РОСТИСЛАВ МИХЕЕВ

VBA И ПРОГРАММИРОВАНИЕ В MS OFFICE ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

СПЕЦИАЛЬНЫЙ КУРС

ЯЗЫК VBA

ОБЪЕКТНЫЕ МОДЕЛИ WORD, EXCEL, ACCESS, OUTLOOK, POWERPOINT, PROJECT

ВЗАИМОДЕЙСТВИЕ С БАЗАМИ ДАННЫХ

РЕШЕНИЕ РЕАЛЬНЫХ ЗАДАЧ ОТЕЧЕСТВЕННЫХ ПРЕДПРИЯТИЙ

Ростислав Михеев

VBA
И ПРОГРАММИРОВАНИЕ В MS OFFICE
ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

Санкт-Петербург
«БХВ-Петербург»
2006

УДК 681.3.06
ББК 32.973.26-018.2
М69

Михеев Р. Н.

М69 VBA и программирование в MS Office для пользователей. — СПб.: БХВ-Петербург, 2006. — 384 с.: ил.

ISBN 5-94157-863-6

В основу книги положен материал учебного курса "Программирование в Microsoft Office для пользователей", который в течение нескольких лет читается сотрудникам крупнейших предприятий России. Рассмотрено программирование на языке VBA с использованием возможностей объектных моделей приложений Microsoft Office. Описан синтаксис языка VBA, основные приемы работы с редактором кода, впервые подробно рассматриваются объектные модели основных приложений Microsoft Office: Word, Excel, Access, Outlook, PowerPoint, Project. Материал сопровождается многочисленными практическими примерами. К каждой главе книги предусмотрены задания для самостоятельной работы с подробными решениями.

Для пользователей MS Office

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Нина Седых*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн обложки	*Инны Тачиной*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.11.05.
Формат $70\times100^{1}/_{16}$. Печать офсетная. Усл. печ. л. 30,96.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-863-6

Оглавление

Предисловие

Несколько лет назад автору — сертифицированному преподавателю Microsoft — поступил заказ: подготовить курс по программированию в Office. Задача была простая: существует группа из нескольких десятков человек, достаточно продвинутых пользователей, которые не имеют никакого опыта программирования. Одни пользователи занимались анализом трафика базовых станций в сети (заказчик был оператором сотовой связи российского масштаба), другие — проектами по развертыванию тех же базовых станций, третьи все это планировали и прогнозировали и т. п. И многие пользователи обращались к IT-подразделению с просьбой автоматизировать выполнение определенных задач, например:

- загрузку в Excel информации из базы данных SQL Server, дальнейший анализ (например, выявление тенденций) и представление результатов в стандартном виде;
- автоматическое создание сводных таблиц и графиков в Excel;
- проход по всем проектам (300—400 единиц) на Project Central (программное средство для корпоративной работы с проектами Microsoft Project) и замена в них каких-то элементов;
- создание стандартных документов Word, в которые бы подставлялись данные из базы данных.

И, конечно, список задач этим не ограничивался.

На этом предприятии были очень квалифицированные, но постоянно занятые программисты. Конечно, они отзывались на просьбы пользователей, но фактически намного больше времени уходило на постановку задачи, чем на ее решение. А через некоторое время задача вполне могла слегка измениться (например, нужно было ввести дополнительную ось на графике в Excel или поменять пару строк в шаблоне Word), и сотрудникам приходилось снова обращаться к разработчикам.

В результате мудрое руководство предприятия пришло к выводу, что проще научить пользователей автоматизировать свою работу самостоятельно, и поэтому был заказан этот курс.

Подготовка любого учебного курса — дело очень трудоемкое, и поэтому автор (у которого к тому времени был опыт преподавания более чем 30 официальных курсов Microsoft) постарался подобрать что-нибудь подходящее из официальных курсов Microsoft Official Curriculum (MOC). Однако его ждала неудача: у Microsoft были предусмотрены только курсы по Access различных версий плюс мини-курсы по очень специфическим вопросам: приложения коллективного использования (курс 2381), решения управления знаниями (курс 1904) и т. п. Сами курсы были скорее обзорными и предназначались для того, чтобы познакомить опытных разработчиков с новыми технологиями.

Тогда автор обратился к книжным полкам. Был куплен десяток книг, которые имели отношение к теме "Программирование в Office", но ни по одной из них сделать учебный курс было нельзя. Некоторые книги под программированием в Office понимали использование математических и финансовых функций в Excel, другие ограничивались рассмотрением элементарных программных конструкций VBA, в-третьих полкниги отводилось на объяснение основ объектно-ориентированного программирования (при этом в реальной работе на VBA определять пользовательские классы приходится достаточно редко). И ни одна из книг не давала возможность "увидеть за деревьями лес" — получить целостное представление об объектных моделях Word, Excel, Access, PowerPoint, Outlook и Project, чтобы дать возможность пользователю самостоятельно находить в них нужные объекты и создавать свои приложения.

Автору пришлось подготавливать этот курс самостоятельно. Был собран и прочитан весь возможный материал, скачано из Интернета и проанализировано несколько сотен приложений VBA, законспектирована официальная документация. Вместе с сотрудниками предприятий решались их проблемы, которые возникали на практике. Уже первый вариант курса был признан заказчиком (и его сотрудниками) очень удачным, а постепенно курс совершенствовался. Вместе с пользователями с разных предприятий на него иногда попадали (из любопытства) опытные разработчики, которые сами рассказывали немало интересного. Постепенно собирался новый материал. За счет общения с большим количеством слушателей удалось отобрать те моменты, которые наиболее важны в практической работе.

Результат — перед вами.

