

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**

Кафедра автоматизированных систем управления (АСУ)

С. Л. Миньков

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ИНФОРМАТИКЕ

2013

Рецензенты:

Тимченко С.В., д-р физ.-мат. наук, профессор, заведующий кафедрой прикладной математики и информатики Томского государственного университета систем управления и радиоэлектроники

Брендаков В.Н., канд. физ.-мат. наук, доцент, заведующий кафедрой высшей математики и информационных технологий Северского технологического института Национального исследовательского ядерного университета «МИФИ»

Корректор: Осипова Е.А.

Миньков С.Л.

Лабораторный практикум по информатике / С.Л. Миньков. — Томск: Факультет дистанционного обучения, ТУСУР, 2013. — 182 с.

© Миньков С.Л., 2013

© Факультет дистанционного обучения, ТУСУР, 2013

АННОТАЦИЯ

Лабораторный практикум по информатике является практической частью учебно-методического комплекса, обеспечивающего преподавание дисциплины «Информатика»

Лабораторный практикум предназначен для ознакомления с основами работы и программирования в электронных таблицах MS Excel.

Изучение возможностей Excel основано на версиях MS Excel 2007/2010.

Весь материал разбит на две комплексные лабораторные работы. В первой используются стандартные встроенные средства Excel, во второй используется система программирования VBA для разработки собственных приложений в Excel. Каждая лабораторная работа содержит теоретический материал, необходимый для выполнения заданий, примеры, показывающие применение функций Excel и средств VBA, и варианты для самостоятельной работы различной степени сложности.

Настоящее издание является переработанным вариантом лабораторных практикумов [1, 2], и автор выражает благодарность коллегам и студентам кафедры автоматизированных систем управления Томского государственного университета систем управления и радиоэлектроники и факультета инновационных технологий Национального исследовательского Томского государственного университета за их пожелания и критические замечания. Они были учтены при подготовке этого издания.

ОГЛАВЛЕНИЕ

Введение.....	7
Лабораторная работа № 1. Основы EXCEL	9
Раздел 1. Таблицы и диаграммы	9
1.1 Книги и листы	9
1.2 Ячейки.....	11
1.3 Данные	12
1.4 Автозаполнение	18
1.5 Примечание	19
1.6 Дублирование данных.....	19
Задание 1.1. Оформление таблицы:.....	20
1.7 Построение диаграмм	28
1.7.1 Построение гистограммы	29
1.7.2 Построение графика функции одной переменной.....	33
1.7.3 Построение графика функции двух переменных	36
Задание 1.2. Построение гистограмм	38
Задание 1.3. Построение графиков	38
Раздел 2. Численное решение уравнений	41
2.1 Решение полиномов	41
Задание 2.1. Корни полинома.....	41
2.2 Решение нелинейных уравнений	44
2.2.1 Решение нелинейного уравнения методом итерации	44
Задание 2.2. Метод итерации	44
2.2.2 Решение нелинейного уравнения методом бисекции	45
Задание 2.3. Метод бисекции	46
2.3 Решение систем линейных уравнений	46
Задание 2.4. Метод обратной матрицы	47
Задание 2.5. Метод Крамера.....	48
2.4 Решение систем нелинейных уравнений	49
Задание 2.6. Метод целевой функции	50
Раздел 3. Обработка данных.....	53
3.1 Сортировка и фильтрация.....	53
Задание 3.1. Сортировка	54
Задание 3.2. Фильтрация.....	55
Задание 3.3. Автофильтр.....	56

Задание 3.4. Расширенный фильтр	59
3.2 Сводные таблицы	59
Задание 3.5. Создание сводной таблицы.....	61
3.3 Промежуточные итоги	62
Задание 3.6. Получение промежуточных итогов	63
3.4 Связь таблиц (консолидация данных)	63
Задание 3.7. Формула связи.....	65
Задание 3.8. Консолидация.....	66
3.5 Форма данных	67
Задание 3.9. Создание формы данных.....	68
Лабораторная работа № 2. Основы VBA.....	69
Раздел 4. Создание макросов с помощью макрорекодера.....	69
Задание 4.1. Штатное расписание.....	73
Задание 4.2. Шаблоны таблицы и диаграммы.....	75
Раздел 5. Среда программирования VBA в Excel:	
процедуры и функции.....	77
5.1 Структура программ на VBA	77
5.2 Скалярные типы данных.....	79
5.3 Структурированные типы данных.....	81
5.4 Выражения	81
5.5 Операторы	83
5.5.1 Оператор присваивания	83
5.5.2 Условные операторы.....	83
5.5.3 Операторы цикла	85
Задание 5.1. Исключить лишнее входное данное	97
Задание 5.2. Процедуры и функции	98
Раздел 6. Объекты пользовательского интерфейса	104
6.1 Объект.....	104
6.2 Метод	110
6.3 Свойство	110
6.4 Событие	113
6.5 Оператор With...End With	113
6.6 Создание вспомогательных процедур.....	114
6.7 Функции MsgBox и InputBox	117
Задание 6.1. Закраска листов.....	120
6.8 Создание кнопок.....	120
6.9 Форматирование листов	122

Задание 6.2. Работа с <i>InputBox</i>	128
Задание 6.3. Формирование титульного листа	135
Раздел 7. Элементы управления	136
7.1 Формы	136
7.2 Элементы управления формы	137
7.3 Элементы ActiveX	139
7.4 Объекты средств рисования	143
Задание 7.1. Работа с элементами управления CheckBox, OptionButton и ToggleButton	146
Задание 7.2. Работа с элементами управления ListBox, ScrollBar и SpinButton	150
Раздел 8. Программирование пользовательских форм	151
8.1 Создание формы с элементами Label и TextBox	152
8.2 Создание формы с RefEdit	155
8.3 Создание формы с ComboBox и ListBox	158
8.4 Создание диалоговой формы «Мастер Путешествий» ...	161
Задание 8.1. Создание формы туристического ваучера	169
8.5 Создание интерфейса работы с базой данных	170
8.6 Построение диаграммы	177
Задание 8.2. Создание базы данных	180
Литература	182

ВВЕДЕНИЕ

Электронные таблицы MS Excel — одна из составных частей пакета прикладных программ Microsoft Office, работающего в среде Windows. Она предназначена для обработки числовых данных, проведения математического моделирования различных процессов (в первую очередь экономических), изготовления различных документов и форм, а также может быть использована в качестве простой базы данных.

Электронные таблицы MS Excel предлагают широкий набор функциональных средств по проведению динамических расчетов и обработке табличных данных:

- создание и редактирование электронных таблиц с применением богатого набора встроенных функций;
- оформление и печать электронных таблиц;
- построение диаграмм и графиков различной степени наглядности и детализации;
- работа с электронными таблицами как с базами данных: фильтрация, сортировка, создание итоговых и сводных таблиц, консолидация данных из различных таблиц, в том числе из внешних баз данных;
- решение экономических задач типа «что — если» путем подбора параметров;
- решение оптимизационных задач;
- численное решение разнообразных математических задач;
- статистическая обработка данных;
- использование интегрированной среды разработки собственных программ-макросов Visual Basic for Applications.

Лабораторный практикум разбит на две комплексные лабораторные работы, каждая из которых, в свою очередь, разбита на разделы. В первой работе их три, во второй — пять. Каждый раздел посвящен той или иной тематике практикума и содержит несколько заданий разной степени сложности. В первой работе — 18 заданий, во второй — 11.

Заданиям обычно предшествуют примеры. Для впервые знакомящихся с возможностями Excel рекомендуется воспроизвести примеры по приведенным указаниям и разобраться, как работает рассматриваемое средство Excel или VBA.

После ознакомления с теоретической частью раздела лабораторной работы студент выполняет соответствующее задание.

Номер варианта для каждого задания выбирается по следующей формуле:

$$V = (N \times K) \text{ div } 100,$$

где V – искомый номер варианта,

N – общее количество вариантов,

K – код варианта,

div – целочисленное деление,

при $V = 0$ выбирается максимальный вариант.

Если варианты не приведены, то задание для всех общее.

В качестве отчета по первой лабораторной работе следует предоставить три файла («книги») Excel с выполненными заданиями: файл «InfoSect1.xlsx» (задания 1.1—1.3), файл «InfoSect2.xlsx» (задания 2.1—2.6), файл «InfoSect3.xlsx» (задания 3.1—3.9). Файлы должны содержать необходимые пояснения хода выполнения работы и описание полученных результатов. Первый лист книги Excel рекомендуется сделать титульным (с указанием фамилии исполнителя, номера и названия задания и номера варианта). Количество листов в книге, на которых будет размещено решение, выбирается самостоятельно исходя из удобства расположения информации.

В качестве отчета по второй лабораторной работе следует предоставить пять файлов («книг») Excel с выполненными заданиями: файл «InfoSect4.xlsm» (задания 4.1—4.3), файл «InfoSect5.xlsm» (задания 5.1—5.2), файл «InfoSect6.xlsm» (задания 6.1—6.3), файл «InfoSect7.xlsm» (задания 7.1—7.2), файл «InfoSect8.xlsm» (задания 8.1—8.3). Эта работа посвящена VBA, поэтому файлы будут содержать макросы, и сохранять их следует с расширением .xlsm (*Книга Excel с поддержкой макросов*). Коды VBA, написанные в процессе выполнения заданий, следует снабдить комментариями, как это приведено в примерах. Перед отправкой протестируйте работу всех написанных макросов и элементов управления на всех возможных режимах и убедитесь в работоспособности и корректности созданного программного кода. Рекомендуемое оформление — как в первой лабораторной работе.

ЛАБОРАТОРНАЯ РАБОТА № 1. ОСНОВЫ EXCEL

Раздел 1. Таблицы и диаграммы

Цель раздела: освоить приемы построения и форматирования таблиц и графиков.

1.1 Книги и листы

При запуске Excel открывается рабочая книга Excel (так называются файлы Excel), которая, начиная с MS Office 2007, по умолчанию имеет имя Книга1.xlsx. Эта книга состоит из трех листов — Лист1, Лист2 и Лист3. Щелкая левой клавишей мыши по ярлычку листа, можно переходить из одного листа в другой. Можно дать им и более осмысленное название. Для этого необходимо щелкнуть по ярлычку листа правой клавишей мыши, вызвав контекстно-зависимое меню, выбрать опцию **Переименовать** и набрать с клавиатуры новое имя, например «План» (рис. 1.1).

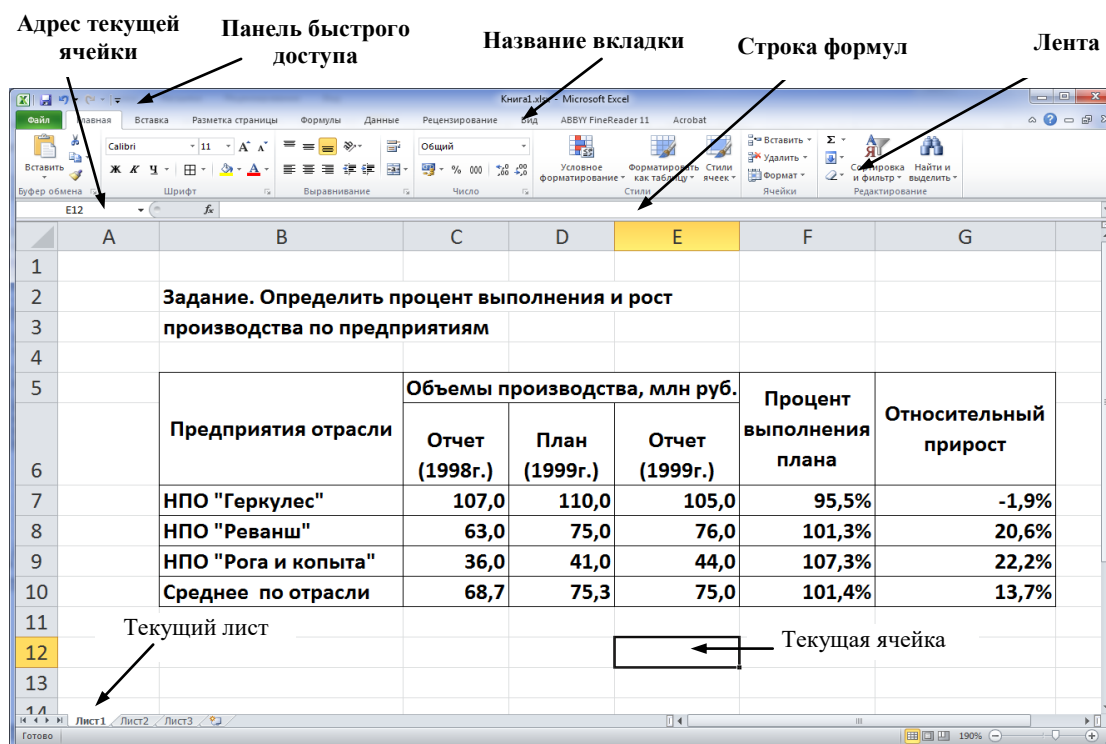


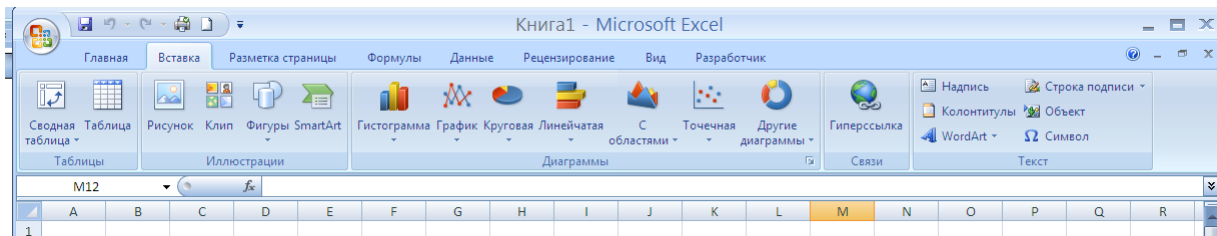
Рис. 1.1 — Рабочий лист Excel План в Excel 2010

