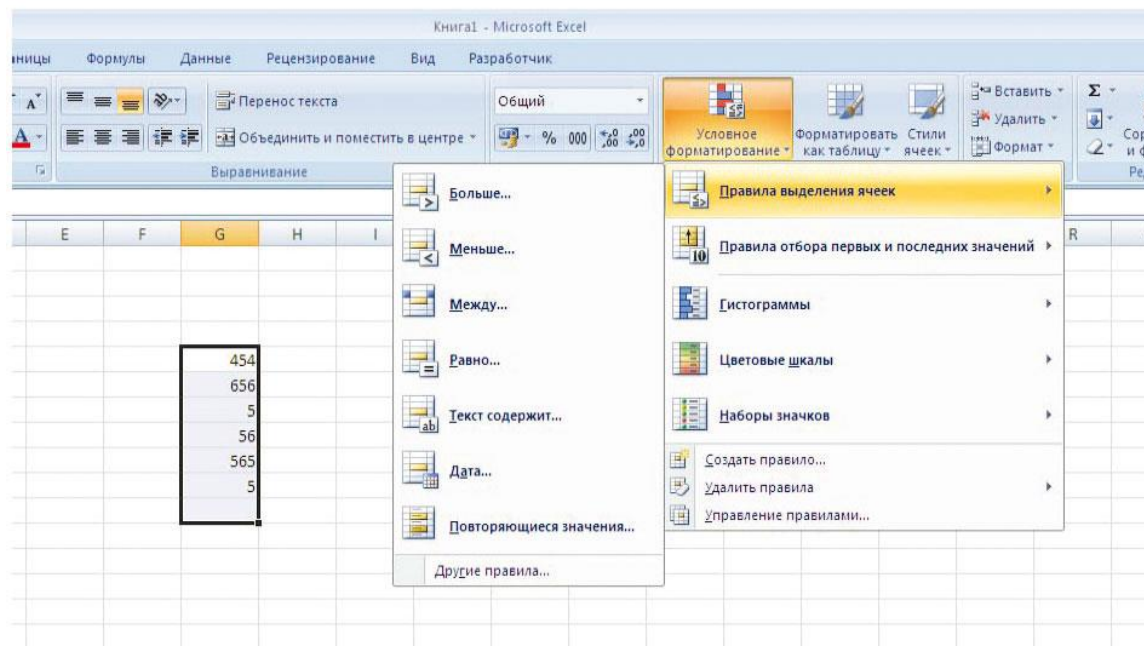


Рис. 7. Позиции меню для отбора и ввода значений

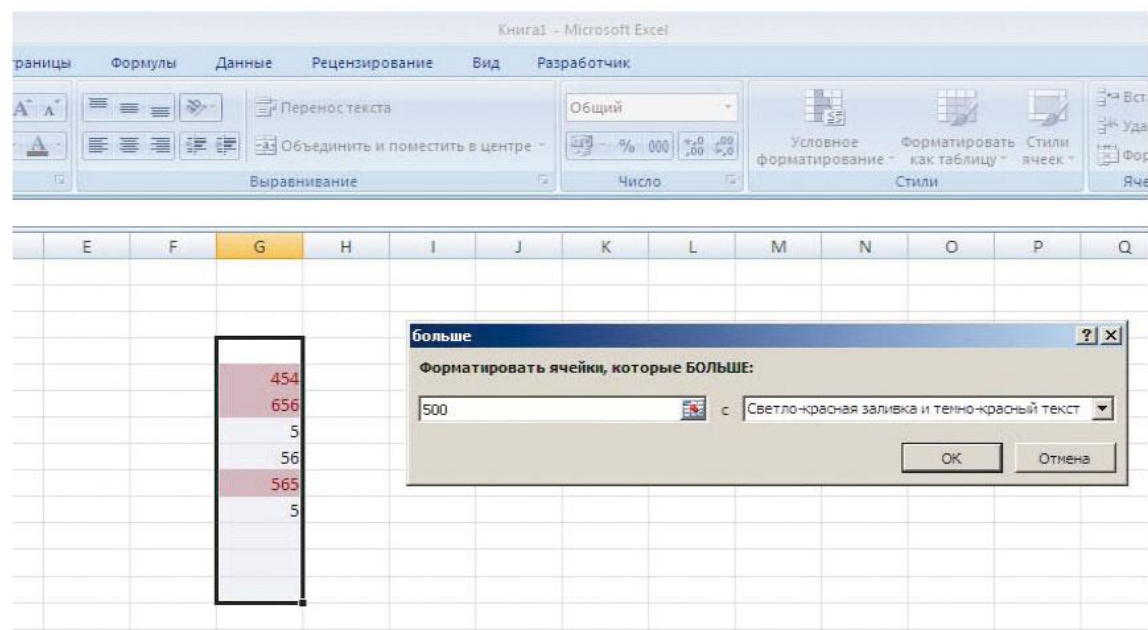


Рис. 8. Диалоговое окно контроля введенного значения больше 500 и результата форматирования

Одним из способов безошибочного ввода данных является использование информации из заранее подготовленных электронных таблиц – справочных таблиц. Здесь следует отметить, что доступ к данным этих справочных таблиц может быть реализован различными способами:

1) Если необходимо просто вводить в ячейку информацию из определенного списка, то лучше всего для этой цели подойдет форматирование ячейки на проверку вводимых данных. Этот подход наиболее эффективен для заполнения ячейки списком постоянных данных. Для этого выберем позиции меню «Данные → Проверка данных → Проверка данных». Появится диалоговое окно «Проверка вводимых данных», в котором необходимо выбрать позицию «Список» (рис. 9, рис. 10):

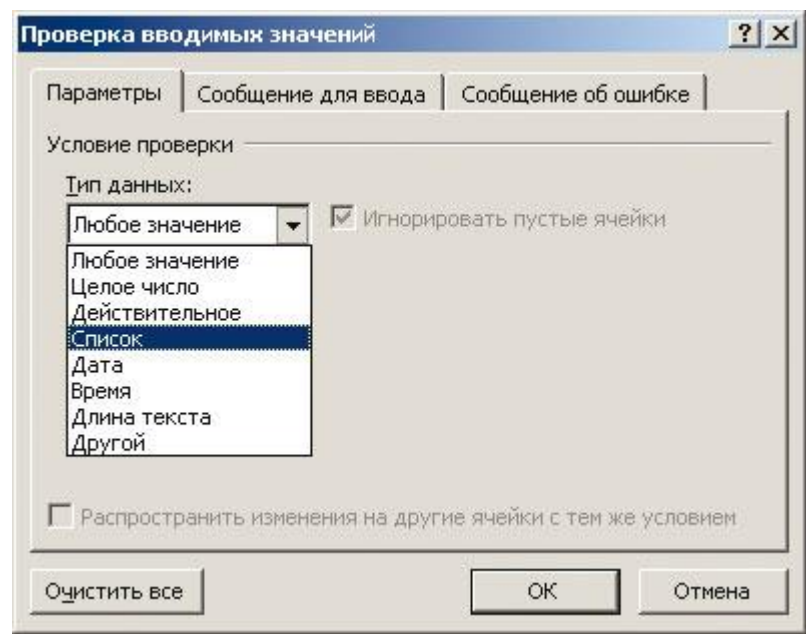


Рис. 9. Позиция «Список» в диалоговом окне «Проверка вводимых данных»

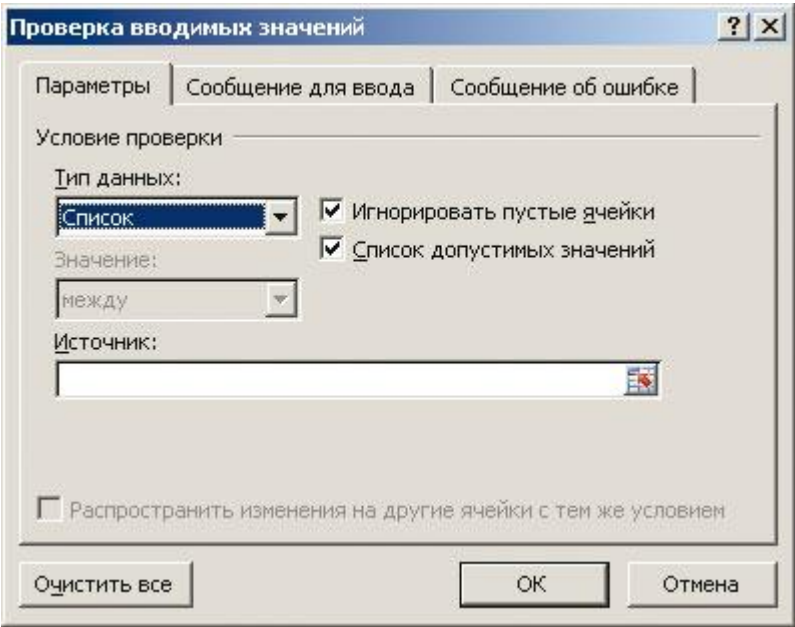


Рис. 10. Диалоговое окно для указания источника заполнения списка

В поле «Источник» необходимо указать диапазон ячеек, в которых находится справочная информация. Разумеется, данные справочные таблицы должны быть на листе за пределами видимости экрана. К сожалению, данный подход не позволяет хранить справочную информацию на другом листе книги (рис. 11).

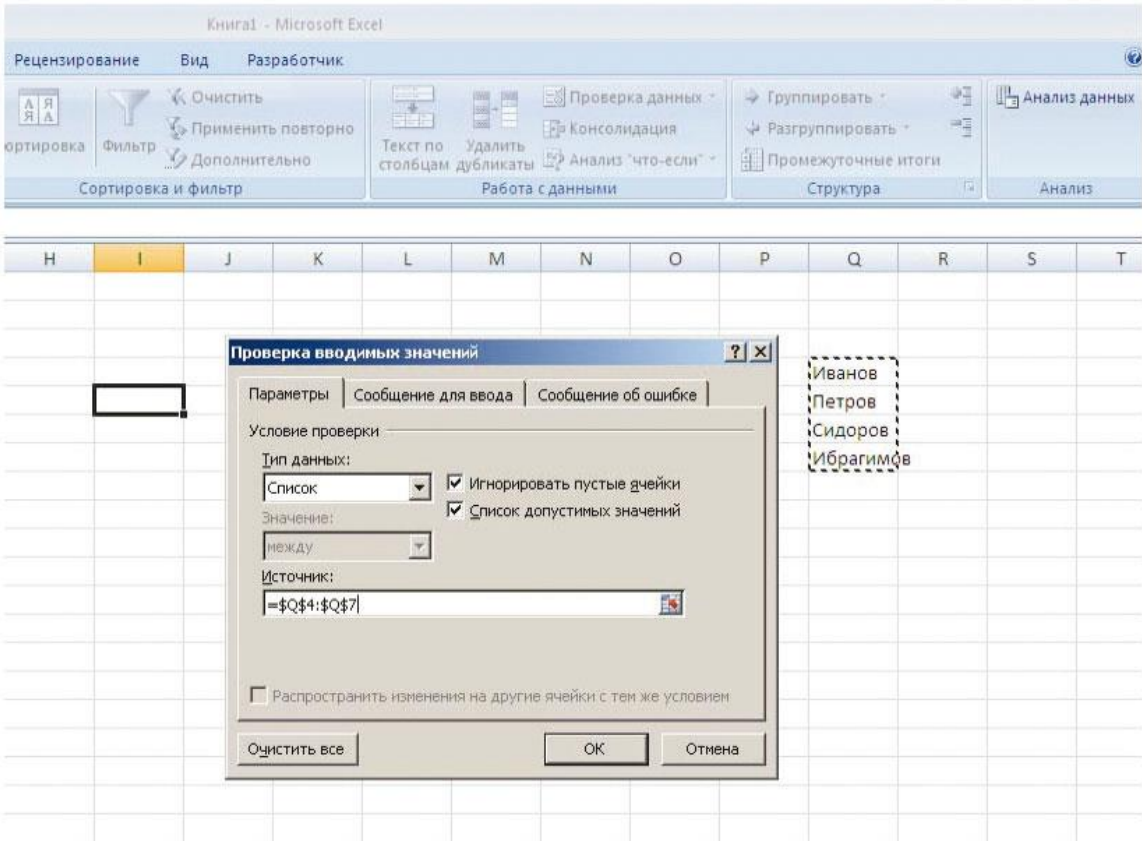


Рис. 11. Рабочая ячейка, окно диалога и справочный диапазон ячеек

После выполнения данных действий, в рабочей ячейке можно выбрать значения из раскрывающегося списка (рис. 12):

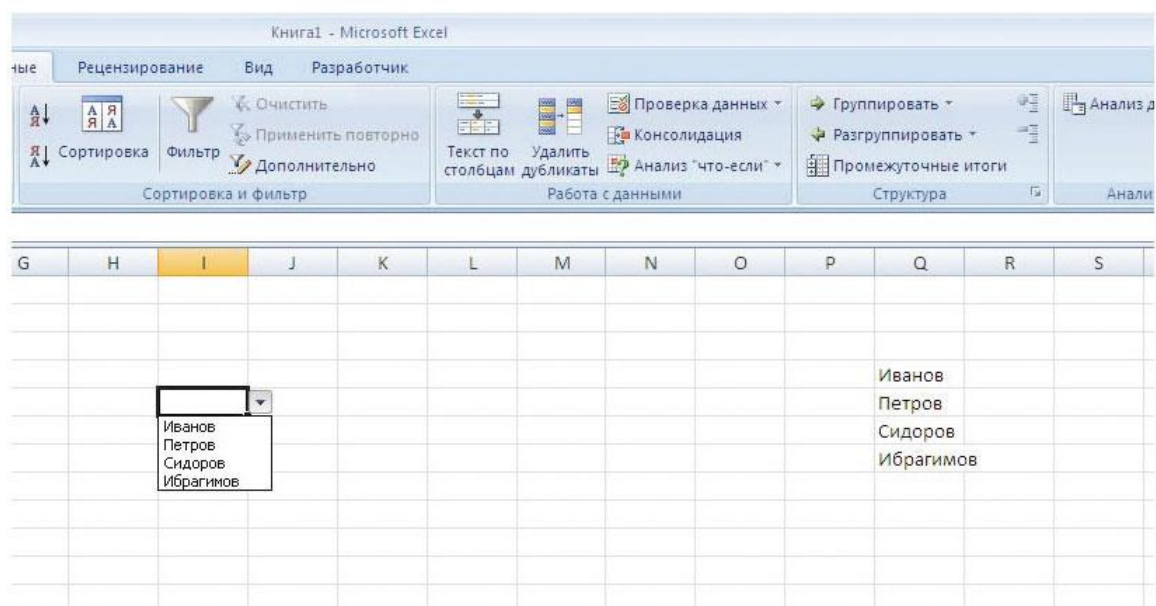


Рис. 12. Рабочая ячейка с возможностью ввода данных из списка и исходная справочная таблица значений.

2) Использование справочных таблиц на листах книги совместно с функциями листа ВПР, ПРОСМОТР.

Удобным инструментом для быстрого и безошибочного ввода данных в электронные таблицы является использование справочных таблиц с заранее введенными связанными данными совместно с функциями листа ВПР и ПРОСМОТР. В таких таблицах и по введенной информации находится информация для следующей ячейки таблицы в пределах строки на основе в данных справочной таблицы.

4.1. Функция лист ВПР для работы со ссылками и массивами

Используя функцию ВПР, вы фактически даете такую команду: «Вот значение. Нужно перейти в другое место, найти такое же значение и показать слова или числа в ячейке, соответствующей ему». Чтобы было проще, третье значение (номер столбца) можно рассматривать как результат поиска. Первые три аргумента функции ВПР являются обязательными. Последний аргумент необязателен, однако если его опустить, по умолчанию ему будет присвоено значение ИСТИНА[1]. Пример использования данной функции представлен на рис. 13. На листе «Протокол» содержится электронная таблица протокола проведения экзамена, а на листе «Исходные данные» – информация о студентах группы в соответствии со справочной таблицей (рис. 14).

[illegible]

Рис. 13. Таблица на листе «Протокол» с выделенной ячейкой, информацией в строке формул и отображенным результатом

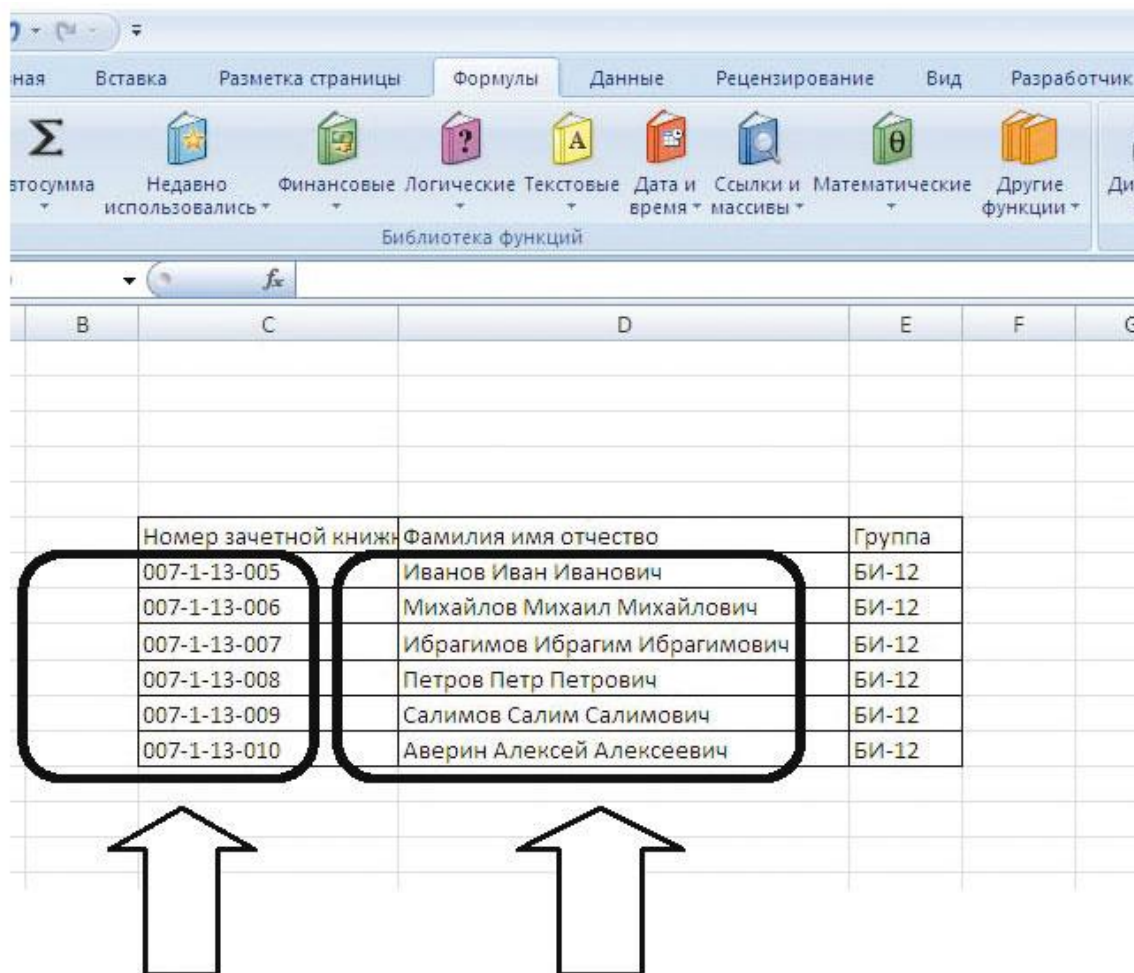
Функция ВПР, введенная в ячейку I3, имеет следующую структуру заполнения:

=ВПР(Протокол! Н12;'Исходные данные'!\$C\$7:\$E\$12;2;ЛОЖЬ).

Работу этой функции для ячейки I3 листа «Протокол» можно изобразить следующей последовательностью:

Протокол! Н12→Исходные данные! C7→Исходные данные! D7

Особенностью функции ВПР является то, что ключевые данные должны находиться в крайнем левом столбце таблицы-справочника (рис. 14):



| Номер зачетной книжки | Фамилия имя отчество | Группа |
|-----------------------|-------------------------------|--------|
| 007-1-13-005 | Иванов Иван Иванович | БИ-12 |
| 007-1-13-006 | Михайлов Михаил Михайлович | БИ-12 |
| 007-1-13-007 | Ибрагимов Ибрагим Ибрагимович | БИ-12 |
| 007-1-13-008 | Петров Петр Петрович | БИ-12 |
| 007-1-13-009 | Салимов Салим Салимович | БИ-12 |
| 007-1-13-010 | Аверин Алексей Алексеевич | БИ-12 |

Рис. 14. Таблица-справочник номеров зачетных книжек студентов группы

4.2. Функция лист ПРОСМОТР для работы со ссылками и массивами

Функция ПРОСМОТР очень похожа на функцию ВПР. Различие заключается в том, что функция ВПР ищет соответствие в первом столбце, а функция ПРОСМОТР ищет в соответствии с размерностями массива. Это означает, что столбцом, в котором находятся ключевые значения, может быть любым из таблицы-справки, и соответственно, результатная информация также может быть выбрана из любого столбца по ключу. Ниже приведены заполненный бланк исходных данных функции ПРОСМОТР (рис. 15) и результат работы функции для Михайлова Михаила Михайловича (рис. 16).

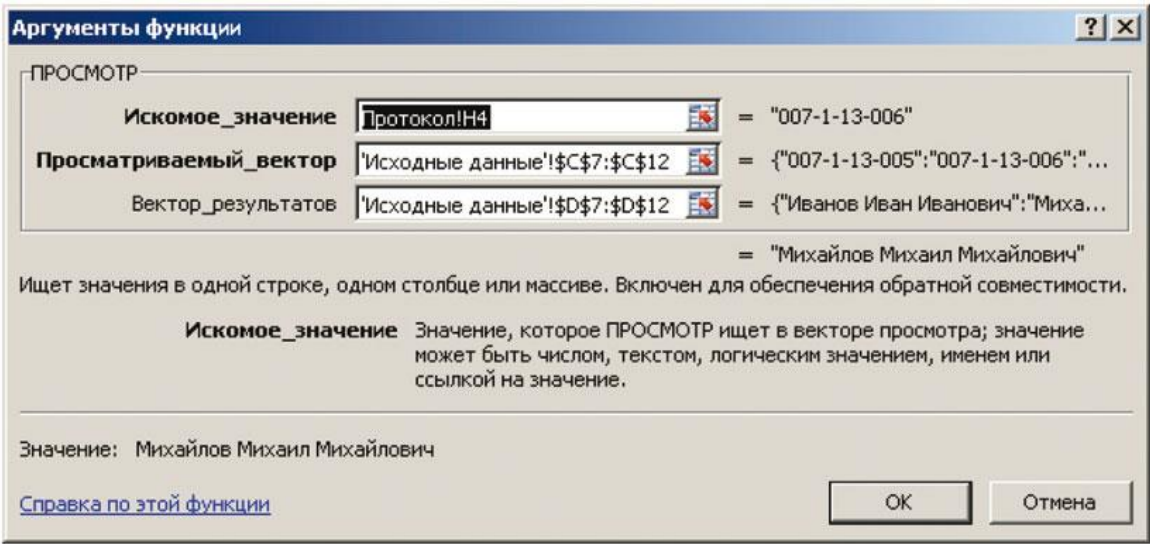


Рис. 15. Бланк исходных данных функции ПРОСМОТР для ячейки I4

Книга3.xlsx - Microsoft Excel

Разметка страницы Формулы Данные Рецензирование Вид Разработчик

Calibri 11 A A⁺ Перенос текста Общий Условное форматирование Форматировать как таблицу Обычный Нейтр Хороший Ввод

Шрифт Выравнивание Число Стили

fx =ПРОСМОТР(Протокол!И4;Исходные данные!\$C\$7:\$C\$12;Исходные данные!\$D\$7:\$D\$12)

| Группа | Предмет | Преподаватель | Дата сдачи | номер зачетной книжки | Фамилия Имя Отчество студента | оценка |
|--------|----------------------------------|-------------------------|------------|-----------------------|-------------------------------|---------|
| Би-12 | Теоретические основы информатики | Сидоров Сидор Сидорович | 07.12.2014 | 007-1-13-005 | Иванов Иван Иванович | хорошо |
| Би-12 | Теоретические основы информатики | Сидоров Сидор Сидорович | 07.12.2014 | 007-1-13-006 | Михайлов Михаил Михайлович | хорошо |
| Би-12 | Теоретические основы информатики | Сидоров Сидор Сидорович | 07.12.2014 | 007-1-13-007 | Ибрагимов Ибрагим Ибрагимович | удовл |
| Би-12 | Теоретические основы информатики | Сидоров Сидор Сидорович | 07.12.2014 | 007-1-13-008 | Петров Петр Петрович | отлично |
| Би-12 | Теоретические основы информатики | Сидоров Сидор Сидорович | 07.12.2014 | 007-1-13-009 | Салимов Салим Салимович | хорошо |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Рис. 16. Формула в ячейке I4 и результат работы функции ПРОСМОТР

4.3. Использование элементов управления на формах листа

При создании приложений с удобным интерфейсом в среде Excel не обойтись без использования элементов управления, которые являются основой любого окна диалога Windows-приложения. Данные элементы размещены едино на панели «Элементы управления» в меню «Разработчик» (рис. 17):

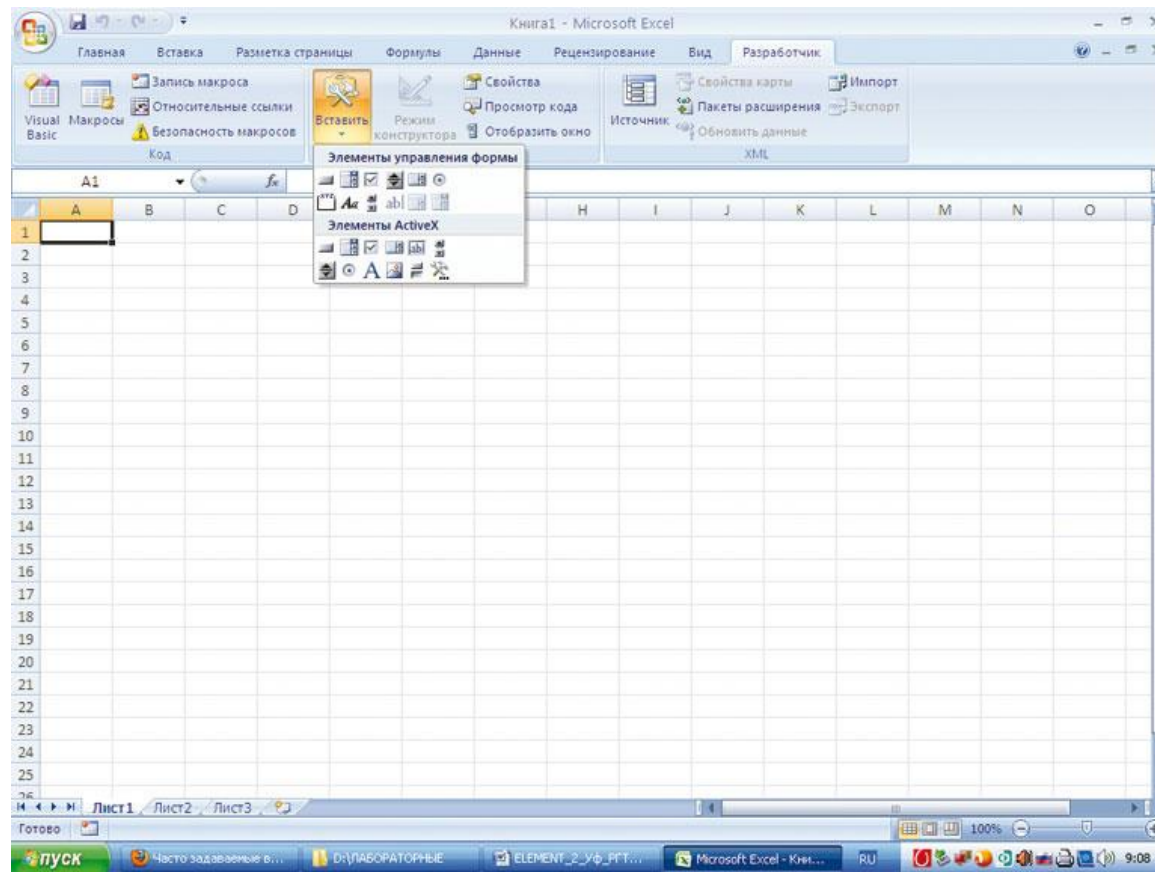


Рис. 17. Панель «Элементы управления» на вкладке «Разработчик»

Если вкладка «Разработчик» отсутствует на ленте, то для его вызова в меню пользователя следует в настройках табличного процессора выделить флажок «Показать вкладку «Разработчик» на ленте (рис. 18, рис. 19):

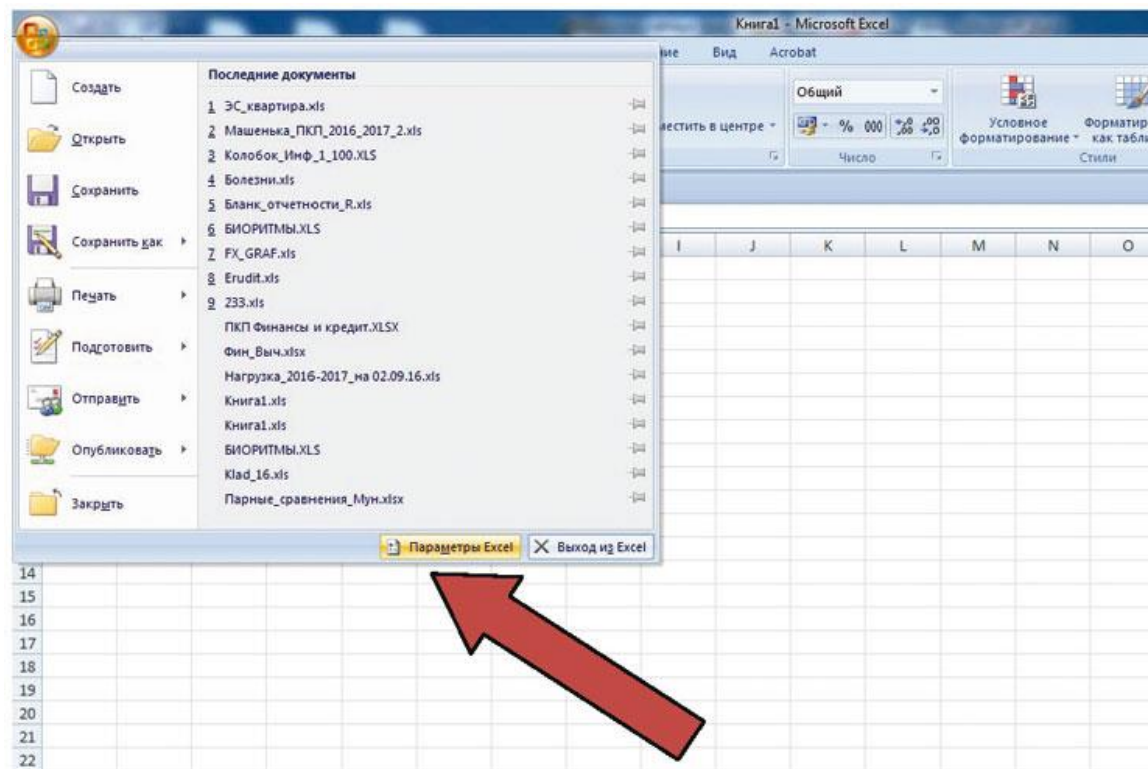


Рис. 18. Позиция меню для настройки параметров табличного процессора

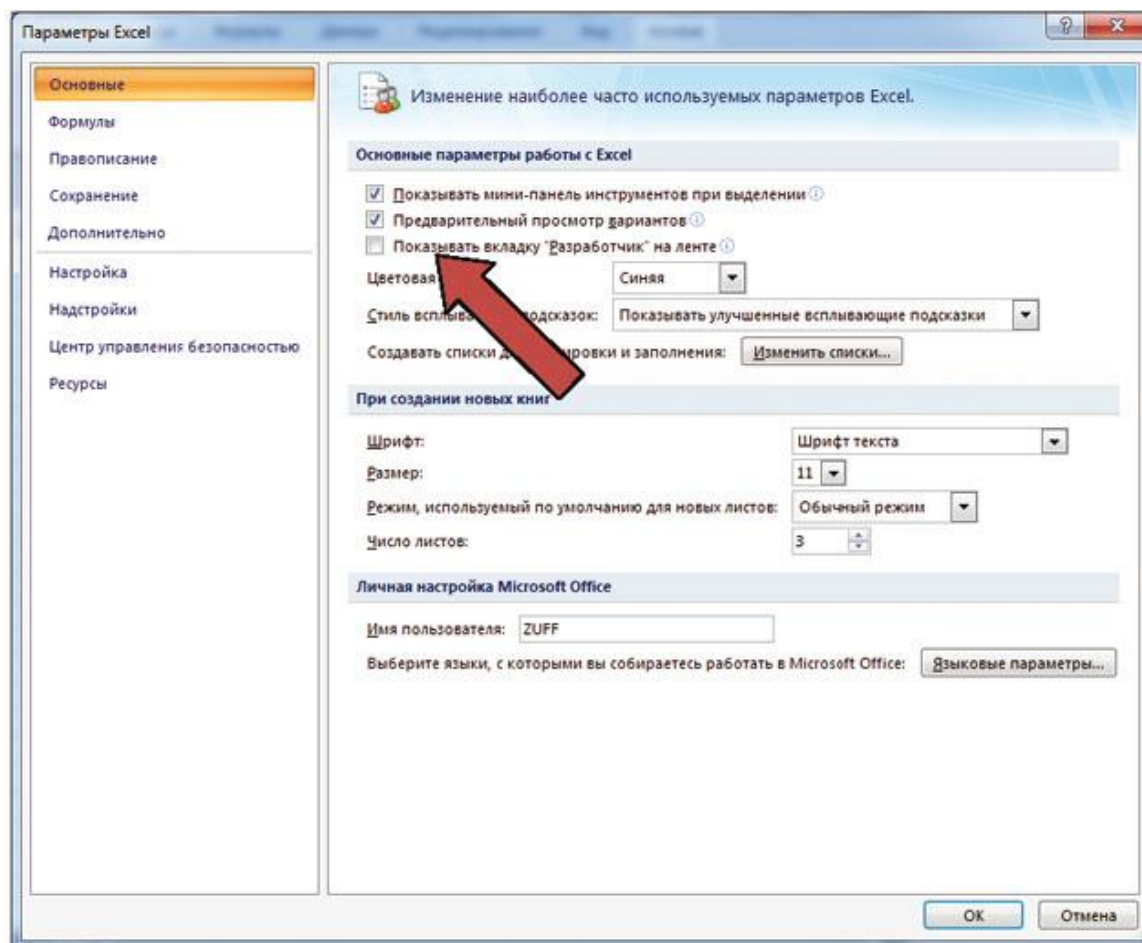


Рис. 19. Выставление флажка для отображения вкладки «Разработчик» на ленте

Следует обратить внимание, что пользователю для реализации своих проектов предоставляются две группы элементов управления:

- элементы управления формы;
- элементы ActiveX.

Главное отличие элементов управления формы от второй группы заключается в том, что пользователю предоставляется возможность работы над удобным интерфейсом приложения практически без знаний языка программирования VBA. Для использования элементов управления необходимо воспользоваться вкладкой Разработчик^Элементы управления^Вставить.

Эта вкладка содержит следующие элементы:

- 1) Подпись.
- 2) Группа элементов.
- 3) Кнопка.
- 4) Флажок.
- 5) Переключатель.
- 6) Список.
- 7) Поле со списком.
- 8) Полоса прокрутки.
- 9) Счетчик.

Следует иметь в виду, что элементы управления являются внешними объектами листа, поэтому могут размещаться произвольно и перекрывать информацию в ячейках листа книги.

Элемент управления «Подпись»

В отличие от информации, занесенной в ячейки таблицы Excel, элемент «Подпись» (рис. 20) позволяет создавать надписи в любом месте рабочего листа, а не только в ячейках.

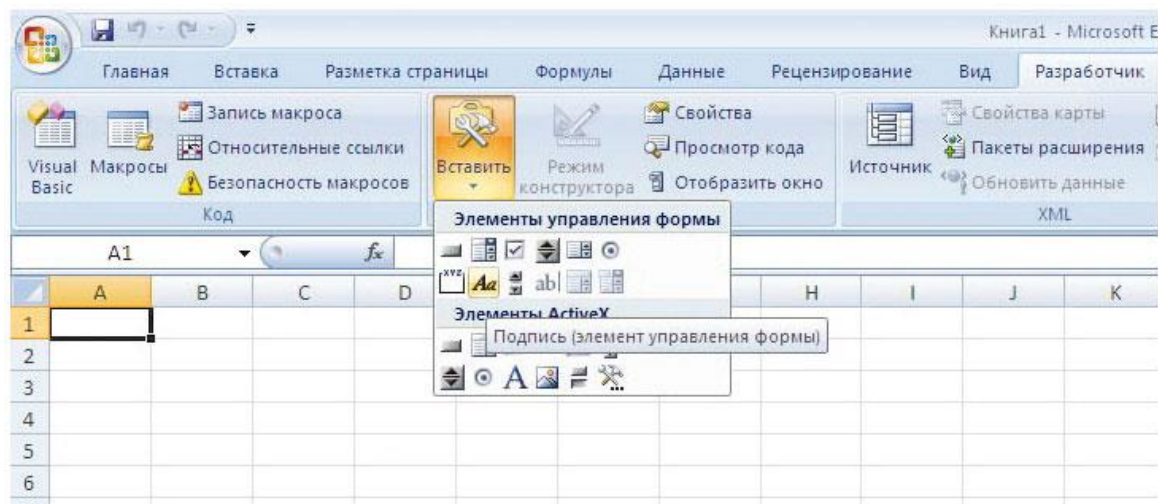


Рис. 20. Элемент «Подпись» на панели «Элементы управления формы»

Ячейка, расположенная под подписью, может быть использована для своего прямого назначения.