Эта книга отличается от многих других следующим:

- она "отлажена" на десятках слушателей с самых разных предприятий. Каждое замечание учитывалось и отражалось в курсе (а потом и в книге),

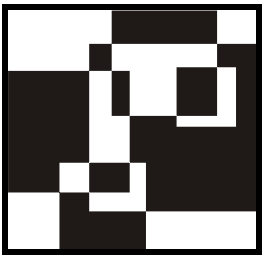
внимание акцентировалось на тех моментах, которые действительно важны для практической работы;

- ❒ учесть все многообразие ситуаций, которые возникают на предприятиях, невозможно. Поэтому в данной книге был сделан акцент не на рассмотрение отдельных случаев ("как вставить диаграмму в лист Excel"), а на том, как самостоятельно найти решение в подобной ситуации;
- ❒ для каждого приложения Office дается общая картина того, как устроена его объектная модель и из каких важнейших объектов она состоит (например, в Word это `Application` — `Document` — `Selection`, `Range` и `Bookmark`, в Excel `Application` — `Workbook` — `Worksheet` — `Range`). Знание этих программных объектов на 80% покрывает потребности при создании реальных приложений VBA;
- ❒ поскольку изначально эта книга была учебным курсом, было бы жаль не использовать некоторые преимущества, которые это дает. Для большинства глав предусмотрены задания для самостоятельной работы. Можно попытаться решить их самостоятельно, а можно использовать их просто как дополнительные примеры. После каждого задания приводится подробное решение с комментариями.

Несколько слов о технических моментах. Все примеры в книге приведены для приложений Microsoft Office 2003 в русскоязычной версии. Практически весь материал применим и к программированию в Office XP. Значительно больше отличий в Office 97 и в Office 2000, хотя основные моменты остаются неизменными во всех версиях. Несмотря на то, что в книге рассмотрены все встроенные функции текущей версии VBA и все главные объекты приложений Office текущей версии, сделан акцент на том, чтобы читатель понял, как можно самостоятельно найти необходимую информацию. Поэтому автор надеется, что книга пригодится и для работы с будущими версиями Microsoft Office, и с программными продуктами других фирм, в которых реализована поддержка языка VBA.

Автор совершенно не возражает против использования этой книги в качестве учебного пособия при проведении курсов в других учебных центрах или при организации обучения на предприятиях. Пользуйтесь на здоровье! И, если понравится, обращайтесь в нашу Академию за другими курсами. Наш адрес электронной почты — **info@askit.ru**, адрес Web-сайта — **www.askit.ru**. Мы с удовольствием проведем для вас курс из числа уже готовых или создадим новый учебный курс (в особенности по нестандартной тематике). И, может быть, из такого заказа возникнет новая книга — подобно тому, как появилась эта.

ГЛАВА 1



Основы программирования в Microsoft Office

1.1. Зачем программировать в Microsoft Office

Ответ на этот вопрос прост: чтобы не делать лишней работы. Программирование в Office — это, прежде всего, уменьшение количества повторяющихся действий (и ручной работы, которая для этого требуется). Вот примеры некоторых типичных ситуаций, когда использование программирования просто напрашивается:

- вам с определенной периодичностью приходится изготавливать документы, очень похожие друг на друга: приказы, распоряжения в бухгалтерию, договоры, отчеты и т. п. Часто информацию можно взять из базы данных, тогда использование программирования может дать очень большой выигрыш во времени. Иногда данные приходится вводить вручную, но и тогда автоматизация дает выигрыш во времени и в снижении количества ошибок;
- разновидность такой же ситуации: одни и те же данные нужно использовать несколько раз. Например, вы заключаете договор с заказчиком. Одни и те же данные (наименование, адрес, расчетный счет, номер договора, дата заключения, сумма и т. п.) могут потребоваться во многих документах: самом договоре, счете, счете-фактуре, акте сдачи выполненных работ и т. д. Логично один раз ввести эту информацию (скорее всего, в базу данных), а затем автоматически формировать (например, в Word) требуемые документы;
- когда нужно сделать так, чтобы вводимые пользователем данные автоматически проверялись. Вероятность ошибки при ручном вводе данных зависит от многих факторов, но, согласно результатам некоторых исследо-

ваний, она в среднем составляет около 2%. "Вылавливать" потом такие ошибки в уже введенных данных — очень тяжелый труд, поэтому лучше сразу сделать так, чтобы они не возникали.

В общем, любое действие, которое вам приходится повторять несколько раз, — это возможный кандидат на автоматизацию. Например, занесение сотен контактов в Outlook, или замена ресурса в десятках проектов Project, или анализ информации из базы данных за разные периоды в таблице Excel — это те ситуации, когда знание объектных моделей приложений Office спасет вас от нескольких часов или даже дней скучного труда.

Конечно, есть еще практиканты и аналогичный бесплатный трудовой ресурс, но хочется ли вам потом заниматься еще и поиском ошибок за ними? Кроме того, программирование несет и другие преимущества для сотрудника, который использует его в работе:

- повышается авторитет сотрудника в глазах руководства и других коллег;
- если программы этого сотрудника активно используются на предприятии (им самим или другими работниками), то этим самым он защищает себя от сокращений, снижения зарплаты и т. п., ведь поддерживать и изменять программы в случае необходимости будет некому.

1.2. Что такое язык VBA

Поскольку эта книга предназначена для обычных пользователей, то без объяснения этого вопроса не обойтись. Формальное определение такое.

VBA (Visual Basic for Applications) — это диалект языка Visual Basic, расширяющий его возможности и предназначенный для работы с приложениями Microsoft Office и другими приложениями от Microsoft и третьих фирм.

В принципе, при программировании в Office можно вполне обойтись и без языка VBA. Подойдет любой COM-совместимый язык, например: обычный Visual Basic, VBScript, Java, JScript, C++, Delphi и т. п. Можно использовать и .NET-совместимые языки программирования: VB.NET, C# и т. п. Вам будут доступны все возможности объектных моделей приложений Office. Например, если сохранить следующий код в файле с расширением vbs и запустить его на выполнение, то будет запущен Word, в котором откроется новый документ и будет впечатан текст:

```
Dim oWord
Set oWord = CreateObject("Word.Application")
oWord.Visible = true
oWord.Documents.Add
oWord.Selection.TypeText ("Привет от VBScript")
```

Тем не менее, VBA — это обычно самый удобный язык для работы с приложениями Office. Главная причина проста — язык VBA встроен в приложения Office, и код на языке VBA можно хранить внутри документов приложений Office: в документах Word, книгах Excel, презентациях PowerPoint и т. п. Конечно же, этот код можно запускать из документов на выполнение, поскольку среда выполнения кода VBA (на программистском сленге — *хост*) встроена внутрь этих приложений.

В настоящее время VBA встроен:

- во все главные приложения Microsoft Office — Word, Excel, Access, PowerPoint, Outlook, FrontPage, InfoPath;
- в другие приложения Microsoft, такие как Visio и Project;
- в более 100 приложений третьих фирм, например, в CorelDRAW и CorelWordPerfect Office 2000, AutoCAD и т. п.