Отметим, что с помощью предлагаемого меню можно также производить изменение цвета ярлычка, удаление листа, добавление нового листа и перемещение его в другой файл. Есть и другой

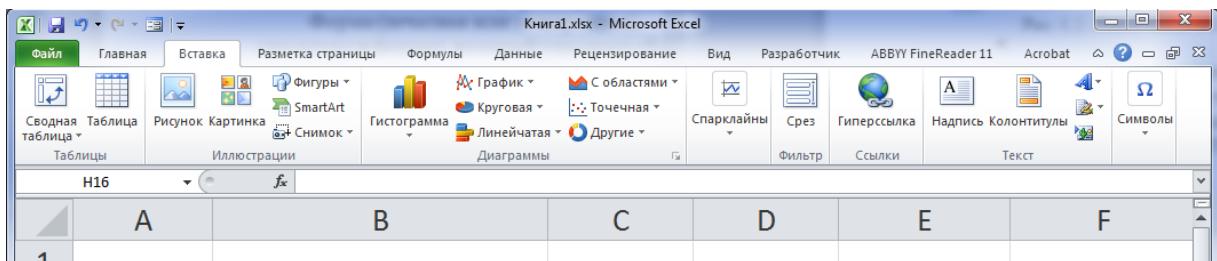
способ переименования — двойной щелчок левой клавишей мыши по ярлычку листа и набор нового имени.

В MS Office, начиная с MS Office 2007 (в т.ч. в Excel), применена новая форма организации командного интерфейса — Лента (Ribbon) Office (рис. 1.2).

По умолчанию на ней расположено 7 вкладок: **Главная**, **Вставка**, **Разметка страницы**, **Формулы**, **Данные**, **Рецензирование**, **Вид**. По желанию пользователя на ней могут появиться новые вкладки, связанные с программами, которые при установке интегрируются с Excel (например, ABBYY FineReader или Acrobat (рис. 1.1)) или уже интегрированы в Excel, но по умолчанию не установлены и их требуется вызвать на Ленту (например, **Разработчик** (рис. 1.2)).



а



б

Рис. 1.2 — Лента Office в Excel 2007 (*а*) и Excel 2010 (*б*)
(Вкладка **Вставка**)

Принципиального отличия между Лентой Office 2007 и 2010 нет. Только в первом случае для вызова меню действий с файлом (рис. 1.3) используется кнопка Office, расположенная в верхнем левом углу, а во втором — привычная по предыдущим версиям MS Office кнопка Файл (рис. 1.1—1.2).

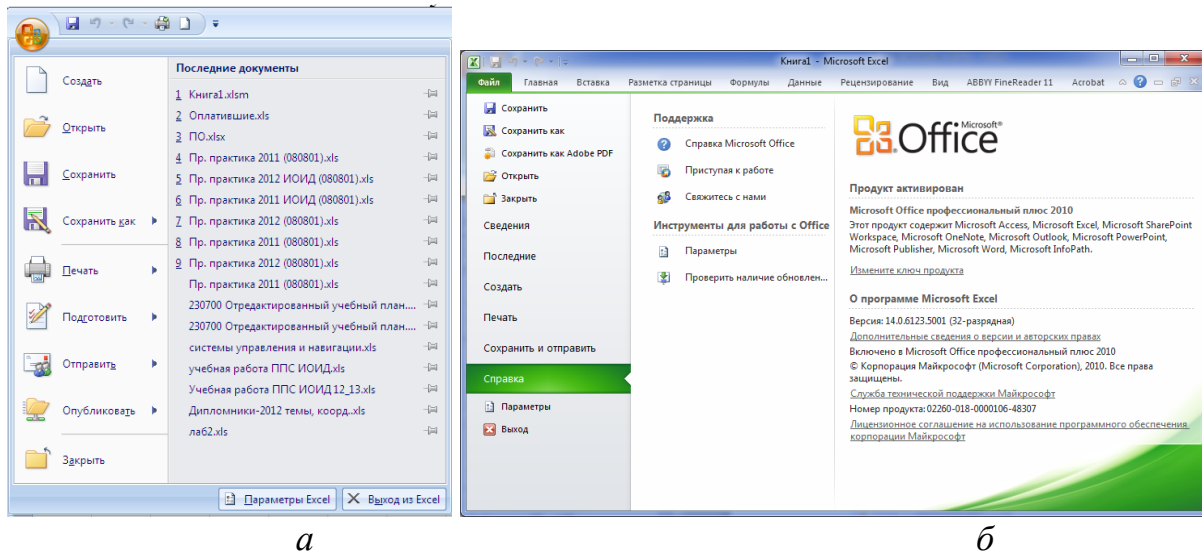


Рис. 1.3 — Меню действий с файлом в Excel 2007 (а) и Excel 2010 (б)

1.2 Ячейки

Каждый лист Excel представляет собой таблицу. Столбцы обозначены буквами от А до XFD (всего 16384 столбцов), а строки — числами от 1 до 1048576. Каждая ячейка таблицы имеет свой номер, например А1, GA200. С помощью мыши или клавиш передвижения курсора (указателя) можно перемещаться из ячейки в ячейку, а выполнив команду **Главная | группа Ячейки | Вставить Строки (Столбцы) на лист / Удалить Строки (Столбцы) с листа**, можно вставлять или удалять в уже созданной таблице пользователя новые строки и столбцы. При этом происходит их автоматическая перенумерация.

Примечание. Не забывайте, что многие команды доступны также через контекстно-зависимое меню, вызываемое щелчком по правой кнопкой мыши. При этом список доступных команд определяется местом, где в это время находится курсор.

Текущая ячейка выделяется черным контуром (Е12, рис. 1.1). Чтобы выделить несколько ячеек (блок), необходимо щелкнуть левой клавишей мыши по начальной (обычно левой верхней) ячейке и, не отпуская ее, протащить указатель до последней (правой нижней) ячейки. Выделенные ячейки (кроме первой) затемняются.

Для выделения нескольких несмежных блоков (бывает полезно при построении диаграмм и графиков) необходимо выделить первый блок, а затем, нажав и удерживая клавишу **Ctrl**, выделить следующий блок и т. д. Чтобы отменить выделение, достаточно еще раз щелкнуть мышью по любому участку листа.

1.3 Данные

В ячейки таблицы можно вводить три типа данных: число, текст, формулу. По первому символу Excel определяет, что введено: если цифра, то число; если это буква или апостроф (знак «'»), то текст, если знак равенства, то формула. Если вы хотите, чтобы число воспринималось как текст, поставьте перед ним апостроф. Для ввода данных необходимо переместиться в нужную ячейку, набрать данные и нажать **Enter** или клавишу перемещения курсора.

Если число не входит в ячейку, то Excel отображает его либо в экспоненциальной форме ($1230000000 \rightarrow 1,23E+09$; $0,0000567 \rightarrow 5,67E-05$), либо вместо числа ставит знаки #####. Тогда необходимо раздвинуть границы ячейки.

В Excel можно выбрать различные форматы представления чисел: **Главная | группа Ячейки | Формат | Формат ячеек | Число | Числовые форматы** (рис. 1.4). Это же самое можно проделать через вкладку **Главная | Число** или через контекстно-зависимое меню, щелкнув по ячейке правой кнопкой мыши и выбрав **Формат ячеек**.

Если при вводе числа допущена ошибка (например, поставлена десятичная точка вместо десятичной запятой¹), то оно будет восприниматься как текст. Это легко заметить: текст по умолчанию выравнивается по левому краю ячейки, а число — по правому.

¹ Какой именно знак используется для разделения целой и дробной частей числа определяется в операционной системе. В Windows это можно сделать, выбрав путь **Пуск | Панель управления | Языки и региональные стандарты | Настройка | Числа | Разделитель целой и дробной части числа** и определив нужный знак.

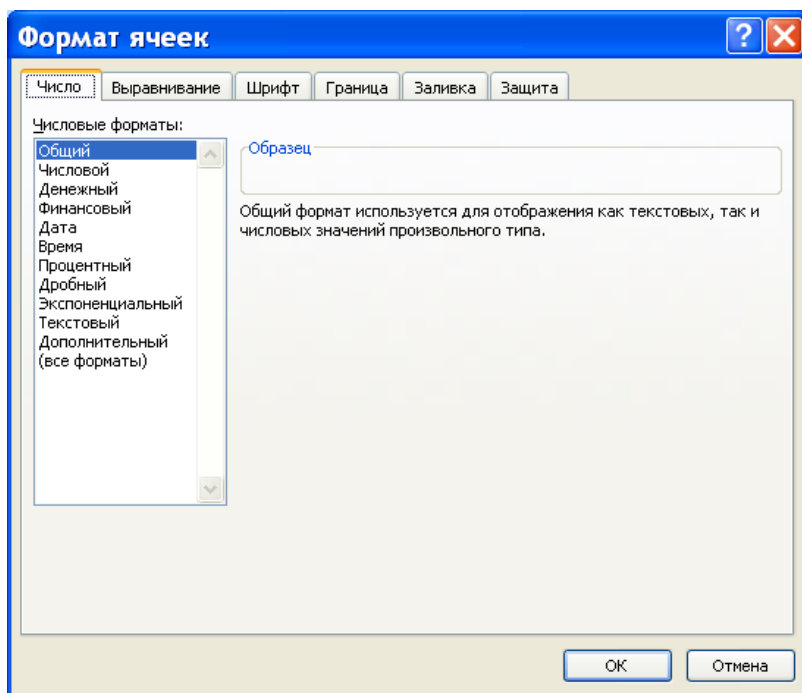


Рис. 1.4 — Диалоговое окно **Формат ячеек**
(закладка **Число**)

Если текст не входит в ячейку, то можно выполнить следующие действия:

- раздвинуть границы ячеек по горизонтали, встав курсором на границу между буквами столбцов (широкий крест курсора превращается в черный крестик со стрелками) и, удерживая нажатой левую клавишу мыши, сдвинуть границу на требуемое расстояние;

- объединить несколько ячеек и в них записать текст. Для этого необходимо выделить несколько соседних ячеек и открыть диалоговое окно **Формат ячеек**, как это было описано выше, и выбрать **Выравнивание | Объединение ячеек** (рис. 1.5). Эта опция также вынесена и на Ленту (вкладка **Главная | Выравнивание | Объединить и поместить в центре**);

- организовать перенос текста в ячейке по словам в диалоговом окне **Формат ячеек: Выравнивание | Переносить по словам**.

Переход в режим редактирования данных также можно осуществить щелчком по строке формул.

Для того чтобы **переместить данные**, следует выделить ячейку или блок, поместить курсор на рамку ячейки или блока (при этом курсор примет форму светлой стрелки со знаком «четырёх направлений»), нажать левую клавишу мыши и, удерживая ее, переместить ячейку или блок в требуемое место. **Копирование данных** производится аналогично перемещению, но с нажатой клавишей **Ctrl** (при этом курсор приобретает вид светлой стрелки со знаком «+»).

Аналогичные действия можно провести с помощью контекстно-зависимого меню (**Копировать, Вырезать, Вставить**) или через вкладку **Главная** | группа **Буфер обмена**.

Пример 1.1. Рассмотрим применение некоторых возможностей Excel при создании таблицы выполнения плана (рис. 1.1).

1. В ячейку B2 вводим текст «Задание. Определить процент выполнения плана и рост», в ячейку B3 — текст «производства по предприятиям». Изменять ширину ячеек нет смысла, т. к. в соседние справа ячейки текст не вносится и ничто не мешает увидеть набранный текст полностью.