Элемент управления «Кнопка»

Наиболее распространенным элементом управления является элемент «Кнопка» (рис. 21). Он служит для запуска на выполнение программного кода при наступлении с ним какого-либо события.

Для создания кнопки необходимо выбрать данный элемент на панели инструментов. Маркер мыши примет вид маленького крестика. Этим крестиком следует указать местоположение левого верхнего угла будущего элемента управления на листе книги и далее, нажав левую клавишу, растянуть вниз и вправо кнопку до нужного размера. На кнопке возможно нанесение надписи, выражающие назначение кнопки. Поэтому при создании надписи требуется указать истинное назначение кнопки, а не абстрактные наименования (рис. 21):

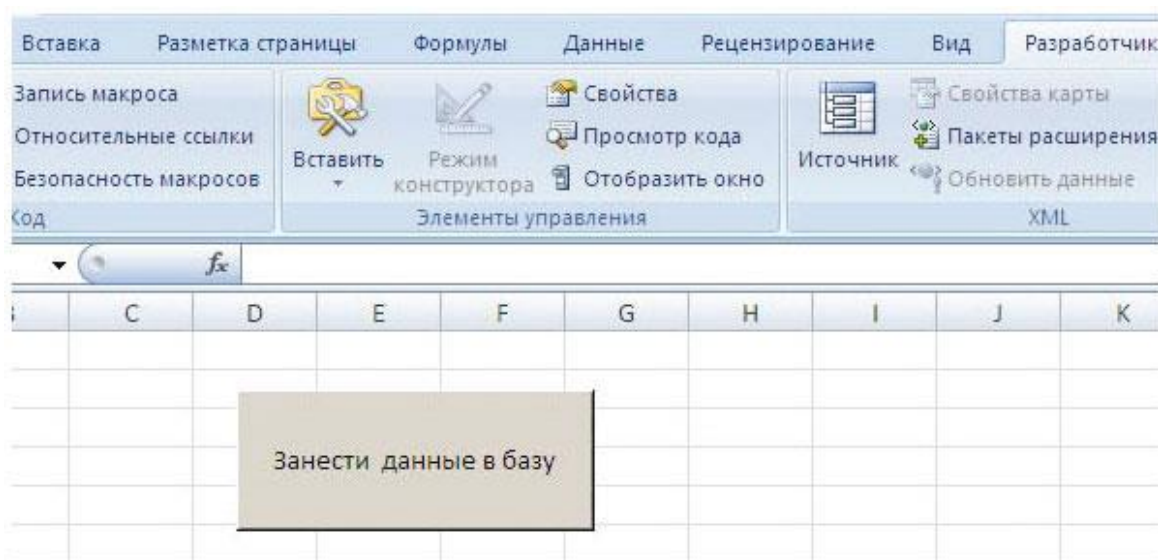


Рис. 21. Элемент «Кнопка» на листе

При нанесении надписи на кнопку действуют такие же правила выбора размера и цвета шрифта, что и при занесении информации в ячейку листа. При этом используется аналогичное окно диалога для выравнивания надписи на кнопке, что и для выравнивания записи в ячейке листа (рис. 22). Для изменения надписи на кнопке необходимо находиться в режиме редактирования текста.

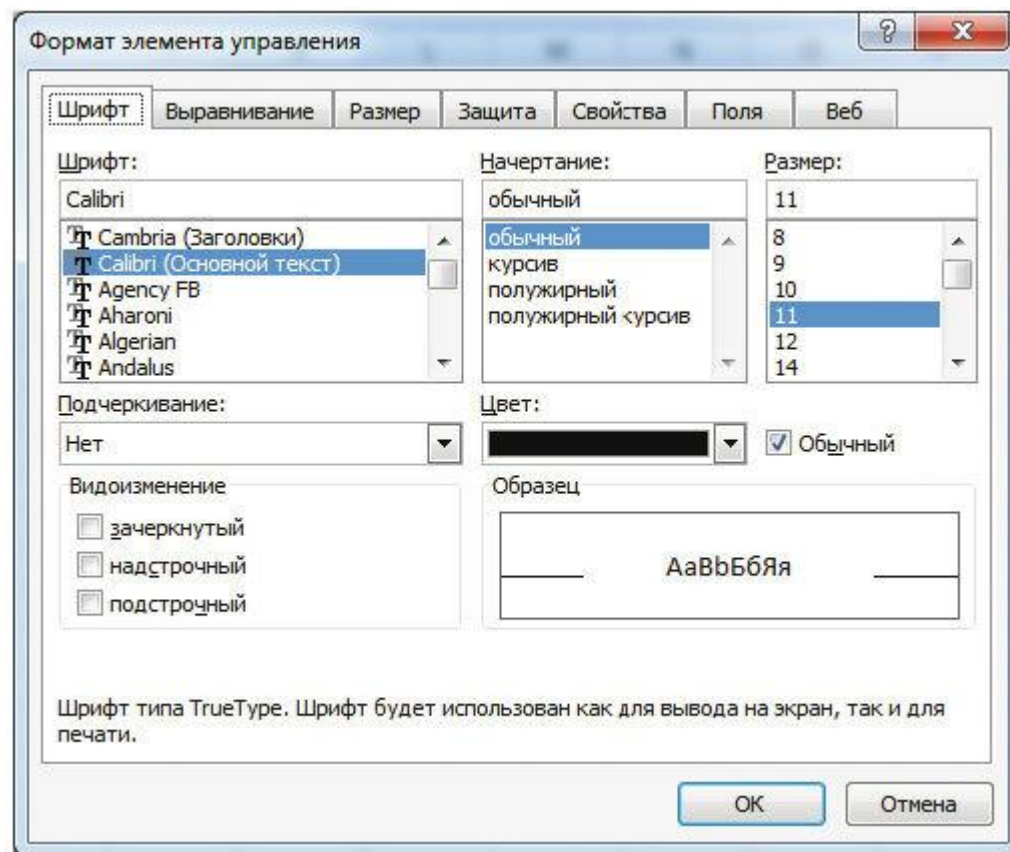


Рис. 22. Окно диалога свойств элемента управления «Кнопка»

Вкладки диалогового окна содержат основные свойства и процедуры, которые доступны при работе с элементом управления.

Для того чтобы настроить кнопки, необходимо выделить кнопку и, нажав правую клавишу мыши, выбрать позиции контекстного меню «Формат объекта...».

Макросы кнопкам можно присваивать как во время их создания, так и после редакции внешнего вида кнопки (рис. 23):

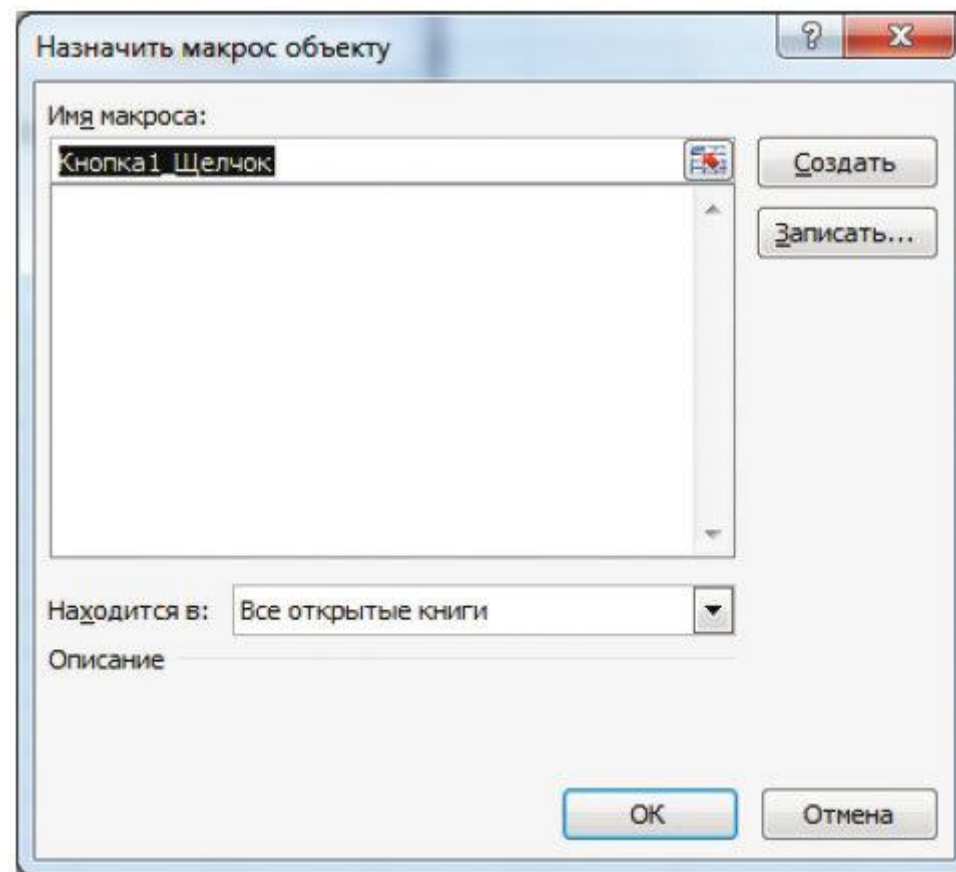


Рис. 23. Назначение макроса кнопке

Для отмены присвоения макроса кнопке во время ее создания необходимо нажать клавишу клавиатуры «ESC» или кнопку «Отмена» окна диалога.

Следует иметь в виду, что элемент управления по желанию может выводиться или не выводиться на печать. Для выбора варианта распечатки необходимо установить флажок в позиции «Выводить на печать» во вкладке «Свойства» позиции контекстного меню «Формат объекта...» (рис. 24):

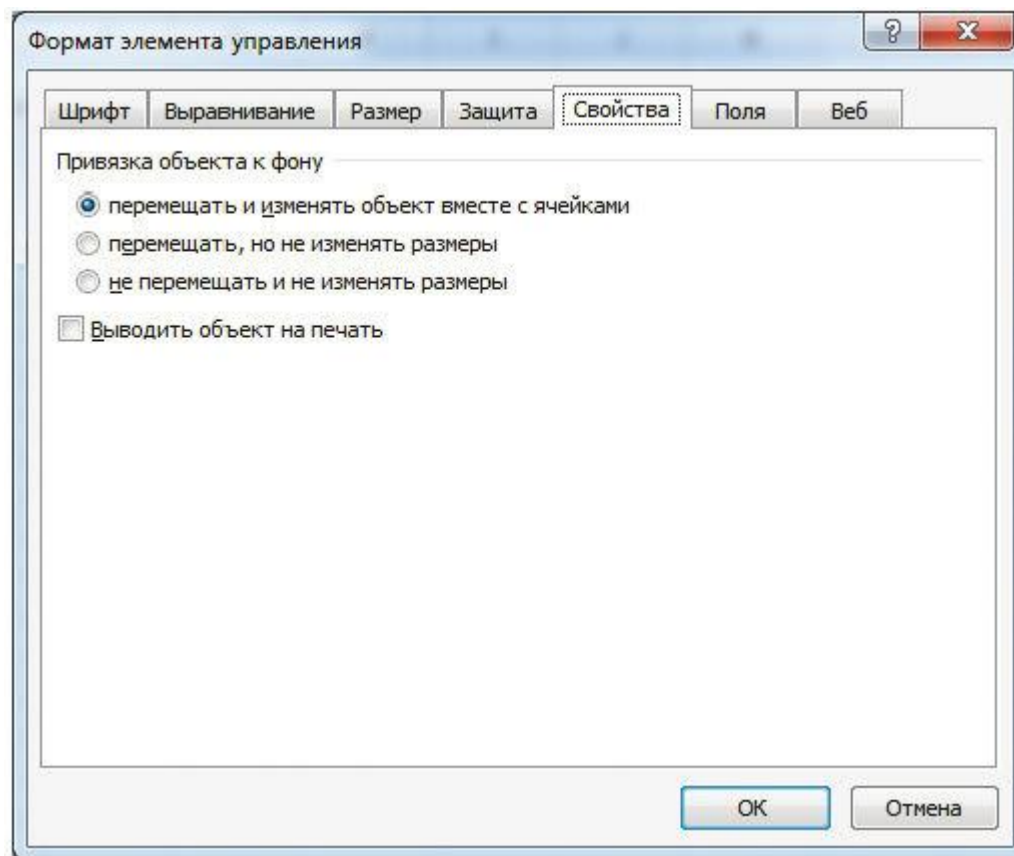


Рис. 24. Вкладка настроек свойств элемента управления

Это относится ко всем рассматриваемым далее элементам управления.

Элемент управления «Группа»

Несмотря на то, что положение переключателей группы может быть произвольным на листе, для удобства работы и наглядности результата их следует расположить вместе. Для этого имеется такой элемент управления, как «Группа» или рамка группы (рис. 25):

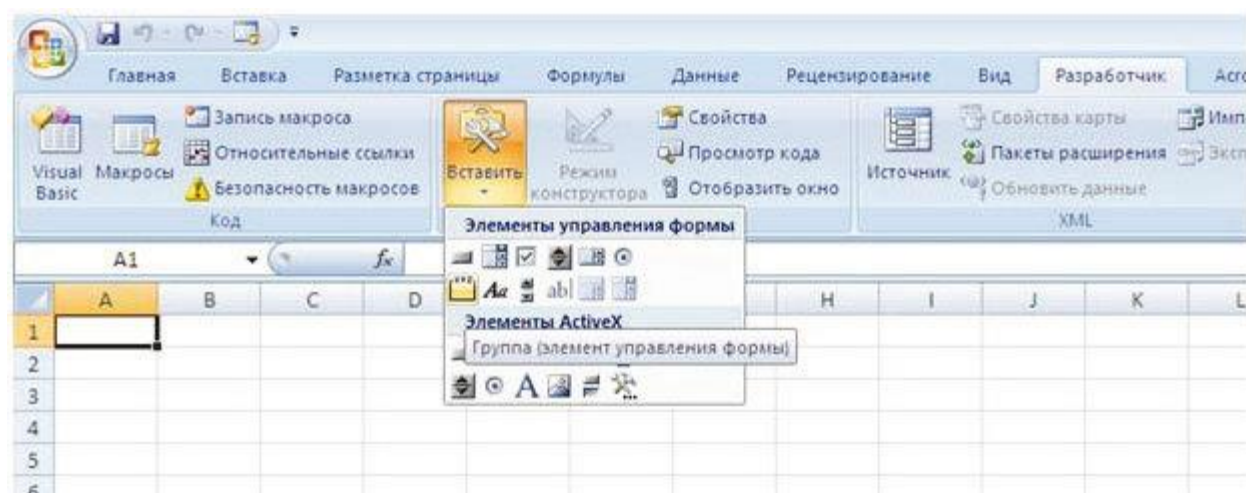


Рис. 25. Элемент управления «Группа»

Особенностью его использования является то, что нельзя уже существующие переключатели группы поместить в рамку группы и заставить переключатели группы работать вместе. Здесь необходимо соблюсти строгую последовательность действий: с начало

создать элемент «Группа», а далее поместить в него переключатели. Только тогда группа нескольких переключателей будут работать корректно (рис. 26):

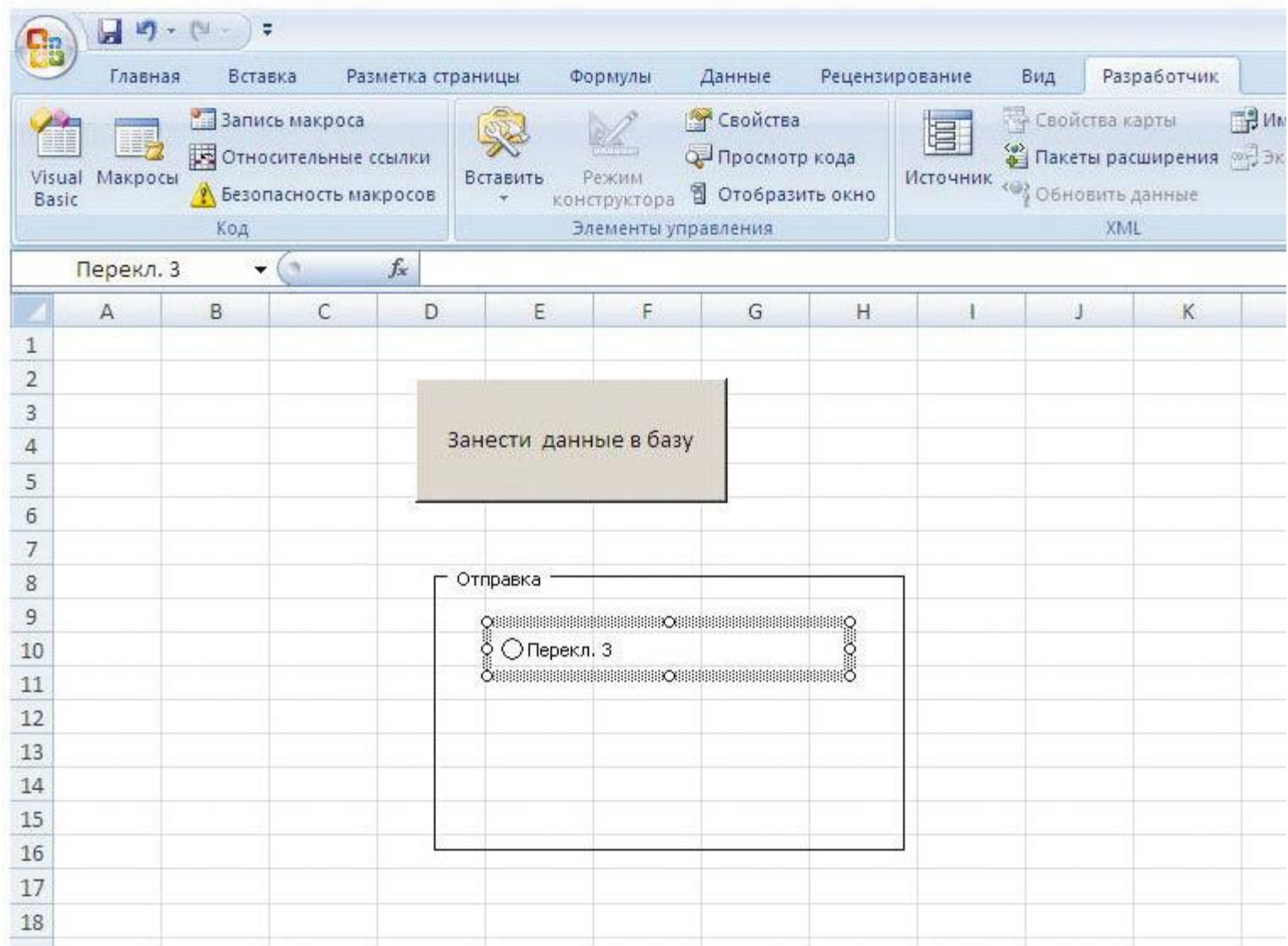


Рис. 26. Элемент управления «Переключатель» в группе

Элемент управления «Переключатель»

Элемент «Переключатель» (рис. 27) предназначен для выбора альтернативного варианта из имеющегося набора в группе:

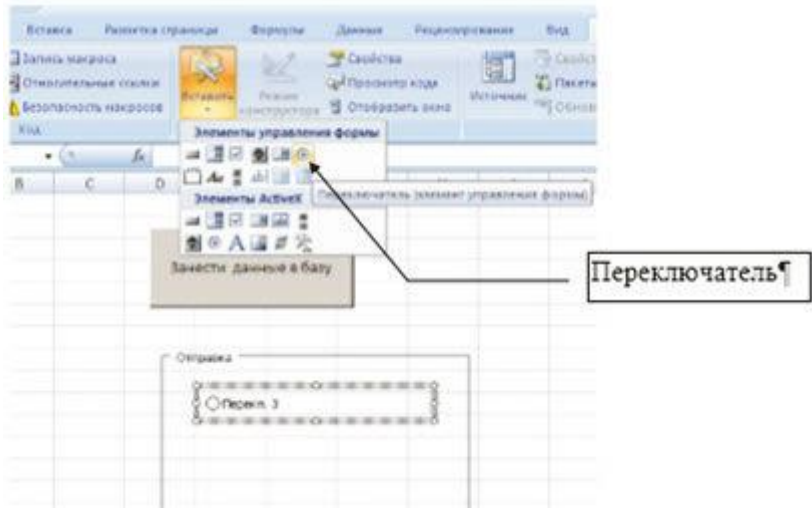


Рис. 27. Элемент управления «Переключатель» на панели «Элементы управления формы»

Поэтому для того чтобы данный элемент мог функционировать, необходимо иметь как минимум два элемента типа переключатель. В связанную ячейку результата работы группы переключателей вначале заносится значение первого переключателя. Местоположение каждого переключателя не влияет на работоспособность всей группы. Порядковый номер переключателя зависит от относительного времени его создания и в зависимости от принадлежности переключателя к той или иной группе элементов, можно определить его номер по адресу ячейки результата (рис. 28):

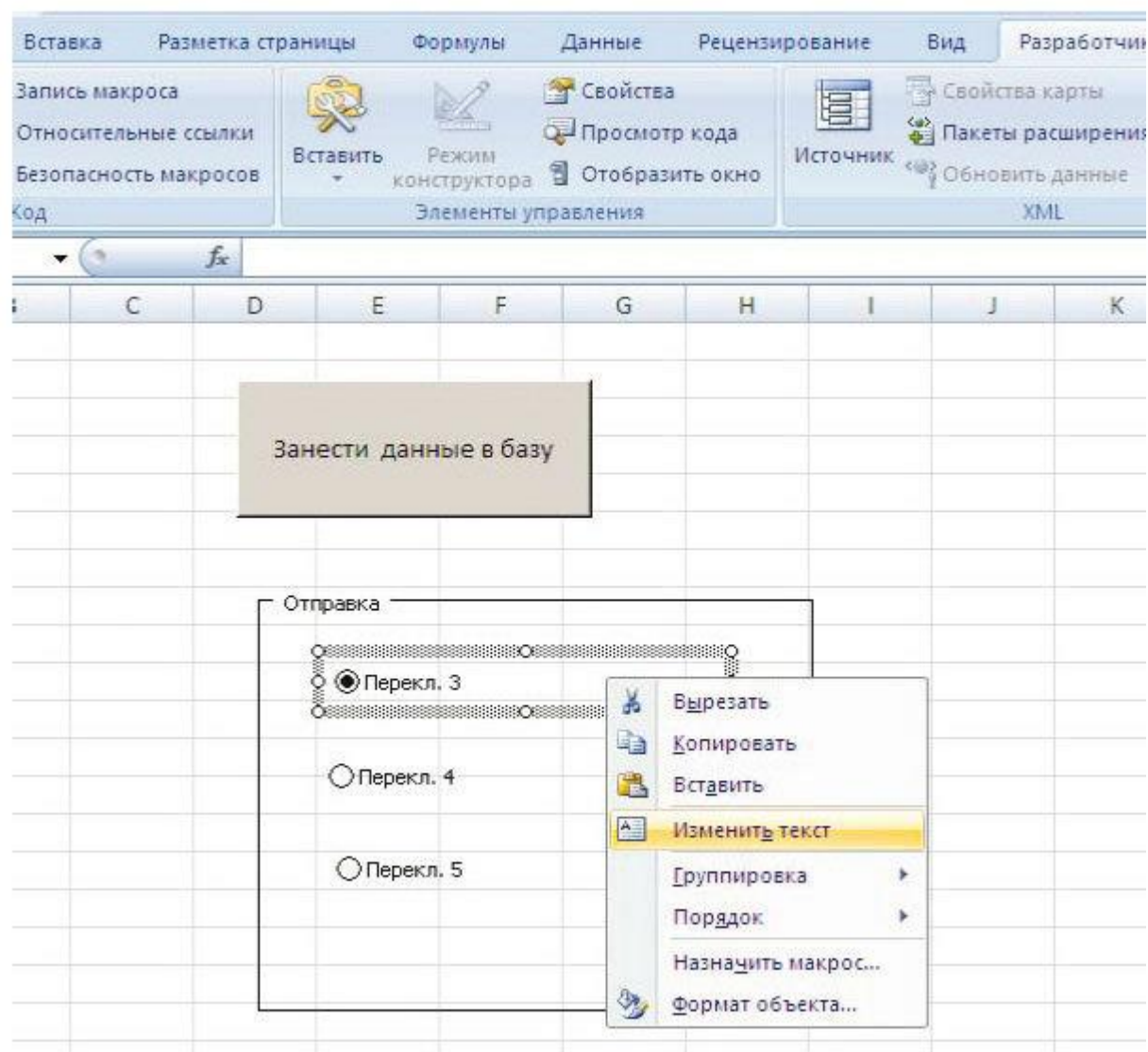


Рис. 28. Группа переключателей

Элемент управления «Флажок»

Элемент управления «Флажок» (рис. 29) также предназначен для выбора нужного варианта.

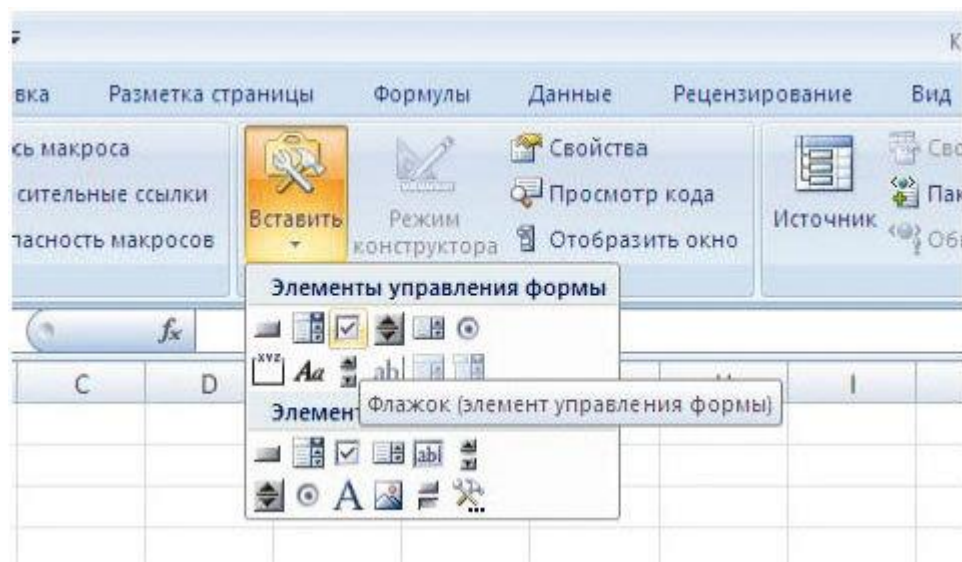


Рис. 29. Элемент управления «Флажок»

В отличие от переключателя он может функционировать самостоятельно, так как предназначен для выбора варианта: позиция «включена» или «выключена». Если даже флажки расположить в тематическую группу, то они все равно будут работать самостоятельно, позволяя набирать комбинацию из различных альтернатив.

Исходя из назначения данного элемента управления, результатом работы будет значение «ИСТИНА» или «ЛОЖЬ». Далее, результаты работы элемента управления можно использовать в логических выражениях, связанных с конкретным действием.

Элемент управления «Список»

Элемент управления «Список» (рис. 30) предназначен для выбора альтернативы из имеющегося списка:

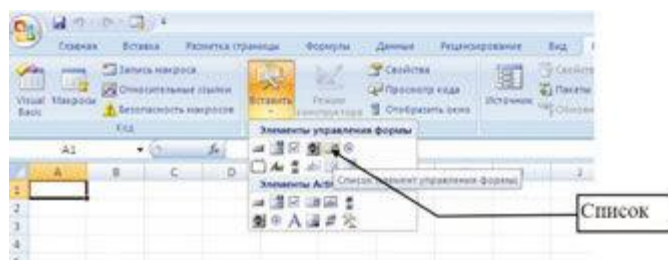


Рис. 30. Элемент управления «Список»

В результате использования данного элемента, порядковый номер выбранной строки можно разместить в любую ячейку листа и в дальнейшем проанализировать. Преимущество этого элемента над рассмотренным альтернативным элементом выбора «Переключатель» состоит в удобстве создания, выражающимся в возможности отдельного размещения элемента диалога и контентной части элемента управления, простоте создания элемента, сравнительной компактности.

Результат работы данного элемента управления, как и других элементов, может быть обработан с помощью условного оператора ЕСЛИ(). При выводе текстовой строки удобно использовать уже существующий список, который применялся для формирования элемента управления. Если список находился в ячейках B2:B5, а результат был помещен в ячейку A1, то для того чтобы использовать результат работы этого элемента управления в соответствующую замыслу пользователя ячейку следует поместить следующее выражение:

=ЕСЛИ(A1=1;B2;ЕСЛИ(A1=2;B3;ЕСЛИ(A1=3;B4;B5))).

Элемент управления «Поле со списком»

Элемент управления «Поле со списком» (рис. 31) предназначен для ввода в поле одного из значений из раскрывающегося списка. Это самый компактный и одновременно информативный элемент управления.

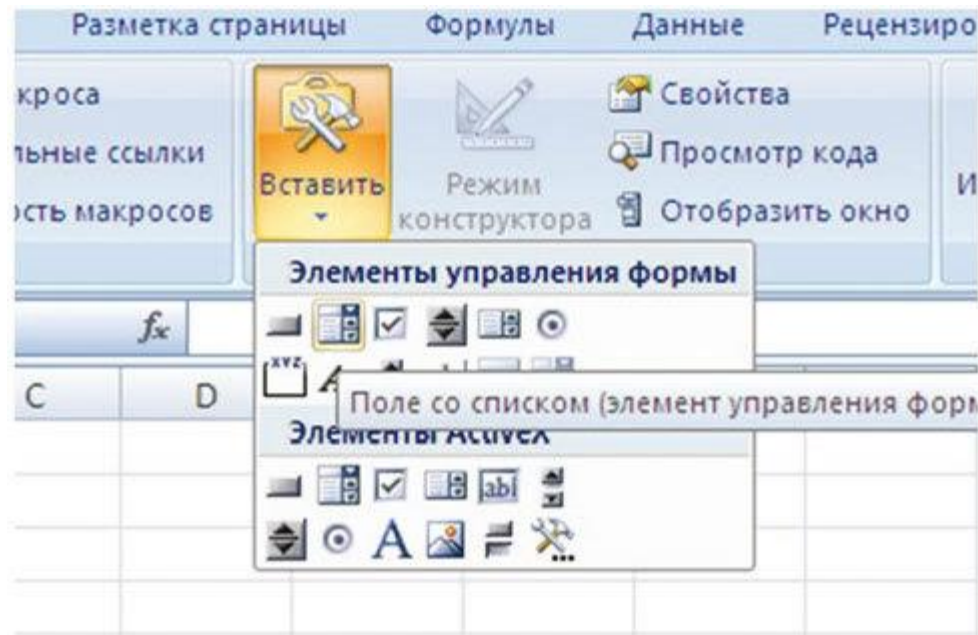


Рис. 31. Элемент управления «Поле со списком»

Поэтому его удобно использовать для построения тестовых и обучающих программ. Результатом работы этого элемента управления также является целое число, которое помещается в ячейку для результата. Элемент также как и «Список» имеет возможность обратного функционирования, т. е. когда по значению числа в ячейке результата в окне вывода элемента появляется соответствующее ему строка значения. Это может быть использовано в некоторых программных реализациях.

Элемент управления «Счетчик»

Элемент управления «Счетчик» (рис. 32) предназначен для наглядного занесения числового значения в связанную с ним ячейку в заданном числовом интервале и заданным шагом.

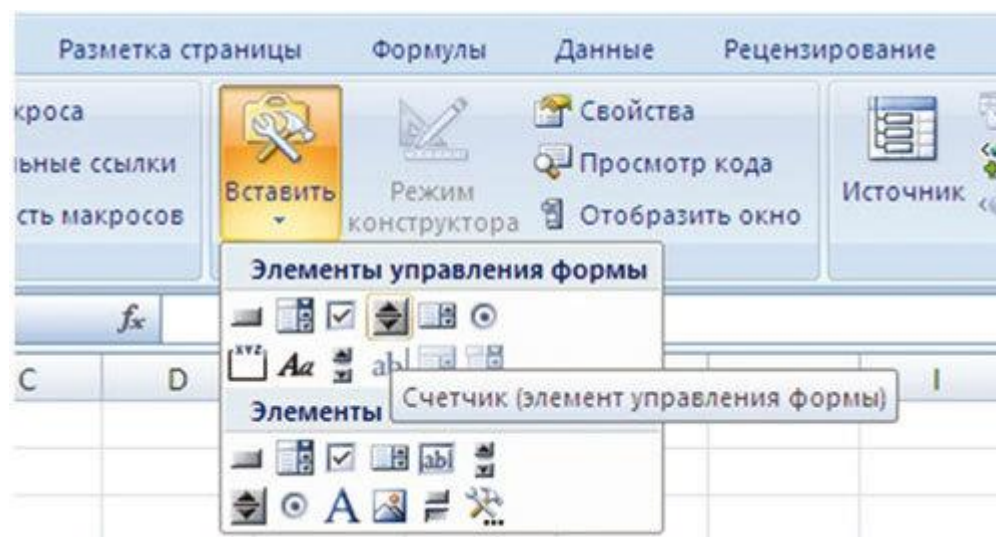


Рис. 32. Элемент управления «Счетчик»

Счетчик обладает отличным от рассмотренных ранее элементов управления окном диалога. Сам элемент управления состоит из двух кнопок со стрелками, направленными вверх и вниз. Стрелка, направленная вверх, служит для увеличения числа ячейки, а направленная вниз – для уменьшения его. После достижения граничного значения числа, установленного при настройке элемента управления в окне диалога, его значение в данном направлении прекращает изменяться. Возвращенное значение счетчика также обрабатывается с помощью функций листа, например функции ЕСЛИ().

Элемент управления «Полоса прокрутки»

Элемент управления «Полоса прокрутки» (рис. 33) предназначен для наглядного и быстрого безошибочного ввода числа в пределах заданного интервала и с установленным шагом.

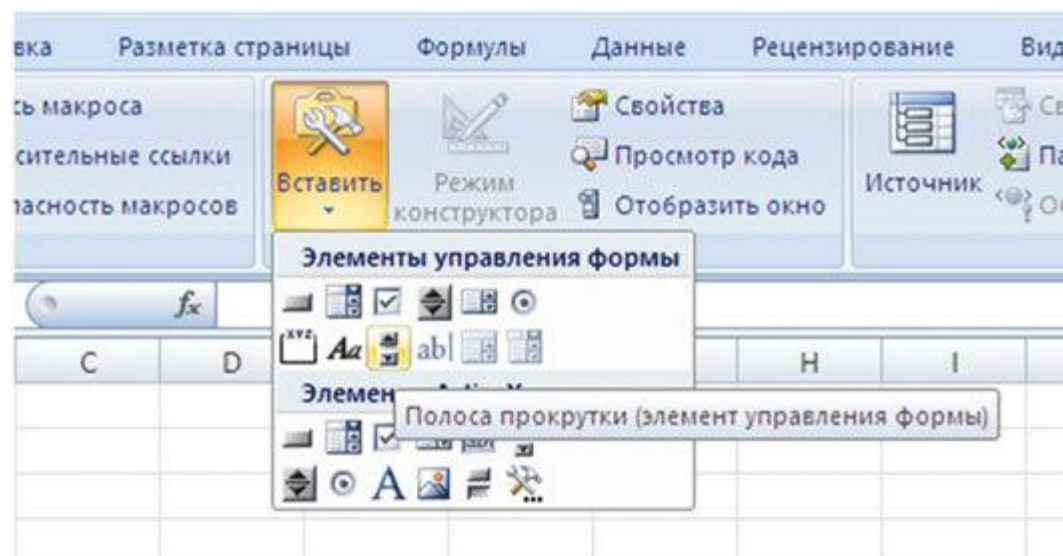


Рис. 33. Элемент управления «Полоса прокрутки»

В отличие от элемента управления «Счетчик», рассматриваемый элемент управления предоставляет пользователю более гибкие возможности при работе. Возвращенное значение элементом управления также обрабатывается с помощью функций листа, например ЕСЛИ().

5. Понятие макроса и различный подход к созданию и использованию их в приложениях MS Office

Часто при работе с офисными приложениями приходится изо дня на день выполнять одни и те же операции над документом. Это может быть и настройка среды работы, традиционные процедуры работы с документом, однообразные многочисленные операции над данными. Такими операциями могут быть: построение таблиц, создание шаблона (бланка) для ввода информации, выполнение постоянной последовательности при вычислительных операциях, построение диаграммы, сохранение файлов в нужной папке и многое другое. И всегда при этом возникает вопрос: «Можно ли автоматизировать эту рутинную работу?». Ответ – «Да, можно!».

Для этого в офисном пакете MS Office имеется замечательный инструмент, называемый макросом.

Макрос (макрокоманда) представляет собой любую последовательность операций над данными или элементами среды работы, записанную в виде программы и готовую к немедленному многократному использованию. Модули макросов записываются на языке программирования VBA (Visual Basic for Application). Особенностью создания несложных макросов в среде MS Word и MS Excel является то, что их можно создать, не зная языка программирования VBA. Код макроса в виде программного модуля формируется автоматически при записи макроса.

Каждый макрос имеет свое имя. По умолчанию макросам присваивается имя **Макрос** и далее его последовательный номер. Для удобства работы рекомендуется макросам присваивать имя, несущую смысловую нагрузку. Тогда будет легко ориентироваться в большом наборе макросов на вашем компьютере и через неделю вы не забудете, за что отвечает макрос под именем **Макрос4**.

Приобретя навыки создания макросов, можно постоянную рутинную работу с документом и приложением смело переложить на плечи персонального компьютера.

5.1. Запись макроса в MS Word и MS Excel

Для создания макроса воспользуемся вкладкой «Разработчик» → «Код» (рис. 34).