Но есть также и множество других преимуществ.

- VBA — универсальный язык. Освоив его, вы не только получите ключ ко всем возможностям приложений Office и других, перечисленных ранее, но и будете готовы к тому, чтобы:
  - создавать полноценные приложения на Visual Basic (поскольку эти языки — близкие родственники);
  - использовать все возможности языка VBScript (это вообще "урезанный" VBA). В результате в вашем распоряжении будут универсальные средства для создания скриптов администрирования Windows, Web-страниц (VBScript в Internet Explorer), Web-приложений ASP, для применения в пакетах DTS и заданиях на SQL Server, а также для создания серверных скриптов Exchange Server и многое-многое другое.
- VBA изначально был ориентирован на пользователей, а не на профессиональных программистов (хотя профессионалы пользуются им очень активно), поэтому создавать программы на нем можно быстро и легко. Кроме того, в Office встроены мощные средства, облегчающие работу пользователя: подсказки по объектам и по синтаксису, макрорекордер и т. п.
- При создании приложений на VBA вам, скорее всего, не придется заботиться об установке и настройке специальной среды программирования и наличии нужных библиотек на компьютере пользователя — Microsoft Office есть практически на любом компьютере.
- Несмотря на то, что часто приложения VBA выполняются медленнее, чем бы вам хотелось, они нересурсоемки и очень хорошо работают, например, на сервере терминалов. Но, как правило, для программ на VBA особых требований на производительность и не ставят: для написания игр, драй-

веров, серверных продуктов он не используется. По моему опыту, возникающие проблемы с производительностью VBA-приложений — это чаще всего не проблемы VBA, а проблемы баз данных, к которым они обращаются. Если проблемы действительно в VBA (обычно тогда, когда вам требуется сложная математика), то всегда есть возможность написать важный код на C++ и обращаться к нему как к обычной библиотеке DLL или встраиваемому приложению (Add-In) для Word, Excel, Access и т. п.

- Программы на VBA по умолчанию не компилируются, поэтому вносить в них исправления очень удобно. Не нужно разыскивать исходные коды и перекомпилировать программы.

В среде программистов-профессионалов считается, что быстрее всего научиться создавать профессиональные приложения можно именно при помощи VBA и объектов приложений Office. Другие языки программирования (C++, Java, Delphi) придется осваивать намного дольше, а их возможности во многом избыточны для большинства повседневных задач, которые встречаются на любом предприятии. Кроме того, использование возможностей объектов Office (графического интерфейса, средств работы с текстом, математических функций и т. п.) позволит резко снизить трудоемкость при создании приложений.

1.3. Макрорекордер: быстрое создание макросов

В большинство программ Microsoft Office (исключая Access и FrontPage) встроено замечательное средство, которое позволит вам создавать программы, вообще ничего не зная о программировании. Это средство называется макрорекордером.

Макрорекордер, как понятно из его названия, — это средство для записи макросов. *Макрос* — всего лишь еще одно название для VBA-программы, а макрорекордер — средство для его автоматического создания.

Внимание!

Приложения Microsoft Office 2003 по умолчанию настроены так, что не позволяют запускать макросы. Поэтому перед тем, как приступать к созданию макросов, в меню **Сервис | Макрос | Безопасность** переставьте переключатель **Уровень безопасности** в положение **Средняя** или **Низкая**, а потом закройте и снова откройте данное приложение. Это потребуется сделать только один раз в начале работы.

Принцип работы макрорекордера больше всего похож на принцип работы магнитофона: мы нажимаем на кнопку — начинается запись тех действий,

которые мы выполняем. Мы нажимаем на вторую кнопку — запись останавливается, и мы можем ее проиграть (т. е. повторно выполнить ту же последовательность действий).

Конечно, макрорекордер позволяет написать только самые простые VBA-программы. Однако и он может принести много пользы. Например, можно "положить" на горячие клавиши те слова, словосочетания, варианты оформления и т. п., которые вам часто приходится вводить (должность, название фирмы, продукт, ФИО директора и ответственного исполнителя и т. д.), этим вы сэкономите много времени.

Как показывает опыт, подавляющее большинство обычных пользователей и не подозревает о существовании макрорекордера, несмотря на то, что его применение позволило бы сэкономить им массу времени.

Перед созданием макроса в макрорекордере:

- ❒ необходимо очень тщательно спланировать макрос, хорошо продумав, что вы будете делать и в какой последовательности. Если есть возможность, определите подготовительные действия. Например, если нужно вставить текущую дату в начало документа, может быть, имеет смысл первой командой макроса сделать переход на начало документа (<Ctrl>+<Home>);
- ❒ посмотрите, нет ли готовой команды, которую можно сразу назначить клавише или кнопке на панели инструментов без создания макроса. Сделать это можно при помощи меню **Сервис | Настройка**. С вкладки **Команды** можно перетащить нужную команду на требуемую панель управления, и, нажав на этой же вкладке кнопку **Клавиатура**, в окне **Настройка клавиатуры** назначить для команды нужную комбинацию клавиш;
- ❒ если вы собираетесь при помощи макроса менять оформление текста, то правильнее вначале создать новый стиль с вашим оформлением, а потом уже применить этот стиль к тексту. В этом случае опять-таки можно обойтись без макроса, просто назначив стиль комбинации клавиш. Делается это при помощи того же диалогового окна **Настройка клавиатуры**, как и в предыдущем случае.

Чтобы создать макрос в макрорекордере (для тех программ Microsoft Office, для которых это средство предусмотрено, например, Word, Excel, PowerPoint, Project):

1. В меню **Сервис | Макрос** выберите команду **Начать запись**. В открывшемся окне **Запись макроса** (рис. 1.1) вам потребуется определить:
   - **Имя макроса**. Правило такое: имя не должно начинаться с цифры, не должно содержать пробелы и символы пунктуации. Максимальная длина в Excel — 64 символа, в Word — 80 символов. Можно писать по-русски;

- будет ли макрос назначен кнопке на панели управления или комбинации клавиш. Выполнить это, а также задать другие средства для вызова макроса можно и потом — об этом будет рассказано в следующем разделе;
- где сохранить макрос. В Word в вашем распоряжении текущий файл и шаблон для всех вновь создаваемых документов — Normal.dot, в Excel — текущая книга, возможность создать макрос одновременно с созданием новой книги и личная книга макросов PERSONAL.XLS (макросы из этой скрытой книги будут доступны во всех книгах). Подробнее про то, где может храниться программный код, мы поговорим в *разд. 2.2*;
- **Описание**. В это поле лучше ввести информацию о том, для каких целей создается этот макрос — это подарок не только для других пользователей, но и для себя (через несколько месяцев).