2. Объединяем ячейки B5:B6 и вводим текст «Предприятия отрасли», центрируя его (**Формат ячеек** | **Выравнивание** | **По вертикали (По горизонтали)** | **По центру**) и изменяя ширину ячейки.

3. Объединяем ячейки C5:E5 и вводим текст «Объем производства, млн руб.». Аналогично объединяем F5:F6 — текст «Процент выполнения плана», и G5:G6 — текст «Относительный прирост». В последних двух блоках также задаем режим **Переносить по словам**. Передвигая границы ячеек, добиваемся требуемого расположения текста в ячейках.

4. В ячейки B7, B8, B9 вводим названия предприятий, а в ячейки C6, D6, E6 — текст «Отчет...», «План...».

5. Заполняем ячейки C7:E9 входными данными.

6. Заполняем строку «Среднее по отрасли». Для этого выделяем ячейку C10 и выбираем вкладку **Формулы**. На появившейся ленте выбираем **Вставить функцию** | **Категория** | **Статистические** | **Выберите функцию:** | **СРЗНАЧ** (рис. 1.6) | **ОК**. В появившемся окне задаем диапазон C7:C9. После нажатия кнопки **ОК**

в ячейке C10 появляется результат нахождения среднего значения содержимого ячеек C7:C9. Сама формула

=СРЗНАЧ (C7:C9)

высвечивается в строке формул.

Окно **Мастер функций** также открывается щелчком по кнопке со знаком f_x (**Вставить функцию**), расположенной слева от строки формул.

Примечание. Excel содержит большое количество встроенных функций: математических, статистических, финансовых и других, сгруппированных по категориям. Знание и умелое применение этих функций облегчает процесс обработки информации. Более подробную информацию о каждой функции можно найти, открыв справку по MS Excel (клавиша F1 клавиатуры) или выбрав ссылку **Справка по этой функции** в нижнем левом углу диалогового окна **Мастер функций** (рис. 1.6).

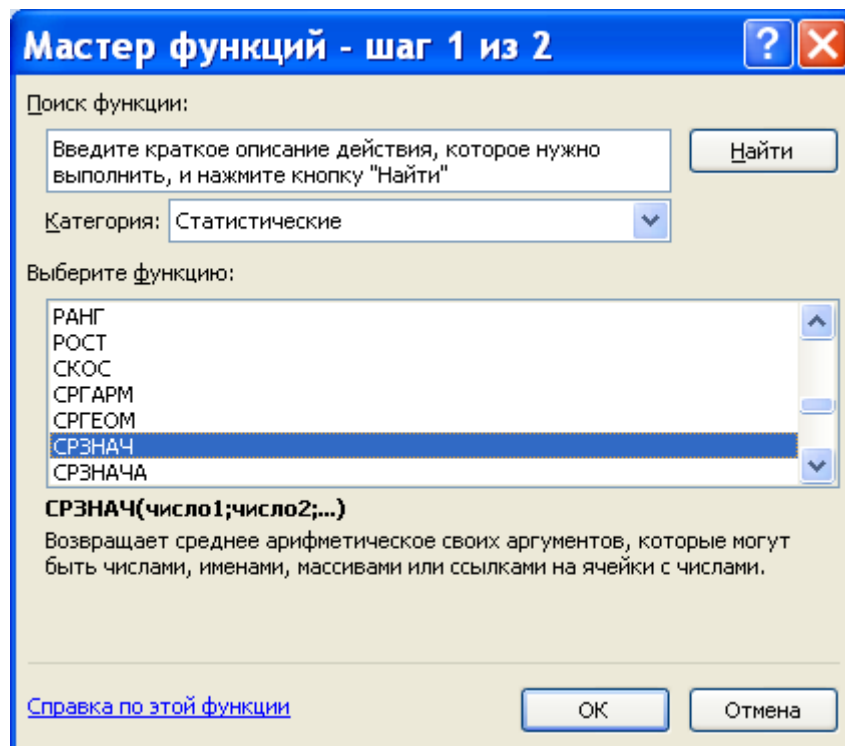


Рис. 1.6 — Диалоговое окно **Мастер функций**

7. Копируем полученную формулу в ячейки D10 и E10. Для этого указываем на маленький черный квадратик в правом нижнем

углу ячейки C10 (курсор при этом превращается в черный крестик — маркер заполнения), нажимаем левую кнопку мыши и, не отпуская ее, двигаем мышь вправо, пока рамка не охватит ячейки D10 и E10. В ячейке D10 появится формула:

=СРЗНАЧ (D7:D9),

а в ячейке E10 — формула:

=СРЗНАЧ (E7:E9).

Таким образом, при копировании произошла **автоматическая замена адресов в формуле**. Это очень полезное свойство Excel, позволяющее заметно упростить рутинные операции по вводу формул.

Примечание. Если же при копировании требуется оставить неизменным адрес какой-нибудь ячейки (или только столбца, или только строки), то перед именем столбца и/или номером строки ставится символ \$, например \$D\$5, H\$4, \$A2. Это называется **абсолютной адресацией ячейки**. Помечать знаком \$ номер строки (имя столбца) или снимать пометку можно также с помощью клавиши F4, поместив курсор рядом с адресом ячейки.

8. Заносим в ячейку F7 формулу:

=E7/D7,

а в ячейку G7 — формулу:

=(E7-C7)/C7.

9. Выделяем ячейки F7 и G7 и копируем сразу две формулы на ячейки F8:F10 и G8:G10 соответственно.

Чтобы задать процентный формат чисел в ячейках F7:G10, выделяем их, переходим в **Формат ячеек | Число | Числовые форматы: | Процентный | Число десятичных знаков | 2**. Это же самое выполняется быстрее при помощи кнопки **Процентный формат** на вкладке **Главная | Число**. Если число десятичных знаков меньше (или больше) требуемого, то следует использовать кнопки **Увеличить разрядность** или **Уменьшить разрядность** (там же). Таблица заполнена.

10. Оформим таблицу, нарисовав внутренние и внешние рамки: **Формат ячеек | Границы** или используя кнопку **Границы** на вкладке **Главная | группа Шрифт**.

Можно подобрать для разных частей таблицы различных фон (тип штриховки, цвет штриховки, цвет фона): **Формат ячейки** | **Заливка** или используя кнопки **Цвет заливки** и **Цвет текста** на вкладке **Главная** | группа **Шрифт**.

Аналогичные возможности предоставляет вкладка **Главная** | группа **Стили** | **Стили ячеек** и **Разметка страницы** | **Темы**.

Примечание. Для распространения формата одной ячейки на другую удобно пользоваться кнопкой **Формат по образцу** на вкладке **Главная** | **Буфер обмена**. Сначала нужно щелкнуть по «родительской» ячейке, затем по кнопке (к маркеру «прилипнет» знак кисти), затем по ячейке, куда нужно перенести формат «родительской» ячейки. При этом переносятся все параметры «родительской» ячейки: шрифт, формат числа, цвет, границы и т. п.

Теперь таблица окончательно готова — и в вычислительном аспекте, и в эстетическом.

При изменении исходных данных в ячейках C7:E9 результаты, находящиеся в ячейках C10:E10 и F7:G10, будут автоматически пересчитываться.

1.4 Автозаполнение

В Excel существует полезная *функция автозаполнения*, рекомендуемая при заполнении рядов данных.

Если ввести в две соседние ячейки последовательно два числа, составляющие начало арифметической прогрессии, например 1 и 2, затем их выделить и, как при копировании, с помощью маркера заполнения «протащить» выделение на несколько ячеек, то ряд продолжится: 1, 2, 3, 4 и т. д.

Excel также позволяет вводить и нечисловые последовательности. Например, если ввести в две соседние ячейки *Январь* и *Февраль* и осуществить описанную выше операцию, то в следующих ячейках появится *Март*, *Апрель* и т. д. Эти последовательности, или *списки*, можно сформировать самому и дать Excel запомнить их. Для этого необходимо открыть меню Excel (**Меню** или **Кнопка Office**) и выбрать **Параметры (Excel)** | **Создавать**

списки для сортировки и заполнения | Изменить списки и в окне **Элементы списка** записать (разделяя **Enter**) элементы, составляющие список.

Режим автозаполнения также включается при вызове команды **Прогрессия** на вкладке **Главная | группа Редактирование | Заполнить**. Перед этим надо задать начальное значение ряда и выделить диапазон заполнения.

1.5 Примечание

Информацию, находящуюся в любой ячейке Excel, можно прокомментировать с помощью вставки **Примечание**. Для этого необходимо выбрать ячейку, к которой следует добавить примечание, вызвать контекстно-зависимое меню, выполнить команду **Вставить примечание** и ввести текст примечания в появившемся поле. После окончания ввода текста следует щелкнуть кнопкой мыши вне области примечания.

Для просмотра примечаний в книге просто наведите указателем мыши на ячейку с примечанием (она помечена красным треугольником). Чтобы сразу увидеть все примечания, нужно нажать кнопку **Показать все примечания** на вкладке **Рецензирование**.

1.6 Дублирование данных

Одно и то же значение можно внести в несколько ячеек одновременно. Для этого выделите ячейки, в которые необходимо внести данные (они могут быть как смежными, так и несмежными), введите данные и нажмите клавиши **Ctrl+Enter**.

Если выделено несколько листов (это производят при нажатой клавише **Ctrl**), то вносимые в активный лист изменения автоматически дублируются в соответствующих ячейках всех остальных выбранных листов.

Если данные были введены на одном листе, то их можно скопировать в соответствующие ячейки других листов. Для этого выделите листы, на которые необходимо скопировать данные, выделите ячейки, содержащие копируемые данные, выберите команду **По листам** на вкладке **Главная | группа Редактирование | Заполнить**.

Чтобы отменить выделение нескольких листов книги, укажите любой невыделенный лист. Если же на экране видны только выделенные листы, наведите указатель на выделенный лист и нажмите правую кнопку мыши. Затем выберите команду **Разгруппировать листы** в контекстном меню.

Задание 1.1. Оформление таблицы:

1) открыть книгу Excel и разместить на листе таблицу из выбранного варианта. Лист переименовать в соответствии с вариантом задания;

2) добавить, если необходимо, новые строки и столбцы;

3) дополнительные исходные данные, не указанные в основной таблице, разместить во вспомогательных таблицах и ссылаться на них через адресацию ячеек;

4) в позиции, помеченные вопросительным знаком, внести формулы в соответствии с требуемым алгоритмом вычисления;

5) оформить таблицу, выделив заголовки, исходные данные и результаты вычислений. Использовать вставку примечаний для пояснения содержимого ячеек.

Вариант 1. Найти коэффициент Энгеля, т.е. рассчитать, какой процент занимают расходы на продукты в общей сумме расходов для разных групп населения. Какие данные лучше коррелируются: расходы на продукты и расходы на жилье или расходы на продукты и расходы на одежду?

Виды расходов	Группа 1	Группа 2	Группа 3	Группа 4	Группа 5
Продукты, руб.	370	580	1300	1950	2300
Жилье, руб.	88	125	180	200	1400
Коммунальные услуги, руб.	44	60	120	120	120
Одежда, руб.	80	220	800	1500	3500
Другие расходы, руб.	350	860	1200	2500	4400
Итого	?	?	?	?	?
Коэффициент Энгеля, %	?	?	?	?	?

Примечание. Использовать функцию КОРРЕЛ.

Вариант 2. Составить таблицу начисления заработной платы работникам МП «Воронья слободка».

Ф.И.О.	Тариф- ный раз- ряд	Выполнение плана, %	Тарифная ставка, руб.	Заработная плата с премией, руб.
Пряхин Н.П.	3	102	?	?
Суходейко А.Д.	2	98	?	?
Лоханкин В.А.	1	114	?	?
Пферд Л.Ф.	1	100	?	?
Севрюгов Л.А.	3	100	?	?
Гигиенишвили Г.С.	2	94	?	?
Птибурдуков А.И.	3	100	?	?

Примечание 1. Тарифная ставка определяется в зависимости от разряда: 1-й разряд — 14000 руб.; 2-й разряд — 16500 руб.; 3-й разряд — 18000 руб. Тарифные ставки оформить отдельной таблицей.

Примечание 2. Размер премиальных определяется в зависимости от выполнения плана:

- ниже 100 % — премия не начисляется;
- 100 % — премия 20 % от тарифной ставки;
- 101.....110 % — премия 30 %;
- 111115 % — премия 40 %.

Для реализации алгоритма начисления используйте вложенные функции ЕСЛИ.

Вариант 3. Рассчитать объем территориального фонда обязательного медицинского страхования в 2001 г. и сравнить показатели с 2000 г.