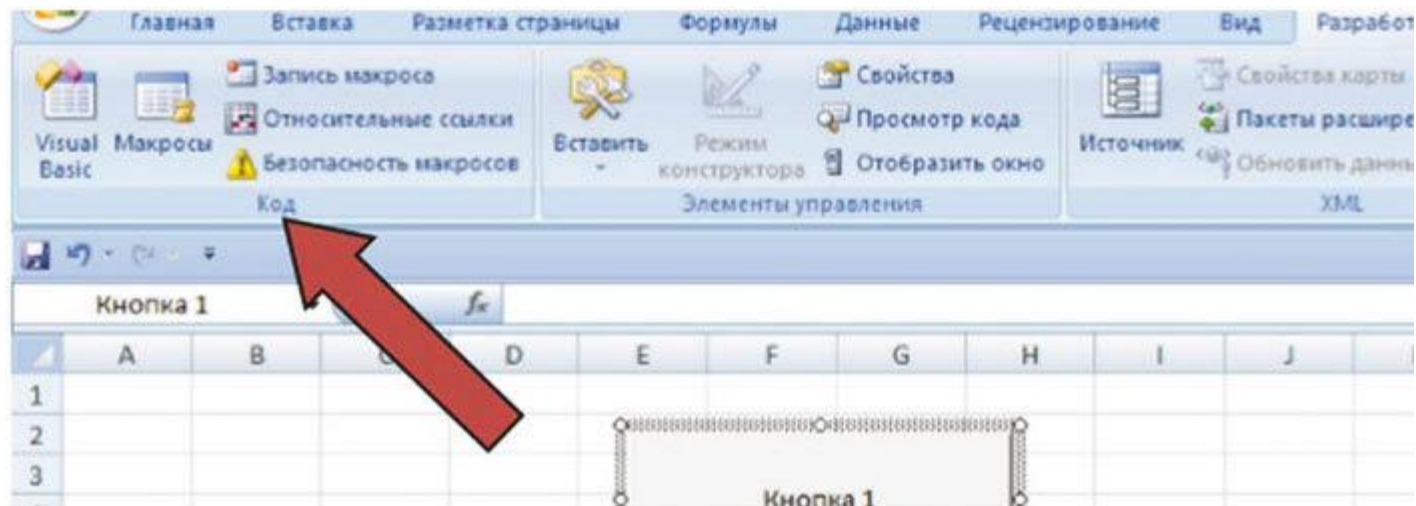


Рис. 34. Инструменты для создания и работы с макросами

Выбрав позицию **Макрос**, можно просмотреть все имеющиеся в наличии макросы и при этом запустить любой из них на выполнение. Выбрав позицию **Запись макроса**, можно создать собственный макрос. После выбора позиции меню **Код** → **Запись макроса** появится окно диалога, в который необходимо занести имя макроса и его описание (рис. 35). Желательно делать описание макроса, т. к. через непродолжительное время трудно будет вспомнить, для каких целей он создавался и какие функции выполняет. Если в этом же окне диалога нажать кнопку «Параметры», то можно выбрать средство вызова макроса: быстрое сочетание клавиш Ctrl+? где вместо знака вопроса следует поставить любую букву. После начала записи макроса на панели инструментов появится новая кнопка (рис. 36), предназначенная для остановки записи команд.

Рис. 35. Форма для задания параметров макроса

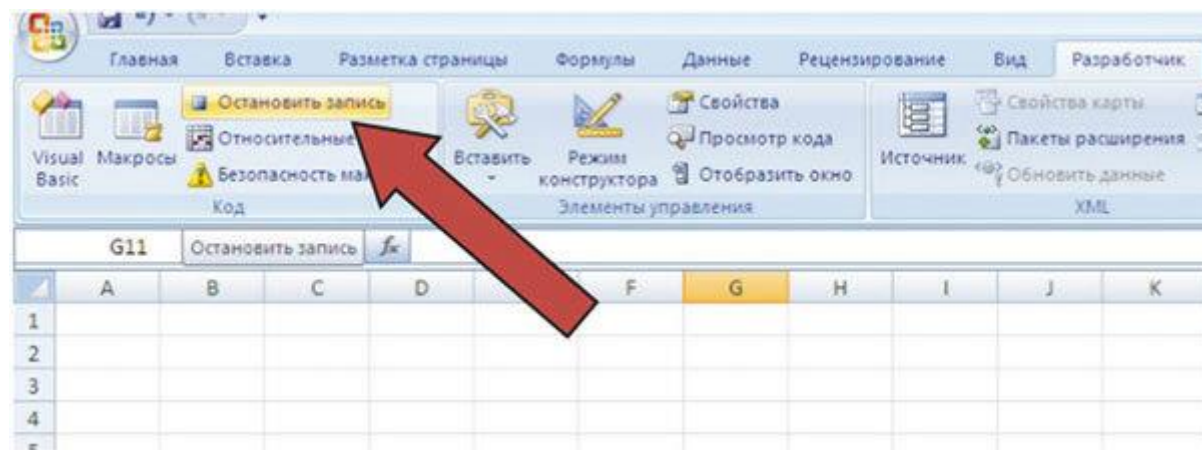


Рис. 36. Кнопка окончания записи макрос

При создании нового макроса следует внимательно отнестись к позиции меню **Относительные ссылки**. От этого зависит правильность работы **Макроса**, т. е. будет ли макрос выполняться относительно первоначально выбранной ячейки или абсолютно независимо от первоначально выделенной ячейки.

5.2. Запуск макроса на выполнение

Существует несколько способов запуска макроса на выполнение:

- Запуск макроса с вкладки «Разработчик».
- Запуск макроса с помощью нажатия сочетания клавиш (рис. 37).
- Запуск макроса при наступлении какого-либо события у элемента управления (рис. 38).
- Запуск макроса при нажатии на вставленный на лист Excel рисунок (рис. 39).
- Запуск макроса при нажатии на вставленный на лист Excel один из элементов построения структуры (рис. 40):

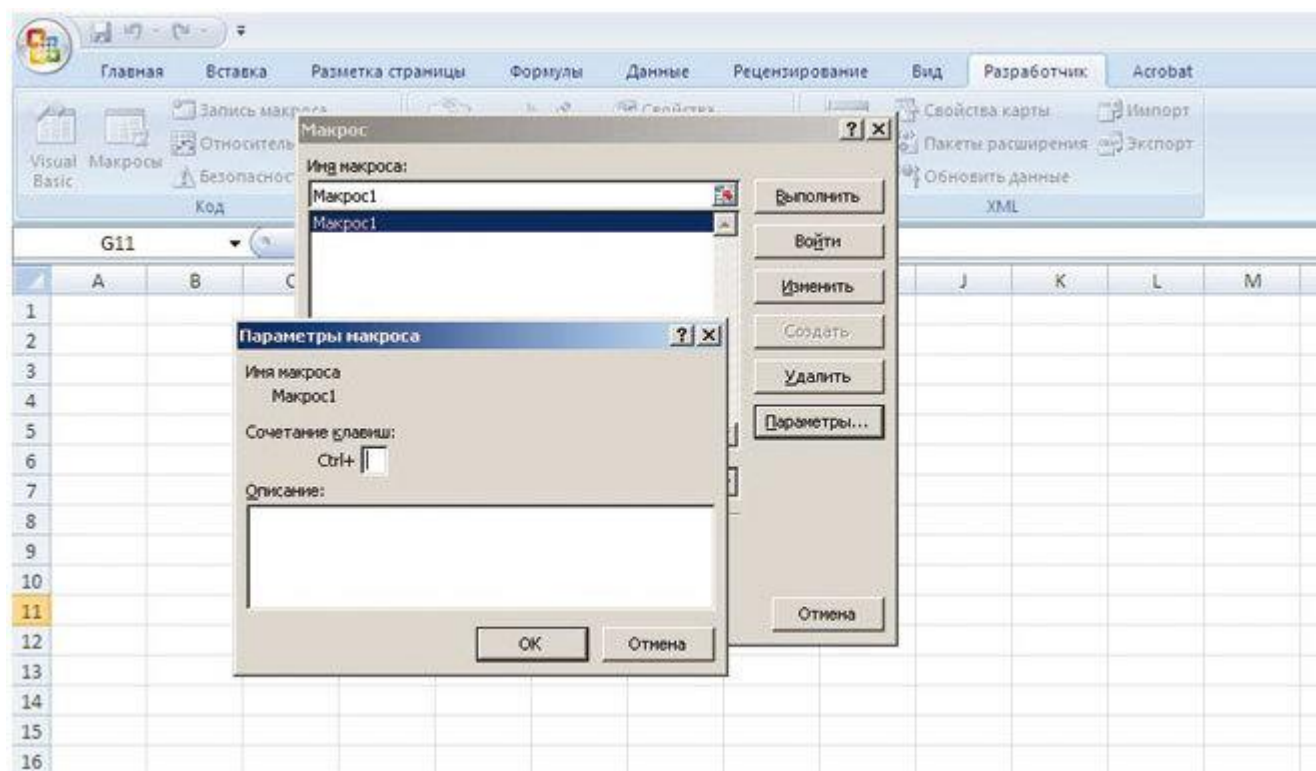


Рис. 37. Запуск макроса с помощью сочетания клавиш

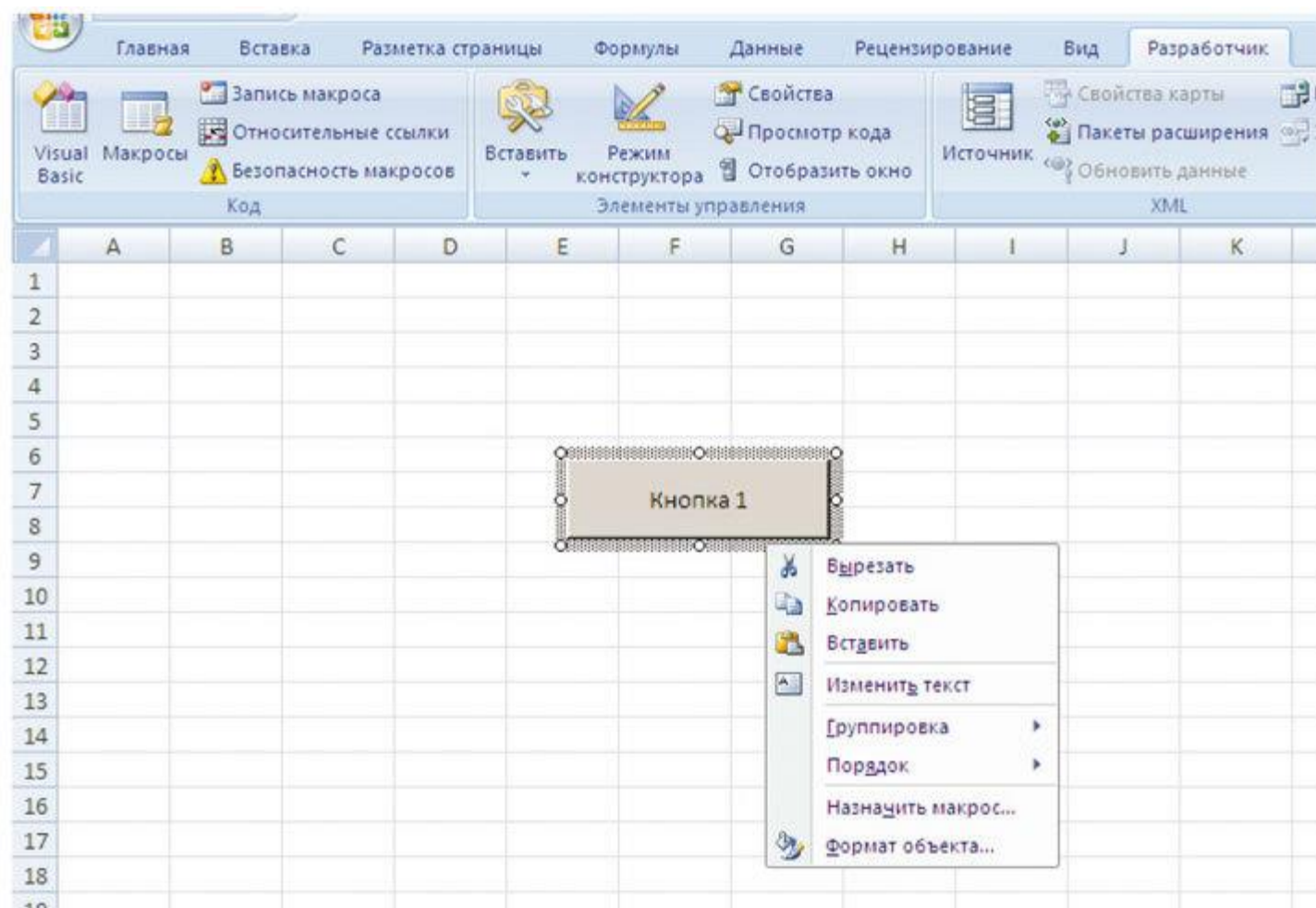


Рис. 38. Запуск макроса при нажатии на элемент управления «Кнопка»

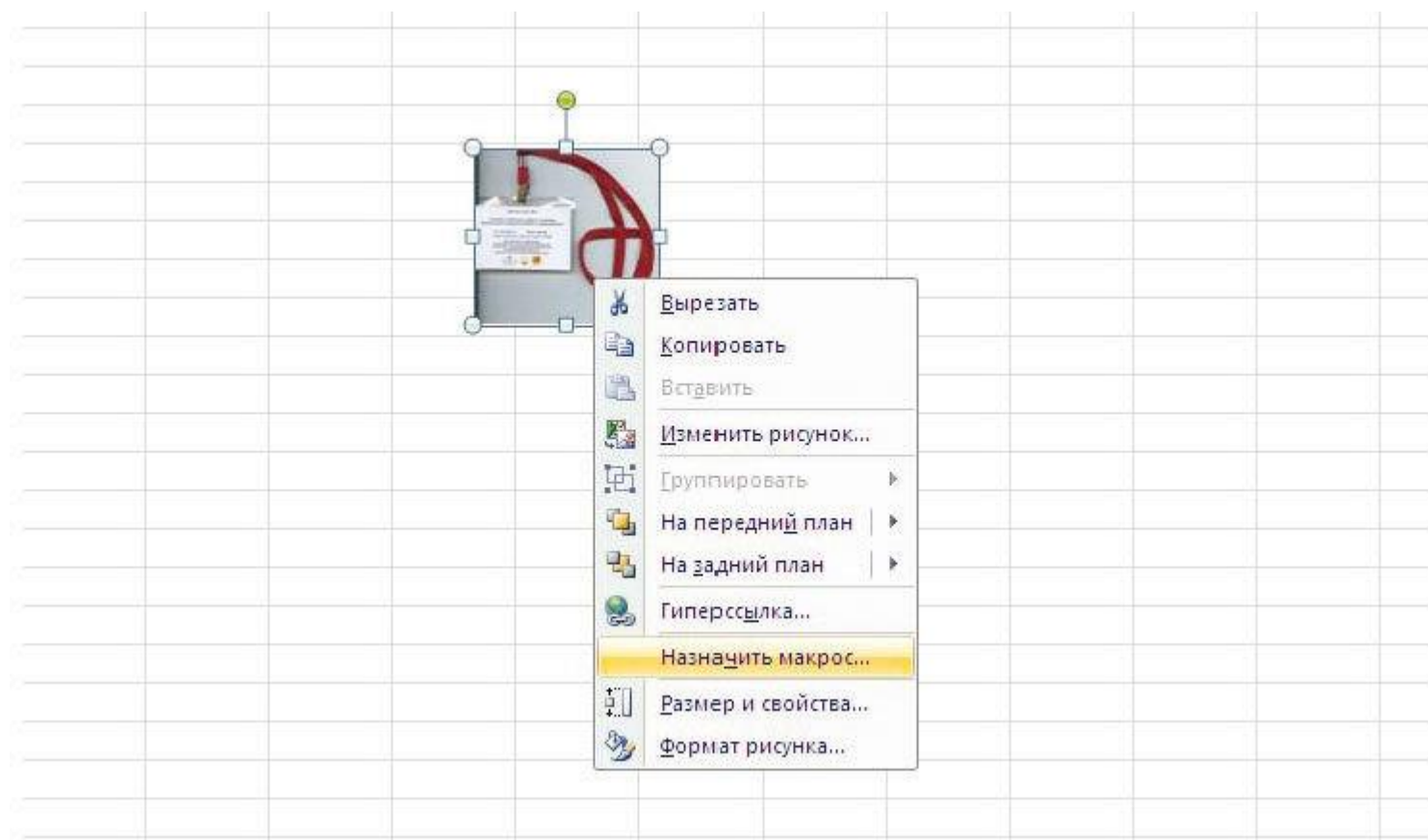


Рис. 39. Запуск макроса при нажатии на рисунок

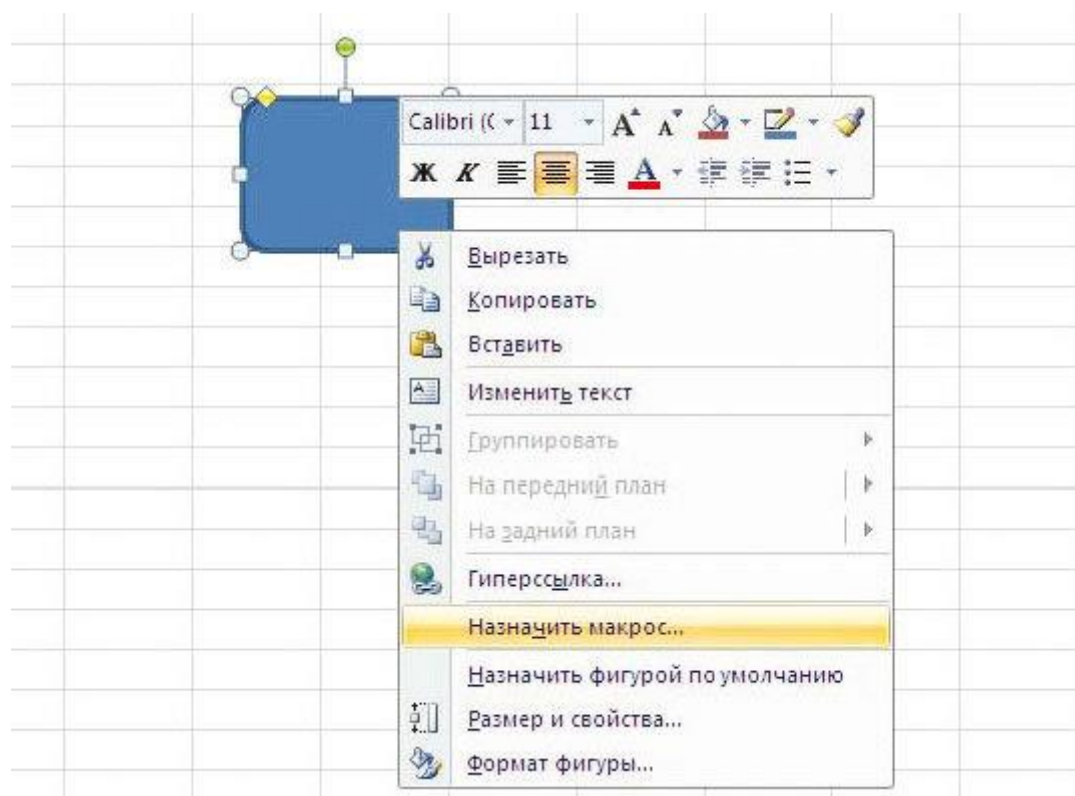


Рис. 40. Запуск макроса при нажатии на графический объект

Важное значение имеет и то, где сохраняются созданные макросы. Макросы могут быть сохранены в личной книге, текущей рабочей книге или в отдельных книгах.

В случае сохранения макроса в личной книге создается скрытая рабочая книга, которая открывается каждый раз, когда осуществляется запуск программы Excel. Макрос, сохраненный в этой книге, становится доступным для других рабочих книг. Это будет глобальный макрос.

В случае сохранения макроса в текущей книге он помещается на отдельном листе под именем **Модуль**, где его всегда можно посмотреть и подправить. Поэтому данный макрос доступен только тогда, когда открыта соответствующая книга.

В связи с этим лучше сохранять макросы в текущей книге (что и сделано по умолчанию) и на каждом листе иметь свою программную кнопку. Но это не является аксиомой, а лишь рекомендацией по использованию.

5.3. Построение макросов в MS Access

Макросы могут существенно облегчить работу с базой данных. При этом процесс работы с базой данных будет более наглядным, гибким.

Как было сказано ранее, макросы – это последовательность макрокоманд, имеющих одно имя. Макрокоманда в MS Access – законченное действие какой-либо функции, например, открытие запроса или формы, закрытие таблицы и так далее. Все функции, которые можно выполнить с базой данных через главное меню, можно реализовать с помощью макросов. Следует иметь в виду и то, что с помощью макросов могут запускаться и модули, написанные пользователем. Тогда, этот мощный инструмент приобретет еще более гибкость.

Создание макросов в MS Access отличается от MS Excel и MS Word. Здесь пользователю не предоставляются различные вольности работы, а определена строгая последовательных действий и настроек. Процесс создания макроса в Access более консервативен, что гарантирует их высокое качество и оптимальность по структуре при достаточной гибкости.

После запуска окна программного продукта, можно увидеть закладку Макросы (рис. 41):



Рис. 41. Позиция «Макрос» на вкладке «Создать»

Если открыть закладку Макросы, то можно увидеть таблицу, строки в которой будут занесены выбранные макрокоманды (рис. 42):

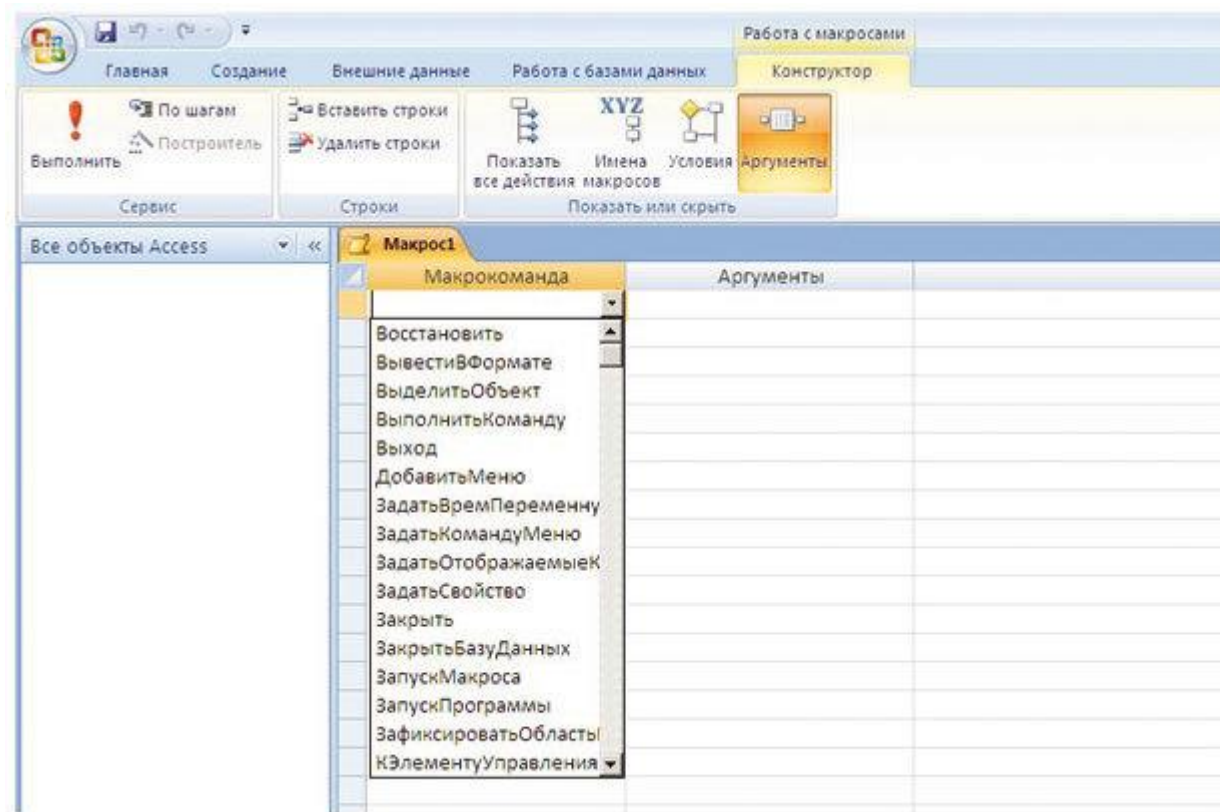


Рис. 42. Список макрокоманд для создания макроса

После выделения какой-либо функции во фрейме комментариев (внизу справа окна) можно прочесть назначение макрокоманд. Кроме этого в данном фрейме можно просмотреть назначение, область действия всех вводимых параметров макрокоманд.

Следует обратить внимание на макрос под именем AutoExec. Не надо его искать в раскрывшемся списке, т. к. под таким именем может быть любой макрос. Макрос AutoExec запускается сразу после открытия базы данных и он очень удобен для открытия главной кнопочной формы, на которой располагаются элементы управления программой.

Хорошим тоном оформления приложения является размещение на диалоговой форме кнопки, закрывающей это приложение. Если макрокоманду на закрытие приложения включить в макрос AutoExec, то используемая программа будет безнадежно потеряна. Во избежание неприятностей не следует включать макрокоманду **Закрыть** в макрос AutoExec.

При создании макросов надо иметь в виду, что под одним именем макроса может располагаться множество макрокоманд. В каждой строке окна макросов можно записать отдельную макрокоманду.

А через позиции Главного меню **Вид** можно в таблице макросов разместить дополнительный столбец для условий. Столбец для ввода условий на макрокоманду можно вывести и с помощью кнопки (рис. 43):

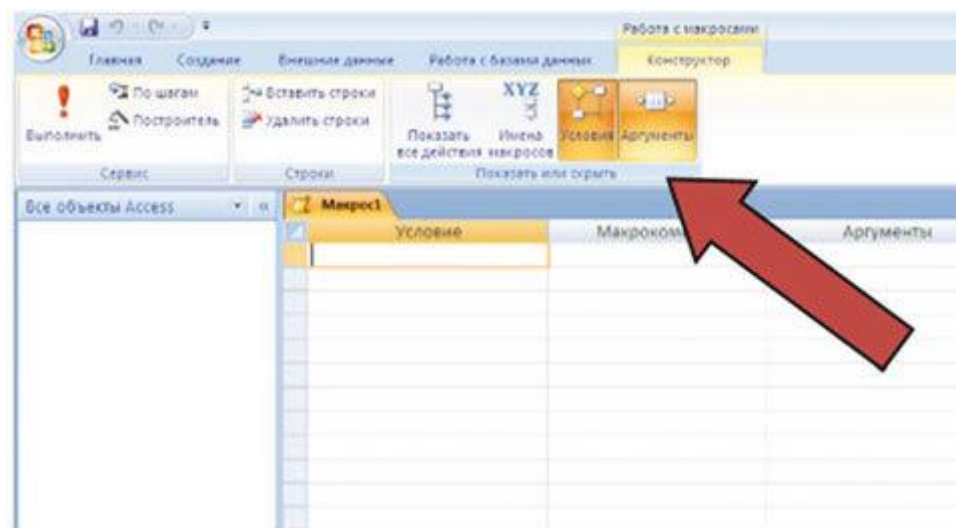


Рис. 43. Команды для добавления в окно диалога по созданию макроса столбца условий и аргументов

6. Использование VBA для автоматизации офисных приложений

Мощным инструментом для автоматизации при работе в офисных приложениях является язык программирования VBA. При записи макросов происходит автоматическое формирование программного кода на языке VBA в виде программной процедуры. Он запускается на выполнение с наступлением различных событий, например: щелчок мыши по кнопке, получение элементом управления фокуса, загрузка приложения и т. д. В MS Access, как было отмечено ранее, программные процедуры и макросы разделены для конечного пользователя.

Надо помнить, что все отражаемое на мониторе компьютера есть результат работы какой-либо программы. В связи с этим понятно, какие громадные возможности несет программный код процедуры, какое мощное удобное, эффективное приложение и удобные средства для расчетов и отображения итогов можно создать, используя в качестве подложки офисные приложения и их собственный инструментальный работы с данными.

7. Практическая реализация учетного приложения в среде MS Word

При разработке приложений с использованием текстового редактора MS Word необходимо рассмотреть все возможности программной реализации:

1. Разработка макросов средствами рекордера.
2. Программная разработка макросов.
3. Разработка и использование шаблонов.
4. Реализация прикладного взаимодействия.

1. Разработка макросов средствами рекордера. В MS Word встроено замечательное средство, которое позволяет создавать программы без каких-либо знаний о программировании. Это средство называется рекордером.

Рекордер – средство для записи макросов. Если макрос – еще одно название для VBA-программы, то рекордер – средство для его автоматического создания.

Принцип работы рекордера больше всего похож на принцип работы магнитофона: мы нажимаем на кнопку – начинается запись тех действий, которые выполняем. Мы нажимаем на вторую кнопку – запись останавливается и можем ее проиграть, т. е. повторно выполнить ту же последовательность действий. В процессе можно воспользоваться паузой, если в этом есть необходимость.

Алгоритм разработки любого макроса, как и любого другого проекта, всегда четко продумывается и состоит из следующей последовательности шагов:

- планирование задачи, которое обеспечивает четкую реализацию при достижении намеченной цели;
- запуск рекордера.

Для этого используется пункт меню ВИД – подпункт Макросы, далее ЗАПИСЬ МАКРОСА (рис. 44).



Рис. 44. Меню ВИД – подпункт Макросы

Далее, необходимо выбрать пункт ЗАПИСЬ МАКРОСА, что должно сопровождаться появлением окна определений, где можно задать имя создаваемого макроса, имя шаблона, в который он будет помещен, и определить метод запуска макроса – по кнопке или по нажатию горячих клавиш в их возможном сочетании. Кроме того, можно внести авторские комментарии, например, определить и описать функциональность данного макроса. Необходимо помнить, что макрос в шаблоне normal.dot действует при загрузке любого внешнего документа. Макрос же документа действует только в конкретном документе (например, договоре или счете-фактуре), (рис. 45)

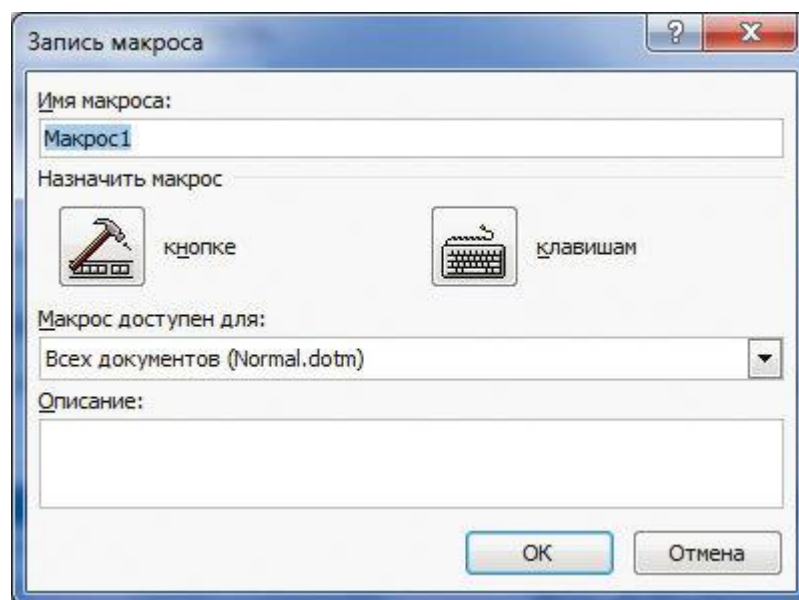


Рис. 45. Окно определений макроса

После определения параметров в окне получаем (рис. 46):

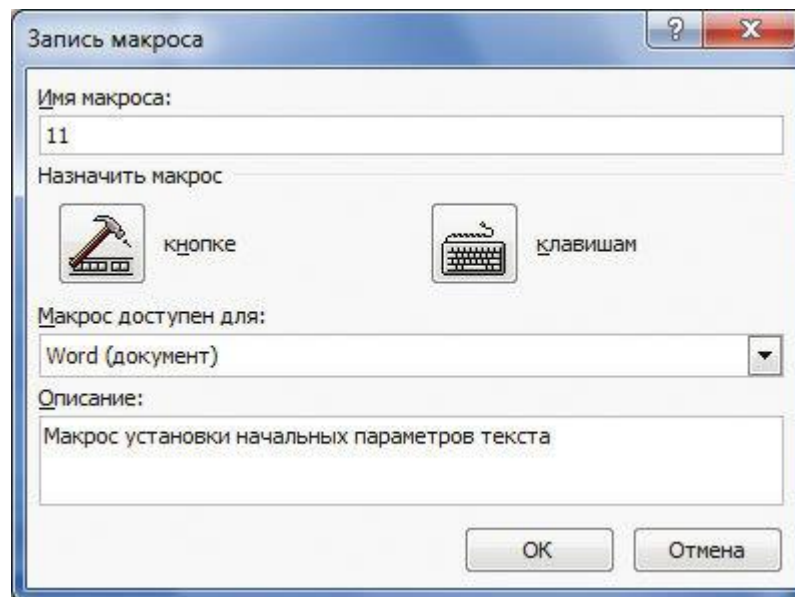


Рис. 46. Описание макроса

Далее начинается запись макроса. Начало записи сопровождается изменением формы курсора – курсор становится стрелкой с кассетой. Начинается запись всех действий разработчика.

После записи процесс останавливается: ВИД → МАКРОСЫ → ОСТАНОВИТЬ ЗАПИСЬ. Если есть необходимость в промежуточной остановке, то нажимают ПАУЗА.

Пример. Существует необходимость в создании макроса, который при создании любого текстового документа устанавливает параметры по умолчанию:

- начальный отступ 1, 25;
- межстрочный интервал 1, 5;
- размер шрифта 14 пт;
- тема шрифта Times New Roman.

Выполнение задачи начинается в соответствии с предложенным алгоритмом:

- запускается рекордер и устанавливаются параметры записи макроса.

Так как макрос создается для работы с любым документом, то местом его размещения надо выбрать шаблон normal.dot. Предположим, что ему дано имя NF (начальное форматирование). Тогда краткий комментарий с «напоминалкой» элемента управления запуска будет иметь следующий вид:

«Производим запуск макроса по кнопке (рис. 47):

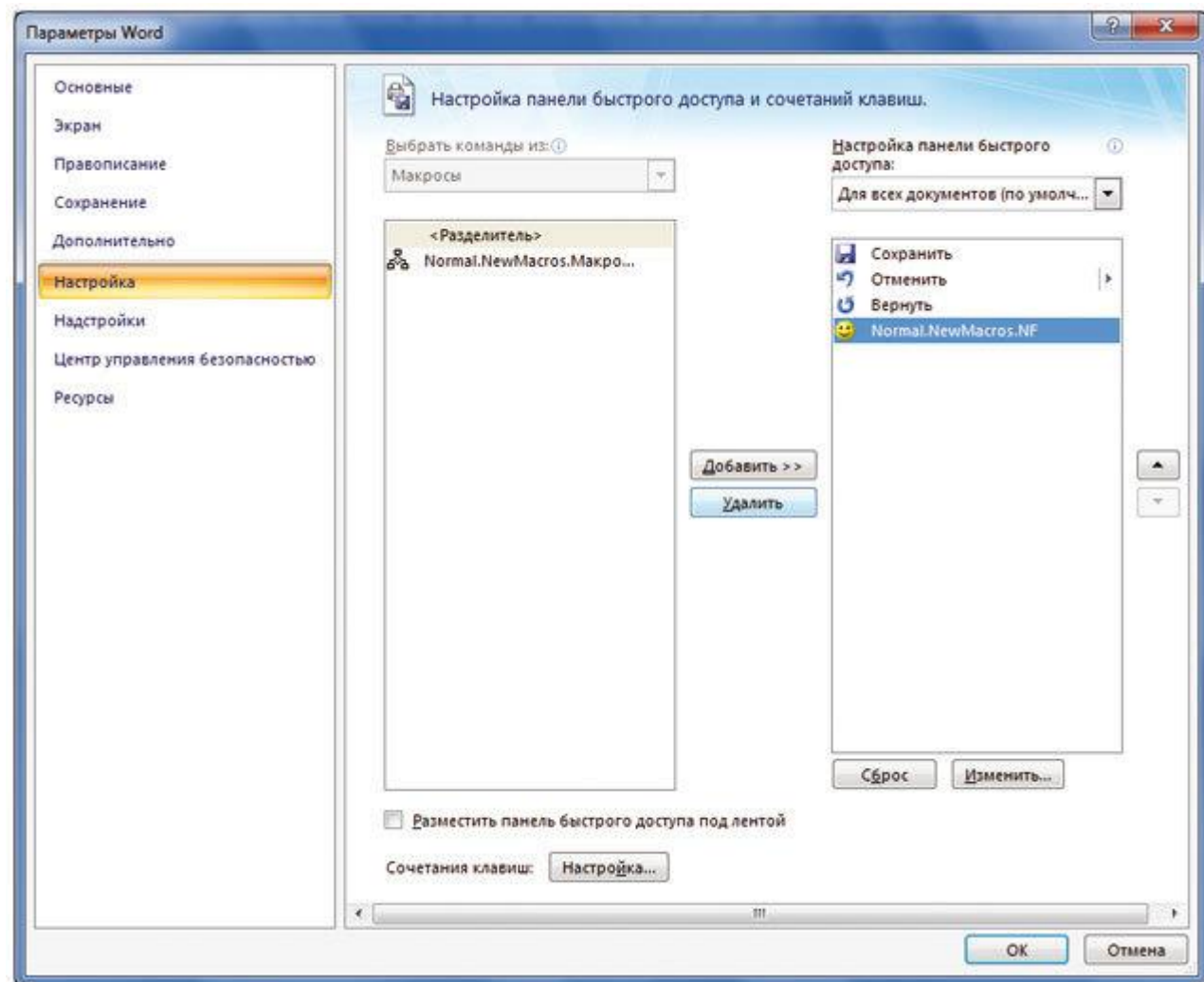


Рис. 47. Настройка панели параметров

В появившемся окне выбираем необходимую панель размещения кнопки (например, панель быстрого доступа) и добавляем туда свой макрос. Далее кнопкой изменить меняем логотип кнопки (рис. 48):

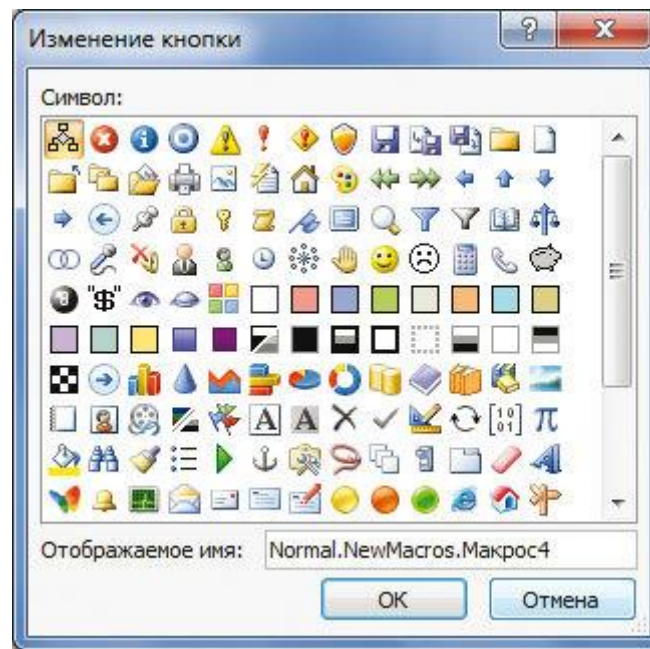


Рис. 48. Изменение логотипа

Вписываем комментарии и сохраняем наш макрос. В выбранной панели появляется заданная кнопка (кнопка номер 4) (рис. 49):

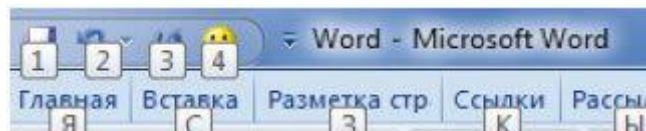


Рис. 49. Сохранение макроса

Макрос готов можно начать работу с ним. Теперь в случае изменения начального запуска создания текстового документа, его всегда можно восстановить.