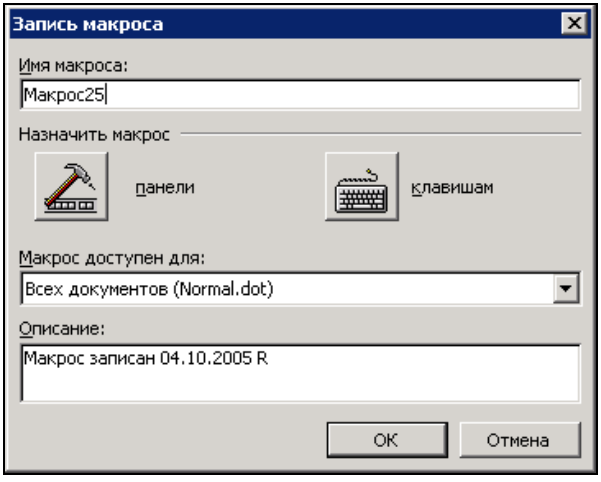


Рис. 1.1. Диалоговое окно **Запись макроса**

2. После нажатия кнопки **OK** или назначения кнопки или клавиатурной комбинации начнется запись макроса. Указатель мыши при этом примет вид магнитофонной кассеты и появится маленькая панель **Остановить запись**. На ней всего две кнопки — **Остановить запись** и **Пауза**. Если вы случайно закрыли эту панель, остановить запись можно через меню **Сервис | Макрос | Остановить запись**.
3. Самый простой способ запустить макрос, которому не назначена кнопка или клавиатурная комбинация, — в меню **Сервис** выбрать **Макрос | Макросы** (или нажать комбинацию клавиш <Alt>+<F8>), в открывшемся окне

Макрос (см. рис. 1.2) в списке выбрать нужный макрос и нажать кнопку **Выполнить**. Из этого же окна можно просматривать и редактировать макросы, удалять или перемещать их и т. п.

Если макросов создано много, то получить список всех назначений клавиш (включая назначения для встроенных макросов Word) можно при помощи меню **Сервис | Макрос | Макросы**, затем в окне **Макрос** в списке **Макросы из** выбрать **Команд Word**, а в списке **Имя** выбрать макрос **ListCommands** и нажать кнопку **Выполнить**. В ответ на приглашение нужно выбрать **Текущие настройки меню и клавиш** (иначе будет выведен полный список команд Word на 26 страниц). В ваш документ будет вставлена таблица с текущими назначениями клавиш, которую можно распечатать.

Если у вас уже есть значительное количество созданных при помощи макрорекордера макросов, то после освоения языка VBA имеет смысл подумать над ними и, может быть, внести изменения. Чаще всего стоит обратить внимание на следующие моменты:

- если в вашем макросе повторяются какие-либо действия, возможно стоит организовать цикл;
- может быть, есть смысл в ходе выполнения уточнить что-либо у пользователя (при помощи встроенной функции VBA `InputBox()` или элементов управления);
- чтобы в ходе выполнения макроса не возникало ошибок, можно реализовать в нем проверку текущих условий.

Как все это сделать, будет рассказано в следующих главах.

И еще один очень важный момент, связанный с макрорекордером. Помимо того, что он позволяет создавать простенькие программы, пригодные для самостоятельного использования без всяких доработок, макрорекордер — это еще и ваш разведчик в мире объектных моделей приложений Office. Опытные разработчики часто пользуются им для того, чтобы понять, какие объекты из огромных объектных моделей приложений Office можно использовать для выполнения тех или иных действий.

Приведу пример: вам нужно автоматизировать создание диаграмм в Excel. Поскольку в русской версии Excel для создания диаграммы вручную вы используете команду **Вставка | Диаграмма**, то, скорее всего, в справке по VBA вы начнете в первую очередь искать объект `Diagram`. И вы его найдете и, возможно, потратите определенное время на его изучение, прежде чем поймете, что это не та диаграмма! Объект `Diagram` представляет то, что в русской версии Excel называется "Схематическая диаграмма" (доступна из того же меню **Вставка**), а обычная диаграмма — это объект `Chart`. А вот если бы вы пустили вперед разведчика (т. е. создали бы диаграмму с записью в макрорекорде-

ре и посмотрели бы созданный код), он бы сразу указал нам нужное направление движения.

1.4. Четыре способа запуска макроса

Предположим, что макрос уже создан (в макрорекордере, как вы уже умеете, или средствами редактора Visual Basic, который вам предстоит освоить), и вы хотите или выполнить его один раз, или настроить возможность вызывать его постоянно. В нашем распоряжении — множество разных способов.

Самый простой, но и самый неудобный способ — воспользоваться окном **Макрос**, которое можно открыть при помощи меню **Сервис | Макрос | Макросы** (рис. 1.2).

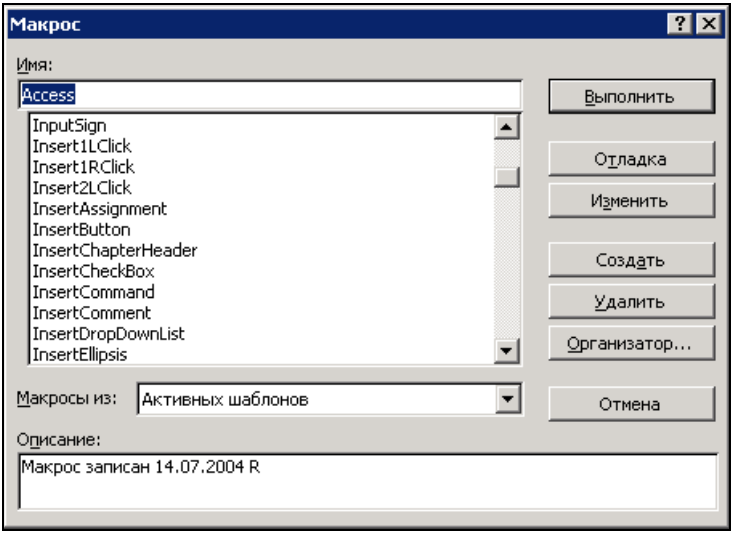


Рис. 1.2. Диалоговое окно **Макросы**

Из этого окна с помощью кнопок можно:

- **Выполнить** — запустить макрос на выполнение;
- **Отладка** — открыть макрос в редакторе Visual Basic и начать его пошаговое выполнение;
- **Изменить** — просто открыть макрос в редакторе Visual Basic;
- **Создать** — необходимо будет ввести имя создаваемого макроса и в редакторе Visual Basic будет автоматически создана процедура с определенным вами именем;

- **Удалить;**
- **Организатор** — поменять описание и назначенное сочетание клавиш.