Виды поступлений в ФОМС	Сумма поступлений, млрд руб.		Удельный вес показателя, %	
	2000 г.	2001 г.	к итогу	к 2000 г.
1. Страховые взносы	101,48	148,88	?	?
2. Платежи на обязательное медицинское страхование неработающего населения	75,03	118,32	?	?
3. Штрафные санкции	12,77	18,0	?	?
4. Другие поступления	10,8	16,92	?	?
Итого	?	?	?	?

Вариант 4. Рассчитать ежедневный предполагаемый доход от деятельности киносети. Найти ежедневный средний доход с одного места.

Кинотеатры	Вместимость зрительного зала, мест	Средний % посещаемости	Цена билета, руб.	Количество сеансов в день	Доход, тыс. руб.
Родина	700	45	140	4	?
Сибиряк	150	40	130	5	?
Аэлита	300	60	125	3	?
Авангард	500	30	140	5	?
Итого	?	?		?	?

Вариант 5. Определить экономические показатели фирмы «Геркулес» в отчетном периоде.

Сегмент рынка	Себестоимость единицы продукции, тыс. руб.	Цена единицы продукции, тыс. руб.	Объем реализации, тыс. шт.	Затраты на производство, тыс. руб.	Выручка от реализации, тыс. руб.	Прибыль, тыс. руб.
A	14	16	10	?	?	?
B	13	15	15	?	?	?
C	14	16	20	?	?	?
D	11	13	10	?	?	?
Итого			?	?	?	?

Вариант 6. Рассчитать сумму затрат на выработку тепла по котельным МП «ТепТоп».

Статьи затрат	Затраты по типам котельных, тыс. руб.		Всего
	Газовая котельная	Угольная котельная	
1. Материалы	84,2	85,5	?
2. Амортизация	165,6	337,5	?
3. Вода	607,1	80,8	?
4. Электроэнергия	339,3	333,9	?
5. Заработная плата с начислениями	621,2	3081,0	?
6. Топливо	1234,5	2194,7	?
7. Ремонтный фонд	590,0	320,6	?
8. Цеховые расходы (11,9 % от ст. 10)	?	?	?
9. Общие эксплуатационные расходы (9,1 % от ст. 10)	?	?	?
10. Итого затрат по котельным	?	?	?

Статьи затрат	Затраты по типам котельных, тыс. руб.		Всего
	Газовая котельная	Угольная котельная	
11. Косвенные затраты	391,5	1709,5	?
12. Всего затрат по котельным с учетом косвенных	?	?	?
13. Рентабельность (15 % от ст. 12)	?	?	?
14. Итого затрат с учетом рентабельности	?	?	?
15. Доля затрат котельных разного типа, %	?	?	?

Вариант 7. Проанализировать динамику поступления товаров от поставщиков.

Поставщики	Поступление товаров, млн руб.		Превышение поступлений		Удельный вес поставщика в общей сумме поступлений, %	
	2010 г.	2011 г.	Сумма, млн руб.	В % к 2010 г.	2010 г.	2011 г.
ООО «Прима»	15,5	16,9	?	?	?	?
АОЗТ «Томь»	23,4	32,1	?	?	?	?
ЧП «Сантик»	0,96	1,2	?	?	?	?
ОАО «Гермес»	7,5	6,4	?	?	?	?
Всего	?	?	?	?	?	?

Вариант 8. Сравнить доходную часть городского бюджета в 2012 и 2013 годах.

Статьи дохода	Доход в 2012 г. (отчет)		Доход в 2013 г. (план)		Превышение доходов	
	Сумма, тыс. руб.	Удельный вес, %	Сумма, тыс. руб.	Удельный вес, %	Сумма, тыс. руб.	В % к 2012 г.
Налоговые доходы	?	?	?	?	?	?
1. Налоги на прибыль (доход), прирост капитала	347660	?	666562	?	?	?
2. Налоги на товары и услуги, лицензионные сборы	396110	?	142887	?	?	?
3. Налоги на совокупный доход	53810	?	35696	?	?	?
4. Налоги на имущество	266900	?	107253	?	?	?
5. Платежи за пользование природными ресурсами	102600	?	382380	?	?	?
6. Прочие налоги, пошлины и сборы	236580	?	274296	?	?	?

Статьи дохода	Доход в 2012 г. (отчет)		Доход в 2013 г. (план)		Превышение доходов	
	Сумма, тыс. руб.	Удельный вес, %	Сумма, тыс. руб.	Удельный вес, %	Сумма, тыс. руб.	В % к 2012 г.
Неналоговые доходы	?	?	?	?	?	?
1. Доходы от имущества, находящегося в государственности	10690	?	37366	?	?	?
2. Административные платежи и сборы	9500	?	4500	?	?	?
3. Штрафные санкции	3500	?	3600	?	?	?
Итого доходов	?	?	?	?	?	?

Вариант 9. Рассчитать начисление стипендии студентам по итогам сессии.

Ф.И.О.	Оценки за экзамены				Начислено стипендии, руб.
	Информатика	Экономическая теория	Теория вероятности	Базы данных	
1. Иванову И.П.	5	5	5	5	?
2. Зверев Д.Б.	4	4	5	5	?
3. Калачов Н.А.	4	5	5	5	?
4. Калачова Е.Б.	4	3	2	4	?
5. Синицына З.С.	4	5	3	5	?
6. Писаревская Л.Г.	5	5	4	4	?
7. Тарасов А.Н.	3	5	2	4	?
8. Паровицкий С.Т.	4	4	5	4	?

Примечание. Размер стипендии составляет 1560 руб. Если все экзамены сданы на «пятерку», то надбавка составляет 50 %. Если есть хотя бы одна «четверка» (при остальных «пятерках»), то надбавка составляет 25 %. Если есть хотя бы одна «двойка», то стипендия не начисляется. Для реализации алгоритма начисления используйте вложенные функции ЕСЛИ.

Вариант 10. Рассчитать сумму прибыли ЧП «Московские баранки».

Показатели	Годы			Превышение (понижение) показателей 2012 г. по сравнению с 2010 г.
	2010	2011	2012	
1. Объем продаж, т	30	47	62	?
2. Цена единицы продукции, руб./кг	70	80	100	?
3. Выручка от реализации продукции, руб.	?	?	?	?
4. Постоянные расходы, тыс.руб.	800	900	1000	?
5. Переменные расходы (60 % от ст. 3), руб.	?	?	?	?
6. Общая себестоимость продукции, руб.	?	?	?	?
7. Удельная себестоимость продукции, руб./кг	?	?	?	?
8. Валовая прибыль, руб.	?	?	?	?

Вариант 11. Рассчитать поступление и расходование денежных средств избирательных фондов зарегистрированных кандидатов в депутаты на должность главы администрации.

Показатели	Состояние избирательного фонда кандидатов					
	Полесов В.М.		Чарушников М.П.		Кислярский М.Б.	
	Сумма, руб.	%	Сумма, руб.	%	Сумма, руб.	%
Поступило средств всего,	?	?	?	?	?	?
в том числе:						
– от избирательной комиссии	2000	?	2000	?	2000	?
– собственные средства кандидата	800	?	800	?	2000	?
– пожертвования юридических лиц	125000	?	1057300	?	4193410	?
– пожертвования физических лиц	–	?	–	?	10590	?
Израсходовано средств всего,	?	?	?	?	?	?
в том числе:						
– радио и телевидение	–	?	33412	?	752600	?
– печатные издания	114418	?	604582	?	1332990	?
– публичные мероприятия	–	?	7780	?	200330	?
– канцелярские расходы	–	?	4169	?	106040	?
– аренда помещений и автотранспорта	8800	?	14392	?	95170	?
– прочие расходы	4940	?	30231	?	1172180	?
Остаток неизрасходованных средств	?		?		?	

Примечание. % — процент от общей суммы поступивших/израсходованных средств

Вариант 12. Рассчитать структуру розничной цены продукции исходя из данных таблицы.

Структурные составляющие розничной цены	Значения показателей по видам продукции, руб.			
	A	B	C	D
1. Себестоимость	500	710	360	120
2. Рентабельность (25 % от п. 1)	?	?	?	?
3. Оптовая цена	?	?	?	?
4. Акциз (70 % от оптовой цены)	?	?	?	?
5. Наценка посредника	10	10	8	8
6. НДС (18 % от отпускной цены фирмы)	?	?	?	?
7. Торговая наценка (10 %)	?	?	?	?
Розничная цена	?	?	?	?

Вариант 13. Рассчитать прибыль, полученную от реализации трех видов продукции.

Показатель	Продукция			Итого
	A	B	C	
1. Цена изделия, тыс. руб.	5	10	20	—
2. Количество изделий, реализуемых в рассматриваемом периоде, шт.	500	700	300	?
3. Выручка от реализации, тыс. руб.	?	?	?	?
4. Удельный вес каждого изделия в общем объеме реализации, %	?	?	?	?
5. Переменные расходы в расчете на одно изделие, тыс. руб.	3	6	12	—
6. Переменные расходы по каждому виду продукции, тыс. руб.	?	?	?	?
7. Постоянные расходы в рассматриваемом периоде, тыс. руб.	—	—	—	6000
8. Итого расходов, тыс. руб.				?
9. Прибыль, тыс. руб.				?

Вариант 14. Рассчитать изменение денежных средств с учетом доходов, полученных от вложения финансовых средств в одно-, трех- и шестимесячные депозиты к началу 7-го месяца. Доход от одномесячного депозита — 1 %, от трехмесячного депозита — 4 %, от шестимесячного депозита — 9 %. Результаты расчета предлагается оформить в виде следующей таблицы.

Статьи доходов (расходов), тыс. руб.	Период, месяц						Итого на начало 7-го месяца
	1	2	3	4	5	6	
Начальная сумма	500	?	?	?	?	?	?
Погашение депозитов	—	?	?	?	?	?	?
Проценты по депозитам	—	?	?	?	?	?	?
1-месячный депозит	50	50	50	50	50	50	—
3-месячный депозит	100	—	—	100	—	—	—
6-месячный депозит	150	—	—	—	—	—	—
Внутренние расходы	30	30	30	30	30	30	—
Сумма на конец месяца	?	?	?	?	?	?	—
Итого							?

Примечание. Процентные ставки по депозитам также оформить в виде таблицы и при ссылке на них использовать адреса соответствующих ячеек.

Вариант 15. Проанализировать затраты на рекламу производителей товаров и услуг по различным секторам рынка. Стоимость рекламных объявлений в газете «Реклама», руб. за 1 кв. см: первая полоса — 75, последняя полоса — 43, полоса с программой ТВ — 30, обычная полоса — 15 (данные оформить отдельной таблицей и ссылаться на них через адреса ячеек).

Для определения места по затратам использовать функцию РАНГ.

Секторы рынка	Площадь объявлений, кв. см				Затраты на рекламу, тыс. руб.	Место по за- тратам
	Первая полоса	Последняя полоса	Полоса с программой ТВ	Обычная полоса		
Информационные технологии и услуги	430,8	135,0	208,8	5363,5	?	?
Продовольственные товары	41,3	0	0	4191,3	?	?
Строительно- хозяйственные то- вары и услуги	149,3	138,5	488,3	3697,8	?	?
Предметы гигиены и санитарии	0	0	0	949,0	?	?
Одежда и обувь	0	0	108,0	534,5	?	?
Мебель и торговое оборудование	134,0	0	0	2071,3	?	?
Лекарства и меди- цинские услуги	0	0	42,3	568,0	?	?
Автотовары и авто- услуги	50,0	0	0	2648,1	?	?

Секторы рынка	Площадь объявлений, кв. см				Затраты на рекламу, тыс. руб.	Место по затратам
	Первая полоса	Последняя полоса	Полоса с программой ТВ	Обычная полоса		
Бытовая техника и ее обслуживание	0	420,0	0	1680,3	?	?
Недвижимость	76,3	0	0	2087,3	?	?
Итого	?	?	?	?	?	?

1.7 Построение диаграмм

С помощью Excel можно превращать сухие и абстрактные строки и столбцы чисел в привлекательные и информативные диаграммы. Визуальное представление информации облегчает ее восприятие, помогает лучше представить поведение функциональных зависимостей.

В Excel используются различные типы диаграмм: гистограммы (изображение данных в виде столбиков различной формы), графики (в виде линий), диаграммы-области (сплошная заливка), круговые, кольцевые, лепестковые, трехмерные и другие диаграммы.

Вызов списка типов на лист Excel происходит по команде **Вставка | группа Диаграммы | Вставка диаграммы** (рис. 1.7).