Если со временем макрос станет не нужен или его необходимо отредактировать, то это можно сделать через пункт Макросы – меню Вид (рис. 50):

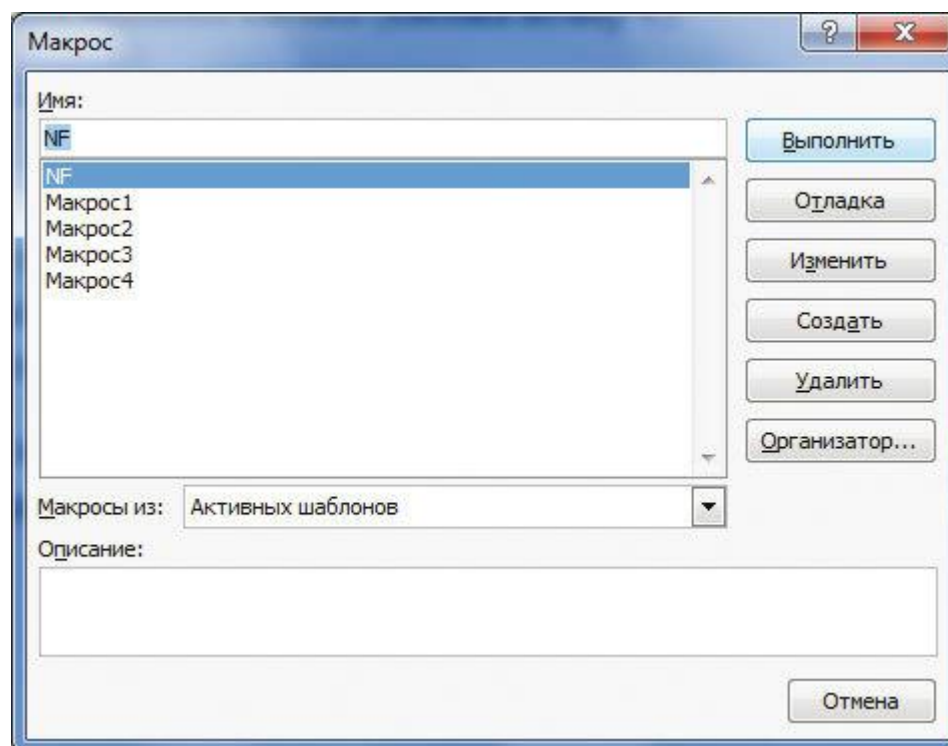


Рис. 50. Пункт Макросы – меню Вид

Редактирование возможно также в режиме отладки – режиме VBA (рис. 51):

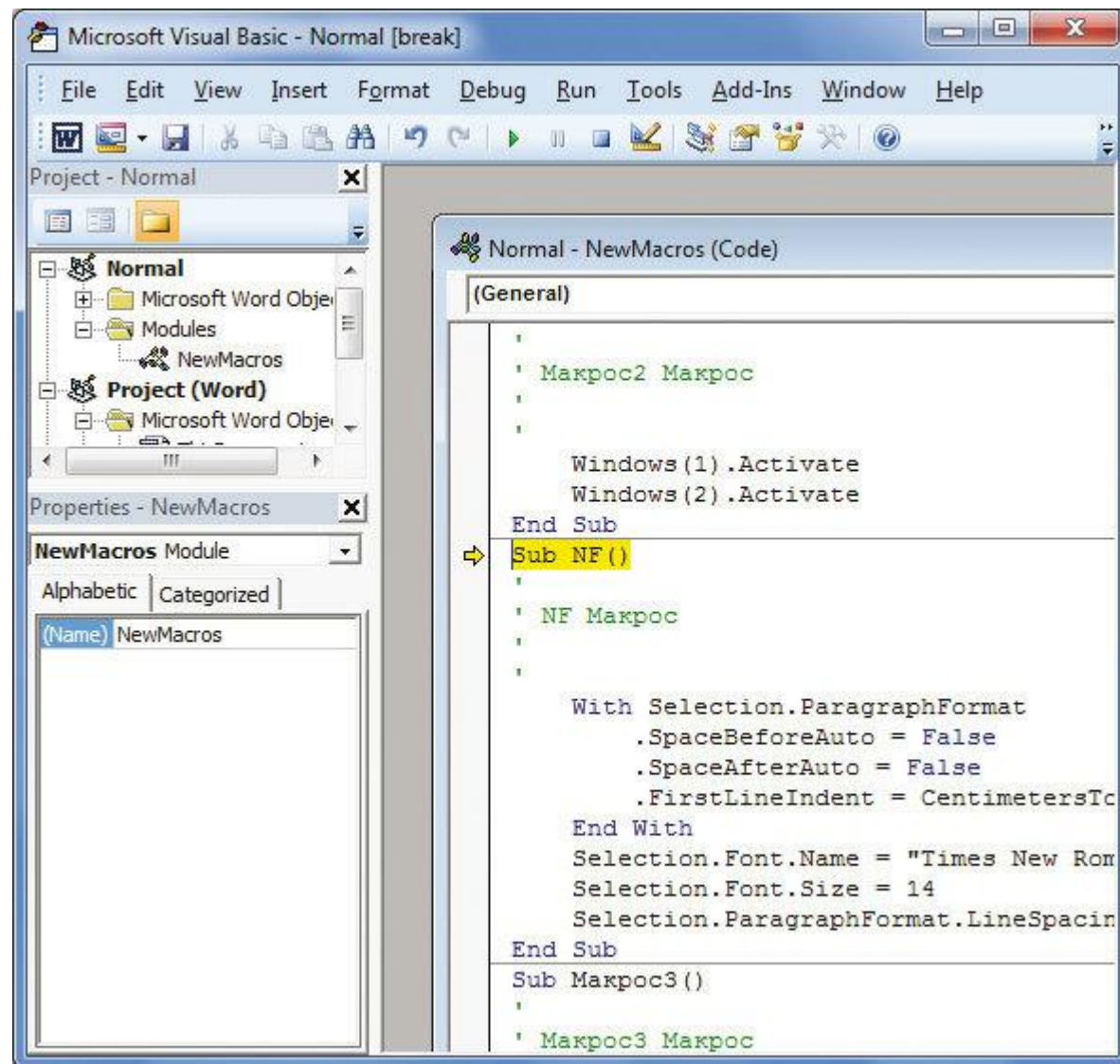


Рис. 51. Редактирование в режиме VBA

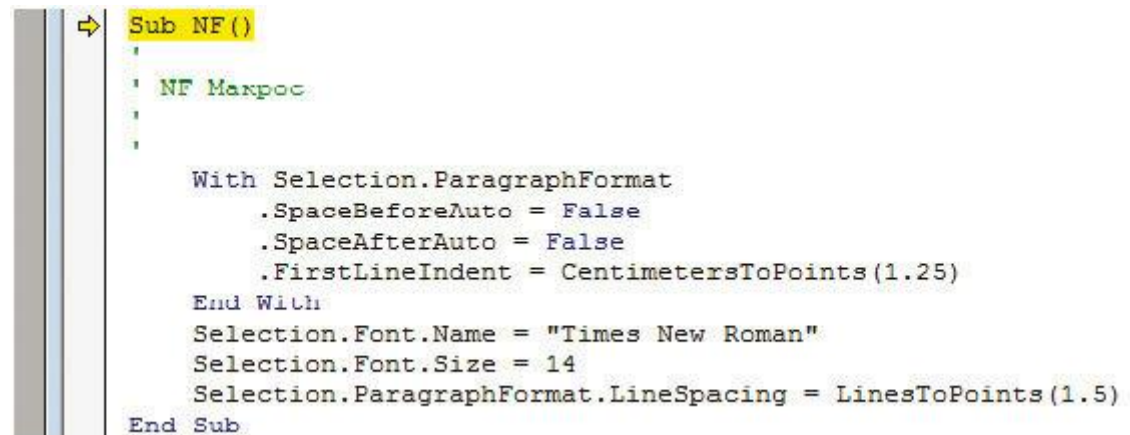
2. Программная разработка макросов. Использование средств VBA всегда может привести к максимально гибким результатам. При этом необходимо ориентироваться в синтаксисе языка, либо уметь пользоваться подсказками.

Вызов окна справки производится из редактора Visual Basic после нажатия кнопки <F1>. Второй вариант – воспользоваться кнопкой **Справка** на панели инструментов **Standard**. В результате откроется окно (рис. 52):



Рис. 52. Вызов окна справки

Для примера подредактируем уже созданный нами макрос начального редактирования. Предположим, что нам необходим разный размер шрифтов при начальной загрузке. Наш исходный код выглядит следующим образом (рис. 53):



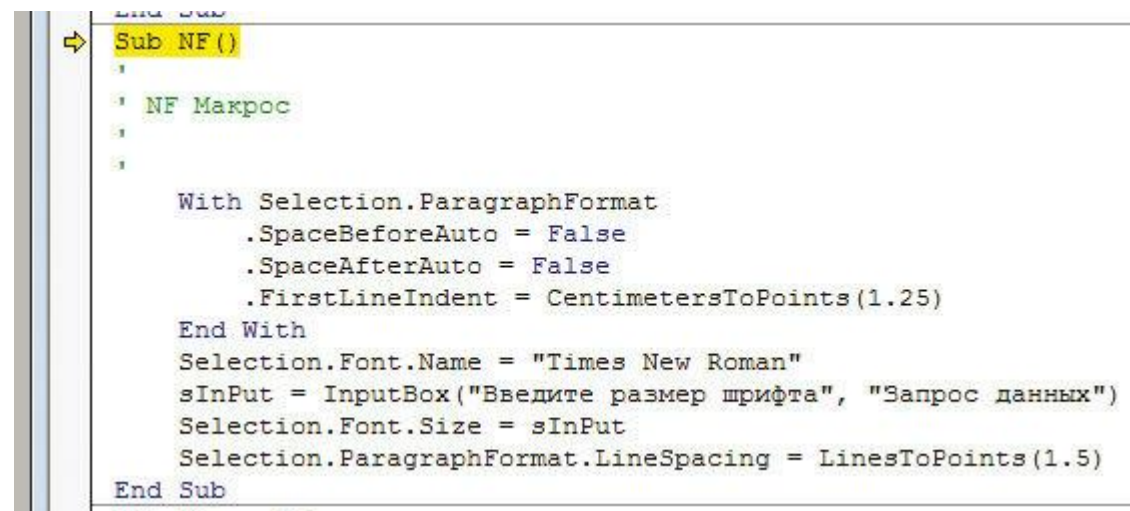
```

Sub NF ()
'
' NF Макрос
'
With Selection.ParagraphFormat
.SpaceBeforeAuto = False
.SpaceAfterAuto = False
.FirstLineIndent = CentimetersToPoints(1.25)
End With
Selection.Font.Name = "Times New Roman"
Selection.Font.Size = 14
Selection.ParagraphFormat.LineSpacing = LinesToPoints(1.5)
End Sub

```

Рис. 53. Вид исходного кода

Добавляем диалоговое окно и ввод данных о шрифте через переменную. В результате код изменится следующим образом (рис. 54):



```

Sub NF ()
'
' NF Макрос
'
With Selection.ParagraphFormat
.SpaceBeforeAuto = False
.SpaceAfterAuto = False
.FirstLineIndent = CentimetersToPoints(1.25)
End With
Selection.Font.Name = "Times New Roman"
sInPut = InputBox("Введите размер шрифта", "Запрос данных")
Selection.Font.Size = sInPut
Selection.ParagraphFormat.LineSpacing = LinesToPoints(1.5)
End Sub

```

Рис. 54. Измененный вид кода

Добавляется запрос ввода данных:

sInPut = InputBox(«Введите размер шрифта», «Запрос данных») и определение размера шрифта через переменную:
Selection. Font. Size = sInPut.

В результате до определения начального форматирования макрос запрашивает размер шрифта (рис. 55):

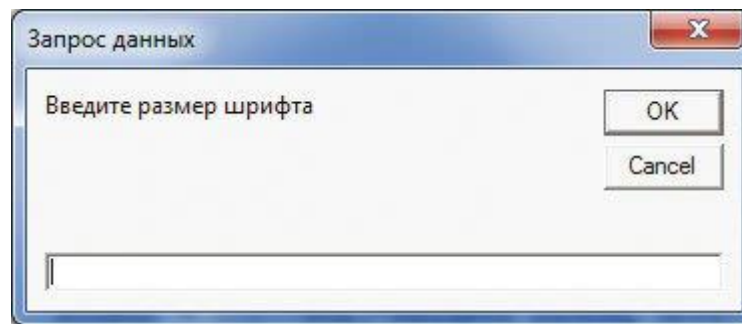


Рис. 55. Размер шрифта

3. Разработка и использование шаблонов. Автоматизацию работы с шаблоном лучше всего рассмотреть на примере взаимодействия с СПС Консультант+. Предоставляемые шаблону данные документа легко экспортируются в Word.

В СПС бланки и формы под текстовый редактор Word предоставляет раздел «Деловые бумаги». Из раздела возьмем бланк договора на предоставление услуг (рис. 56):

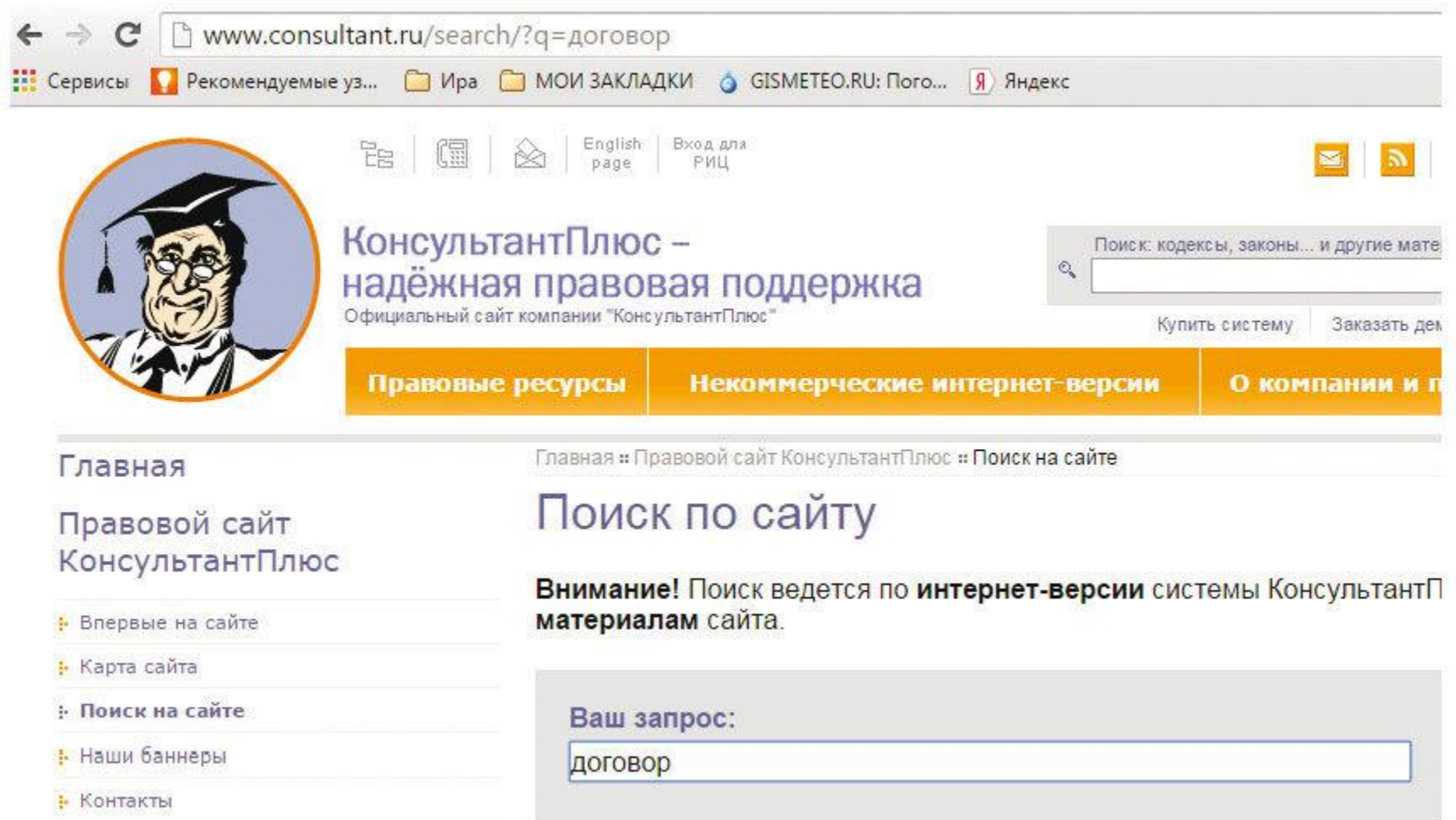


Рис. 56. Поиск по сайту СПС Консультант+

Далее, конвертируем его в MSWord (рис. 57):

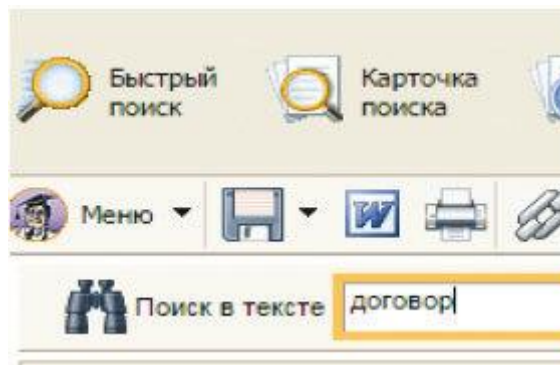


Рис. 57. Конвертация документа в MSWord

В итоге получаем документ, загруженный в текстовый редактор. В этом документе расставляем закладки в тех местах, где располагаются параметры заполнения документа:

1. Номер договора.
2. Дата заключения.
3. Имя участника договора.
4. Название организации-участника договора. Каждой закладке дадим имя (nd, dz, iu, po) и пометим их на рисунке 58:

ДОГОВОР № «»

г. Уфа

«» именуемый в дальнейшем Заказчик в лице «», действующего на основании «» с одной стороны ООО «Планета», именуемое в дальнейшем Исполнитель, в лице директора Симонова И. Т., действующего на основании

Рис. 58. Форма договора

При вставке закладок используем пункт меню ВСТАВКА и меню ЗАКЛАДКА (рис. 59):

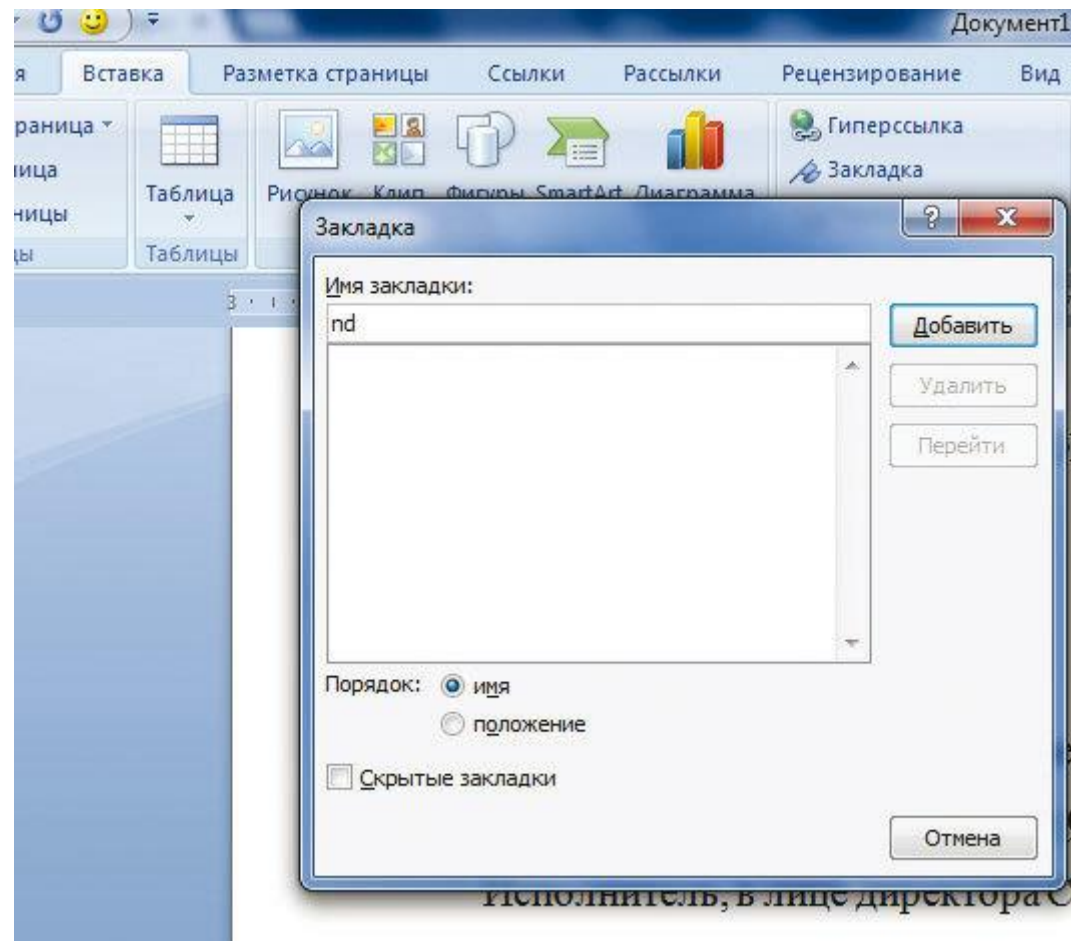


Рис. 59. Имя закладок

Полученный таким образом документ, сохраняем как шаблон, например, с именем dog. dot. Нужный шаблон готов и его можно использовать в дальнейшей работе.

Автоматизацию работы с данным шаблоном можно рассмотреть в следующем контексте:

- в организации существует необходимость в быстром оформлении договоров;
- необходимо написать модуль автоматизации этой работы на основании шаблона, исходной формы и макроса с кнопкой.

При этом шаги алгоритма выглядят следующим образом при создании исходной формы:

Так как регулируемые являются 4 основных параметра по закладкам, наша форма будет включать 4 текстовых поля. Откроем окно редактора Visual Basic для MS Word и щелкнем правой кнопкой мыши по проекту Normal в Project Explorer, а затем выберем в контекстном меню **Insert^UserForm**(рис. 60).

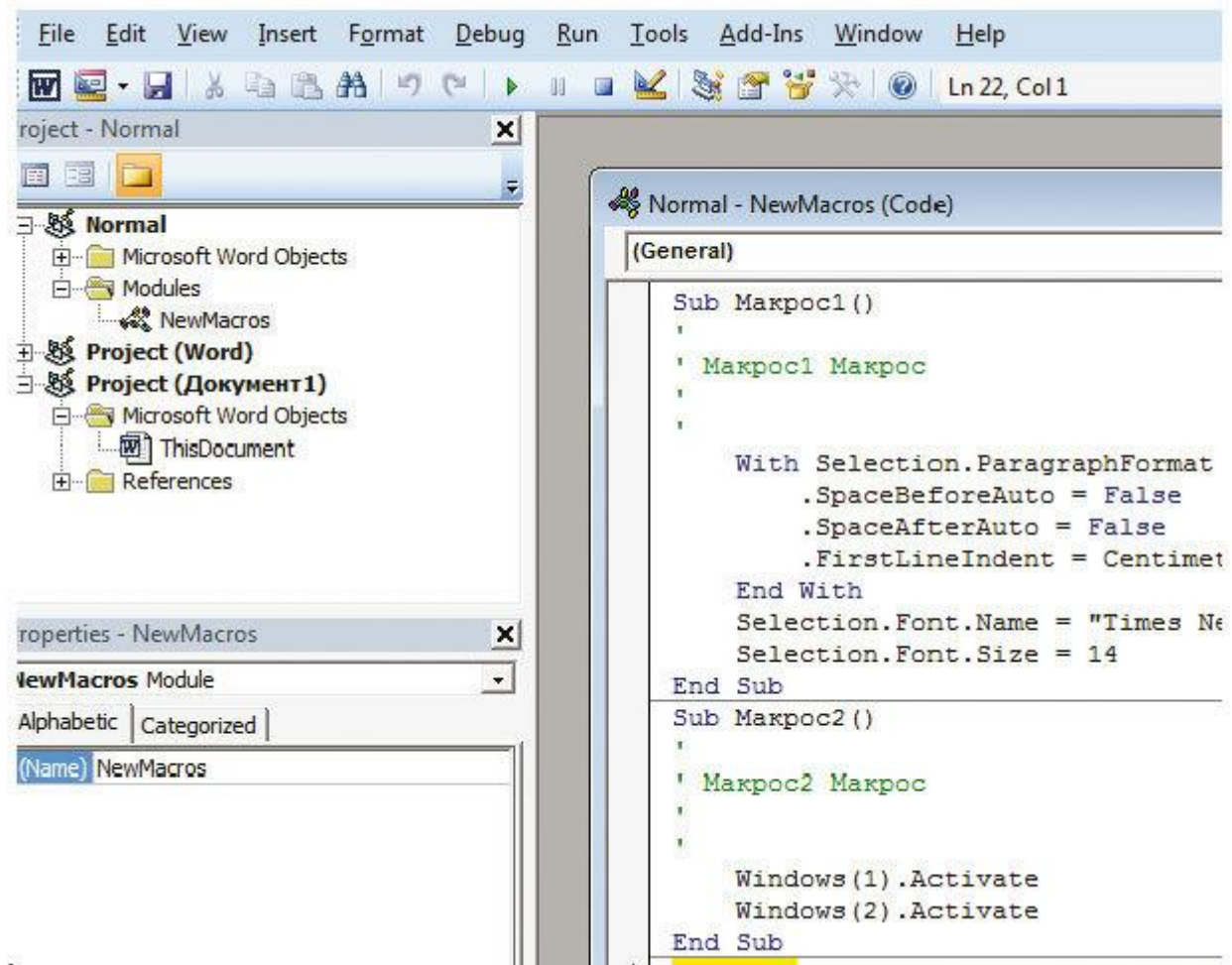


Рис. 60. Процесс создания исходной формы

В дизайнера форм сконструируем форму (рис. 61):

данные договора

номер

дата

имя

организация

отмена

ввести данные и сформир

Рис. 61. Данные по договорам

Определим свойства всех элементов управления. Установим значение для свойства Caption для формы как «Данные договора». Для кнопки cmdDog установим True для значения свойства Default, а для кнопки cmdCancel установим True для значения свойства Cancel. Для свойства (Name) самой формы введем значение Dog.

В стандартном модуле NewMacros проекта Normal создадим новую процедуру с именем Dog(). Код ее может быть таким (рис. 62):

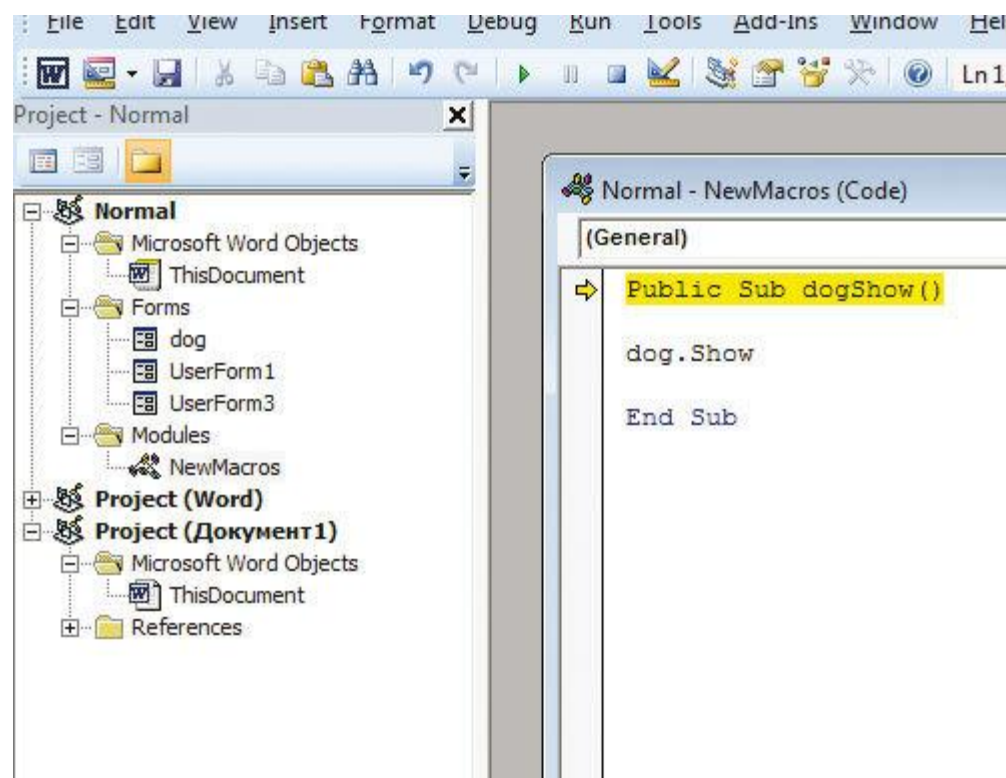


Рис. 62. Код новой формы

Убедимся, что при запуске макроса наша форма открывается. Для создания кнопки присвоим созданному макросу просмотра формы кнопку на произвольной панели. Далее воспользуемся кнопками НАСТРОЙКА в ПАРАМЕТРАХ WORD – МАКРОСЫ – ИЗМЕНИТЬ (если хотим придать нашему макросу другую пиктограмму) (рис. 63, рис. 64):

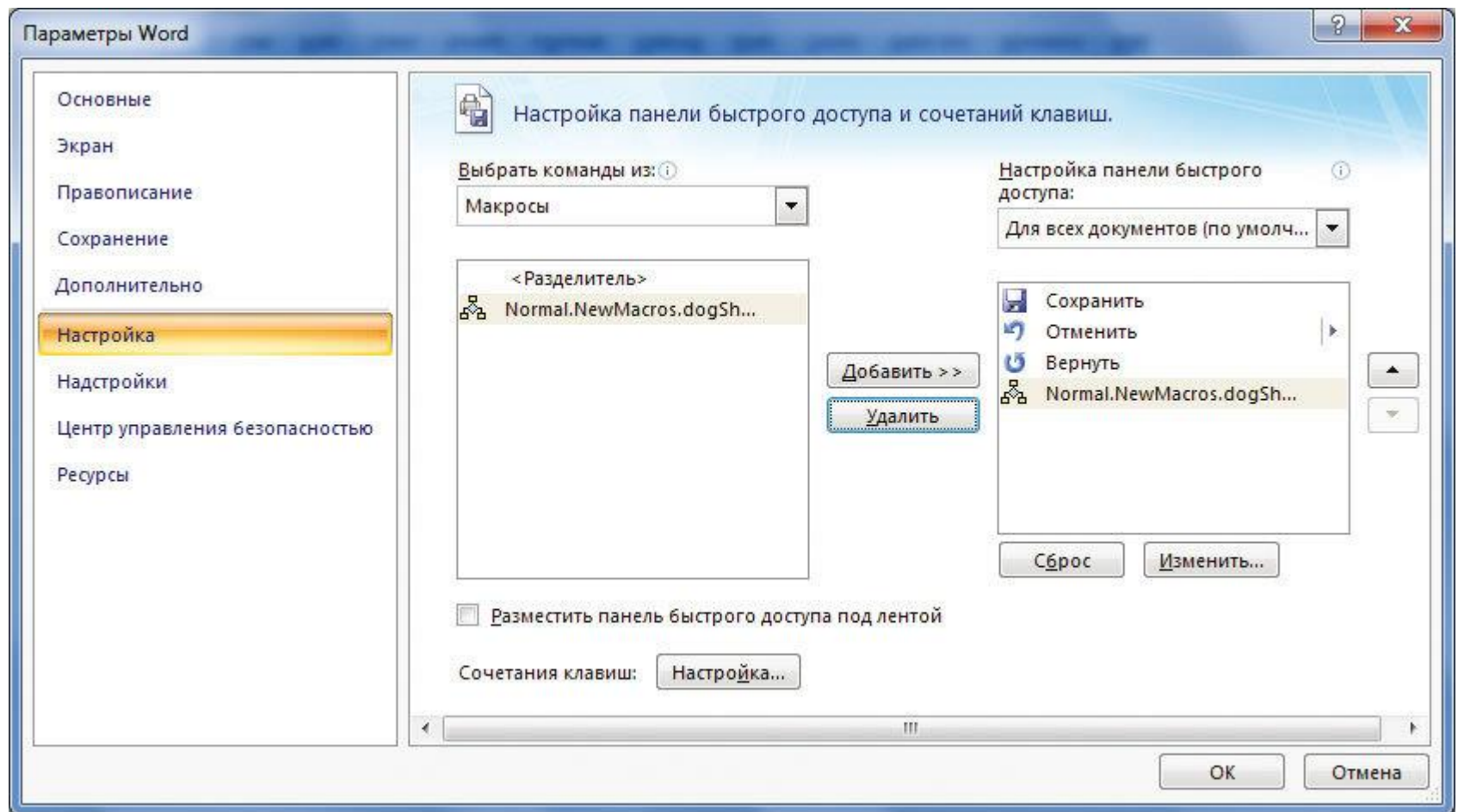


Рис. 63. Настройка макроса

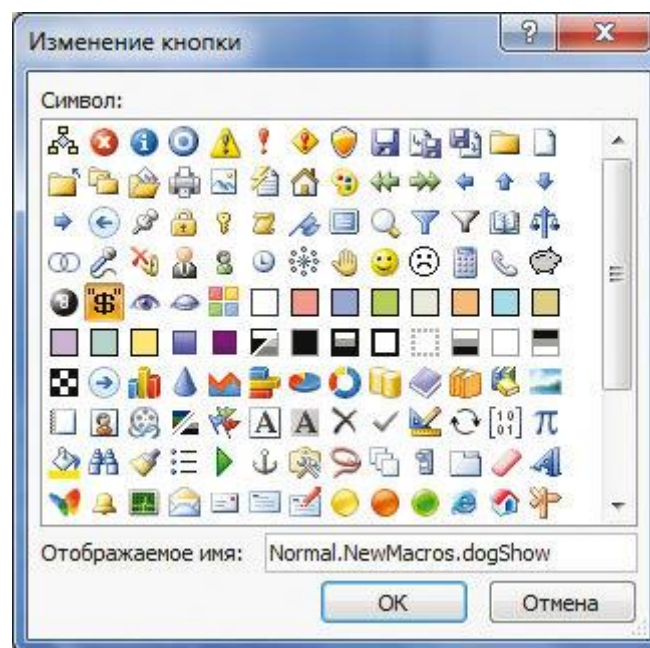


Рис. 64. Изменение кнопки макроса

Теперь необходимо заранее подготовленный и представленный выше шаблон прикрепить к созданным элементам формы.