Каждый раз открывать это окно, находить нужный макрос (а их может быть, например, несколько десятков) и нажимать на кнопку **Выполнить** — не самый быстрый вариант. Вряд ли он очень понравится вашим пользователям, да и вам самим работать будет неудобно. Поэтому в вашем распоряжении несколько более удобных вариантов.

Если вы пользуетесь макросом постоянно, то можно использовать самый быстрый способ его вызова — клавиатурную комбинацию. Например, сейчас, когда я пишу эту книгу, я "положил" на клавиатурные комбинации простые макросы, которые вводят нужный мне текст. Если мне нужно набрать "Visual Basic", я нажимаю <Alt>+<V>, если Microsoft Office — <Alt>+<M>. На клавиши (правда, уже без макросов) у меня разложены и все стили — заголовки, маркированные списки и т. п. Очень удобно!

На практике клавиатурным комбинациям есть смысл назначать только те макросы, которые используются каждый день, например, ввод информации об ответственном исполнителе, о руководителе, которому пойдет документ на подпись, о полном названии вашей организации и т. п. Главное — чтобы вы использовали их постоянно, иначе вы просто забудете, какое сочетание клавиш за что отвечает.

Назначить сочетание клавиш макросу можно очень просто.

В Word это выглядит так: с помощью меню **Сервис | Настройка** открываем одноименное окно и переходим на вкладку **Команды**. Затем нажимаем на кнопку **Клавиатура**: откроется окно **Настройка клавиатуры** (рис. 1.3).

В списке **Категории** нужно выбрать **Макросы**, в списке **Команды** — созданный вами макрос, установить указатель ввода в поле **Новое сочетание клавиш** и нажать требуемое сочетание клавиш. Помимо обычных сочетаний типа <Alt>+<1>, <Alt>+<M> и т. п., можно использовать и более сложные. Например, вы вставляете при помощи макросов два варианта названия вашей организации: полное и краткое. Этим макросам можно назначить клавиатурные комбинации вида <Alt>+<N>, <F> и <Alt>+<N>, <S>. Это значит, что если вы вначале нажмете вместе клавиши <Alt>+<N>, а затем <F>, то соответствующий макрос будет выполнен. Вводить такое сочетание клавиш в поле **Новое сочетание клавиш** нужно точно так же, как вы будете его применять.

После того как нужное сочетание клавиш будет введено, нажмите кнопку **Назначить**, а потом **Закрыть.**

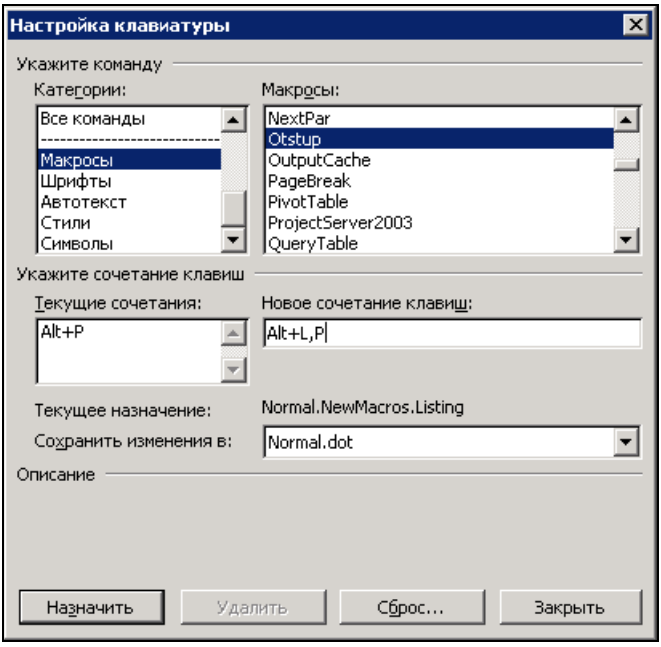


Рис. 1.3. Окно настройки клавиатурных комбинаций

Внимание!

Следите за надписью **Текущее назначение** в этом диалоговом окне. Вполне возможно, что выбранному вами сочетанию клавиш уже назначен другой макрос или встроенная команда. Если вы проигнорируете это сообщение, то вы переназначите эту комбинацию вашему макросу. Но если пользователь привык использовать эти клавиши для других целей (<Ctrl>+<S>, <Ctrl>+<N> и т. п.), он может быть очень недоволен.

В Excel кнопки **Клавиатура** в окне **Настройка** (меню **Сервис | Настройка**) вы не найдете. Здесь придется назначать клавиатурные комбинации по-другому: в меню **Сервис** выбрать **Макрос | Макросы**, выбрать в списке нужный макрос и нажать кнопку **Параметры**. Откроется окно **Параметры макроса** (рис. 1.4), в котором вы сможете выбрать нужную клавиатурную комбинацию (только в сочетании с клавишей <Ctrl>) и ввести описание макроса. Вообще говоря, любое сочетание клавиш можно назначить макросу и в Excel, но простыми способами этого сделать нельзя — придется писать программный код, в котором будут перехватываться события приложения.

Как мы уже говорили, клавиатурные комбинации есть смысл назначать только тем макросам, которыми вы пользуетесь каждый день. А что делать с полезными макросами, которые активно используются, к примеру, в отчетный

период, а потом опять ждут своего часа целый месяц? Подавляющее большинство пользователей за этот месяц забывает все назначенные клавиши и теряет те бумажки, на которых вы им записали эти клавиатурные комбинации. Да и сам разработчик (у меня это случается регулярно) вполне может забыть, что именно нужно нажимать для запуска старого макроса.