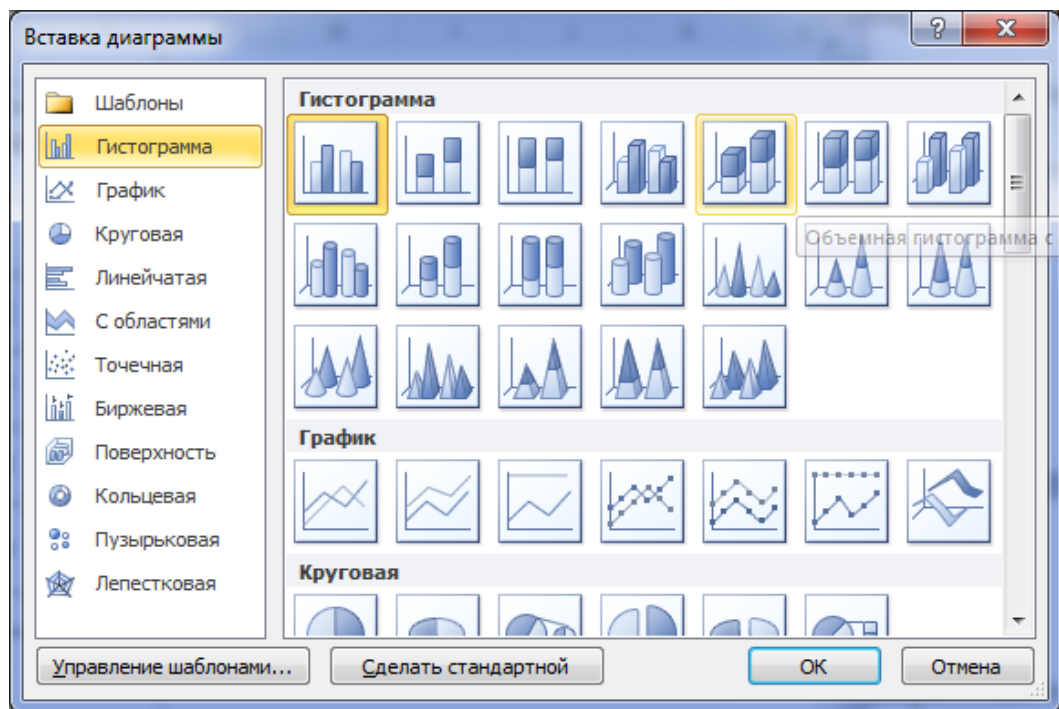


Рис. 1.7 — Типы диаграмм

1.7.1 Построение гистограммы

Пример 1.2. Визуализируем некоторые данные из таблицы 1.1, показывающей доход от реализации колбасных изделий ООО «Рога и копыта».

Таблица 1.1

Наименование изделий	Объем производства, т	Цена за кг, руб.	Торгово-сбытовая скидка, %	Цена со скидкой, руб.	Сумма, тыс. руб.
Колбаса арбатовская п/к, 1 сорт	5	350	8	322	1750
Колбаса остаповская п/к, 1 сорт	12	360	8	331,2	4322
Колбаски восточные, п/к, в/с	2	580	8,5	530,7	1160
Колбаса «Геркулес» п/к, 2 сорт	14	250	7,8	230,5	3500
Колбаса черноморская п/к, в/с.	3	660	8,5	603,9	1980
Итого	36				12710

Порядок действий следующий:

Вариант 1

1. Выделим первые два столбца вместе с заголовками, но без «Итого». В рассматриваемом примере выделять всю таблицу не рекомендуется, т. к. на одном поле построения диаграммы следует показывать строки или столбцы, содержащие *группы данных одного наименования* (или руб., или кг, или %). «Итого» вместе со слагаемыми обычно не показывают.

2. Выбираем вкладку **Вставка** | группа **Диаграммы** | **Вставка диаграммы** (рис. 1.7).

3. Выбираем тип диаграммы **Гистограмма** и вид диаграммы **Объемная диаграмма с группировкой** (название проявляется при наведении курсора на объект).

4. Щелчок по выбранному типу приводит к появлению гистограммы заказанного вида на выбранном массиве данных (рис. 1.8).



Рис. 1.8 — Гистограмма, построенная по двум столбцам табл. 1.1 (легенда удалена)

Вариант 2

Можно начинать построение диаграммы и без выделения области построения, выбрав двойным щелчком вид диаграммы в окне **Вставка диаграммы** (рис. 1.7). При этом в области листа Excel появится пустое поле, а в области ленты — лента «Работа с диаграммами» (рис. 1.9), вкладка **Выбрать данные** которой дает возможность выбрать необходимые данные для построения диаграммы (рис. 1.10). Также обратите внимание на вкладки этой ленты **Макет** и **Формат**, предоставляющие богатые возможности для форматирования созданной диаграммы.

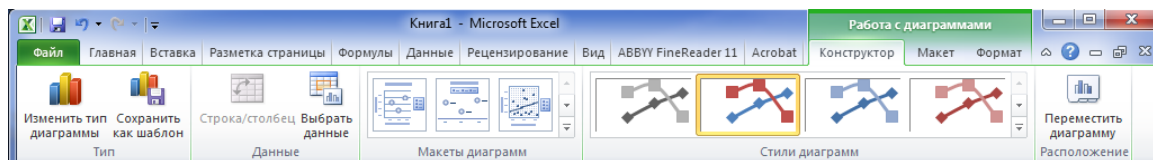


Рис. 1.9 — Лента Работа с диаграммами

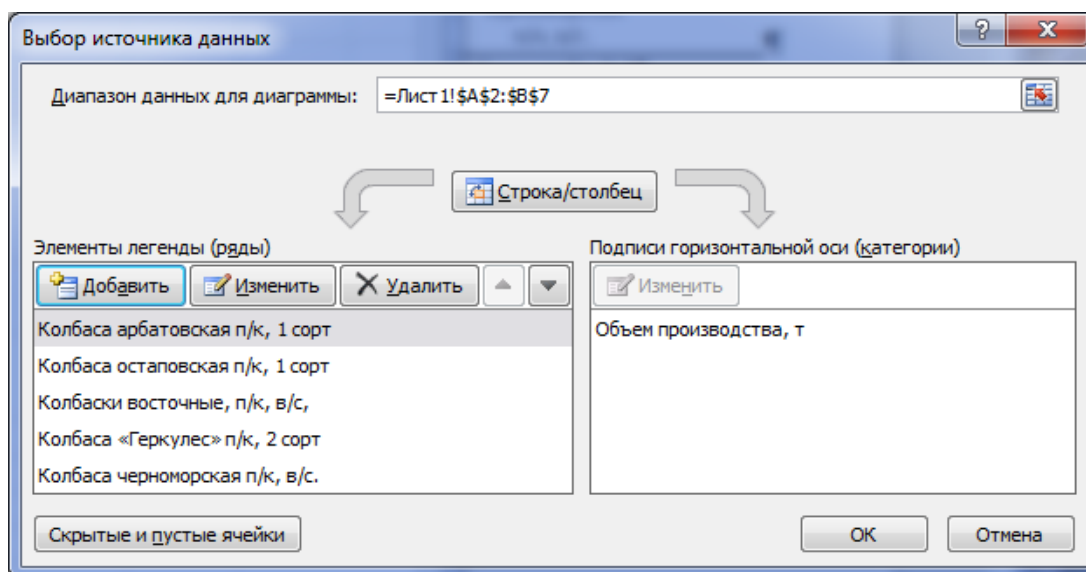


Рис. 1.10 — Диалоговое окно **Выбор источника данных**

Выбираем ту же область, но (для разнообразия) меняем местами строки и столбцы (кнопка **Строка/Столбец**). Гистограмма принимает вид (рис. 1.11): подписи строк стали легендой диаграммы, а заголовок — подписью строк.

Столбики гистограммы имеют разные цвета. Но если предполагается размещение полученной гистограммы в черно-белом варианте, то вместо цветовой гаммы рекомендуется использовать различную штриховку.

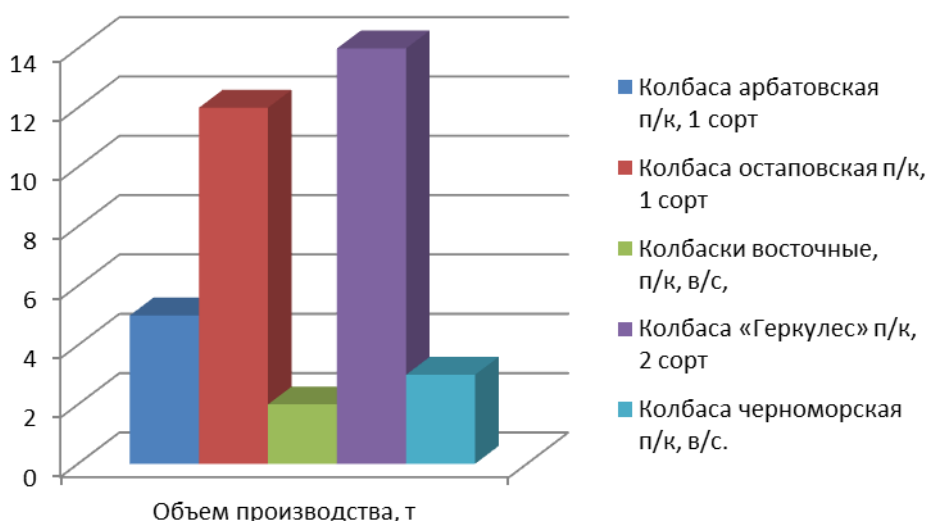


Рис. 1.11 — Гистограмма, построенная по двум столбцам табл. 1.1

Для этого, щелкнув по выбранному столбику правой кнопкой мыши, выбираем в контекстном меню команду **Формат ряда**

данных. В появившемся окне выбираем **Заливка | Узорчатая заливка** (рис. 1.12) и соответствующий тип заливки столбика. Операцию повторяем для каждого столбца.

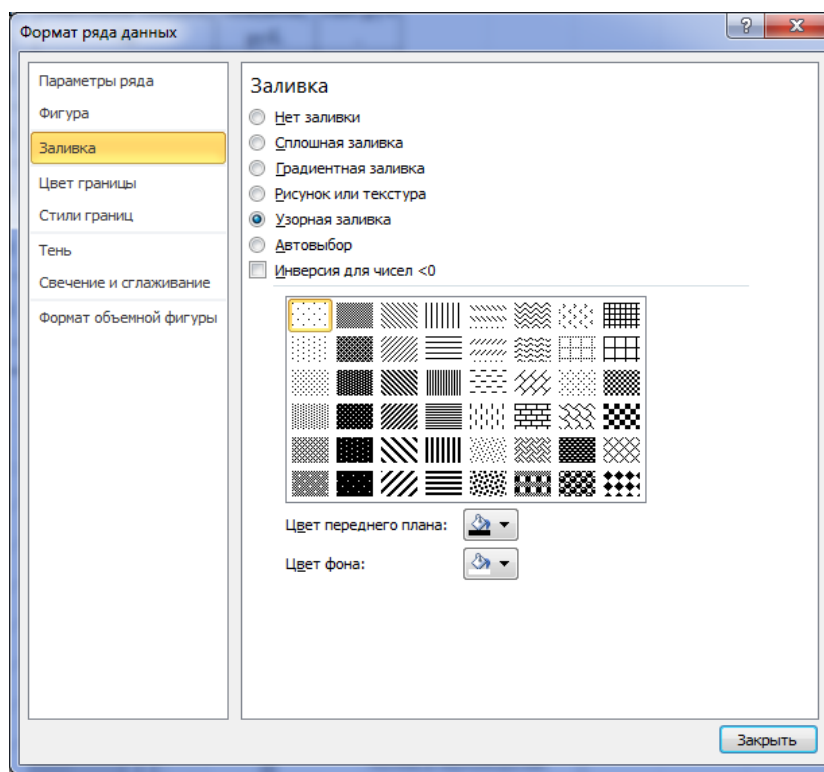


Рис. 1.12 — Диалоговое окно
Формат ряда данных

Гистограмма приобретает следующий вид (рис. 1.13).

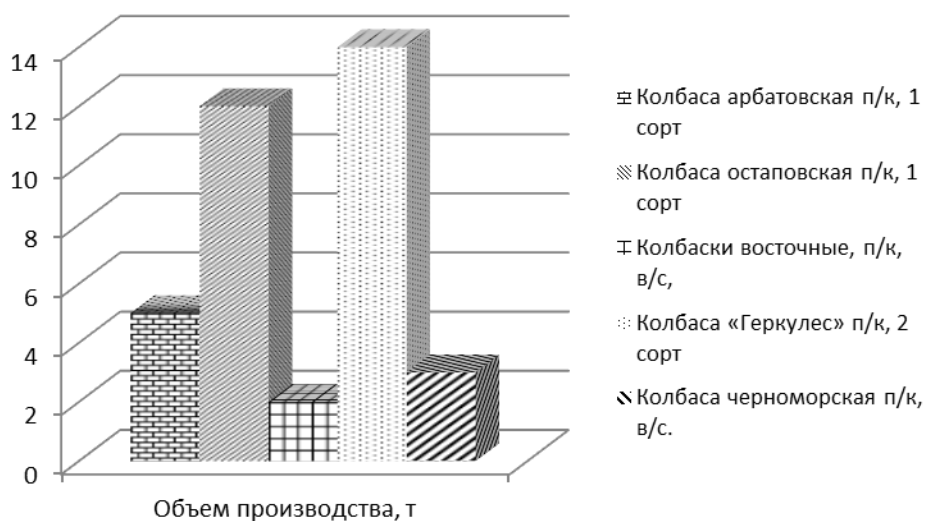


Рис. 1.13 — Вариант гистограммы с узорчатой заливкой

Также в диалоговом окне **Формат ряда данных** можно изменить и другие характеристики построенной гистограммы.