Оформление кнопки «Ввести данные и сформировать договор» осуществляется следующим образом. В созданной форме активируем кнопки с помощью разработанных процедур, так для кнопок «Отмена» и «Ввести данные и сформировать договор» создаем код (рис. 65):

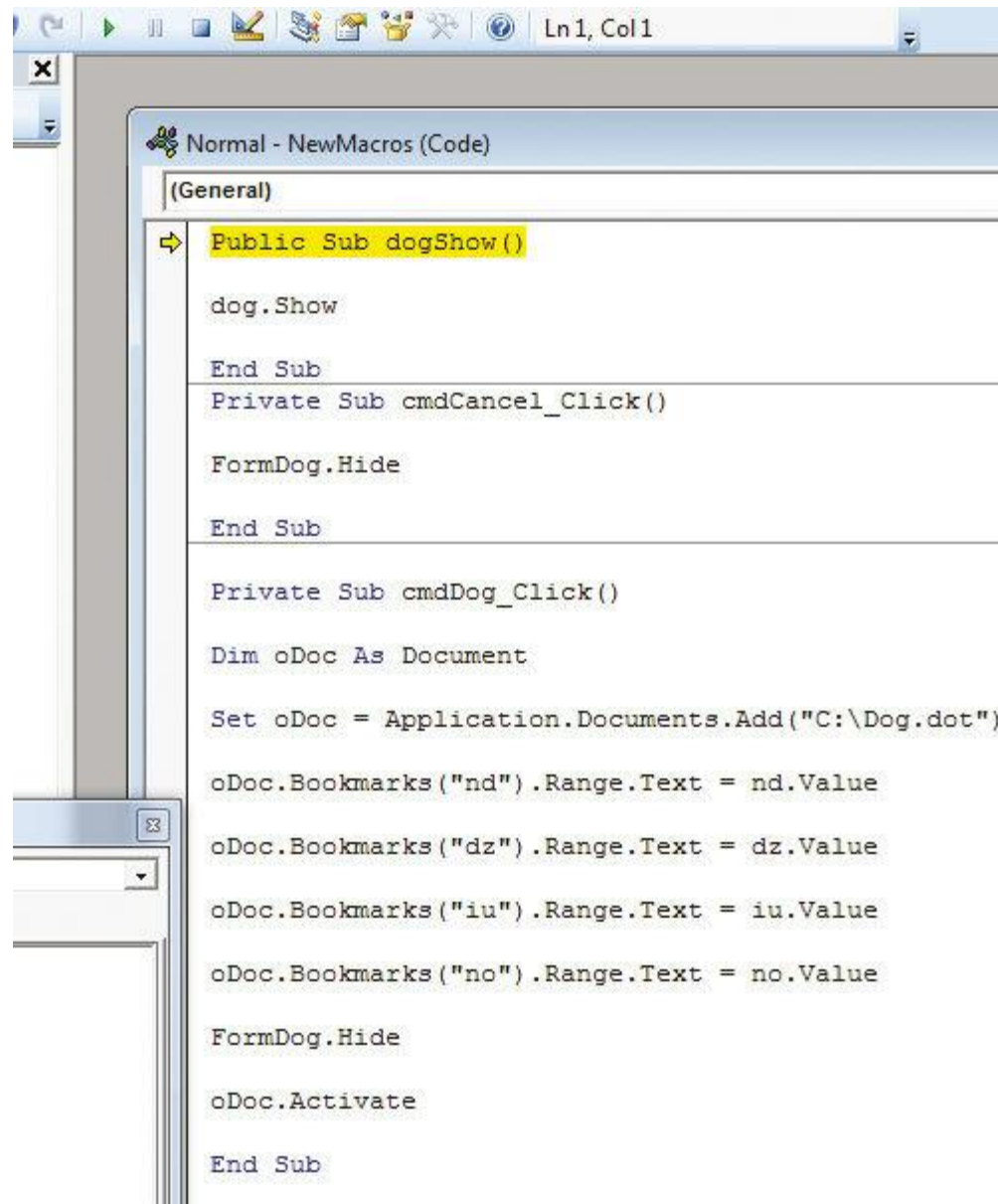


Рис. 65. Активация кнопок

Далее, проверим работоспособность полученного результата (рис. 66):

72837198

номер

12.04.2015

дата

Симонов И.Б.

имя

"Поспелово"

организация

отмена

ввести данные и
сформировать
договор

| ДОГОВОР № «72837198»

Г.Уфа 12.04.2015

«"Поспелово"» именуемый в дальнейшем Заказчик, в лице «Симоно

Рис. 66. Вид созданного договора

4. Реализация прикладного взаимодействия. Вызов специальной вставки осуществляется из пункта «Вставить панели» – ГЛАВНАЯ. Она позволяет связать несколько документов в единое целое, при этом автоматизация работы может быть представлена автоматизацией заполнения (рис. 67):

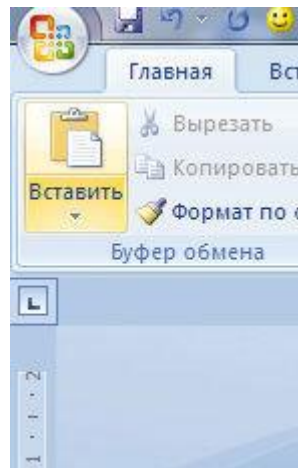


Рис. 67. Пункт «Вставить панели»

Окно специальной вставки позволяет связывать разные документы (рис. 68):

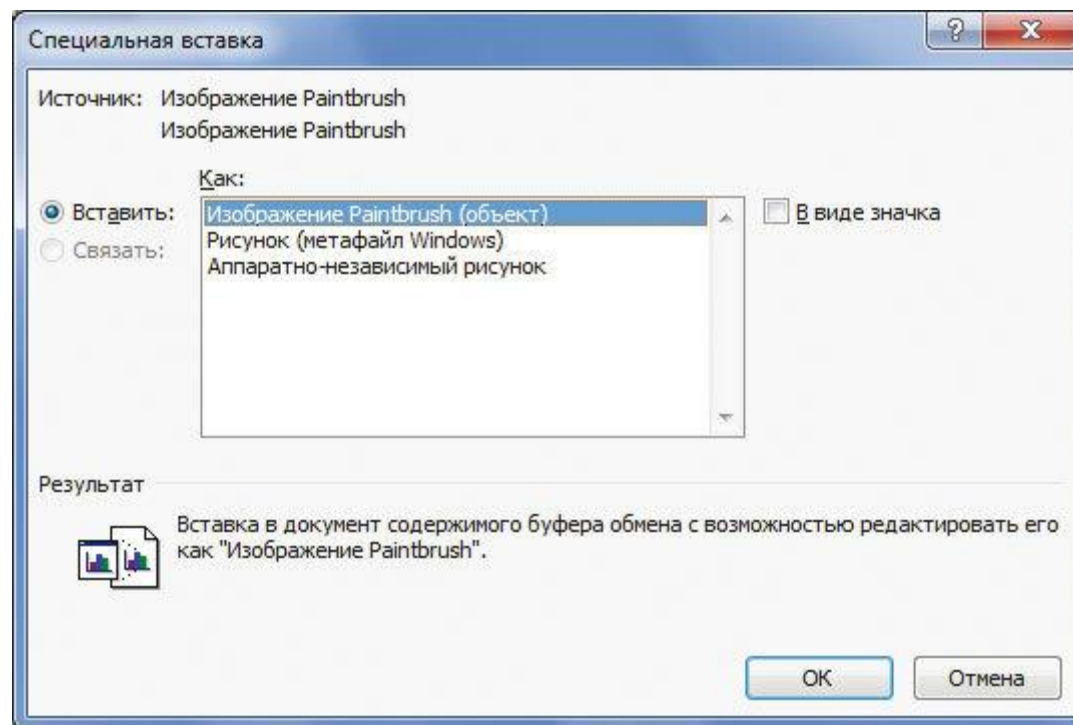


Рис. 68. Окно специальной вставки

Алгоритм работы достаточно прост. Необходимый компонент копируется в буфер обмена. Затем, вставляется в нужное место документа по специальной вставке. При этом обязательным является активизация кнопки связать.

Например, имеем таблицу с исходными данными договора, бланк договора, бланк счета-фактуры и бланк акта выполненных работ. Заполняем таблицу исходных данных (рис. 69):

| | |
|---------------|------------|
| исполнитель | Иванов |
| подразделение | Отдел1 |
| дата | 22.04.2015 |

Рис. 69. Таблица исходных данных

Каждое из полей последовательно помещаем в буфер обмена и вставляем в необходимое место бланка документа (рис. 70):

| | |
|-------------------------|--------------|
| Директор предприятия | Сидоров И.К. |
| | |
| Отв. Исполнитель Иванов | |
| Отдел1 | |
| 22.04.2015 | |

Рис. 70. Вставка в бланк документа

При использовании специальной вставки обязательно выбираем тип вставляемого фрагмента и активируем функцию СВЯЗАТЬ (рис. 71):

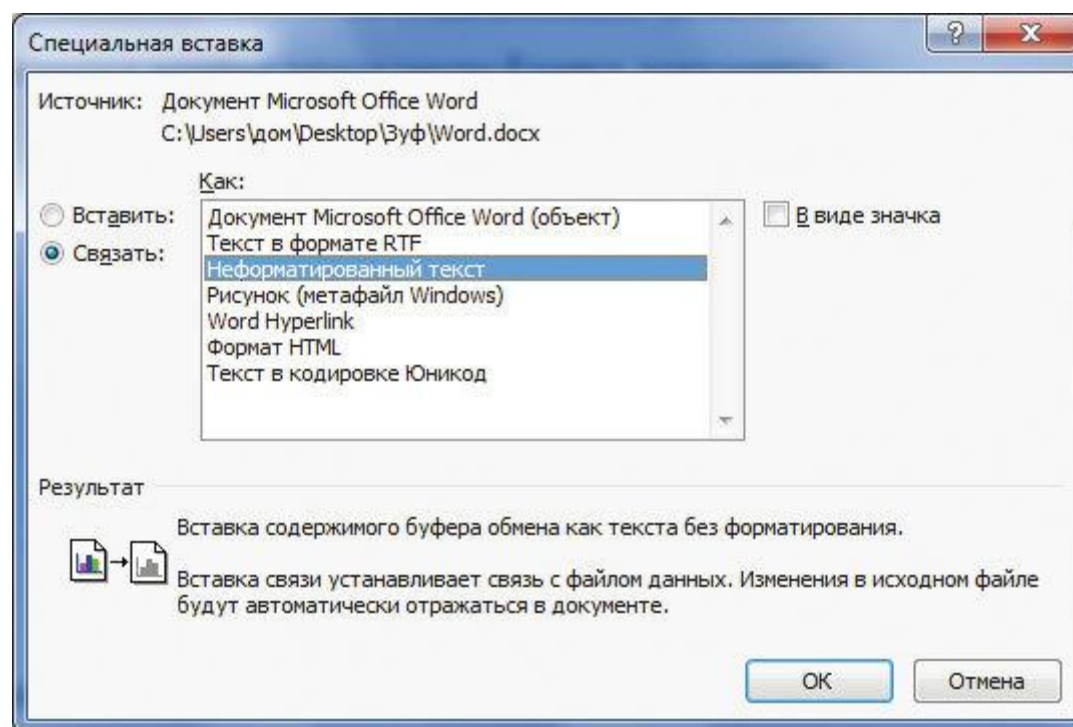


Рис. 71. Активизация функции СВЯЗАТЬ

Результат состоит в том, что изменение исходных данных автоматически влечет изменение параметров в выходных документах (рис. 72, рис. 73):

| | |
|---------------|------------|
| исполнитель | Васин М.Б. |
| подразделение | Отдел2 |
| дата | 22.04.2015 |

Рис. 72. Изменение параметров документа

Директор предприятия | Сидоров И.К.

Отв. Исполнитель Васин М.Б.

Отдел2

22.04.2015

Рис. 73. Изменение параметров в выходных документах

8. Практическая реализация учетного приложения в среде MS Excel

Рассмотрим на примере регистрации участников научно-практической конференции разработку учетного приложения. При подготовке конференции проводится большая работа:

- 1) подготовка информационного письма с последующей рассылкой будущим участникам конференции;
- 2) регистрация участников конференции;
- 3) сбор статей для публикации в сборнике материалов конференции;
- 4) подготовка программы конференции;
- 5) организация оплаты за участие в конференции;
- 6) подготовка бейджиков участникам конференции;
- 7) издание сборника материалов конференции;
- 8) почтовая рассылка материалов конференции.

При разработке учетного приложения в нашем примере будут автоматизированы следующие работы:

- 1) регистрация участников конференции;
- 2) регистрация представленных статей для сборника материалов конференции;
- 3) подготовка бейджиков участникам конференции;
- 4) подготовка наклеек на конверты для почтовой рассылки материалов конференции;
- 5) автоматизированное формирование платежного поручения для оплаты участия в работе конференции.

8.1. Выбор формуляра-образца бланка для учетного приложения

Учетное приложение – это, прежде всего, база данных в том или ином виде, где регистрируется вновь поступившая информация. В зависимости от приложения MS Office эта база может быть реализована в виде от простой таблицы-регистра с возможностью сортировки строк и элементарного поиска информации до создания полноценной базы СУБД. Если база представляет собой плоскую таблицу, то говорить об отдельных сущностях представляется достаточно сложным. Но и в этом случае мы должны определиться, какие поля (столбцы таблицы) необходимы для хранения информации.

Отчасти это можно узнать по бланкам учетного приложения, которые будут печататься по данным из таблицы – базы. В нашем случае это платежное поручение для оплаты услуги через банковские структуры (рис. 74):

Microsoft Excel - 232.xls

Файл Правка Вид Вставка Формат Сервис Данные Справка

Введите вопрос

100%

Arial Cyr 10

R55C68

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

1 Платежное поручение

2 Поступ. в банк плат. Списано со сч. плат.

3 0401060

4

5

6

7 ПЛАТЕЖНОЕ ПОРУЧЕНИЕ №

8 Дата Вид платежа

9

10

11 Сумма прописью Пятьсот рублей 00 копеек

12

13 ИИН КПП Сумма 500.00

14

15

16

17

18 Платательщик

19

20

21 Банк плательщика

22

23 Сбербанк России г. Москва

24

25 Банк получателя

26

27 ИИН 7729406900 КПП 772901001 Сч. № 40702810038180102376

28

29 ООО "Бухсофт.ру" г. Москва

30

31 Вернадское ОСБ 7970 г. Москва

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

Готово

Пуск

Разработка учётных п...

Документ1 - Microsoft ...

C:\Documents and Settin...

СКАЧАТЬ ОБРАЗЕЦ БПА...

Microsoft Excel - 232...

14:55

Рис. 74. Бланк скаченного платежного поручения

Поиск формы платежного поручения в формате MS Excel можно найти как в справочно-правовой системе «КонсультантПлюс», так и в сети Интернет (рис. 75). Существует множество сайтов, где можно найти такую информацию:

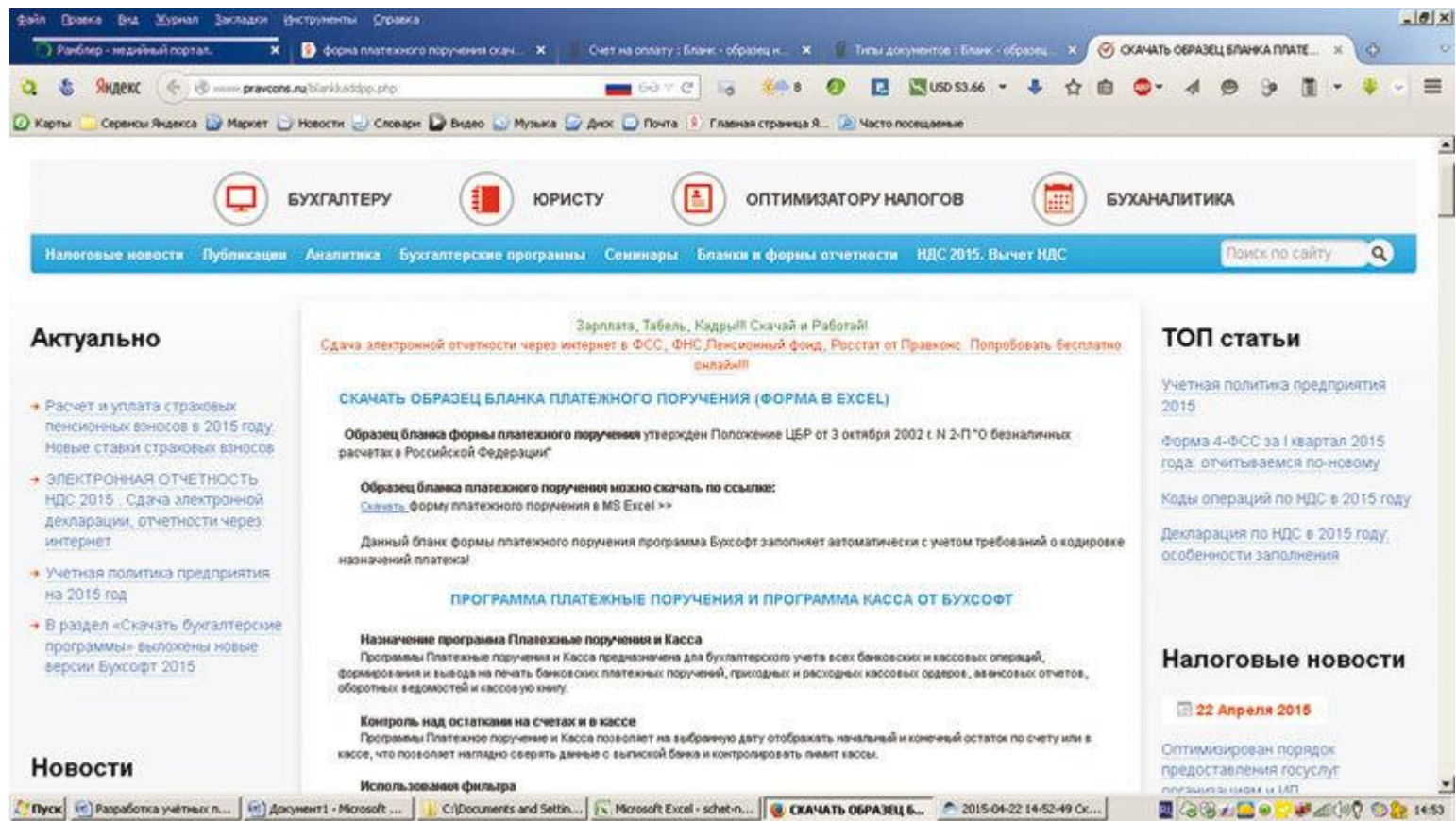


Рис. 75. Один из сайтов, где можно найти бланки документов

Кроме бланка платежного поручения в нашем примере необходимо каждому участнику конференции распечатать и выдать бейджик единого образца. Исходные данные для формирования бейджика должны браться из таблицы-регистра участников конференции. Внешний вид бейджика может иметь следующий вид (рис. 76):

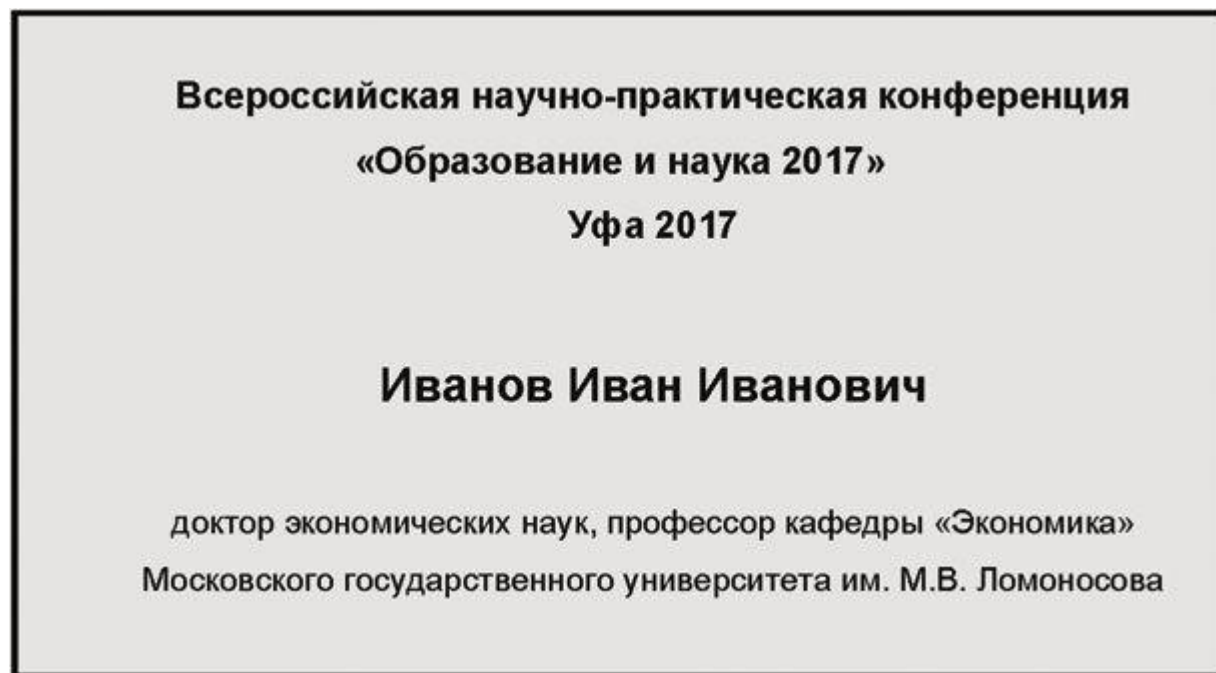


Рис. 76. Вариант оформления бейджика для участника конференции

В качестве одной из функций нашего учетного приложения является генерация бейджиков для участников конференции. Это достигается с использованием технологии слияния документов, когда исходный шаблон конечного документа соединяется (сливается) с данными из таблицы MS Word или MS Excel или базы данных MS Access. Эти данные автоматически вставляются в каждый экземпляр бейджика в нужную позицию и остается только их распечатать. Возможен и более сложный подход к решению данной задачи – написание программного кода для переноса каждой записи таблицы в соответствующие позиции шаблона бейджика. Для приложения, которое создается за короткий промежуток времени и выполняет часто разовую задачу слияния документов, алгоритм использования стандартных средств MS Office является наиболее приемлемым.

8.2. Разработка в MS Excel структуры таблицы для хранения данных в учетном приложении

В соответствии с используемым в нашем примере бланка платежного поручения, чтобы зафиксировать процедуру прохождения платежного поручения необходима таблица со следующими столбцами:

- 1) Номер платежного поручения.
- 2) Плательщик
 - Фамилия
 - Имя
 - Отчество.
- 3) Получатель
 - Наименование организации
 - ИНН
 - Номер счета.
- 4) Сумма платежа
- 5) Назначение платежа – «За участие в работе научно-практической конференции»
- 6) Дата заполнения.

Сумма платежа может быть разной в зависимости от количества получаемых экземпляров и способа получения данного сборника. Выделенные жирным позиции маркированного списка – обязательные и неизменные, поэтому они могут быть вбиты в бланк заранее.

8.3. Три варианта реализации диалога учетного приложения

Для создания диалоговой части при работе с электронными таблицами возможны три подхода к исполнению:

А. Таблица-регистр является основной рабочей областью приложения. Очередная информация заносится в нижнюю строку таблицы-регистра. При этом ячейки таблицы, если необходимо, могут использовать справочную информацию из вспомогательных таблиц.

В. Создание формы листа с использованием различных элементов управления. Элементы управления привязаны к справочным данным на листе. Перенос введенных данных в таблицу-реестр осуществляется с помощью макроса.

С. Использование VBA для создания диалоговой формы для последующего переноса введенных данных из формы в ячейки очередной строки электронной таблицы.

Приведенные варианты размещены по мере сложности реализации и трудозатрат. Рассмотрим все варианты.

8.3.1. Создание учетного приложения. Вариант А

При создании учетного приложения в данном варианте оформления исключается как таковая диалоговая форма. Информация сразу заносится в рабочую строку таблицы – регистра. Но при этом информация из справочных таблиц используется уже не в диалоговой форме, а в самой таблице-реестре. Данный подход приемлем, когда имеются ограничения по времени создания приложения и срок его эксплуатации является не длительным, т. е. только для решения задачи на данном временном интервале, например, в течение одного или двух месяцев.

При этом основное внимание уделяется созданию условий удобной работы с таблицей-реестром. В нашем примере назовем лист, на котором расположена таблица-регистр, «Базой». Во всех последующих примерах реализации обозначение листов останется неизменным.

Создадим таблицу-регистр следующей структуры (рис. 77):

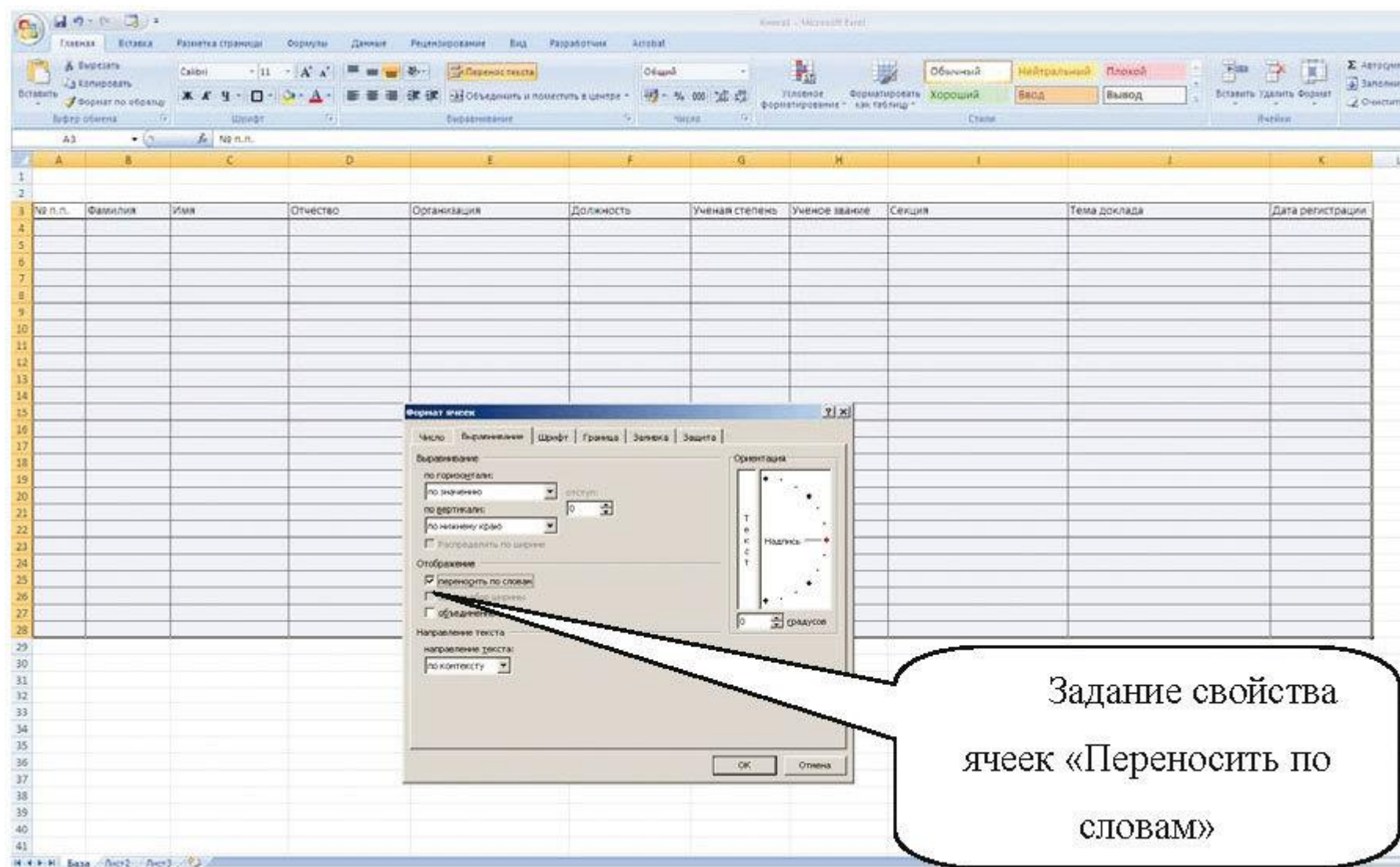


Рис. 77. Таблица-регистр на листе для регистрации участников конференции

В данную таблицу будем заносить всех участников конференции с темами их докладов. Так как информация, заносимая в ячейки может быть достаточно большой, то для удобства работы с таблицей и наглядного отображения занесенной информации зададим в формате всех ячеек таблицы свойство «Переносить по словам» (см. рис. 77).

На том же листе с базой разместим таблицы-справочники, которые облегчат ввод данных в таблицу-регистр, повысят безошибочность ввода, однообразие ввода, что позволит автоматизировать обработку по подведению итогов (рис. 78):

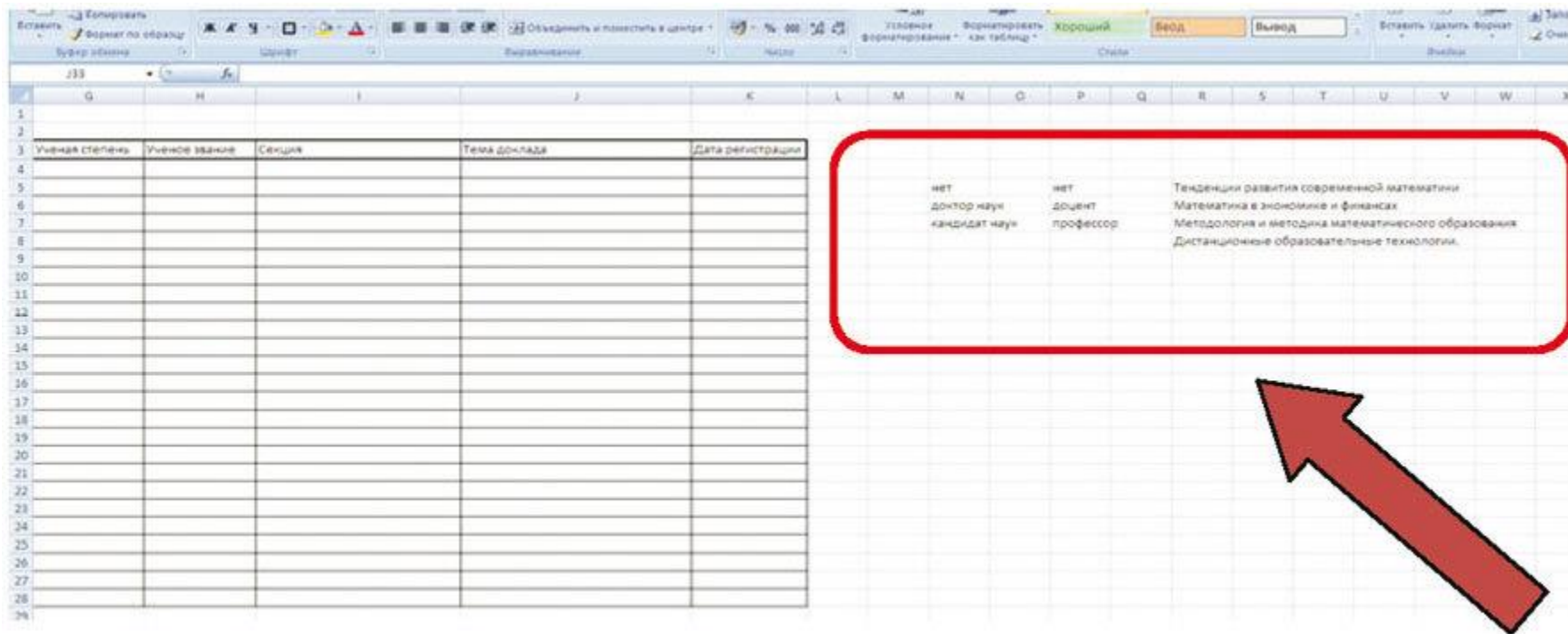


Рис. 78. Справочные таблицы для контроля исходных данных (выделены рамкой)

Таблицы-справочники нужны, прежде всего, когда информация постоянно повторяется. В нашем примере это ученая степень, ученое звание, названия секций конференций (рис. 79):

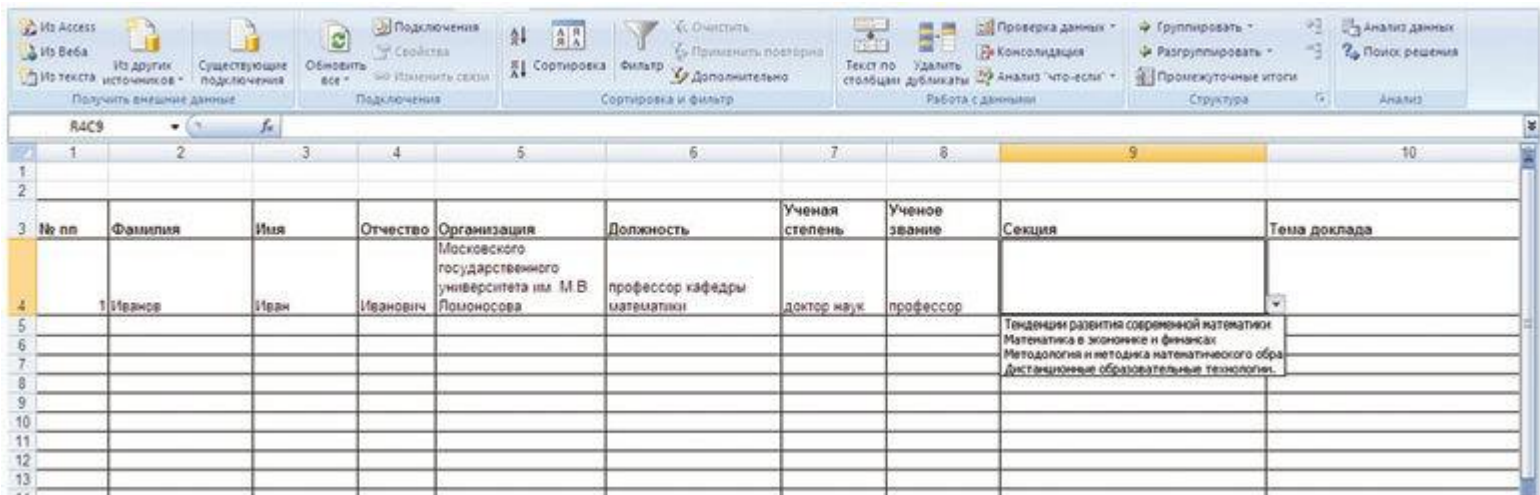


Рис. 79. Раскрывающийся список сформированный из справочной таблицы

Заполнив таблицу, в которой зарегистрированы наши участники конференции, заполним платежное поручение на любого из них. Для этого создадим две небольшие процедуры на VBA, которые будут автоматически заносить данные выбранной строки в бланк платежного поручения (рис. 80):

AW1

| Поступ. в Банк плат. | | Списано со сч. плат. | | 0401060 | |
|--|---------------|----------------------|----------------------|---------|--|
| ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № | | 16.04.2017 | | Дата | |
| Сумма прописью | | Вид платежа | | | |
| ИНН | КПП | Сумма | 300,00р. | | |
| Иванов Иван Иванович | | Сч. № | | | |
| Платательщик | | БИК | | | |
| Банк плательщика | | Сч. № | | | |
| Сбербанк России г. Москва | | БИК | 044525225 | | |
| Банк получателя | | Сч. № | 30101810400000000225 | | |
| ИНН 7729406900 | КПП 772901001 | Сч. № | 40702810038180102378 | | |
| Уфимский филиал Финансового университета г. Москва | | Вид оп. | Срок плат. | | |
| Вернадское ОСБ 7970 г. Москва | | Наз. пл. | Очер. плат. | | |
| Получатель | | Код | Рез. поле | | |
| За участие в конференции | | | | | |
| Назначение платежа | | | | | |
| Подпись | | | Отметки банка | | |
| М.П. | | | | | |

Рис. 80. Бланк платежного поручения

Ниже представлены два программных кода, которые при совместной работе позволяют перенести данные с выбранной строки таблицы-регистра в бланк платежного поручения при двойном щелчке левой клавишей мыши по номеру строки (рис. 81, рис. 82). Следует обратить внимание на то, что первый модуль расположен на листе «База» и предназначен для работы в пределах листа. Поэтому для передачи данных в платежное поручение, которое находится на другом листе, необходимо вызвать следующую процедуру модуля «Заполнение». Результат переноса можно увидеть на рисунке 83:

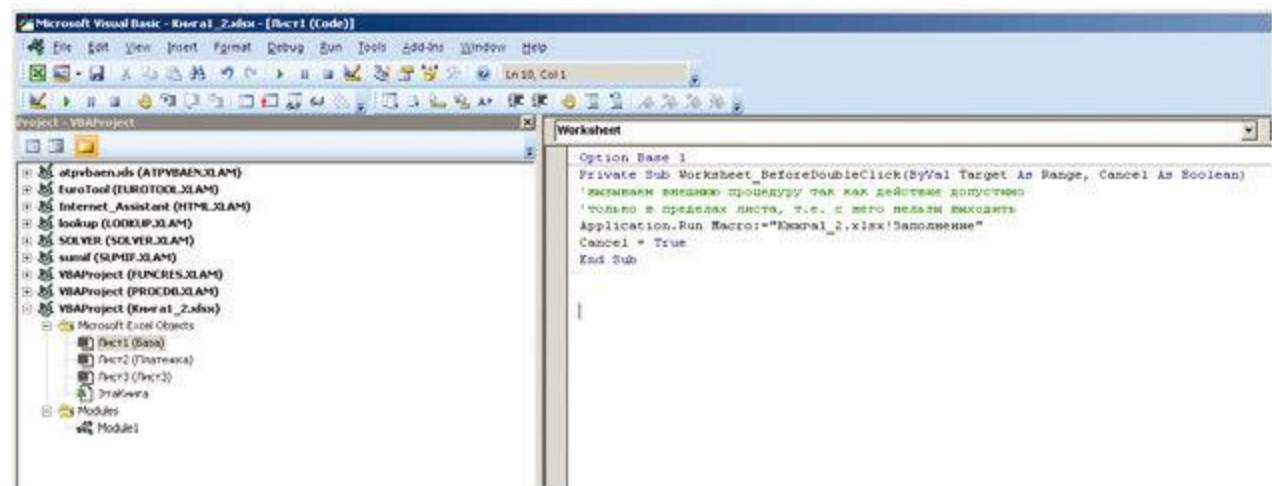


Рис. 81. Программная процедура листа «База»

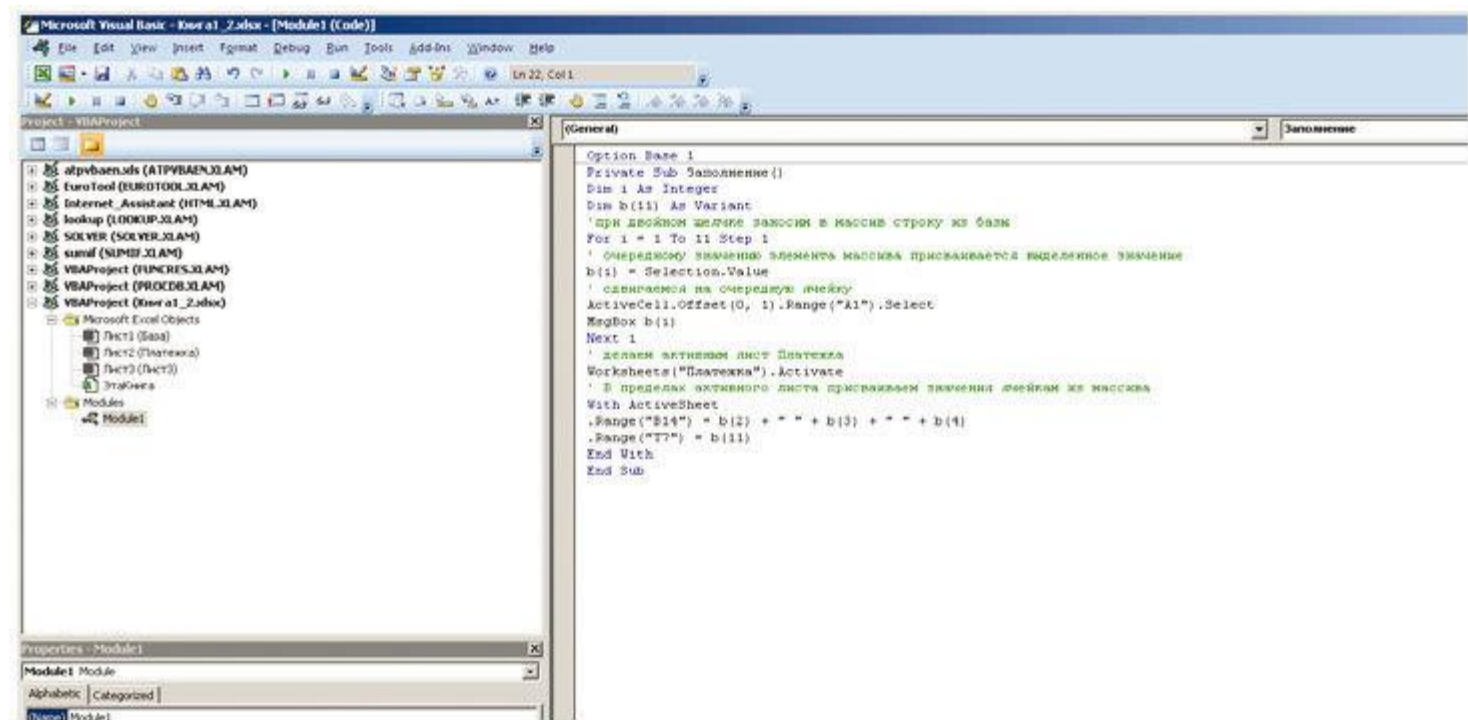


Рис. 82. Процедура модуля для переноса данных из таблицы-регистра в платежное поручение

ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № 11.04.2017

Сумма прописью: Триста рублей

ИНН: 7729406900 КПП: 772901001 Сумма: 300

Плататель: Петров Петр Петрович

Платательское учреждение: Сбербанк России г. Москва

БИК: 044525225 Сч. №: 30101810400000000225

Банк получателя: Финансовый университет г. Москва

ИНН: 7729406900 КПП: 772901001 Сч. №: 40702610038180102376

Вид пл.: Наз. пл.: Срог. плат.: Очер. плат.: Рек. поле:

Подпись: И.П.