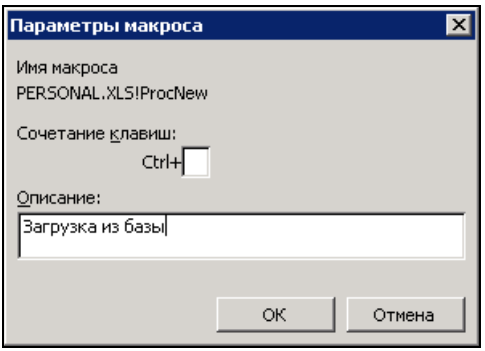


Рис. 1.4. Диалоговое окно **Параметры макроса**

Лучший выход в такой ситуации — назначить макрос пункту меню или кнопке на панели управления. Пожалуй, назначение пункту меню даже лучше — больше возможностей упорядочить макросы и использовать понятные названия. Однако нажимать кнопки на панели инструментов быстрее, так что выбирайте сами, что вам больше нравится. Создание и настройка новой панели инструментов для вызова макросов в Word может выглядеть так:

1. В меню **Сервис** выберите **Настройка** и перейдите на вкладку **Панели инструментов**.
2. Нажмите кнопку **Создать**, введите название панели (например, **Мои_макросы**) и выберите тот документ, в котором она будет создана. Если вы выберете Normal.dot, то панель будет доступна для всех документов Word на этом компьютере (что чаще всего и необходимо). Другой вариант — создать панель инструментов в том документе Word, который у вас открыт. В этом случае панель будет доступна только из этого файла.
3. После того как вы нажмете кнопки **OK** и **Закрыть**, будет создана новая пустая панель (которая будет находиться где-нибудь прямо поверх документа). Чтобы было удобней, перетащите ее к стандартным панелям инструментов, а потом вновь воспользуйтесь командой главного меню **Сервис | Настройка**. В окне **Настройка** перейдите на вкладку **Команды**, в списке **Категории** выберите **Макросы** и просто перетащите на панель инструментов нужные макросы из списка **Команды**. Если на панель инструментов нужно поместить не один, а несколько макросов, то, возможно,

удобнее будет нажать кнопку **Упорядочить команды** и воспользоваться очень удобным диалоговым окном **Изменение порядка команд** (рис. 1.5).

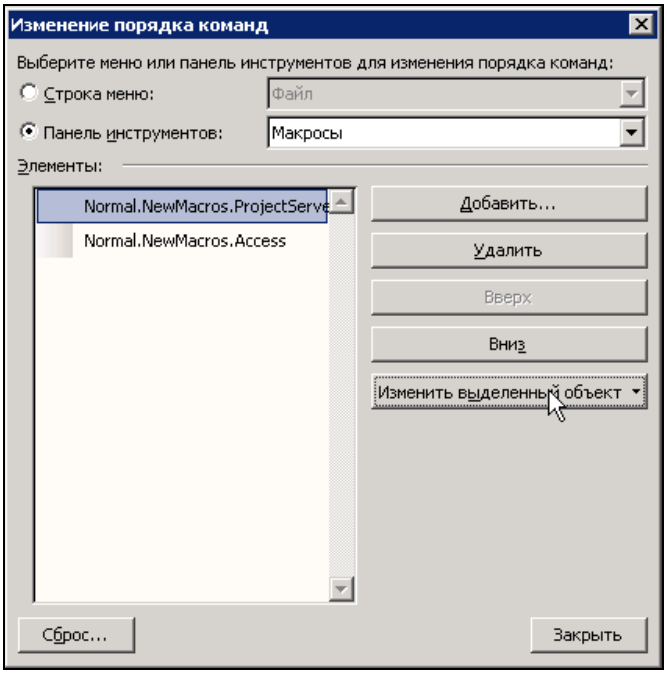


Рис. 1.5. Диалоговое окно **Изменение порядка команд**

Мы добавили нужные кнопки на панели инструментов, но пока они выглядят не очень интересно (например, **Normal.NewMacros.QueryTable**). Вряд ли такое название что-то скажет пользователю. Поэтому следующее действие — настройка кнопок. Для этого при открытом окне **Настройка** (это условие обязательно!) просто щелкните правой кнопкой мыши по кнопке панели инструментов, которую надо настроить. Откроется специальное контекстное меню (рис. 1.6).

С помощью этого меню можно:

- **Удалить** — удалить кнопку (также можно просто перетащить ее обратно с панели на окно **Настройка**);
- **Имя** — ввести имя, т. е. надпись на кнопке или пункте меню. Для меню использование надписи удобно, для кнопки на панели инструментов — не очень, поскольку это занимает много места;
- **Копировать значок на кнопке** и **Вставить значок для кнопки** — воспользоваться понравившимся вам значком с другой кнопки;

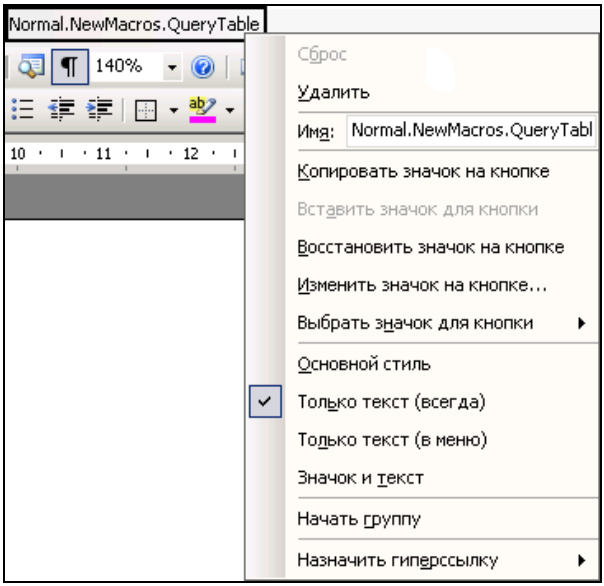


Рис. 1.6. Контекстное меню для настройки кнопки на панели инструментов

- **Изменить значок на кнопке** — откроется скромный редактор, в котором вы сможете сами нарисовать нужный значок;
- **Выбрать значок для кнопки** — выбрать один из 42 стандартных значков. На самом деле только в Word значков, которые можно использовать, несколько тысяч;
- **Основной стиль** — под этой надписью скрывается то, что нам обычно и нужно: чтобы кнопка была представлена только рисунком, безо всяких надписей;
- **Только текст (всегда)**, **Только текст (в меню)**, **Значок и текст** — определяют, что именно из набора надпись/рисунок будет показано на кнопке. Естественно, наиболее часто используемый вариант — **Основной стиль**;
- **Начать группу** — слева от кнопки появится вертикальная черта (разделитель);
- **Назначить гиперссылку** — назначить ссылку на другое место в вашем документе или на страницу в Интернете.