После построения диаграммы имеется возможность изменить все ее параметры. Для этого достаточно щелкнуть правой клавишей мыши по области построения диаграммы и выбрать в появившемся контекстно-зависимом меню необходимую опцию (обратите внимание, что меню будет иметь разный вид в зависимости от того, по какому месту щелкнуть: по легенде, по оси, по стенке, по области диаграммы).

1.7.2 Построение графика функции одной переменной

Тип диаграммы **Точечная** показывает отношения между численными значениями в нескольких группах. По умолчанию первая группа (столбец или строка) принимается за аргумент (х), а другие — за соответствующие значения функций.

Точечная диаграмма имеет две оси значений, при этом одни числовые значения выводятся вдоль горизонтальной оси (оси X), а другие — вдоль вертикальной оси (оси Y). Точечные диаграммы обычно используются для иллюстрации и сравнения числовых значений, например научных, статистических или технических данных.

Пример 1.3. В табл. 1.2 приведена зависимость цены единицы некоторого товара от объема его продаж (известная в экономике «кривая спроса $D-D$ »).

Таблица 1.2

Объем продаж, тыс. шт.	8	10	18	20	32	40	50
Цена, руб.	510	430	350	280	200	100	80

Для построения кривой спроса:

1) выделяем всю таблицу, вызываем **Вставка | группа Диаграммы | Вставка диаграммы | Точечная | Точечная с гладкими кривыми | ОК**;

2) убираем легенду, а вместо названия диаграммы «Цена, руб.» вписываем «Спрос»;

3) на ленте **Работа с диаграммами** выбираем вкладку **Макет** и исполняем команду **Названия осей | Название основной горизонтальной оси | Название под осью**. Вместо текста «Название оси», появившегося на диаграмме, пишем «Объем продаж, тыс. шт.»;

4) аналогично для вертикальной оси выбираем опцию «Повернутое название» и вписываем «Цена, руб.».

Построение и форматирование графика закончено (рис. 1.14).

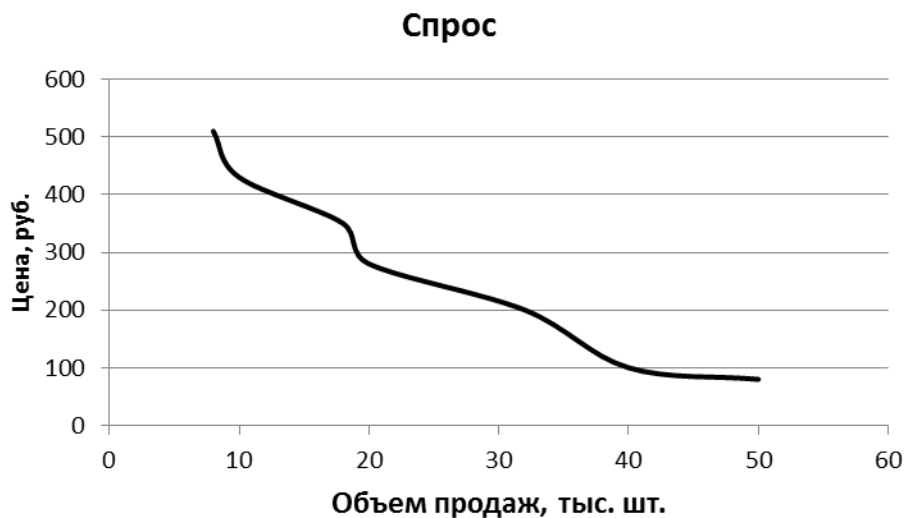


Рис. 1.14 — График спроса, построенный по табл. 1.2

Если функция задана аналитической зависимостью $y = f(x)$, то нужно предварительно ее **протабулировать**, т. е. построить таблицу из двух строк (или столбцов) $\{x_i, y_i\}$, где $x_i = x_0 + i \cdot h$ — узловые точки; $h = (x_n - x_0) / n$ — шаг табуляции; $i = 0, \dots, n$, а $y_i = f(x_i)$. Для заполнения ряда x можно использовать режим автозаполнения или формулу увеличения значения x на один шаг.

Пример 1.4. Построить на одном поле графики зависимости функций $y = \sin(x)$ и $y = \cos(x)$ в диапазоне $x \in [-2; 2]$.

Табулируем функции, т. е. строим таблицу значений функций $y = \sin(x)$ и $y = \cos(x)$ в указанном диапазоне (рис. 1.14).

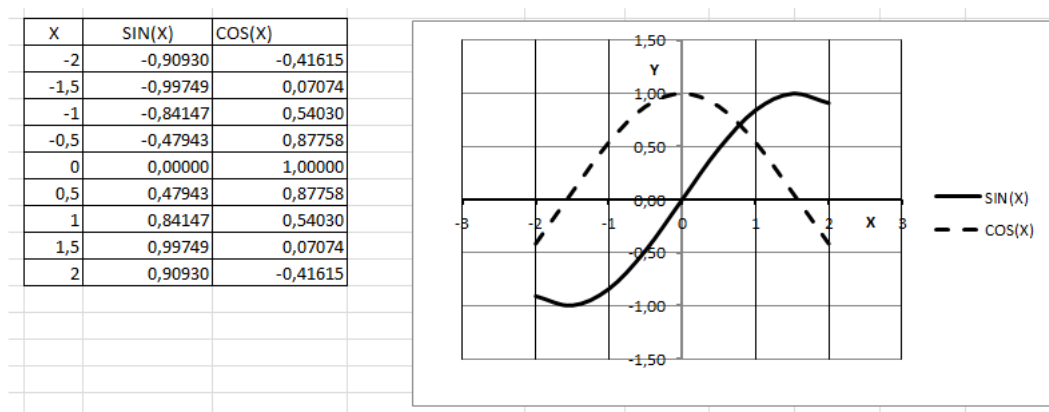


Рис. 1.14 — Таблица и график функций $y = \sin(x)$ и $y = \cos(x)$

Выделяем всю таблицу, выбираем **Вставка** | группа **Диаграммы** | **Точечная с гладкими кривыми** | **ОК**. На листе появляется диаграмма с графиками (рис. 1.15). Проводим ее форматирование.

1) Наводим указатель мыши на линию графика $y = \cos(x)$, щелкаем правой кнопкой мыши, выбираем **Формат ряда данных** | **Тип линии** | **Тип штриха** | **Пунктир** | **Заккрыть**. Линия становится пунктирной.

2) Наводим указатель мыши на оцифровку вертикальной оси, щелкаем правой кнопкой мыши, выбираем **Формат оси** | **Число** | **Число десятичных знаков** | **2** | **Заккрыть**. Установлен показ 2-х знаков после запятой в числах вертикальной оси.

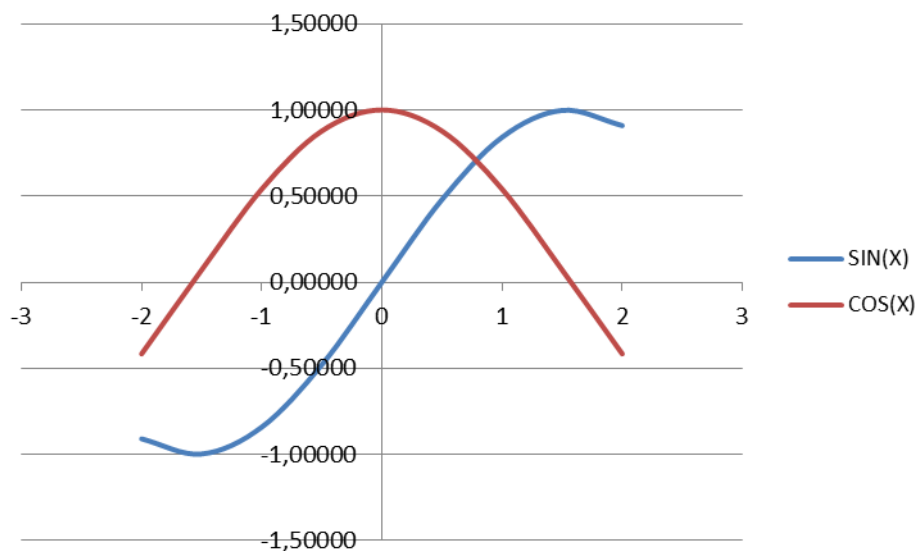


Рис. 1.15 — График функций $y = \sin(x)$ и $y = \cos(x)$
(до форматирования)

3) Наводим указатель мыши на оцифровку горизонтальной оси, щелкаем правой кнопкой мыши, выбираем **Добавить основные линии сетки**. Появились вертикальные линии сетки (рис. 1.14).

1.7.3 Построение графика функции двух переменных

Графическое представление функции двух переменных — поверхность в трехмерном пространстве.

Пример 1.5. Построим график производственной функции Кобба—Дугласа

$$Y = A \times K^{\alpha} \times L^{\beta},$$

где A, α, β — константы, K — объем производственных фондов, L — объем трудовых ресурсов, Y — выпуск продукции предприятием или отраслью. Эти переменные могут выражаться либо в стоимостном выражении, либо в натуральном количестве.

Пусть функция имеет вид:

$$Y = 900 \times K^{0,5} \times L^{0,25},$$

где $K = 100 \dots 300$ тыс. руб.; $L = 30 \dots 200$ тыс. руб.

Табулируем функцию, располагая значения K по горизонтали с шагом 10, а L — по вертикали с шагом 5. На пересечении столбца со значением K_i и строки со значением L_j вводим значение функции $Y_{i,j} = 900 \times K_i^{0,5} L_j^{0,25}$ (рис. 1.16). Верхнюю левую ячейку оставляем пустой.

при наборе формулы необходимо зафиксировать знаком \$ номер строки переменной, изменяющейся по горизонтали (т. е. K), и номер столбца переменной, изменяющейся по вертикали (т. е. L).

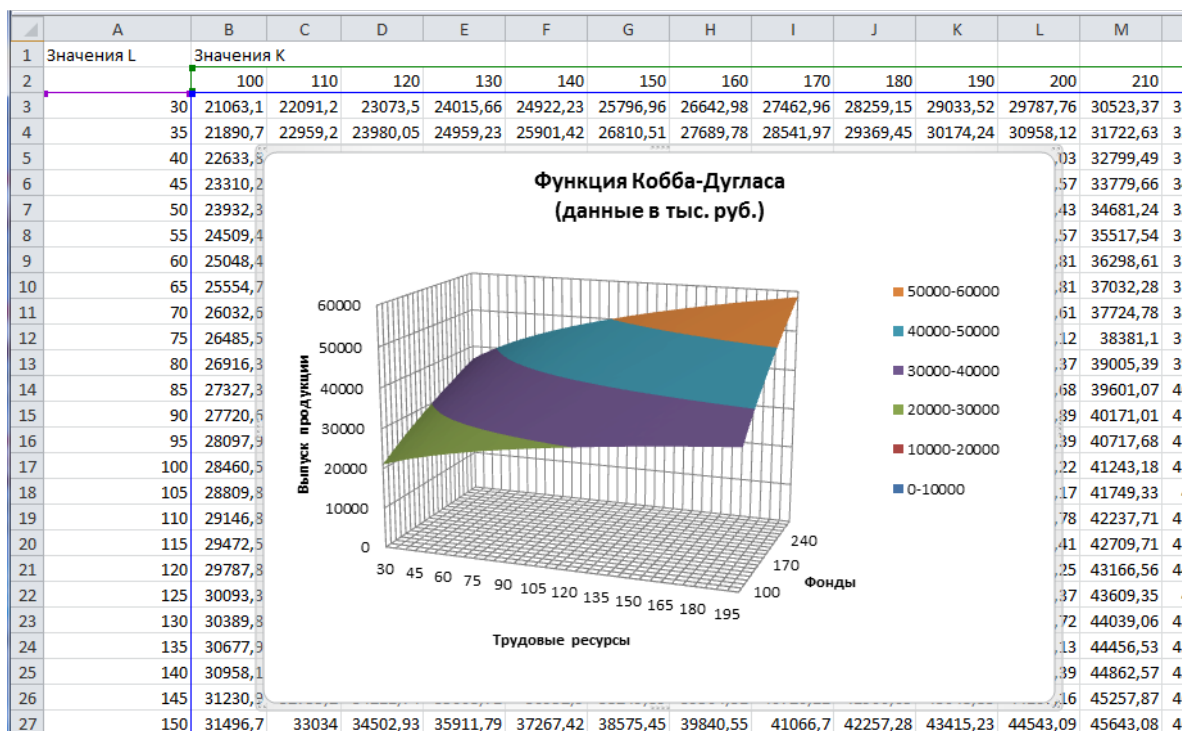


Рис. 1.16 — Элемент листа Excel с табулированием функции двух переменных

Например, в ячейке C4 находится формула вида

$$=900 * C\$2^{0,5} * \$A4^{0,25}.$$

Тогда достаточно один раз ввести формулу, например в ячейку В3, и затем скопировать ее на все ячейки таблицы. Произойдет корректная автоматическая смена адресов переменных (проверьте!).