Рис. 83. Заполненное платежное поручение

А теперь платежное поручение можно выводить на печать. Средства табличного процессора удобны для организации процедуры вывода на печать и нет необходимости создания для этого отдельного макроса (рис. 84, рис. 85).

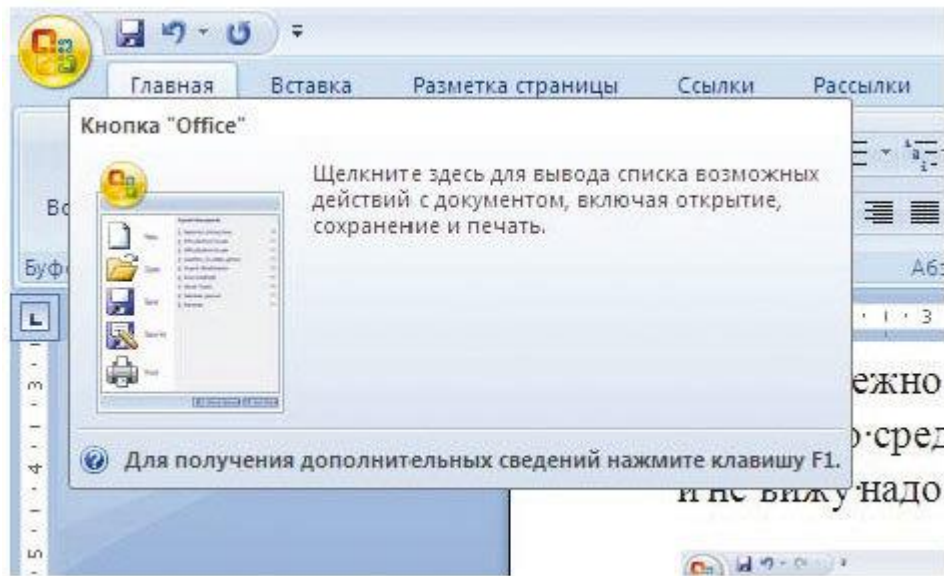


Рис. 84. Позиция меню кнопки «Office»

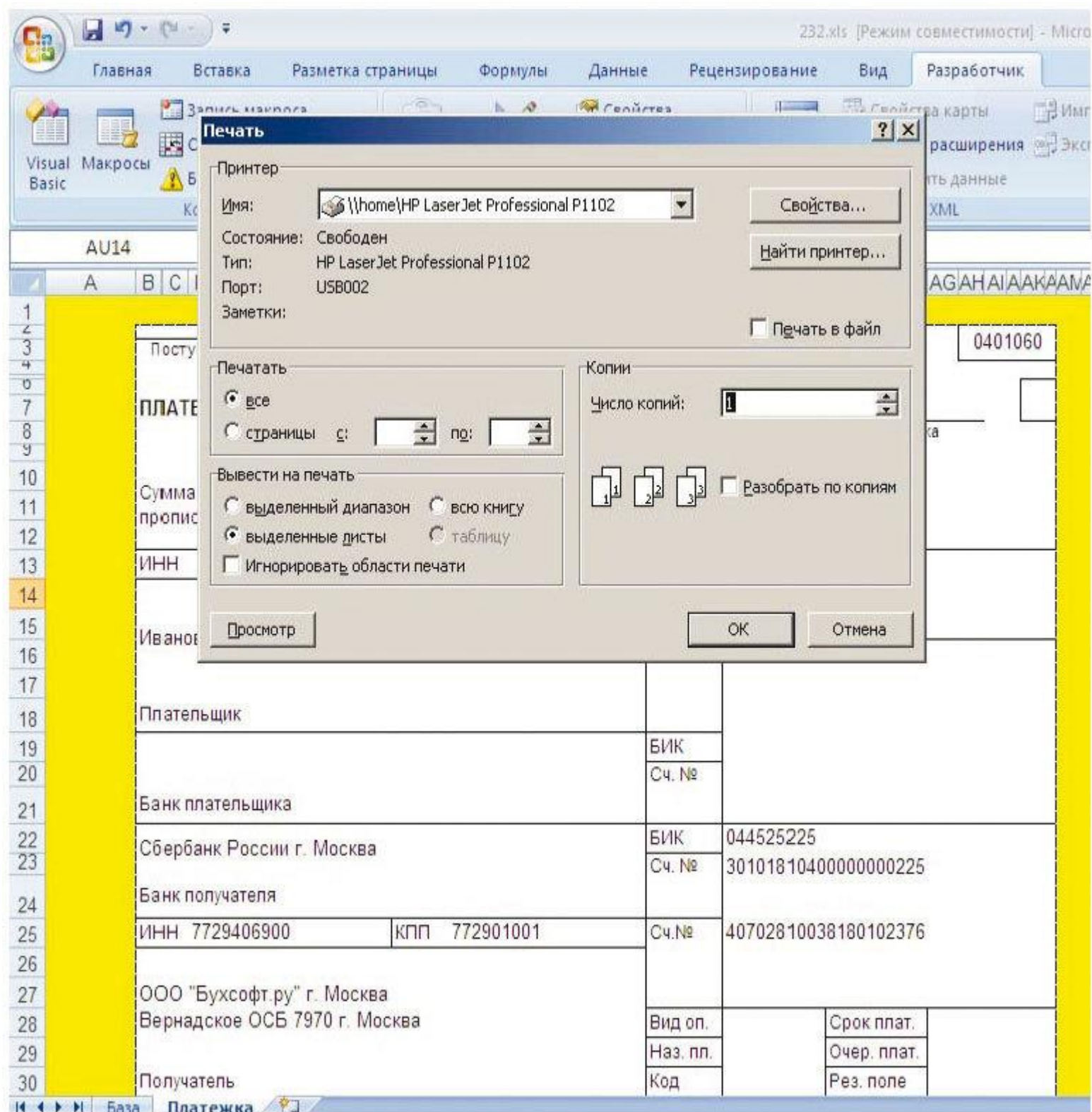


Рис. 85. Панель настройки Печать

Мы рассмотрели вариант учетного приложения, состоящего из таблицы-регистра, справочных таблиц и платежного поручения в электронном виде. Для автоматизации переноса данных из строки регистра используются макросы.

Учетное приложение такой конфигурации достаточно для решения широкого круга задач, когда надо быстро наладить учет каких-либо событий и при этом автоматизировать выпуск платежного поручения. Например, при регистрации участников конференции, а также для нужд малого бизнеса, когда используется безналичный расчет.

Следует иметь в виду, что офисные приложения могут эффективно обмениваться между собой данными и совместно использовать их. Данные, занесенные в таблицу, могут быть использованы для автоматизированного создания массовой рассылки писем, приглашений, поздравлений.

8.3.2. Создание учетного приложения с использованием формы листа. Вариант В

В данном разделе мы рассмотрим создание учетного приложения, в котором содержатся следующие составляющие компоненты:

- диалоговая форма листа с элементами управления;
- справочные таблицы, содержащие информацию о товаре и его стоимости;
- таблица-регистр для учета осуществленных продаж;
- бланк платежного поручения;
- модуль на языке VBA для автоматизации процесса работы.

8.3.2.1. Анализ предметной области

Пусть требуется разработать учетное приложение для автоматизации учета продажи товаров частного кондитерского цеха. Система должна предусматривать диалоговое окно, отображающее предлагаемые товары, их цены и вес.

В кондитерском цехе ведется учет оказанных услуг и на каждого клиента в базу вносятся следующие сведения:

- наименование;
- товары;
- количество товаров;
- цена по каждому товару;
- стоимость покупки.

В платежное поручение автоматически должны вписываться данные клиента, сумма к оплате, дата и назначение платежа.

Учетное приложение имеет следующие компоненты: диалоговая часть, основная электронная таблица для сбора и хранение данных на листе «База», справочные таблицы и печатный бланк утвержденной формы (платежное поручение). Перенос исходных данных из формы листа и расчетных данных в таблицу-регистр будет реализовано с помощью процедуры, написанной на языке программирования VBA. Передача строки данных из основной таблицы в печатный бланк будет осуществляться также программно.

8.3.2.2. Этапы создания приложения

1. На листе «Диалог» создадим диалоговую форму листа с использованием элементов управления «Поле со списком» и «Кнопка» (рис. 86):

| | |
|--------------------|-------------------------|
| Клиент | ООО "Смак" |
| Товар | Торт "Встреча" ▼ |
| Цена за единицу | 250,00р. |
| Количество | 40 |
| К оплате | 10 000,00р. |
| Назначение платежа | За кондитерские изделия |
| Дата | 16.04.2017 |

Занести в базу

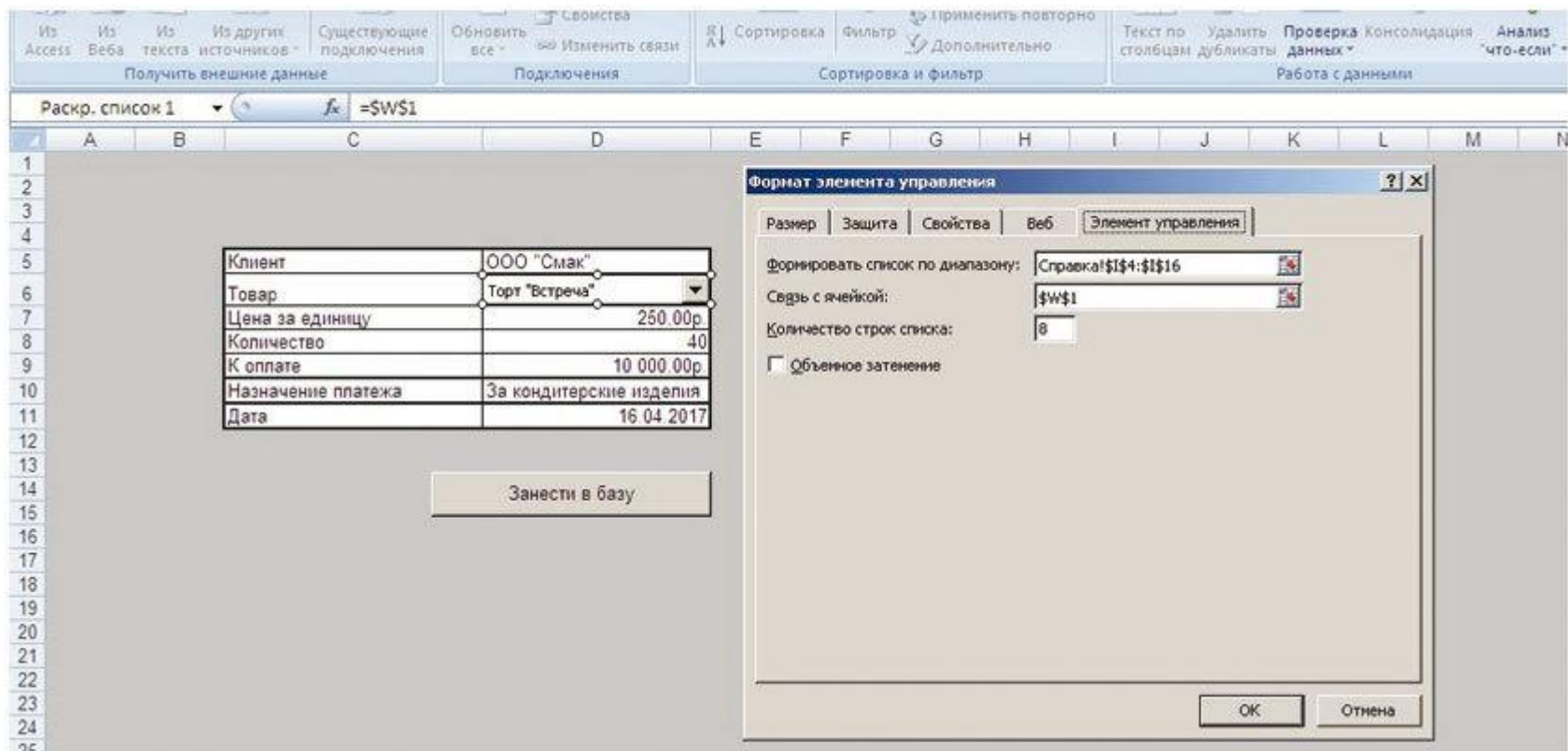
Рис. 86. Диалоговая форма учетного приложения

2. На листе «Справка» создадим таблицу-справочник, где укажем наименование услуг и их стоимость (рис. 87):

| группа | наименование | Вес | Цена за единицу |
|--------|---------------------|------|-----------------|
| 1 | Пирожное "Северное" | 0,2 | 20 |
| 2 | Пирожное "Шу" | 0,15 | 25 |
| 3 | Пирожное "Эклер" | 0,2 | 15 |
| 4 | Пирожное "Лакомка" | 0,25 | 30 |
| 5 | Торт "Наполеон" | 0,9 | 200 |
| 6 | Торт "Встреча" | 1 | 250 |
| 7 | Торт "Полет" | 0,9 | 150 |
| 8 | Торт "Лапоть" | 0,5 | 100 |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |

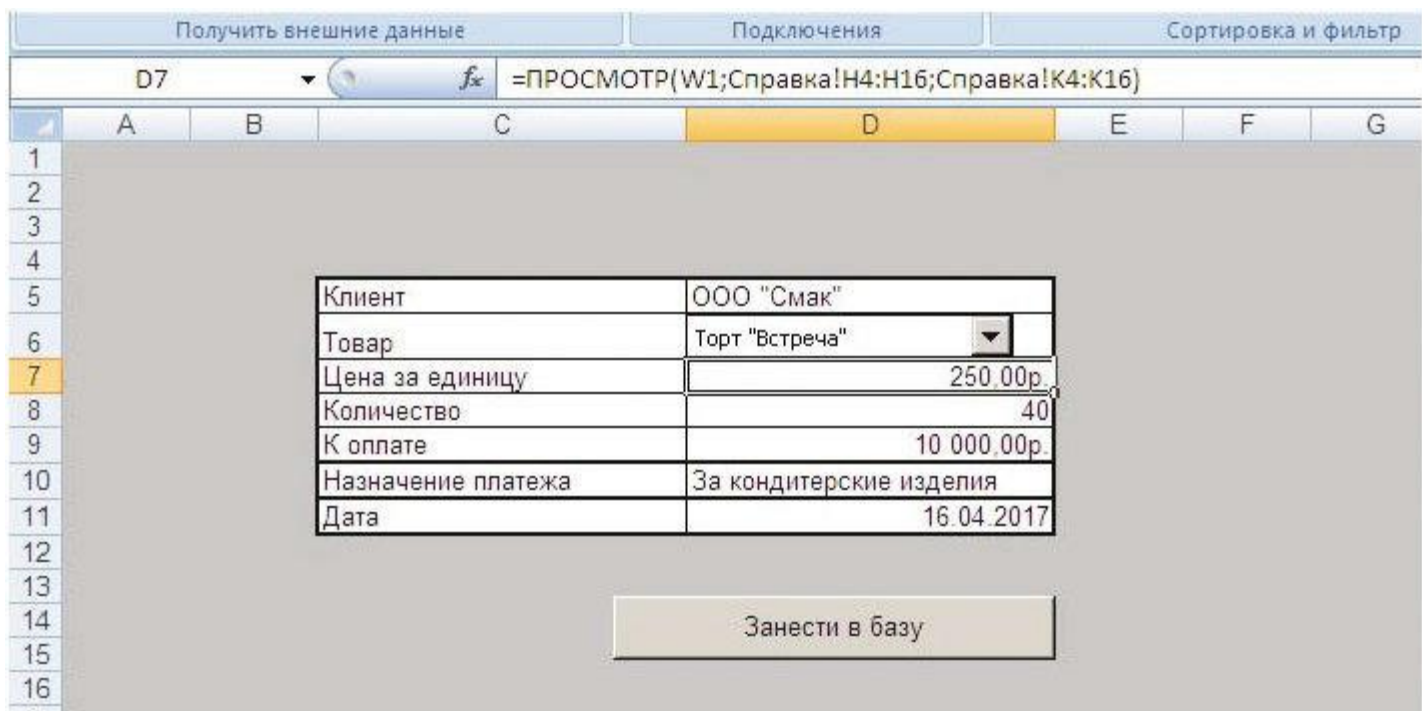
Рис. 87. Справочная таблица

Настроим элемент управления «Поле со списком», воспользовавшись контекстным меню (правая клавиша мыши) ^ Формат объекта, сформируем списки элементов управления, задав источник исходных данных (рис. 88):



3. Для отображения текущей даты продажи в ячейку D11 занесем функцию листа =СЕГОДНЯ().

4. Для отображения цены выбранного продукта из справочной таблицы в ячейку D7 занесем функцию листа =ПРОСМОТР(W1;Справка!Н4:Н16;Справка!К4:К16), что позволит автоматически отображать цену товара в зависимости от выбранной позиции элемента управления «Поле со списком», результат работы которого возвращается в ячейку W1 (рис. 89):



5. Добавим столбец «Количество», где пользователь может указать количество товаров.
6. В столбце «К оплате» подсчитаем общую сумму за выбранные товары (рис. 90):

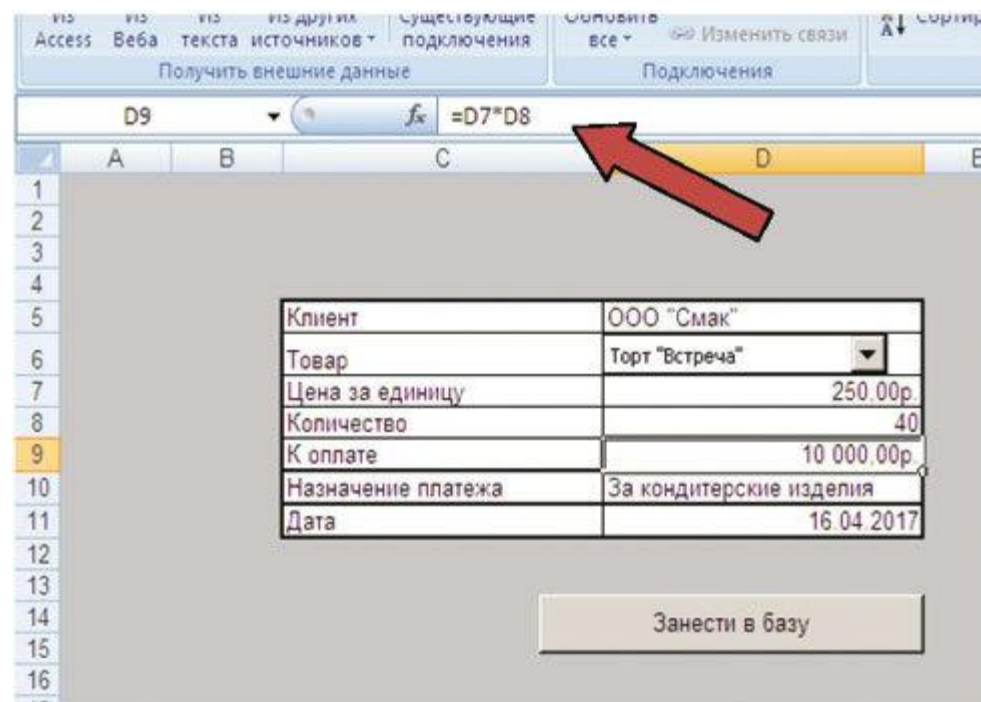


Рис. 90. Расчетная формула для вычисления суммы к оплате

7. Добавим строку «Назначение платежа» для ручного ввода.
8. На листе «База» создадим следующую таблицу-регистр (рис. 91):

The screenshot shows the Microsoft Excel 2010 interface. The 'Данные' (Data) ribbon is active, displaying options for getting external data, connections, sorting and filtering, and data tools. Below the ribbon, the active sheet is 'База' (Base), and the current cell is H8, containing the date 01.05.2016. The data table is as follows:

| | A | B | C | D | E | F | G | H |
|----|------|------------|-----|----------|------------|----------------|-------------------------|------------|
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | № пп | Клиент | Что | Цена | Количество | Сумма к оплате | Назначение платежа | Дата |
| 4 | 1 | ООО "Лига" | 8 | 6.00р. | 3 | 18.00р. | За кондитерские изделия | 04.02.2016 |
| 5 | 2 | ООО "Смак" | 3 | 5.00р. | 20 | 100.00р. | За кондитерские изделия | 04.02.2016 |
| 6 | 3 | ООО "Тор" | 8 | 6.00р. | 40 | 240.00р. | За кондитерские изделия | 05.02.2016 |
| 7 | 4 | ООО "Смак" | 8 | 100.00р. | 40 | 4 000.00р. | За кондитерские изделия | 30.04.2016 |
| 8 | 5 | ООО "Смак" | 8 | 100.00р. | 40 | 4 000.00р. | За кондитерские изделия | 01.05.2016 |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | | | | | | | | |

Рис. 91. Таблица-регистр на листе «База»

9. На листе «Платежка» поместим бланк платежного поручения, который нашли в сети Интернет (рис. 92). Следует отметить, что в сети Интернет достаточно большое количество сайтов, предлагающих различные бланки в формате MS Excel.

Access Беба текста источников подключения все Сортир

В32 За кондитерские изделия

| | | | | | |
|-------------------------------|---------------|----------------------|----------------------|--------------------------|--|
| Платеж в банк плат | | Списание со сч. плат | | 0401000 | |
| ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № | | 16.04.2017 | | <input type="checkbox"/> | |
| | | Дата | | Вид платежа | |
| Сумма прописью | | | | | |
| ИНН | КПП | Сумма | 10 000,00р. | | |
| ООО "Смак" | | Сч. № | | | |
| Платежник | | В/К | | | |
| Банк плательщика | | Сч. № | | | |
| Сбербанк России г. Москва | | В/К | 044526226 | | |
| Банк получателя | | Сч. № | 30101810400000000225 | | |
| ИНН 7729406900 | КПП 772901001 | Сч. № | 40702810038180102378 | | |
| ООО "Лаконика" г. Москва | | Вид оп. | | | |
| Варшавское ОДБ Т970 г. Москва | | Наз. пл. | Срок плат. | | |
| Получатель | | Код | Овер. плат. | | |
| | | | Рез. поле | | |
| За кондитерские изделия | | | | | |
| Назначение платежа | | | | | |
| М.П. | | Подпись | | Отметка банка | |
| | | | | | |
| | | | | | |

Рис. 92. Бланк платежного поручения на листе

10. Воспользуемся кнопкой макрос «Перенос» (рис. 93):

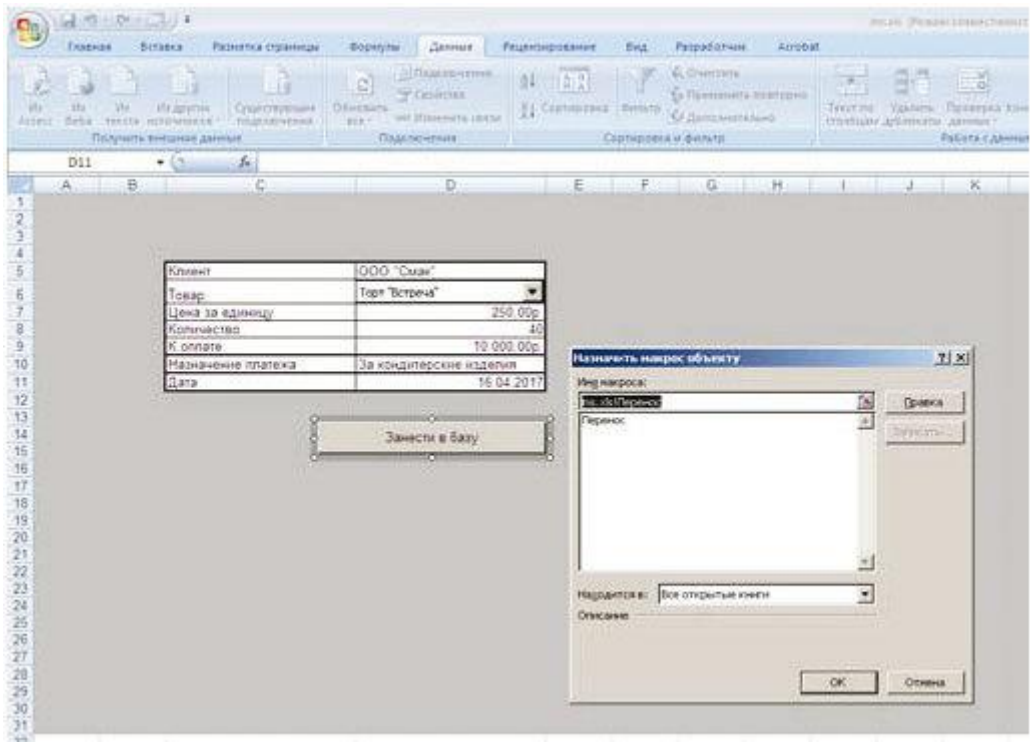


Рис. 93. Назначение элементу управления «Кнопка» макроса «Перенос»

Программный код макроса показан на рисунке 94. Он содержит большое количество комментариев, которые помогут легче понять алгоритм его работы. Назначение макроса (процедуры) – перенос данных из диалоговой формы в последнюю строку таблицы-регистра и платежное поручение.

При использовании языка программирования открываются большие возможности по управлению приложением. В нашем примере мы всего лишь переносим информацию из одной электронной таблицы в другую, используя для этого массив в программе. Данные, после того как попадут в программную среду могут быть обработаны по любому алгоритму по желанию разработчика. Тогда в ячейках листа не будет никаких формул, все расчеты могут быть выполнены программно, а результаты после этого выставлены в любую ячейку книги.

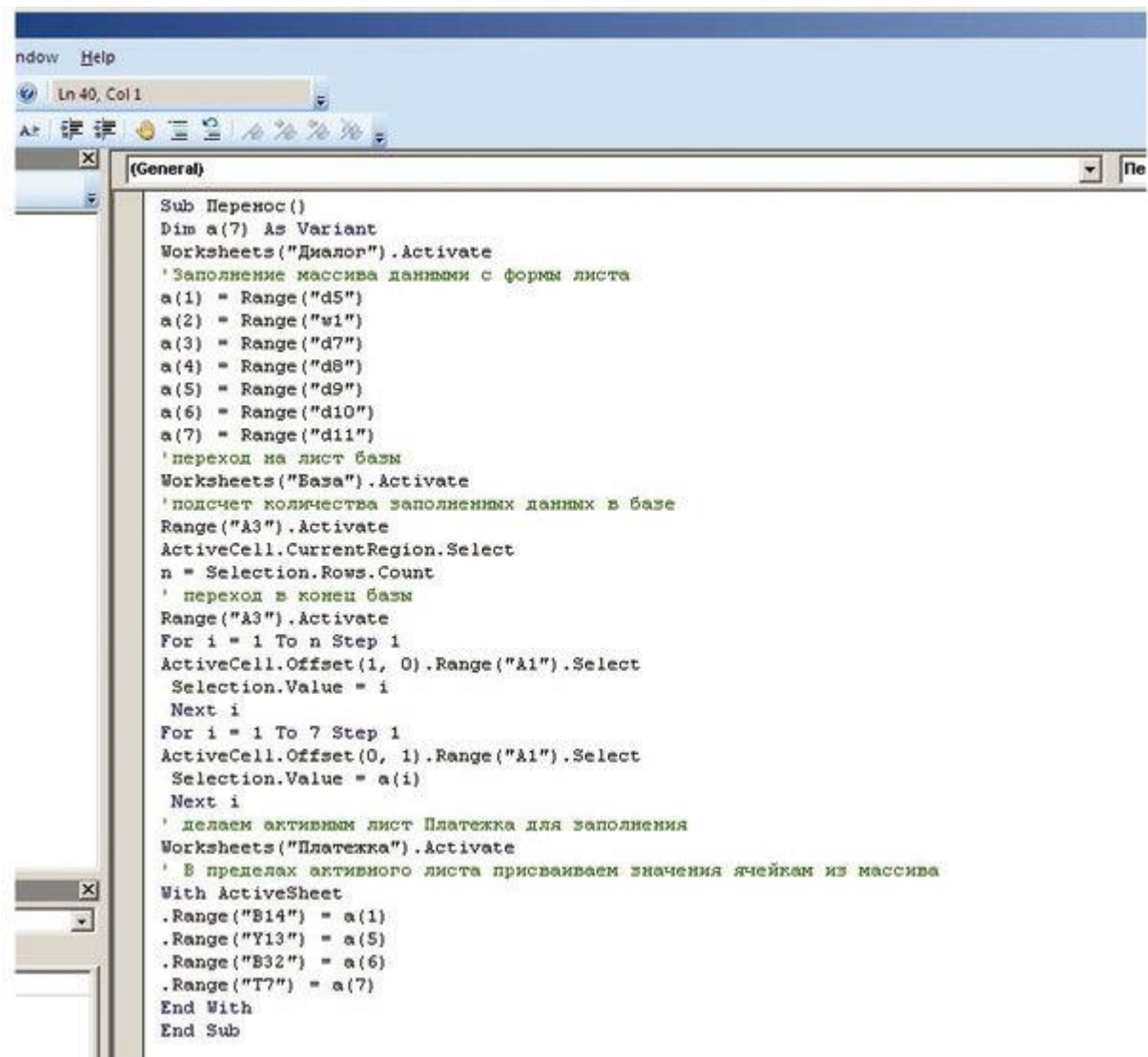


Рис. 94. Программный код учетного приложения

После разработки диалоговой части учетного приложения, для автоматизации заполнения платежного поручения и вывода его на печать, воспользуемся разработанными ранее в варианте А модулями на VBA. Возьмем исходные данные из таблицы-регистра, а она у наших вариантов учетного приложения одинаковая.

8.3.3. Создание учетного приложения с использованием диалоговой формы пользователя. Вариант С

В данном разделе пособия мы рассмотрим этапы создания учетного приложения, у которого интерфейсная часть реализована в виде диалоговой формы пользователя. Приложение предназначено для небольшой частной пекарни. Бланк платежного поручения приведен в пример только для отображения возможностей реализации данной операции – выписки платежного поручения.

Внешний, начальный интерфейс программы представлен в виде формы листа, хоты можно было бы и сделать открытие формы пользователя для ввода исходных данных при наступлении события: загрузка приложения (рис. 95):

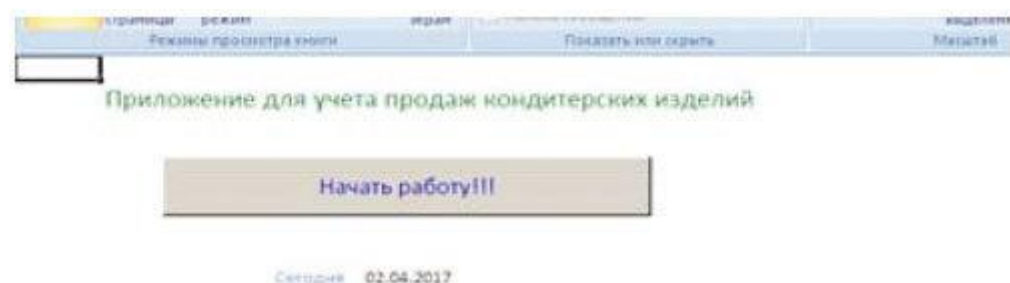


Рис. 95. Начальное окно приложения

В начальном окне приложения находится кнопка, привязанная к макросу и отрывающая диалоговое окно для ввода начальных значений (рис. 96):

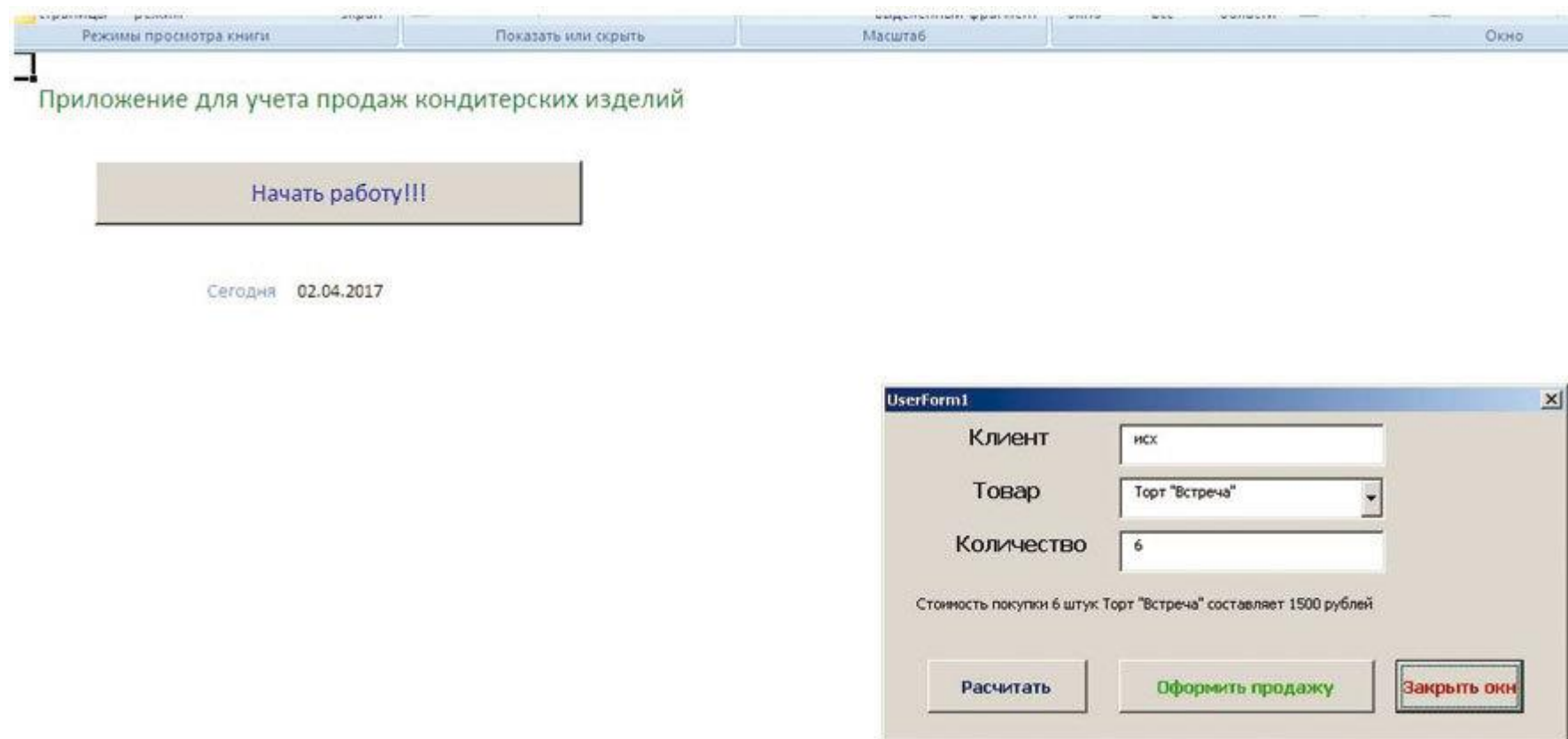


Рис. 96. Форма диалога после нажатия на кнопку «Начать работу!!!»

Элемент «Раскрывающийся список» автоматически программно заполняется из справочной таблицы при выполнении строки кода:

UserForm1. ComboBox1. RowSource = “Диалог! AN6:An13”.