Конечно же, мы могли обойтись и без создания своей панели управления, просто добавив новые кнопки в существующие (точно таким же перетаскиванием). Аналогичным образом мы можем преобразовать стандартные панели инструментов. Однако не забывайте, что все эти преобразования доступны только при открытом диалоговом окне **Настройка**.

Создание меню производится немного по-другому:

1. Откройте то же диалоговое окно **Настройка** (меню **Сервис | Настройка**).
2. В списке **Категории** выберите **Новое меню**.
3. Перетащите команду **Новое меню** из списка **Команды** того же окошка (рис. 1.7) в нужное место основного меню.

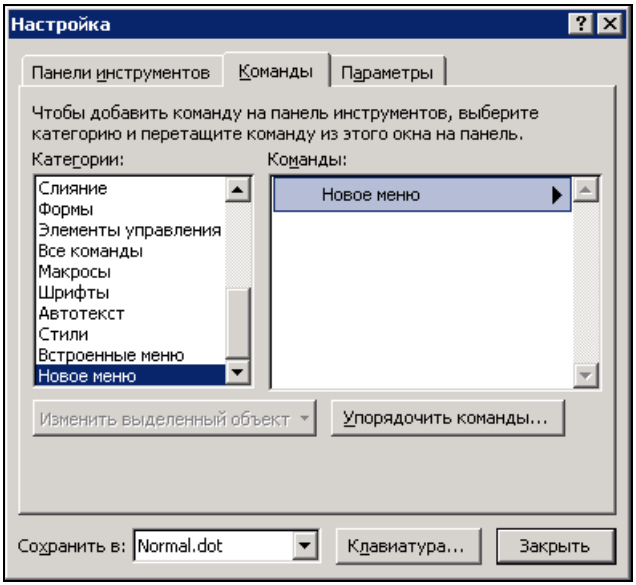


Рис. 1.7. Диалоговое окно **Настройка**

4. Далее точно так же при открытом окне **Настройка** щелкните правой кнопкой мыши по созданному вами пункту меню и переименуйте его (в нашем примере назовите его **Макросы**).

Далее нужно нажать кнопку **Упорядочить команды** в окне **Настройка**. В открывшемся диалоговом окне **Изменение порядка команд** (рис. 1.8) нужно в списке **Строка меню** выбрать **Макросы** и добавить в него нужные элементы (т. е. созданные вами макросы). Переименовать их можно при помощи кнопки **Изменить выделенный объект** прямо из этого окна.

В результате у вас может получиться очень милое меню, в котором пользователю запутаться будет трудно (рис. 1.9).

В Excel все очень похоже, но чуть-чуть по-другому. Если в Excel мы откроем окно **Настройка** (меню **Сервис | Настройка**) и в списке **Категории** выберем **Макросы**, то вместо списка макросов в списке **Команды** будет две возможности: **Настраиваемая команда меню** и **Настраиваемая кнопка** (рис. 1.10).

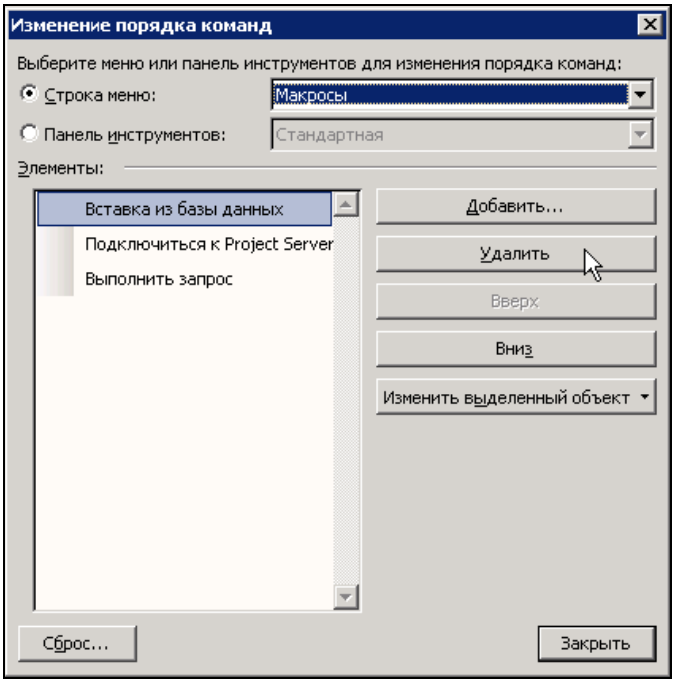


Рис. 1.8. Окно **Изменение порядка команд**

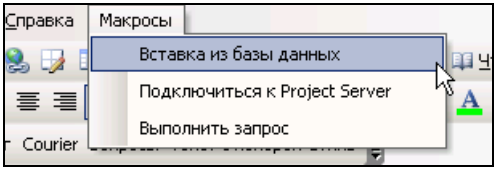


Рис. 1.9. Пример меню для запуска макросов

Настраиваемая кнопка — это готовая кнопка, которую можно перетащить на панель инструментов, а потом открыть для нее контекстное меню и воспользоваться командой **Назначить макрос**. Конечно же, для выбора иконки, формата отображения и т. п. можно воспользоваться и другими возможностями контекстного меню, которые доступны и в Word.

Для создания нового меню в Excel нужно точно так же создать новое меню, как и в Word, а потом нажать на кнопку **Упорядочить команды** и добавить в это меню несколько элементов **Настраиваемая команда меню**. Их реальная настройка (в том числе и назначение макросов) производится по нажатию кнопки **Изменить выделенный объект**.