Порядок построения этой поверхности следующий.

1. Выделяем всю таблицу (и значения аргументов, и значения функции), начиная с пустой левой верхней ячейки и заканчивая правой нижней (A2:V37).

2. Выбираем **Вставка** | группа **Диаграммы** | **Поверхность** (если отсутствует, то **Другие** | **Поверхность**) ОК.

3. Проводим необходимое форматирование, элементы которого описаны выше.

Получили поверхность, представленную на рис. 1.16.

Используя опцию **Макет** | **Поворот объемной фигуры**, можно двигать область диаграммы, рассматривая поверхность в различных ракурсах.

Задание 1.2. Построение гистограмм

Построить гистограммы, иллюстрирующие табличные данные из задания 1.1. Тип диаграммы выбрать исходя из степени наглядности представления информации.

Задание 1.3. Построение графиков

Построить графики функции одной переменной на отрезке $x \in [-2; 2]$ для одного из выбранных вариантов, приведенных ниже, для разных шагов табуляции: 0,5; 0,2; 0,1. Сравнить вид графиков, сделать выводы.

Примечание. При записи формулы использовать функцию ЕСЛИ. Она возвращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если ЛОЖЬ.

Синтаксис этой функции (рис. 1.17):

ЕСЛИ(условие;значение_если_истина;значение_если_ложь).

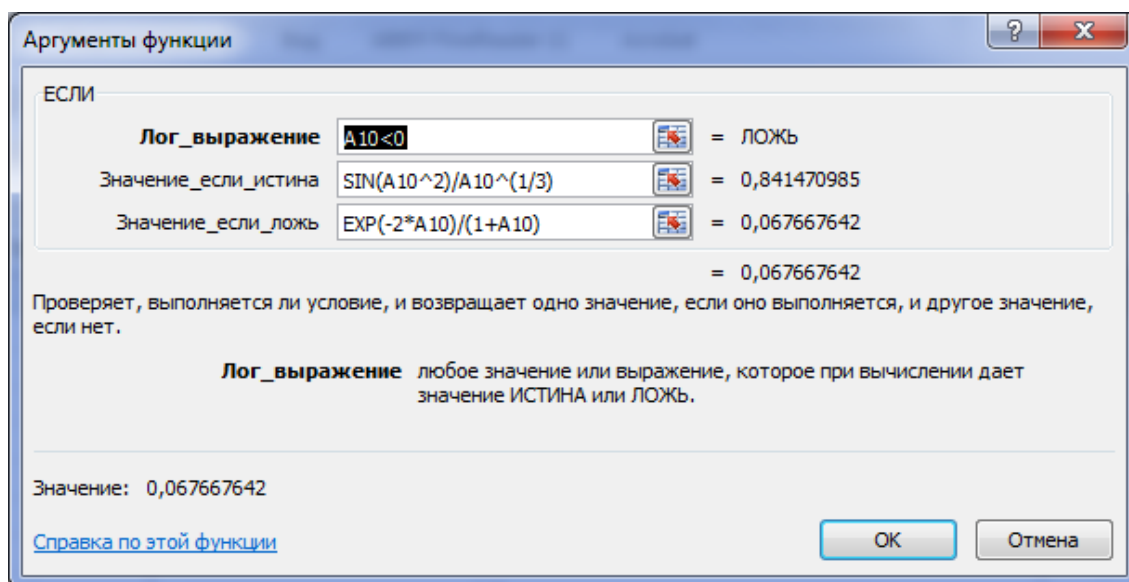


Рис. 1.17 — Диалоговое окно функции ЕСЛИ.
Значение ячейки A10 равно 1

В условии для сравнения двух значений используются операторы сравнения: = (равно); > (больше); < (меньше); >= (не меньше); <= (не больше); <> (не равно).

Например, формула =ЕСЛИ(A1>=0;КОРЕНЬ(A1);"число отрицательное!"), записанная в какой-либо ячейке, помещает в нее значение квадратного корня из числа, находящегося в ячейке A1, при его неотрицательном значении, и выдает предупредительный текст в случае отрицательного значения.

Сравнивать можно как арифметические, так и текстовые выражения.

Если после набора формулы в одном из полей (рис. 1.17), она вдруг окажется охваченной двойными кавычками, т. е. интерпретирована как текст, — ищите ошибку. Это может быть несоответствие скобок, неверно записанное или использованное имя функции, адрес ячейки, содержащий русские буквы, пропущенный знак умножения и т. п.

1	$y = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, & x \leq 0 \\ 2x + \frac{\sin^2(x)}{3+x}, & x > 0 \end{cases}$	2	$y = \begin{cases} 3\sin(x) - \cos^2(x), & x \leq 0 \\ \frac{3\sqrt{1+x^2}}{\ln(x+5)}, & x > 0 \end{cases}$
3	$y = \begin{cases} \frac{3+\sin^2(2x)}{1+\cos^2(x)}, & x \leq 0 \\ 2x + \frac{\sin^2(x)}{3+x}, & x > 0 \end{cases}$	4	$y = \begin{cases} \frac{3x^2}{1+x^2}, & x \leq 0 \\ \sqrt{1 + \frac{2x}{e^{0,5x} + x^2}}, & x > 0 \end{cases}$
5	$y = \begin{cases} \frac{3+\sin^2(x)}{1+x^2}, & x \leq 0 \\ 2x^2 \cos^2(x), & x > 0 \end{cases}$	6	$y = \begin{cases} \sqrt{1+2x^2 - \sin^2(x)}, & x \leq 0 \\ \frac{2+x}{\sqrt[3]{2+e^{-0,1x}}}, & x > 0 \end{cases}$
7	$y = \begin{cases} \sqrt{1+x^2}, & x \leq 0 \\ \frac{1+x}{\sqrt[3]{1+e^{-0,2x}} + 1}, & x > 0 \end{cases}$	8	$y = \begin{cases} \sqrt{1+ x }, & x \leq 0 \\ \frac{1+3x}{\sqrt[3]{1+x+2}}, & x > 0 \end{cases}$
9	$y = \begin{cases} 2\ln(1+x^2), & x \leq -1 \\ (1+\cos^2(x))^{\frac{3}{5}}, & x > -1 \end{cases}$	10	$y = \begin{cases} \sqrt[3]{1+x^2}, & x \leq 0 \\ \sin^2(x) + \frac{1+x}{1+e^x}, & x > 0 \end{cases}$

11	$y = \begin{cases} \frac{1+ x }{\sqrt[3]{1+x+x^2}}, & x \leq -1 \\ \frac{1+\cos^4(x)}{3+x}, & x > -1 \end{cases}$	12	$y = \begin{cases} \frac{\sqrt{1+ x }}{2+ x }, & x \leq 0 \\ \frac{1+x}{2+\cos^3(x)}, & x > 0 \end{cases}$
13	$y = \begin{cases} \frac{1+x}{\sqrt[3]{1+x^2}}, & x \leq 0 \\ -x+2e^{-2x}, & x > 0 \end{cases}$	14	$y = \begin{cases} 3x+\sqrt{1+x^2}, & x \leq 0 \\ 2\cos(x)e^{-2x}, & x > 0 \end{cases}$
15	$y = \begin{cases} \sqrt{1+\frac{x^2}{1+x^2}}, & x \leq 0 \\ 2/ \cos(x) , & x > 0 \end{cases}$	16	$y = \begin{cases} x ^{\frac{1}{3}}, & x \leq 0 \\ -2x+\frac{x}{3+x}, & x > 0 \end{cases}$
17	$y = \begin{cases} \frac{1+x}{1+x^2}, & x \leq 0 \\ \sqrt{1+\frac{\cos(x)}{3+x}}, & x > 0 \end{cases}$	18	$y = \begin{cases} \frac{1+x+x^2}{1+x^2}, & x \leq 0 \\ \sqrt{1+\frac{2\sin(x)}{1+x^2}}, & x > 0 \end{cases}$
19	$y = \begin{cases} 1+\frac{3+x}{1+x^2}, & x \leq 0 \\ \sqrt{1+(1-\sin(x))^2}, & x > 0 \end{cases}$	20	$y = \begin{cases} \frac{1+2x}{1+x^2}, & x \leq 0 \\ \sin^2(x)\sqrt{1+x}, & x > 0 \end{cases}$

Раздел 2. Численное решение уравнений

Цель раздела: освоить приемы решения нелинейных уравнений, систем линейных и нелинейных уравнений в среде Excel, познакомиться с возможностями сервисных программ **Подбор параметра** и **Поиск решения**.

С помощью Excel можно решать разнообразные математические задачи, возникающие при реализации методов экономико-математического моделирования (и не только). Например, балансовая модель производства и потребления совокупного общественного продукта представляет собой замкнутую систему линейных алгебраических уравнений, финансовое моделирование сводится к решению нелинейных уравнений, реализация методов экстраполяции на основе кривых роста приводит к решению систем линейных и нелинейных уравнений и т. п. [5].

2.1 Решение полиномов

Для полиномов $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ модули всех действительных корней $x_k, k = 1 \dots n$ расположены в диапазонах

$$A < |x_k| < B, \quad (2.1)$$

где

$$A = \frac{1}{1 + \frac{\max\{|a_n|, |a_{n-1}|, \dots, |a_1|\}}{|a_0|}};$$

$$B = 1 + \frac{\max\{|a_{n-1}|, |a_{n-2}|, \dots, |a_0|\}}{|a_n|}.$$

Следовательно, все действительные положительные корни лежат в интервале $[A, B]$, а все действительные отрицательные корни — в интервале $[-B, -A]$.

Задание 2.1. Корни полинома

Найти корни полинома с заданной точностью ε .

Предлагается следующий алгоритм.

1. В диалоговом окне **Параметры Excel** задать относительную погрешность вычислений корней $\varepsilon = 0,00001$: **Файл | Параметры | Формулы | Относительная погрешность**.

2. Определить A и B по формуле (2.1), разместив предварительно на листе Excel таблицу коэффициентов выбранного полинома (табл. 2.1).

3. Составить таблицу $\{x, P(x)\}$, табулируя полином в найденных интервалах, например с шагом $h = (B - A)/10$ (см. п. 1.7.2 лабораторной работы 1).

Таблица 2.1

Вариант	Уравнение	Вариант	Уравнение
1	$x^4 + 6x^3 + 11x^2 - 2x - 28 = 0$	11	$x^4 + 3x^3 + 8x^2 - 5 = 0$
2	$x^4 + 5x^3 + 9x^2 + 5x - 1 = 0$	12	$x^4 + 6x^3 + 11x^2 + 2x - 28 = 0$
3	$x^4 + 3x^3 + 3x^2 - 2 = 0$	13	$x^4 + 5x^3 + 9x^2 - 5x - 1 = 0$
4	$x^4 + x^3 - 11x^2 + 8x - 6 = 0$	14	$x^4 + 3x^3 + 3x^2 - 2 = 0$
5	$x^4 - 10x^3 + 16x + 5 = 0$	15	$x^4 - x^3 - 7x^2 - 8x - 6 = 0$
6	$x^4 - 3x^3 - 4x^2 - x - 3 = 0$	16	$x^4 - 10x^2 - 16x + 5 = 0$
7	$x^4 + 4x^3 + 4x^2 + 4x - 1 = 0$	17	$x^4 + 3x^3 + 4x^2 + x - 3 = 0$
8	$x^4 + 6x^3 + 13x^2 + 10x + 1 = 0$	18	$x^4 - 4x^3 - 4x^2 - 4x - 1 = 0$
9	$x^4 + x^3 - 4x^2 + 16x - 8 = 0$	19	$x^4 + 2x^3 + 3x^2 + 2x - 2 = 0$
10	$x^4 - x^3 - 4x^2 - 11x - 3 = 0$	20	$x^4 - 6x^3 + 13x^2 - 10x + 1 = 0$

4. Определить две соседние ячейки столбца x , где функция меняет свой знак, и выделить их цветом. Одно из значений, для которого значение функции ближе к нулю, принять за *начальное приближение* к корню полинома.