Результат работы элемента управления видно на рисунке 97:

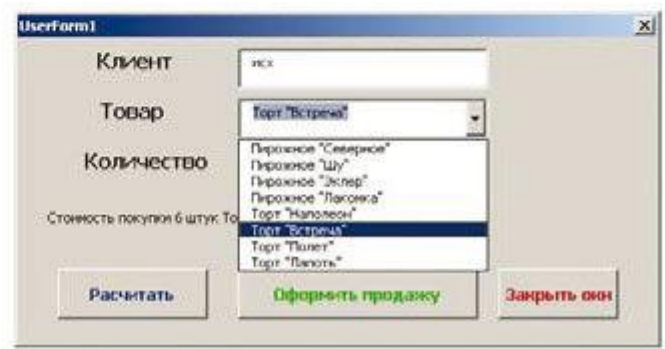
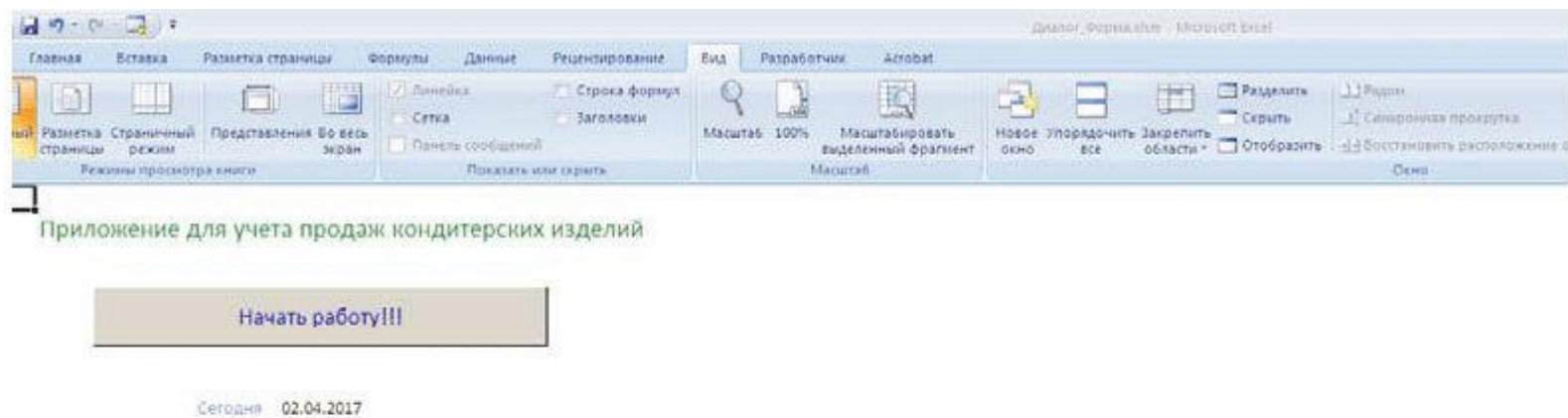


Рис. 97. Заполненный элемент управления формы

После нажатия кнопки «Рассчитать» свойству Caption элемента управления Label будет присвоена текстовая строка: UserForm1. Label4. Caption = “Стоимость покупки “&UserForm1. TextBox2. Value& “ штук “ &b& “ составляет “ &c& “ рублей”. Ее можно увидеть на рисунке 98:

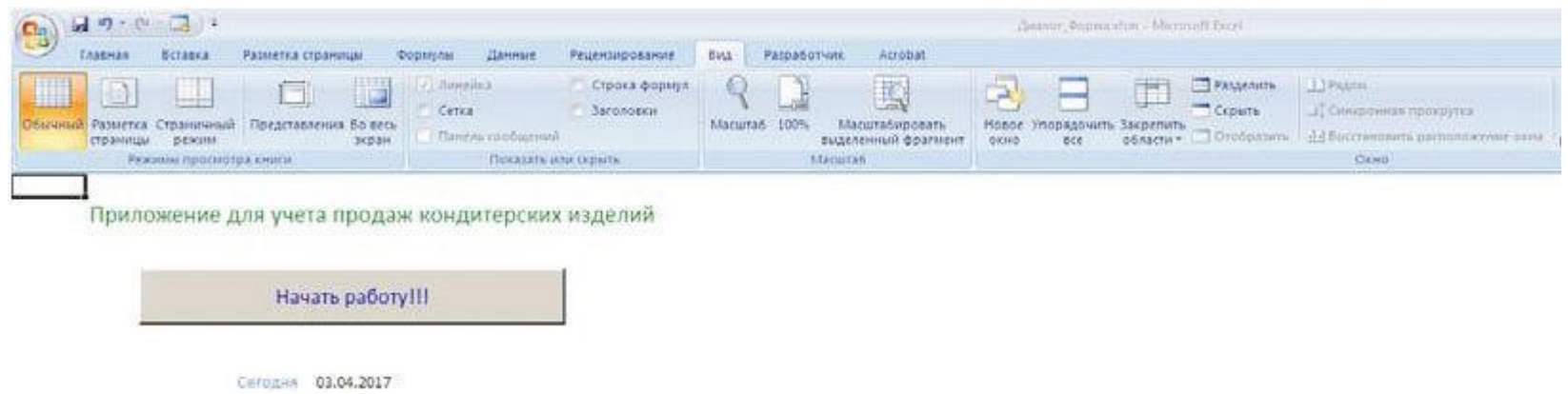


Рис. 98. Расчетные значения, выведенные в Окно диалога

После нажатия кнопки «Оформить продажу» произойдет перенос данных из формы в таблицу-регрстр (рис. 99) и платежное поручение (рис. 100):

| № пп | Клиент | Что | Цена | Количество | Сумма к оплате | Дата |
|------|-------------|--------------------|---------|------------|----------------|------------|
| 1 | прпрпр | Пирожное "Эклер" | 15,00р. | 5 | 15.03.1900 | 03.04.2017 |
| 2 | прпрпр | Пирожное "Эклер" | 15,00р. | 5 | 75,00р. | 03.04.2017 |
| 3 | Иванов Иван | Пирожное "Лакомка" | 30,00р. | 5 | 150,00р. | 03.04.2017 |

Рис. 99. Таблица-регрстр с занесенными продажами

ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № _____ от 03.04.2017

Сумма: 150

Клиент: Иванов Иван

Товар: Пирожное "Лакомка"

Количество: 5

Стоимость покупки 5 штук Пирожное "Лакомка" составляет 150 рублей

Рассчитать | Оформить продажу | Закрыть окно

Рис. 100. Платежное поручение с занесенными данными текущей продажи

Ниже приводится текст программного кода реализации данного варианта исполнения учетного приложения: OptionBase 1

```
Dim a As Single
```

```
Dim b As String
```

```
Dim c As Single
```

```
Sub CommandButton1_Click()
```

```
Worksheets(“Диалог”). Activate
```

```
‘ подсчет количества заполненных данных в таблице
```

```
Range(«AN5»). Activate
```

```
ActiveCell. CurrentRegion. Select n = Selection. Rows. Count
```

```
‘ присвоение значения из элемента формы переменной b
```

```
b = UserForm1. ComboBox1. Value
```

```
‘ переход в начало таблицы
```

```
Range(“AN5”). Activate
```

```
‘ поиск в таблице соответствующей цены в зависимости от названия продукта
```

```
For i = 1 To n – 1 Step 1
```

```
ActiveCell. Offset(1, 0). Range(«A1»). Select
```

```
If b = Selection. Value Then
```

```
‘ перемещаемся на два столбца вправо и считываем значение
```

```
ActiveCell. Offset(0, 2). Range(«A1»). Select a = Selection. Value
```

```
End If
```

```
Next i c = UserForm1. TextBox2. Value * a
```

```
Range(«A1»). Select
```

```
UserForm1. Label4. Caption = «Стоимостьпокупки «& UserForm1. TextBox2. Value & «штук «& b & «составляет «& c & «рублей»
```

```
End Sub
```

```
Sub CommandButton2_Click()
```

```
Dim a1(6) As Variant
```

```
Worksheets(«Диалог»). Activate
```

```
‘ заполнениемассивапеременных a1(1) = UserForm1. TextBox1. Value a1(2) = b a1(3) = a a1(4) = UserForm1. TextBox2. Value a1(5)  
= c a1(6) = Range(«E8»). Value
```

```
‘ переход на лист базы
```

```
Worksheets(“База”). Activate
```

```
‘ подсчет количества заполненных данных в базе
```

```
Range(«A3»). Activate
```

```
ActiveCell. CurrentRegion. Select n = Selection. Rows. Count
```

```
‘ переходвконецбазы
```

```
Range(«A3»). Activate
```

```
For i = 1 To n Step 1
```

```
ActiveCell. Offset(1, 0). Range(«A1»). Select
```

```
Selection. Value = i Next i
```

```
For i = 1 To 6 Step 1
```

```
ActiveCell. Offset(0, 1). Range(«A1»). Select
```

```
Selection. Value = a1(i)
```

```
Next i
```

```
Worksheets(“Платежка”). Activate
```

```
‘ В пределах активного листа присваиваем значения ячейкам из массива
```

```
With ActiveSheet
```

```
Range(«B14») = a1(1)
```

```
Range(«U11») = a1(2)
```

```
Range(«Y13») = a1(5)
```



```

Range(«T7») = a1(6)
Range(«B32») = «Стоимостьпокупки «&UserForm1. TextBox2. Value& «штук «&b& «составляет «&c& «рублей»
End With
End Sub

```

```

Sub CommandButton3_Click()

```

```

    ' звкрытиеформы

```

```

    UserForm1. Hide

```

```

EndSub

```

```

PrivateSubUserForm_Activate()

```

```

    ' заполненияраскрывающегосясписказначениямиприоткрытииформы

```

```

    ' издиапазонаячееклиста

```

```

    UserForm1. ComboBox1. RowSource = «Диалог! AN6:An13»

```

```

EndSub.

```

При сохранении учетного приложения следует выбрать формат сохранения файла *. xlsx – Книга Excel с поддержкой макросов (рис. 101):



Рис. 101. Сохранение файла с макросами

8.4. Выводы по разделу

Отметим, что варианты А, В, С отличаются только интерфейсной частью. Программные модули для автоматизации заполнения и распечатки платежного поручения у них одинаковые. Если учетное приложение содержит расчетную часть, то она также будет одинаковой для различных вариантов исполнения. Так как расчетный раздел может быть представлен отдельной расчетно-аналитической электронной частью на дополнительном листе приложения или приведен в виде расчетного модуля, написанного языке программирования VBA. В последнем случае результат выполнения также будет помещен в аналитическую таблицу. Визуализация результатов достигается построением различных диаграмм и графиков на основе итоговой таблицы. Для эффектной и наглядной демонстрации результатов возможно создание анимации с помощью программного кода, что возможно при наличии определенных навыков работы.

9. Практическая реализация учетного приложения в среде MS Access

Еще одним наиболее удобным инструментом для создания учетных приложений в составе MS Office является СУБД MS Access. Надо отметить, что начиная с версии пакета 2007, он входит не во все релизы. Данный инструмент мощный, гибкий и хорошо подходит для решения задач, рассматриваемых в нашем пособии. При этом считаем, что пользователи пособия уже имеют представление и начальные навыки работы с текстовым процессором, табличным процессором и СУБД, входящими в данный пакет программ.

При разработке учетного приложения часто основным лимитирующим фактором становится время его создания. Иногда на это выделяется максимум один день, и основной задачей приложения является эффективно и удобно выполнять свои функции. Никто не отрицает, что можно создать и более «продвинутой» систему. Но для этого надо другое время и другие ресурсы. Поэтому нашей формулой при разработке учетного приложения должна стать последовательность «Быстро → Эффективно → Удобно → Недорого».

9.1. Создание базы данных для учета участников конференции

Для регистрации участников конференции создадим небольшую базу их двух таблиц. Учет участников конференции будем вести в первой таблице – назовем ее «Регистрация», а вторая таблица необходима нам для организации передачи данных в бейджик и памятку участника конференции назовем «Шаблоны».

Априори считаем, что читатели данного пособия имеют навыки работы с офисными приложениями MS Office. В связи с чем, подробно не расписываем процесс создания таблиц базы данных в MS Access. Так как создание таблиц – самый первый, начальный этап создания любой базы. Структура этих таблиц показана на рисунках (рис. 102, рис. 103).

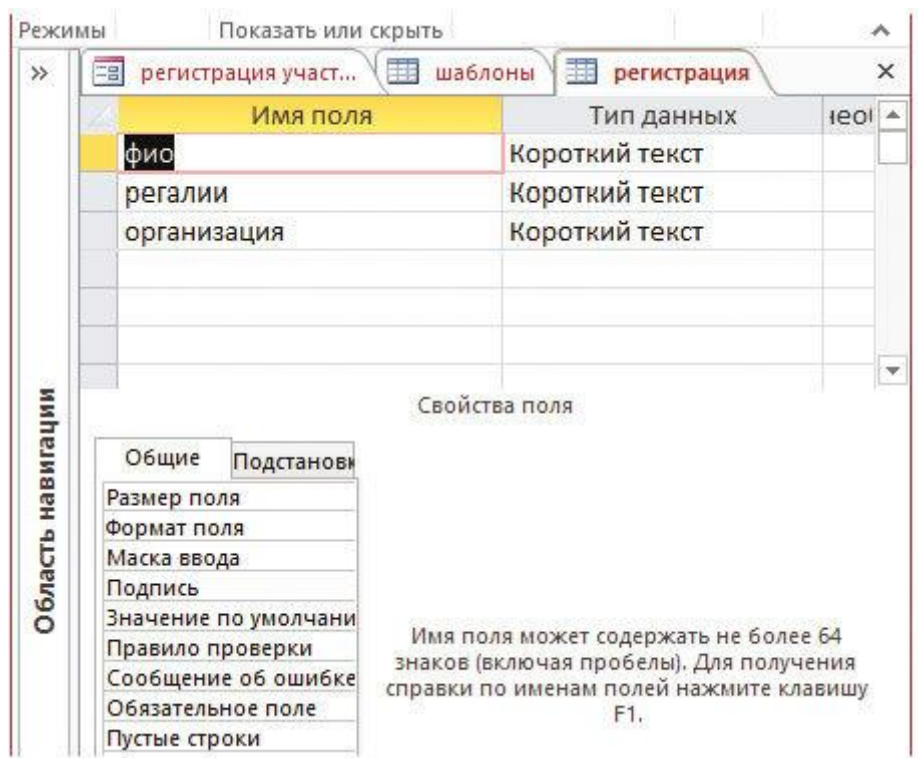


Рис. 102. Таблица «Регистрация» в режим конструктора

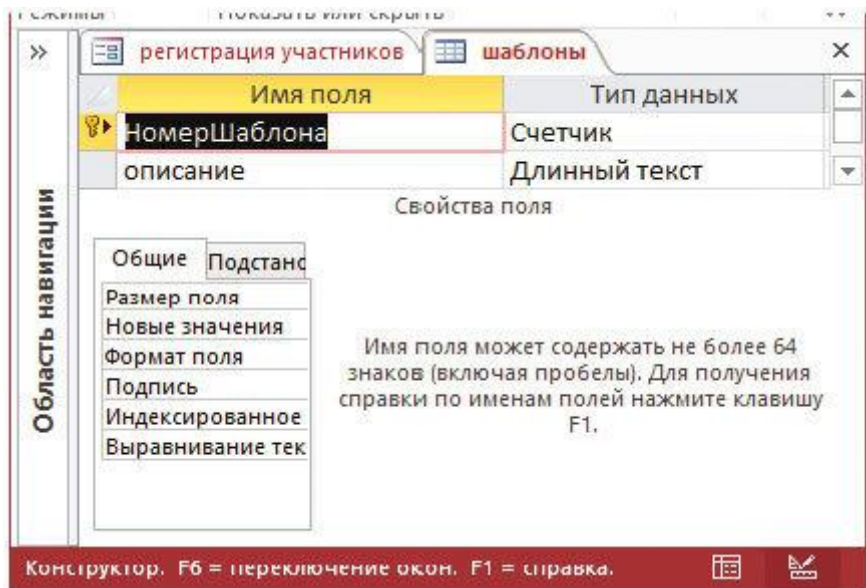


Рис. 103. Таблица «Шаблоны» в режиме конструктора

При работе в среде MS Access большую помощь оказывают помощники программной среды. Они в MS Access, наверное, самые удачные и продуктивные, по сравнению с другими помощниками других офисных приложений. Пошаговая последовательность действий в диалоге с помощником приводит к созданию законченного объекта базы данных.

9.2. Экспорт исходных данных из базы данных MS Access в документ MS Word или MS Excel

В случае необходимости формирования документов с общими свойствами:

1. один и тот же шаблон;
2. сохранение параметров заполнения;
3. привлечение списков клиентов, возможно совместное использование разных приложений MS Office, например текстового процессора Word с СУБД Access.

При этом возникают дополнительные преимущества:

1. обеспечивается сохранение данных, введенных пользователем через форму;
2. данные можно использовать повторно.

Алгоритм решения задач подобного рода предполагает использование связки Word – Access. Причем реализованная база данных будет источником заполнения шаблона.

Пример. Необходимо разработать систему подготовки и печати организационных документов (бейджик, памятка участника) для участников конференции.

Для начала разработаем необходимый шаблон1 и в необходимые места добавляем закладки. В нашем случае таких мест 3: фамилия имя отчество, звания и регалии, организация. Создаем соответственно, закладки BFIO, BDOL, BORG.

Сохраняем полученный шаблон в формате *. dot (например, BD. dot) (рис. 104)

Затем разрабатываем шаблон памятки участника. Кроме основного текста добавляем закладку с реквизитами участника (в нашем случае Фамилия, Имя, Отчество) (рис. 105).

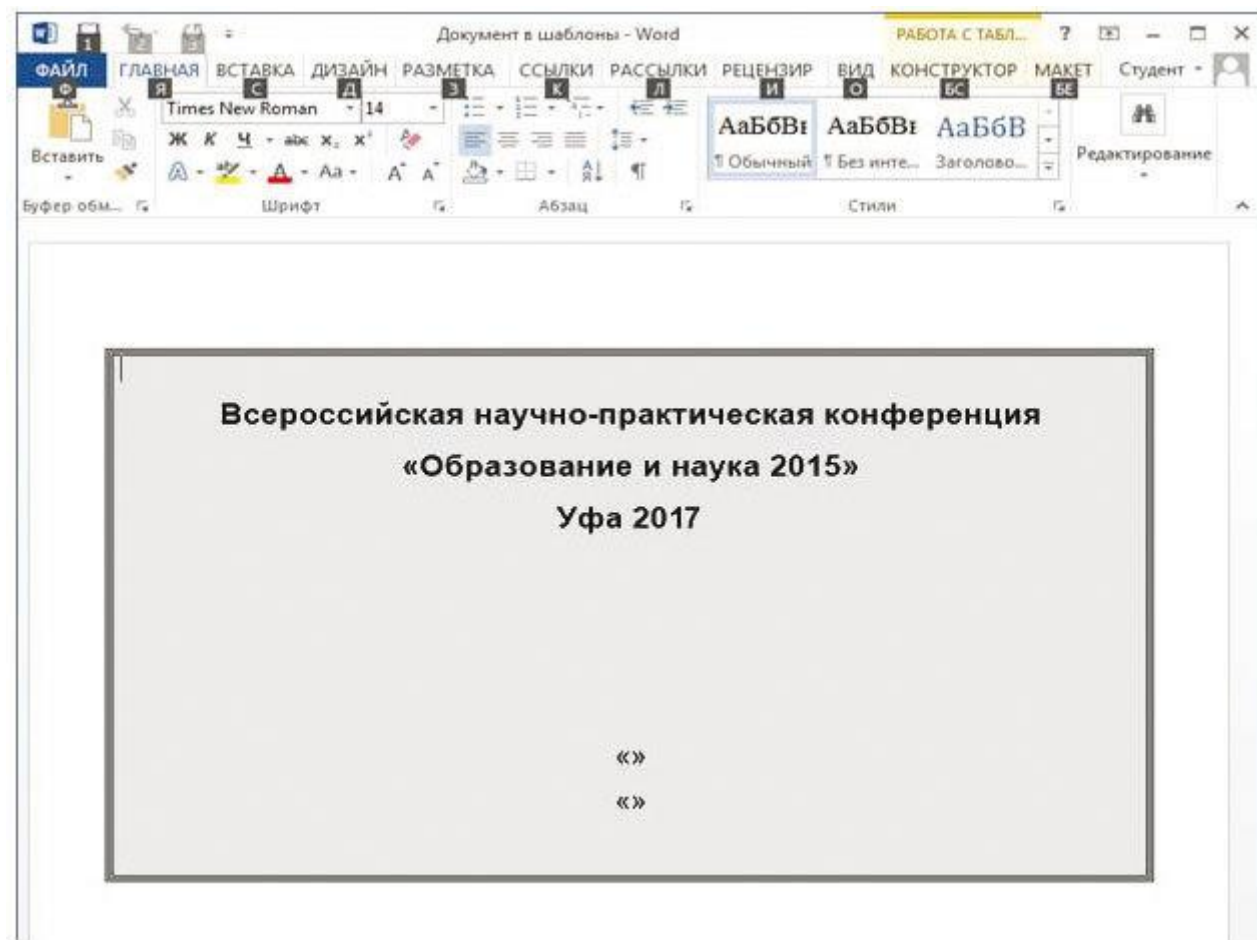


Рис. 104. Шаблон бейджика участника конференции

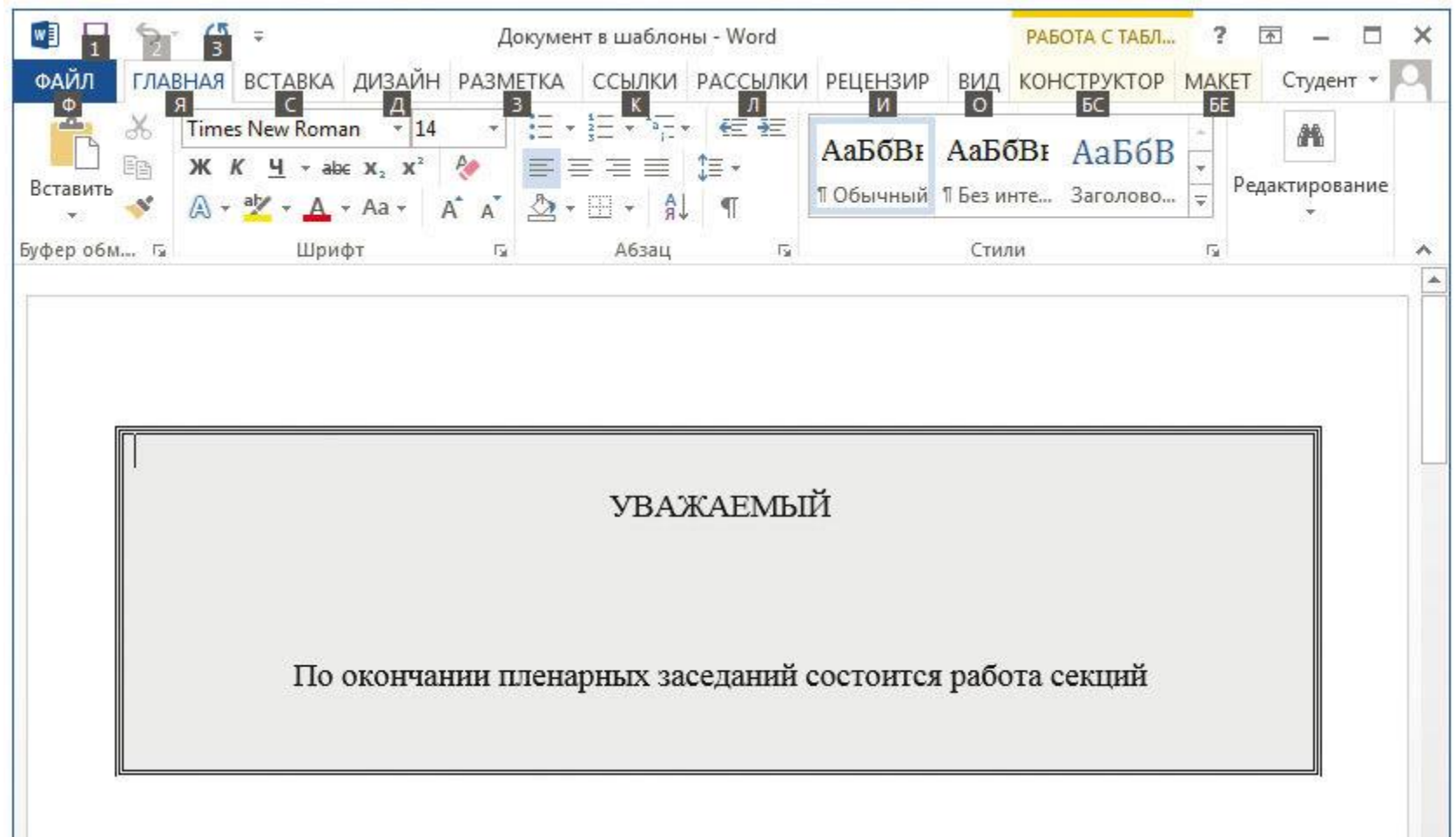


Рис. 105. Шаблон Памятки участника конференции

При работе по оформлению шаблона в необходимое место документа добавляем закладку BFIO. Имя шаблона выбираем произвольное (например, ram. dot). Полученные шаблоны являются основой взаимодействия рассматриваемых приложений. Затем переходим в СУБД и проектируем необходимую базу данных.

База данных будет состоять из 2 таблиц: таблица хранения шаблонов документов (особенно удобно, если существует несколько документов для формирования) и таблица с параметрами заполнения нашего шаблона (реквизиты участников конференции) (рис. 106).

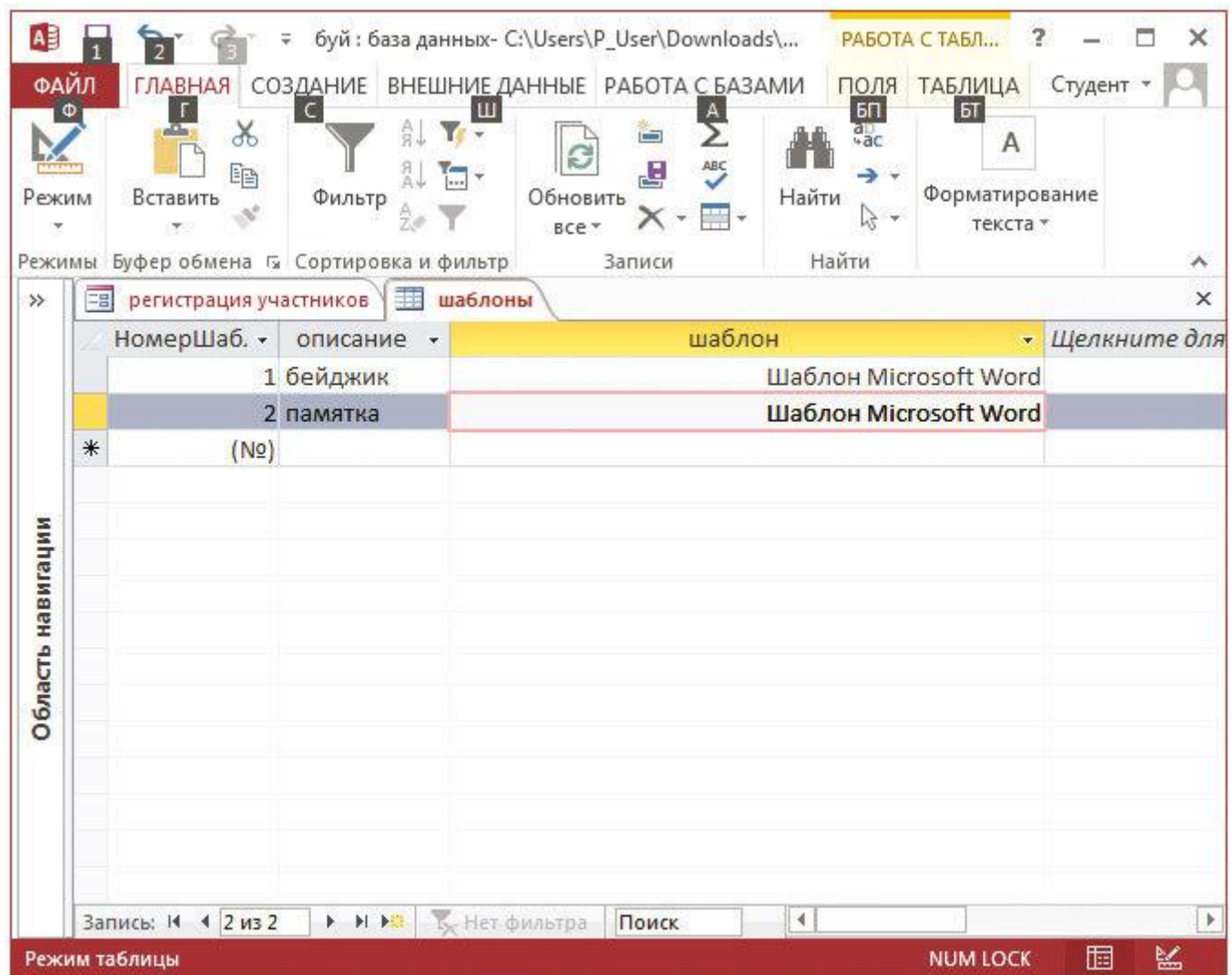


Рис. 106. База данных с шаблонами бейджика и памятки участников конференции

Для подключения шаблонов к базе данных выделяем ячейку в столбце **Шаблон** и в меню **Вставка** выбираем **Объект**. В открывшемся окне переставляем переключатель в положение **Создать из файла**, затем нажимаем кнопку **Обзор** и выбираем последовательно шаблон `C:\BD.dot`, созданный заранее. Далее нажимаем на кнопку **ОК**, чтобы поместить шаблон внутрь базы данных. Для записи 2 проделываем ту же операцию. Сначала в меню **Вставка** выбираем **Объект**, затем – **Создать из файла**, после выбираем шаблон `C:\ram.dot`.

При заполнении поля шаблона необходимые к использованию файлы помещаются в ячейки простым копированием. В результате проведенных операций каждая ячейка заполняется объектами типа шаблон в соответствии с именем шаблона и его назначением. В конструкторе это поле объекта OLE (рис. 107). Для проверки правильности выполнения этих операций можно двойным кликом запустить каждый из шаблонов и проверить правильность их срабатывания. Обязательно необходимо проследить правильность присвоения номеров записей соответствующим шаблонам.

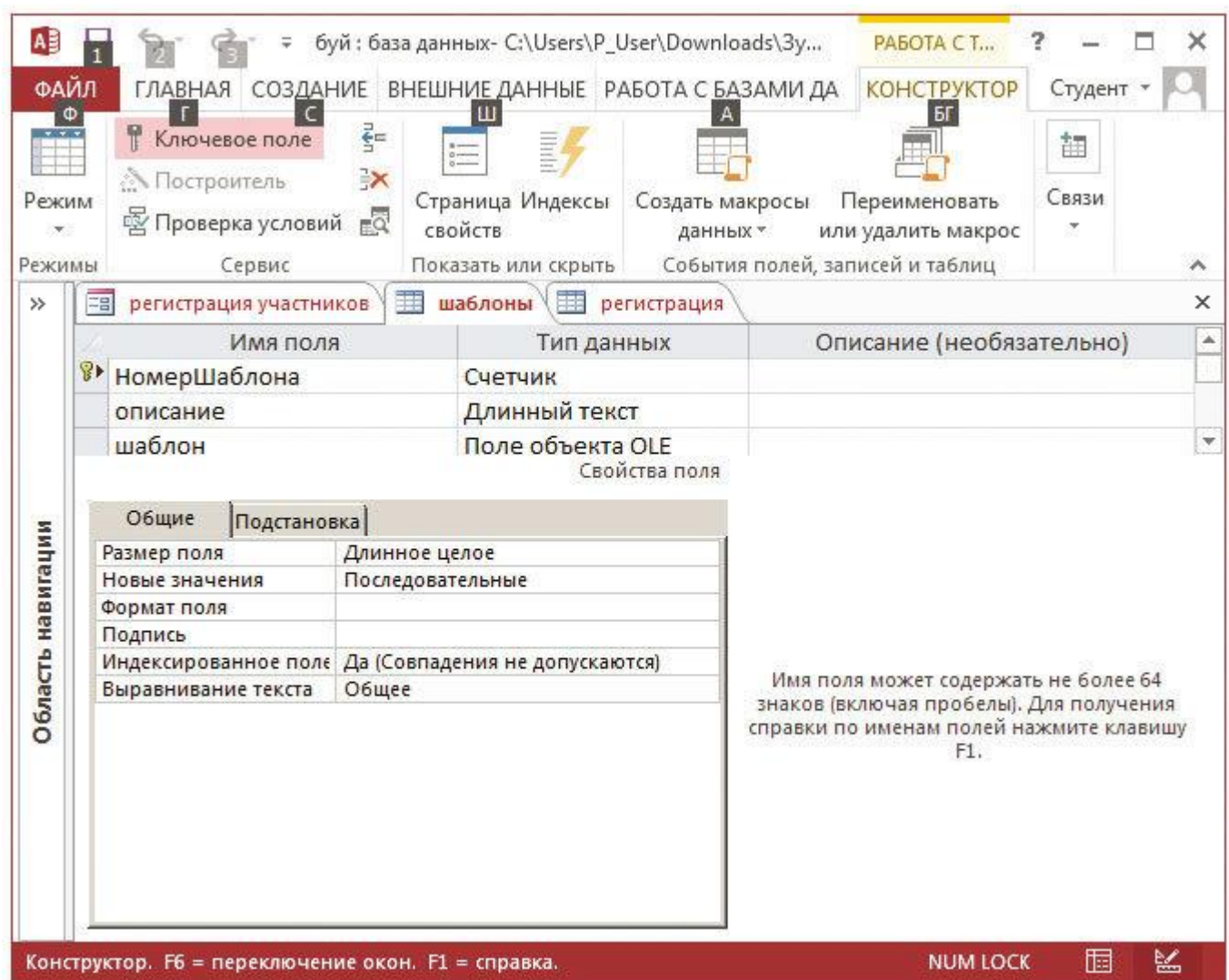


Рис. 107. Таблица «Шаблоны» в режиме конструктора

Таблица 2 заполняется данными участников конференции. Полнота заполнения определяется объемом необходимой информации. Если необходима база учета лишь регистрационных параметров участников, то таблица может состоять лишь из полей регистрации, соответствующих закладкам в шаблонах для бейджика и памятки участника. Таким образом, наша таблица состоит из трех полей типа текст с возможными настройками размера поля и маски ввода данных (рис. 108).

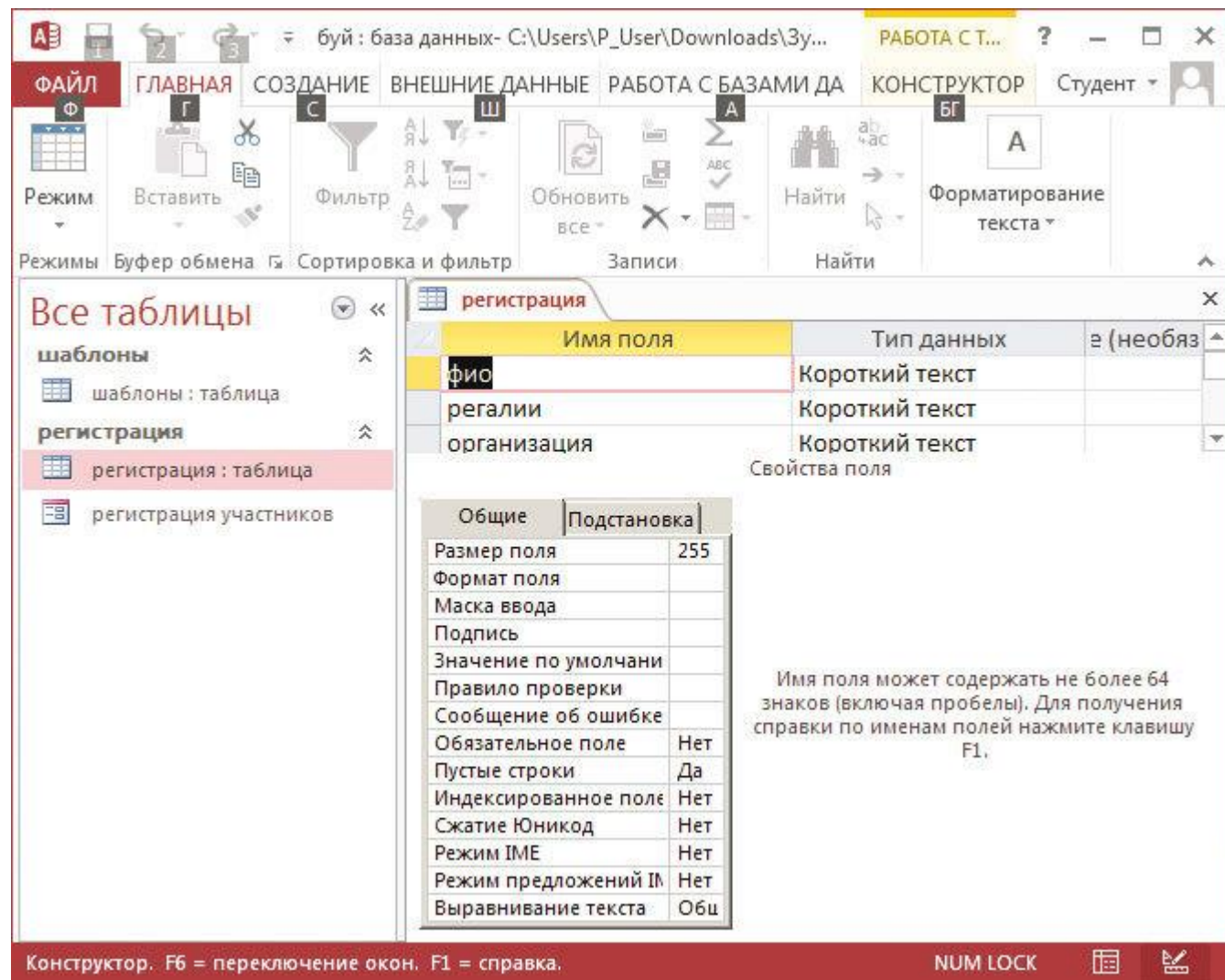


Рис. 108. Таблица «Регистрация» в режиме конструктора

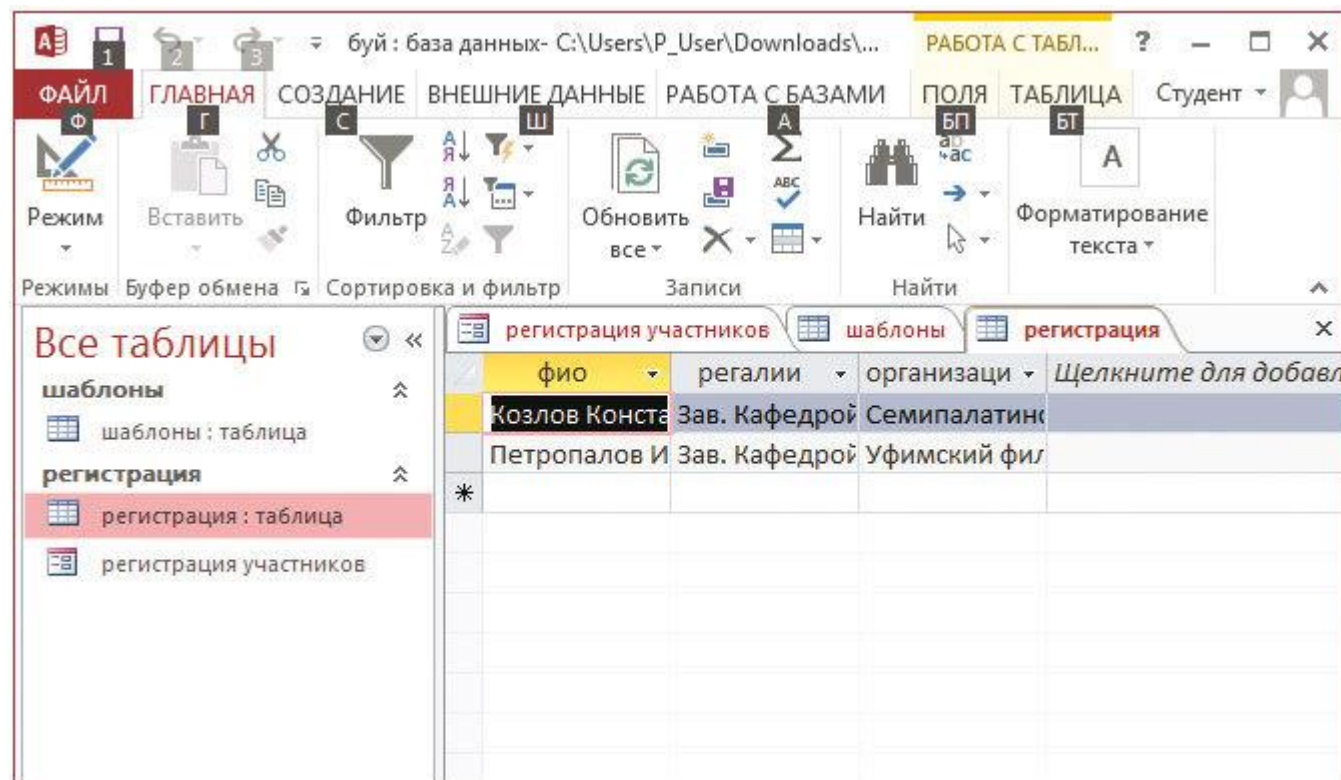


Рис. 109. Заполненная таблица «Регистрация»

После разработки соответствующих таблиц заполняем их необходимой информацией (рис. 109). Проще и полезнее для этой цели сгенерировать форму ввода исходных данных, которую в дальнейшем можно будет снабдить компонентами управления.