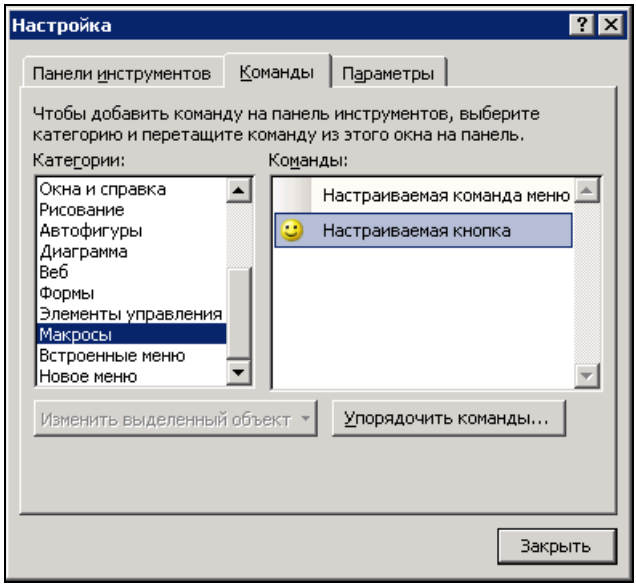


Рис. 1.10. Создание кнопки для запуска макроса в Excel

В подавляющем большинстве остальных приложений Office (PowerPoint, Project, Outlook и т. п.) работа с макросами происходит так же, как и в Word.

Есть еще один способ предоставить пользователю возможность запускать макросы — самый функциональный, но и самый трудоемкий: создать специальную графическую форму, на которую можно поместить, например, ниспадающий список макросов. При использовании этого способа можно предусмотреть дополнительные элементы управления для ввода параметров, которые макросы смогут "подхватывать" во время выполнения (напрямую параметры макросам передаваться не могут, поскольку макрос — это процедура, не принимающая параметров). Однако использование этого способа требует написания программного кода. Как работать с формами и элементами управления на них, будет рассказано в *гл. 5*. После этого создание такой формы не составит никакого труда.

Есть еще одна специальная возможность для запуска макросов: сделать так, чтобы они запускались при возникновении специального события. Таким событием может стать, например, внесение изменений на лист Excel, открытие книги Excel или документа Word и т. п. Подробнее про работу с событиями будет рассказано также в *гл. 5*. Однако можно обеспечить автоматический запуск макроса и без программирования: достаточно просто назначить ему специальное имя. Например, для Word список таких специальных названий представлен в табл. 1.1.

Таблица 1.1. *Специальные названия макросов для Word*

Имя процедуры	Когда запускается
`AutoExec`	При запуске Word (этот макрос должен храниться в Normal.dot)
`AutoNew`	При создании нового документа
`AutoOpen`	При открытии любого документа (если в Normal.dot) или при открытии документа, в котором находится макрос с таким именем
`AutoClose`	При закрытии документа
`AutoExit`	При выходе из Word

В Excel предусмотрены специальные имена макросов для рабочей книги `Auto_Open`, `Auto_Close`, `Auto_Activate` и `Auto_Deactivate`. Однако Microsoft предупреждает, что эти возможности оставлены только для обратной совместимости, и рекомендует пользоваться событийными процедурами.

Еще один момент, связанный с макросами `Auto`: поскольку раньше эти возможности активно использовались вирусами, в Office 2003 по умолчанию эти макросы запускаться не будут. Для того чтобы обеспечить им возможность запуска, необходимо изменить установленный уровень безопасности в меню **Сервис | Макрос | Безопасность** на **Низкий**.

Ну и последняя, по моему опыту, самая малоизвестная, но, тем не менее, очень полезная возможность для запуска макросов. Вы можете запустить их из командной строки при запуске Word или Excel, указав имя макроса в качестве параметра командной строки. Например, чтобы открыть Word и сразу выполнить макрос `MyMacros` из Normal.dot, можно воспользоваться командой:

```
winword.exe /mMyMacros
```

Очень удобно использовать эту возможность, если создать несколько ярлыков для запуска приложения Office, например, на рабочем столе, изменить в них командную строку и использовать для запуска приложения с одновременным запуском макросов.

Задание для самостоятельной работы 1: Запись макроса для автоматического ввода текста в Word

Ситуация:

Вам несколько раз в день необходимо передавать распоряжения в бухгалтерию. Каждое распоряжение должно заканчиваться строками, аналогичными представленным на рис. 1.11.

Генеральный директор:	Иванов А. А.
Отв. исполнитель Петрова М. М. т. 55-55	

Рис. 1.11. Строки, ввод которых нужно автоматизировать

ЗАДАНИЕ:

Напишите при помощи макрорекордера макрос, который бы автоматически создавал такие строки (вместо "Петрова М. М." подставьте ваши данные).

Созданный макрос должен быть доступен для всех создаваемых вами документов. Он должен запускаться по нажатию кнопки с рожицей (рис. 1.12).

Создайте новый документ, запустите макрос на выполнение и убедитесь, что он работает.

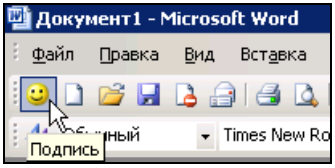


Рис. 1.12. Веселая кнопка для пользователя

Ответ к заданию 1

1. Откройте новый документ в Word. В меню **Сервис** выберите **Макрос | Начать запись**. В окне **Запись макроса** в поле **Имя макроса** введите "Подпись" (без кавычек), убедитесь, что в поле **Макрос доступен для** стоит значение **Всех документов (Normal.dot)**, и нажмите кнопку **Назначить макрос панели**.
2. В окне **Настройка** на вкладке **Команды** перетащите элемент **Normal.NewMacros.Подпись** в нужное место на панели управления. Затем щелкните по перемещенному элементу правой кнопкой мыши, в контекстном меню выберите пункт **Выбрать значок для кнопки**, затем выберите изображение улыбающейся рожицы. Еще раз щелкните правой кнопкой мыши по этому элементу и в контекстном меню выберите **Основной стиль**. Нажмите на кнопку **Закрыть** окна **Настройка**. Начнется запись макроса.
3. Введите нужный текст, а затем нажмите на кнопку **Остановить запись** (или в меню **Сервис | Макрос** выберите команду **Остановить запись**).
4. Создайте новый документ Word и убедитесь, что новая кнопка работает и там.