В окне базы данных перейдем на вкладку **Формы** и щелкнем два раза мышью по строке **Создание формы с помощью мастера**. Откроется окно мастера создания форм.

На первом экране мастера в списке **Таблицы и Запросы** выберем **Таблицу: регистрация**, затем поместим в список **Выбранные поля** все поля из этой таблицы и нажмем на кнопку **Далее**.

На следующем экране выберем внешний вид формы в один столбец и нажмем на кнопку **Далее**.

На следующем экране выберем наиболее понравившийся стиль и нажмем на кнопку **Далее**.

На последнем экране в окне определения имени формы введем имя формы **Форма для занесения договоров**, переставим переключатель в положение **Изменить макет формы** и нажмем на кнопку **Готово**. Форма будет открыта в режиме конструктора.

Произведем расстановку и дооформление созданных элементов на форме средствами конструктора по нашему вкусу.

Добавим при помощи **Toolbox** на свободную часть формы элемент управления **Присоединенная рамка объекта**. Удаляем автоматически сгенерированную вместе с ним надпись, а затем откроем свойства этого объекта. Для свойства **Имя** настроим значение **OLEObject1**, а для свойства **Вывод** на экран установим значение **Нет**.

Добавим на форму три кнопки: **Отмена**, **Бейжик** и **Памятка**. Элемент управления для первой кнопки назовем **cmdCancel**, для второй – **cmdReg**, а для третьей **cmdPam**. В открывающемся окне мастера при создании кнопки нажимаем на кнопку **Отмена** (рис. 110).

Убедимся, что для элементов управления текстовых полей оставлены имена по умолчанию (**ФИО**, **Регалии**, **Организация**). В итоге форма в окне конструктора должна выглядеть, например, так, как представлено на рисунке 111.



Рис. 110. Диалоговая форма базы данных в режиме конструктора

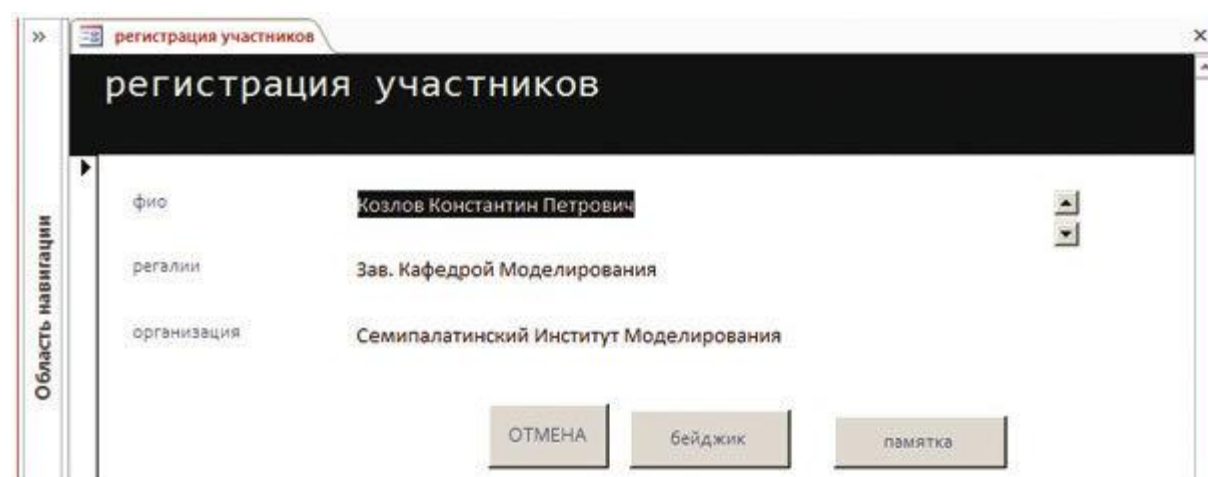


Рис. 111. Готовая форма диалога базы данных

Щелкнем правой кнопкой мыши по кнопке **Отмена** и в контекстном меню выберем **Обработка событий**. В открывшемся окне **Построитель** выберем **Программы** и нажмем **ОК**. Введем следующий код для события **Click** этой кнопки:

```
Private Sub cmdCancel_Click()  
Form. Undo  
End Sub
```

При помощи меню **Tools** → **References** в окне редактора кода добавим ссылку на объектную библиотеку Microsoft Word 12.0 Object Library.

Откроем код для события **Click** для кнопки **Бейджик** и введем следующий код:

```
Private Sub cmdDog_Click()  
Dim dDate As Date  
Dim ФИО, регалии, организация As String
```

Присваиваем значения переменным при помощи элементов управления формы
If Form. Controls(«ф.и.о»). Value <> «» Then _
Город = Form. Controls(«ф.и.о»). Value

```
If Form. Controls("Организация"). Value <> «» Then _ Организация = Form. Controls(«Организация»). Value If Form. Controls("пегалии»). Value <> «» Then _ Представитель = Form. Controls("пегалии"). Value
```

```
'Получаем шаблон – теперь из базы данных
Dim oBOF As BoundObjectFrame
Set oBOF = Form. Controls("OLEObject1 ")
oBOF = DLookup("[Шаблон]", «Шаблоны», «[НомерШаблона] = 1»)
oBOF. Verb = acOLEVerbOpen oBOF. Action = acOLEActivate
```

```
'Получаем ссылки на запущенный нами Word и открытый в нем документ
Dim oWord As Word. Application
Set oWord = GetObject( "Word. Application")
Dim oDoc As Word. Document
Set oDoc = oWord. ActiveDocument oWord. Visible = True oWord. ActiveWindow. WindowState = wdWindowStateMaximize oDoc. Activate
```

```
'Вставляем данные в закладки oDoc. Bookmarks(«bfi о»). Range. Text = фюо oDoc. Bookmarks("bdol"). Range. Text = пегалии
oDoc. Bookmarks("borg"). Range. Text = организация
```

```
EndSub
```

Откроем код для события **Click** для кнопки **Памятка** и введем следующий код:

```
Private Sub cmdPam_Click()
```

```
Dim dDate As Date
```

```
Dim ФИО, пегалии, организация As String
```

```
'Присваиваем значения переменным при помощи элементов управления формы
```

```
If Form. Controls(«фюо»). Value <> «» Then _
```

```
Город = Form. Controls(«фюо»). Value
```

```
If Form. Controls("Организация"). Value <> «» Then _
```

```
Организация = Form. Controls(«Организация»). Value
```

```
If Form. Controls("пегалии»). Value <> «» Then _
```

```
Представитель = Form. Controls("пегалии"). Value 'Получаем шаблон – теперь из базы данных
```

```
Dim oBOF As BoundObjectFrame
```

```
Set oBOF = Form. Controls("OLEObject1 ")
```

```
oBOF = DLookup("[Шаблон]", «Шаблоны», «[НомерШаблона] = 2»)
oBOF. Verb = acOLEVerbOpen oBOF. Action = acOLEActivate
```

```
'Получаем ссылки на запущенный нами Word и открытый в нем документ
```

```
Dim oWord As Word. Application
```

```
Set oWord = GetObject( "Word. Application")
```

```
Dim oDoc As Word. Document
```

```
Set oDoc = oWord. ActiveDocument oWord. Visible = True oWord. ActiveWindow. WindowState =
```

```
wdWindowStateMaximize oDoc. Activate
```

```
'Вставляем данные в закладки oDoc. Bookmarks(«bfi о»). Range. Text = фюо
```

```
EndSub
```

Запустите созданный нами код на выполнение и убедимся в его работоспособности. В результате должен получиться текстовый документ – Бейджик (рис. 112).



Рис. 112. Образец бейджика

Заполненный бейджик данными из базы и текстовый документ памятка участника (рис. 113).

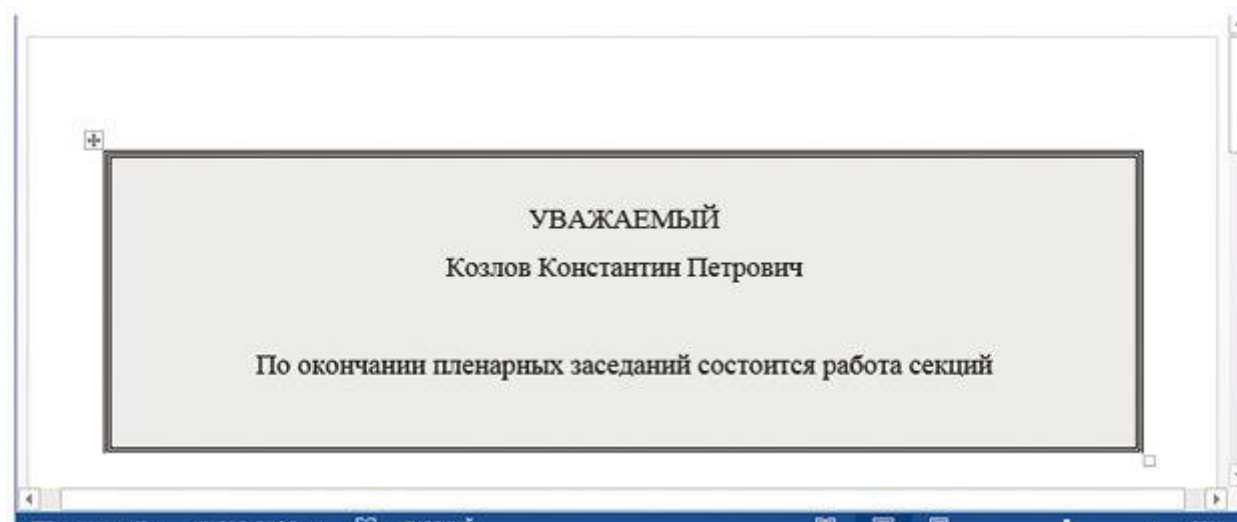


Рис. 113. Памятка участника конференции, заполненная данными из базу

В заключении следует отметить, что эффективная работа в программном комплексе MS Office во многом зависит от уровня подготовки пользователя. Инструментарии пакета позволяют реализовывать самые разнообразные пожелания заказчика, а не только математические расчеты и статистическую обработку данных. Пользователю, владеющему начальными навыками VBA данная задача не должна представлять сложность.

10. Создание массового документа. Создание бейджиков. Создание наклеек на конверты для рассылки материалов с использованием технологии слияния документов

Для создания массового письма можно использовать и другие инструменты, которые интегрированы в приложения MS Office. В нашем случае мы воспользуемся вкладкой «Рассылки» тестовом процессоре Word и данными из электронной таблицы учетного приложения. В качестве источника данных для вставки в подготовленный шаблон письма, наклейки, конверта и т. п. (документа, подготовленного в MS Word) могут служить данные в табличной форме заранее подготовленные или в текстовом процессоре или в табличном процессоре и данные из таблицы базы данных. Также следует отметить, что данные могут быть введены и непосредственно в процессе слиянии шаблона и источника данных.

Работать с инструментом «Рассылки» очень просто: надо последовательно перемещаться по панелям инструментов вкладки «Рассылки» слева – направо. Некоторые позиции панелей активизируются только после выбора параметра в предыдущей, слева, позиции. На рисунке 114 показана позиция выбора рассылаемого формата документа: Письмо, Сообщения электронной почты, Конверты, Наклейки, Каталог.

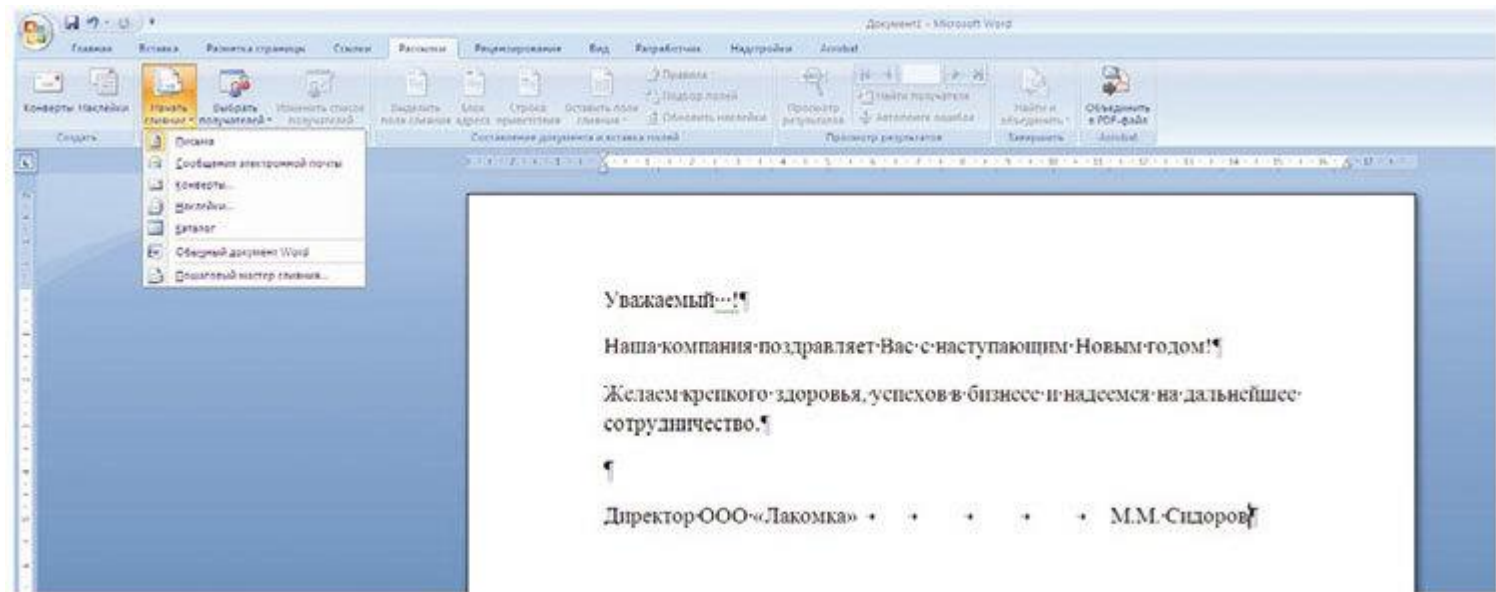


Рис. 114. Выбор формата рассылаемого документа

Далее идет кнопка «Выбрать получателя». Её позиции позволяют или ввести новый список, или использовать существующий список или выбрать из контактов Outlook (рис. 115).

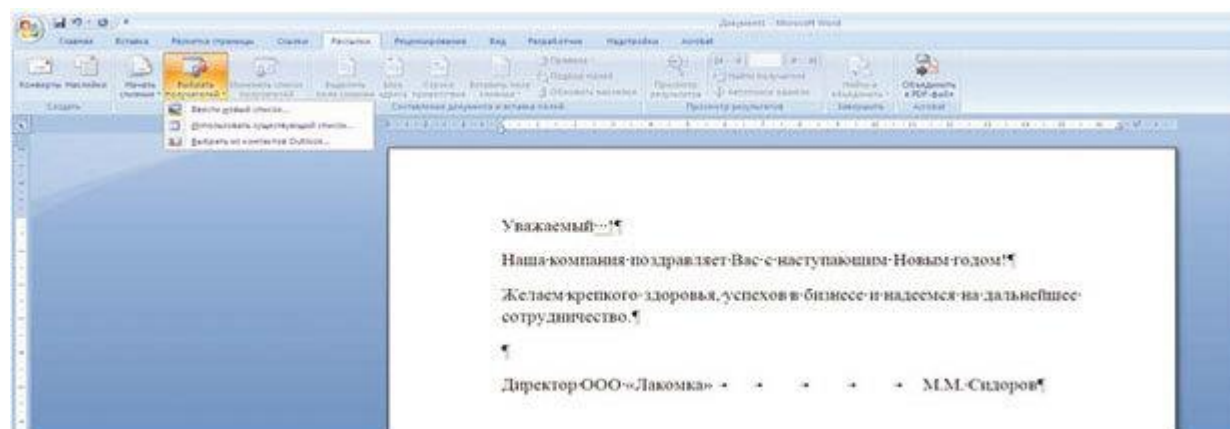


Рис. 115. Выбор источника данных

В нашем примере воспользуемся данными из учетного приложения по продаже кондитерских изделий. Для этого найдем файл приложения в файловой системе нашего компьютера и откроем его двойным щелчком мыши и увидим следующее (рис. 116).

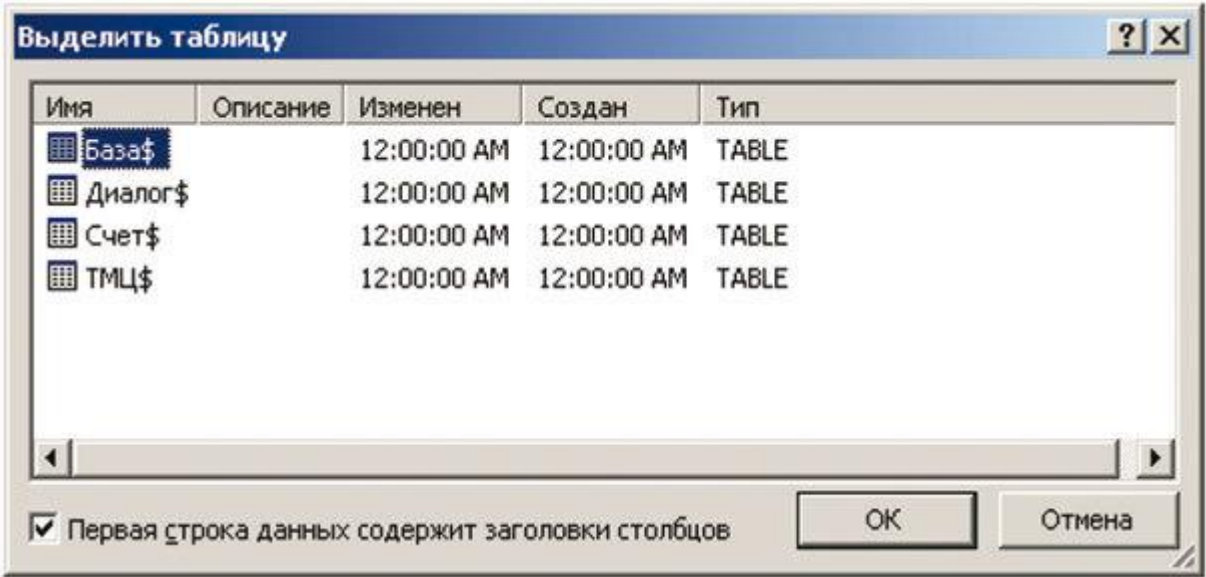


Рис. 116. Листы учетного приложения

Оставим только интересующие строки электронной таблицы (рис. 117)

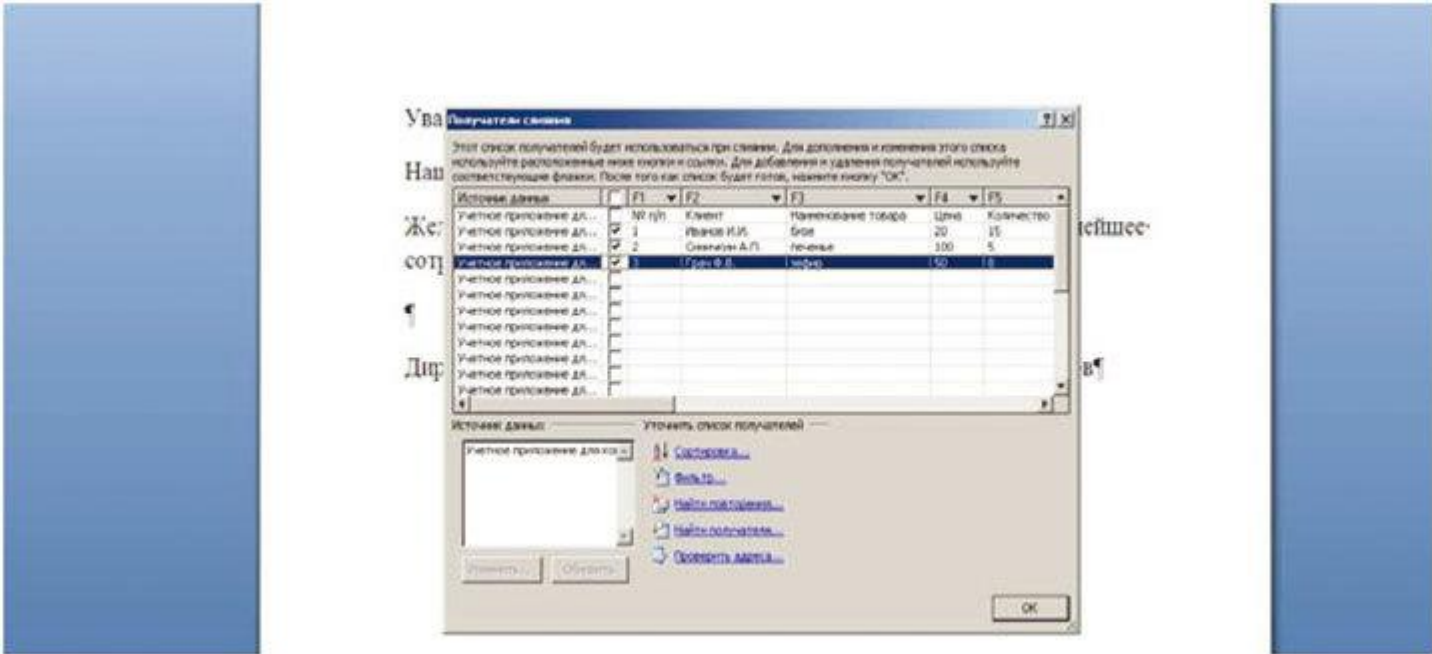


Рис. 117. Строки электронной таблицы для слияния

Вот как эти данные выглядят в электронной таблице – оригинале (рис. 118).

| | | | | | | | | | |
|--|-------|---------------|---------------------|------|------------|---------------------|------------|---|---|
| Microsoft Excel - Учетное приложение для кондитерской организации Харак А.А. 12ИПБ3БУ.xlsm | | | | | | | | | |
| Файл Правка Вид Вставка Формат Сервис Данные Служба Справка Adobe PDF | | | | | | | | | |
| D7 fx | | | | | | | | | |
| | A | B | C | D | E | F | G | H | I |
| 1 | | | | | | | | | |
| 2 | № п/п | Клиент | Наименование товара | Цена | Количество | Сумма к оплате, руб | Дата | | |
| 3 | 1 | Иванов И.И. | бизе | 20 | 15 | 300 | 05.02.2017 | | |
| 4 | 2 | Синичкин А.П. | печенье | 100 | 5 | 500 | 10.02.2017 | | |
| 5 | 3 | Грач Ф.В. | зефир | 50 | 8 | 400 | 16.02.2017 | | |
| 6 | | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |

Рис. 118. Исходные данные в таблице – регистре

Далее выбираем, какой столбец (см. рис. 117) выбираем для занесения в бланк после слова «Уважаемый». В нашем случае это F2 (рис. 119).

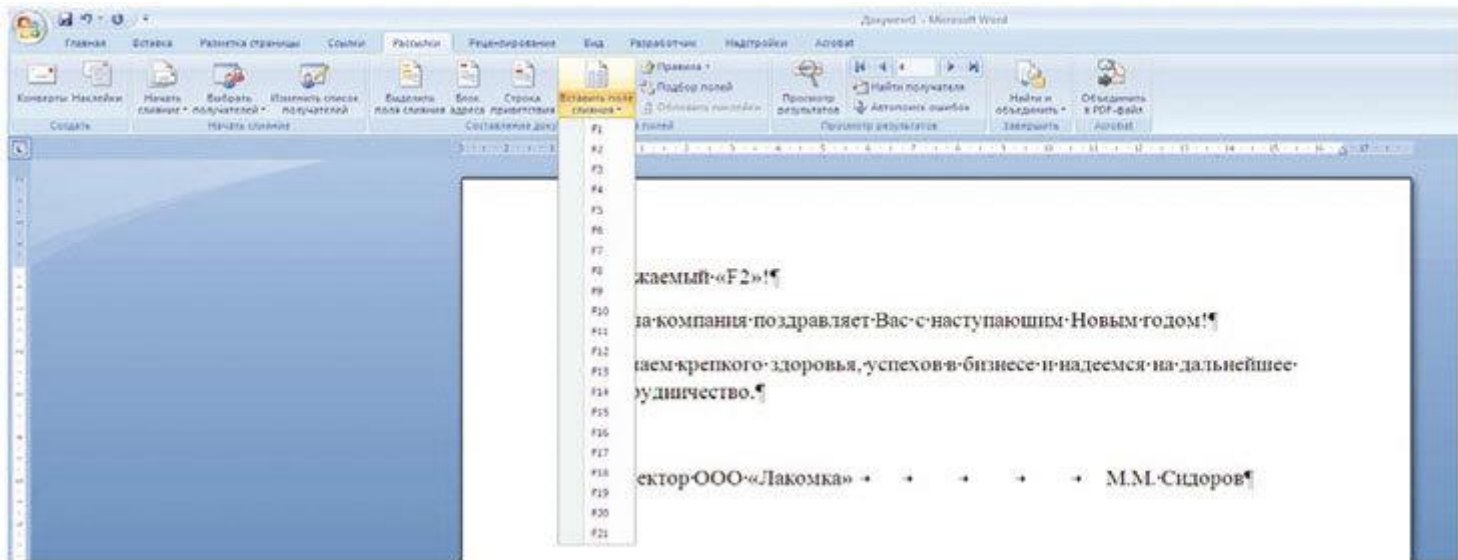


Рис. 119. Выбор столбца из таблицы исходных данных

После нажатия на кнопку «Просмотр результата» произойдет слияния подготовленного бланка-шаблона исходных данных (рис. 120).

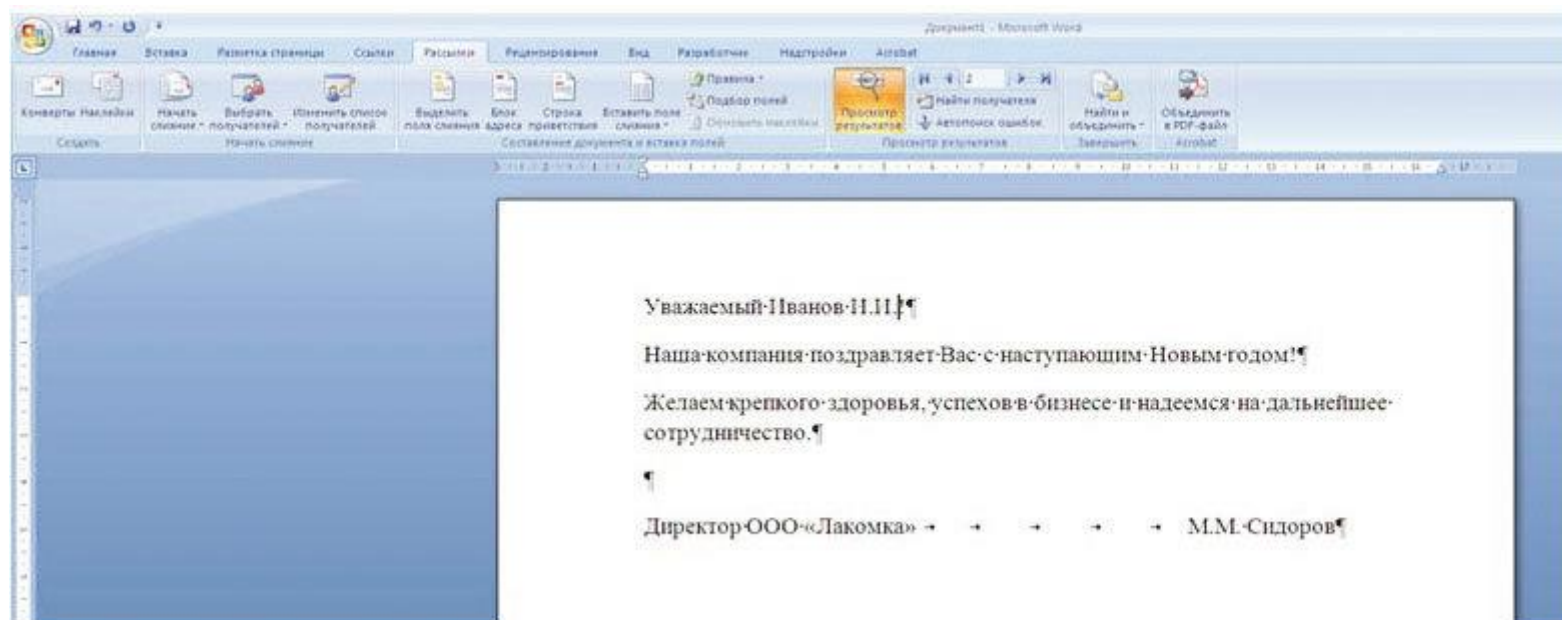


Рис. 120. Результат слияния шаблона поздравительного письма и данных из учетного приложения

На приведенном примере видно, что многие задачи могут быть решены различными способами: как с использованием встроенного языка программирования VBA, так и с использованием инструментария текстового и табличного процессоров и других приложений пакета MS Office. Все зависит от того, какие условия задачи, каковы потребности уровня автоматизации необходим для решения прикладной задачи; каковы сроки исполнения основной, прикладной задачи, что хотим увидеть на выходе при создании нашего приложения; каковы должны быть возможности приложения по автоматизации решения прикладной задачи – все это влияет на глубину проработки учетного приложения, реализованного в нем функционала и уровня использования программного кода.

Контрольные вопросы

1. Каковы инструменты разработки макросов?
2. Перечислите основные типы шаблонов реализации документов в MS Word. Можно ли разработать собственный шаблон?
3. Каков алгоритм разработки макроса в текстовом редакторе MS Word, чем он отличается от алгоритмов других программных средств?
4. Какие элементы управления можно назначить макросу в MS Word? Чем отличаются эти элементы управления?
5. Как загрузить редактор VBA в MS Word? Какие функции он выполняет?
6. Можно ли настроить инструменты взаимодействия между MS Word и СПС «Консультант+»? Каким образом реализовать это взаимодействие?
7. Что такое закладка в тексте? Как она реализуется и для чего используется в MS Word?
8. Каков алгоритм разработки формы в MS Word? Какие элементы управления можно включать в разрабатываемые формы?
9. Что такое специальная вставка в MS Word? Как выбирается тип документа при организации связи?
10. Как создается кнопка управления макросом? Выбор кнопки из библиотеки пиктограмм.

Практические задания

| № Задания | Содержание задания |
|--------------|---|
| | Разработать учетное приложение поддержки системы работы с договорами. Учет ведется по параметрам: дата заключения договора и исполнитель. |
| | Разработать учетное приложение поддержки системы работы с договорами. Учет ведется по параметрам: банковские реквизиты и ИНН исполнителя. |
| | Разработать учетное приложение поддержки системы работы с договорами. Учет ведется по параметрам: Имя заказчика и Имя руководителя Заказчика. |
| | Разработать учетное приложение поддержки системы работы с договорами. Учет ведется по параметрам: Основание заказчика и Имя руководителя заказчика. |
| | Разработать учетное приложение поддержки системы работы с договорами. Учет ведется по параметрам: реквизитов заказчика и реквизитов исполнителя. |
| | Разработать учетное приложение совместного автоматического заполнения документов, прилагаемых к договору: акт выполненных работ и отчет по процедуре исполнения заказа. |
| | Разработать учетное приложение совместного автоматического заполнения документов, прилагаемых к договору: товарная накладная и перечень поставленных товаров. |

| № Задания | Содержание задания |
|--------------|---|
| | Разработать учетное приложение совместного автоматического заполнения документов, прилагаемых к договору: техническое задание и акт согласования изменений. |
| | Разработать учетное приложение совместного автоматического заполнения документов, прилагаемых к договору: накладная и акт согласования поставки комплектующих. |
| | Разработать учетное приложение совместного автоматического заполнения документов, прилагаемых к договору: требование предпроектных исследований и акт согласования этапов разработки. |

Заключение

В данном пособии были рассмотрены вопросы создания ученых приложений. В начале пособия рассмотрены основные требования к учетному приложению, его составные части и его отличие от комплекта электронных таблиц.

Отдельными разделами выделены вопросы, обозначенные в рабочей программе дисциплины «Разработка учетных приложений в MS Office» изучаемой студентами на направлении «Экономика» Финансового университета при Правительстве РФ.

Показаны примеры реализации учетных приложений как в отдельном офисном приложении, так и комплексное использование офисных приложений для решения прикладных задач различной сфер деятельности.

Во всех примерах показан реальный программный код на языке программирования VBA, что также поможет читателю его легче понять, а может и в дальнейшем использовать его фрагменты в своих задачах.

В пособии приведены примеры альтернативной подхода к решению ряда задач и этапов реализации учетного приложения.

Предложения авторов к созданию учетных приложений носит рекомендательный характер, абсолютно не исключает творческого похода читателя к решению вопроса автоматизации своей прикладной задачи.

Современное информационное общество требует от специалистов финансового профиля всесторонних знаний не только по предметной области, но и в области информационных технологий. Особенно это касается процессов внедрения новых решений и использования новых инновационных технологий. Вопросам использования цифровых технологий в различных сферах бизнеса в нашей стране уделяют особенное, большое внимание. На современном этапе необходимо обладать компетенциями не только по своей прикладной предметной области, но в аспектах внедрения и разработки программного обеспечения, в том числе в вопросе малой автоматизации рабочего места – в создании учетных приложений. Решение задач контекстного программирования учетных приложений позволяет не только упростить работу каждого отдельного пользователя, но и оптимизировать работу информационной среды предприятия в целом.

Знания, полученные в процессе изучения данного учебного пособия, позволяют бакалаврам экономических направлений более конкретно подходить как к созданию учетных приложений, так и автоматизации других задач прикладного характера.

Знание процесса создания учетного программного обеспечения и инструментария для его создания позволят подходить к вопросам автоматизации более профессионально, а решение задач информатизации корпоративных процессов сделать более качественным.

Список литературы

1. Лукьянов П. Б. Разработка учетных приложений в MS Office: Учебное пособие. – М.: КНОРУС, 2015. – 56 с.
2. Гарнаев А. Ю., Рудикова Л. В. Microsoft Office Excel 2010: разработка приложений. – СПб.: БХВ-Петербург, 2011. – 528 с. (ЭБС Znanium).
3. Слепцова Л. Д. Программирование на VBA в Microsoft Office 2010. – М.: Вильямс, 2010. – 432с.
4. Гобарева Я. Л., Городецкая О. Ю., Золотарюк А. В. Бизнес-аналитика средствами Excel: Учебное пособие. – М.: Вузовский учебник: ИНФРА-М, 2014. – 336 с. (ЭБС Znanium).
5. Киселев Г. М., Бочкова Р. В., Сафонов В. И. Информационные технологии в экономике и управлении (эффективная работа в MS Office 2007): Учебное пособие. – М.: Дашков и К°, 2013. – 272 с. (ЭБС Znanium).