5. Уточнить значение корня с помощью сервисной команды **Подбор параметра (Данные | Анализ «что если» | Подбор параметра)** (рис. 2.1). В поле **Установить в ячейке** ввести адрес ячейки, где вычисляется значение полинома, соответствующее выбранному начальному приближению. В поле **Значение** ввести 0 (т. е. искомое значение полинома). В поле **Изменяя значение ячейки** ввести адрес ячейки, где находится начальное приближение к корню полинома.

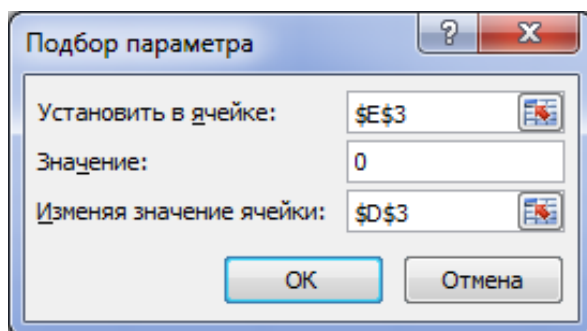


Рис. 2.1 — Диалоговое окно
Подбор параметра

Примечание. В этой ячейке (D3 на рис. 2.1) должно содержаться числовое значение, а не формула, его вычисляющая. Для того чтобы заменить в ячейке формулу на ее числовое значение, необходимо, находясь в этой ячейке, вызвать контекстно-зависимое меню и выбрать **Копировать**. Затем, находясь в той же ячейке, снова вызвать контекстно-зависимое меню и выбрать **Специальная вставка** (рис. 2.2).

В появившемся диалоговом окне отметить **Вставить значения**. После этого ячейка готова к использованию в поле **Изменяя значение ячейки** диалогового окна **Подбор параметра**.

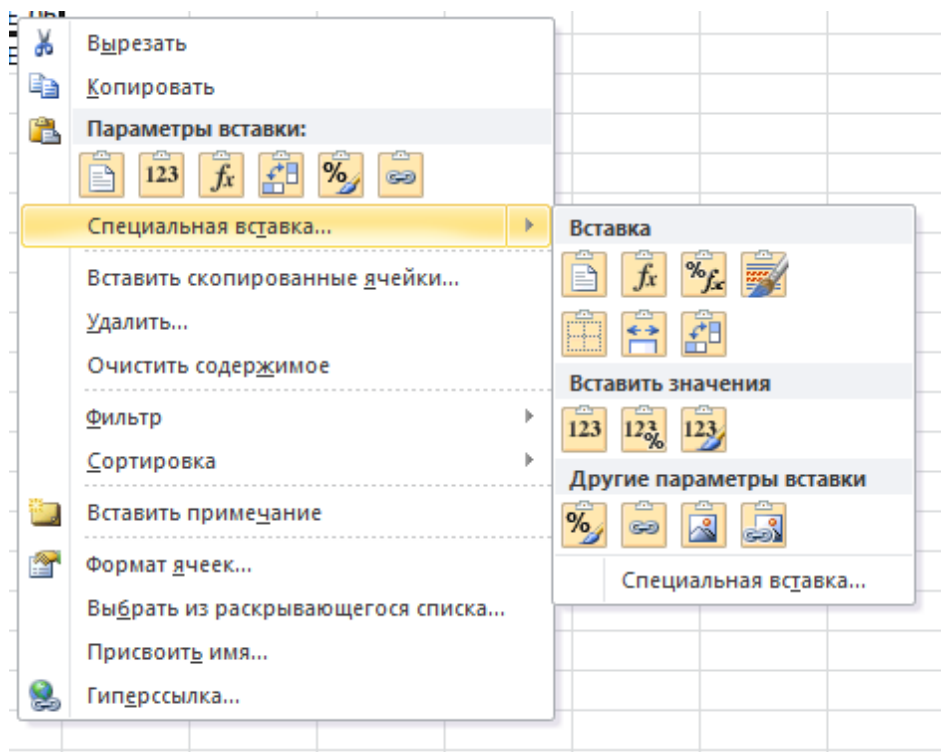


Рис. 2.2 — Специальная вставка

6. После подбора параметра (нажать **ОК**) получит значение корня с заданной ранее степенью точности. Процесс повторяется для всех найденных начальных приближений в диапазонах, определяемых формулой (2.1).

2.2 Решение нелинейных уравнений

2.2.1 Решение нелинейного уравнения методом итерации

Пусть дано уравнение $f(x) = 0$. Для нахождения его корней методом итераций уравнение представляют в виде $x = F(x)$ (очевидно, что это можно сделать не единственным способом) и записывают итерационную схему

$$x_{k+1} = F(x_k), \quad (2.2)$$

с помощью которой строится итерационный процесс уточнения корней, начиная с начального значения x_0 , выбираемого самостоятельно. Достаточное условие сходимости процесса: в окрестности корня $|F'(x)| < 1$.

Если процесс расходится (получающиеся приближения удаляются друг от друга) или сходится очень медленно, то необходимо сменить вид представления $x = F(x)$. В этом может оказать помощь другой итерационный метод решения нелинейных уравнений — *метод Ньютона*. Его итерационная схема имеет вид

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (2.3)$$

Сравнивая формулы (2.2) и (2.3), замечаем, что в качестве функции $F(x_k)$ можно взять правую часть из формулы (2.3). В большинстве случаев метод Ньютона сходится быстрее.

Задание 2.2. Метод итерации

Решить нелинейное уравнение методом итераций.

Порядок действий в Excel может быть следующий.

1. Представить данное уравнение в виде $x = F(x)$, взяв его из табл. 2.2. Задать точность решения $\varepsilon = 0,0001$.

Таблица 2.2

1	$\ln(x) + (x+1)^3 = 0$	11	$x - \sin(x) = 0,25$
2	$x2^x = 1$	12	$\operatorname{tg}(0,58x + 0,1) = x^2$
3	$\sqrt{x+1} = 1/x$	13	$\sqrt{x} - \cos(0,387x) = 0$
4	$x - \cos(x) = 0$	14	$3x - \cos(x) - 1 = 0$
5	$3x + \cos(x) + 1 = 0$	15	$\lg(x) - 7/(2x+6) = 0$
6	$x + \ln(x) = 0,5$	16	$x + \lg(x) = 0,5$
7	$2 - x = \ln(x)$	17	$x^3 - 4\sin(x) = 0$
8	$(x-1)^2 = \exp(x)/2$	18	$\operatorname{ctg}(1,05x) - x^2 = 0$
9	$(2-x)\exp(x) = 0,5$	19	$x\lg(x) - 1,2 = 0$
10	$2,2x - 2^x = 0$	20	$\operatorname{ctg}(x) - x/4 = 0$

2. Создать таблицу с заголовками столбцов **Номер шага**, **Очередное приближение к корню**, **Проверка на точность**.

3. В первую ячейку первой строки таблицы занести значение 0, во вторую — начальное приближение.

4. В следующие строки занести соответственно номер очередного шага, итерационную формулу, вычисляющую правую часть итерационной схемы, и условную формулу, позволяющую помещать в ячейку текст «Стоп» или «Дальше» в зависимости от выполнения заданной точности решения (см. п. 5 алгоритма).

5. Процесс копирования формулы продолжать до получения необходимой точности: разность двух рядом стоящих приближений по модулю должна быть меньше заданного значения ε .

6. После получения решения построить график, иллюстрирующий процесс сходимости: по оси абсцисс отложить номер шага, по оси ординат — очередное приближение к корню.

7. Ответить на вопрос: любое ли начальное приближение можно задавать в вашем варианте? Определить (примерно) диапазон возможных начальных значений, проведя численный эксперимент.

2.2.2 Решение нелинейного уравнения методом бисекции

Если метод итераций сходится не всегда, то метод бисекции (или метод деления отрезка пополам, или метод дихотомии) — безусловно сходящийся метод нахождения корней нелинейного

уравнения $f(x) = 0$, лишь бы был известен отрезок, на котором расположен один корень уравнения.

Пусть непрерывная функция $f(x)$ меняет знак на концах отрезка $[a, b]$, т. е. $f(a) \times f(b) < 0$. Назовем такой отрезок *отрезком локализации корня*: на нем есть, по крайней мере, один корень. Найдем координату середины этого отрезка $c = (a + b)/2$ и рассмотрим два получившихся отрезка $[a, c]$ и $[c, b]$. Если $f(a) \times f(c) < 0$, то корень находится на отрезке $[a, c]$, в противном случае — на отрезке $[c, b]$. Процесс деления пополам все новых и новых отрезков локализации корня продолжаем до тех пор, пока длина отрезка не станет меньше заданной величины точности решения ε .

Задание 2.3. Метод бисекции

Решить нелинейное уравнение методом бисекции.

Для решения уравнения этим методом в Excel достаточно внести в некоторые ячейки, лежащие в одной строке, формулы, осуществляющие следующие операции:

- вычисление значений левой и правой границ отрезков локализации;
- нахождение середины отрезка;
- вычисление произведения значений функций на левой и правой границах отрезка (для контроля правильности алгоритма);
- проверку на точность решения (аналогично предыдущему заданию).

Затем формулы копируются вниз по столбцам до тех пор, пока не будет найден корень с заданной степенью точности, например $\varepsilon = 0,0001$.

Данные для решения взять из табл. 2.2, то есть решить одно и то же уравнение двумя способами.

2.3 Решение систем линейных уравнений

Дана система линейных алгебраических уравнений (СЛАУ) вида

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2; \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n; \end{cases} \quad (2.4)$$

или в матричном виде $A \cdot X = B$, где $A = \{a_{ij}\}$ — матрица коэффициентов при неизвестных; $B = \{b_{ij}\}$ — вектор-столбец правых частей уравнений; $X = \{x_{ij}\}$ — вектор-столбец неизвестных.

Ее решение — вектор столбец $X^* = \{x_{ij}^*\}$ в Excel можно найти разными способами, например используя специальные функции для работы с матрицами (**Формулы | Математические**):

МОБР — вычисление обратной матрицы A^{-1} ;

МОПРЕД — вычисление определителя матрицы D;

МУМНОЖ — нахождение произведения двух матриц.

Задание 2.4. Метод обратной матрицы

Решение имеет вид $X = A^{-1} \cdot B$, где A^{-1} — матрица, обратная по отношению к матрице A.

С помощью функции МОБР находится обратная матрица, а затем с помощью функции МУМНОЖ она перемножается с вектором-столбцом правых частей уравнений.

Можно проверить найденное решение умножением матрицы коэффициентов на вектор-столбец решения. Должен получиться вектор-столбец правых частей.

СЛАУ взять из таблицы 2.3.

Примечание. Если функция в качестве ответа должна выдавать не одно значение, а вектор или матрицу, то перед вводом формулы необходимо *выделить область* на рабочем листе, куда будет выведен результат вычисления, а после задания исходных данных в поле функции выйти *не как обычно*, нажатием клавиши **Enter** или кнопки **ОК**, а нажатием клавиш **Ctrl + Shift + Enter**.

Задание 2.5. Метод Крамера

Если определитель Δ матрицы A , составленной из коэффициентов при неизвестных, отличен от нуля, то решение имеет вид

$$x_j = \Delta_j / \Delta, \quad j = 1, \dots, n, \quad (2.5)$$

где Δ_j — определитель вспомогательной матрицы, полученной из матрицы A путем замены ее j -го столбца вектором-столбцом правых частей уравнений B (дополнительный определитель).

Рекомендуется сформировать на листе три вспомогательные матрицы, поочередно заменяя столбцы матрицы из коэффициентов столбцами правых частей, затем с помощью функции МОПРЕД найти главный определитель Δ и дополнительные определители Δ_j , а затем по формуле (2.5) вычислить корни СЛАУ.

Таблица 2.3 — Системы линейных алгебраических уравнений

Вариант	СЛАУ	Вариант	СЛАУ
1	$2,7x_1 + 3,3x_2 + 1,3x_3 = 2,1;$ $3,5x_1 - 1,7x_2 + 2,8x_3 = 1,7;$ $4,1x_1 + 5,8x_2 - 1,7x_3 = 2,1$	2	$0,34x_1 + 0,71x_2 + 0,63x_3 = 2,08;$ $0,71x_1 - 0,65x_2 - 0,18x_3 = 0,17;$ $1,17x_1 - 2,35x_2 + 0,75x_3 = 1,28$
3	$1,7x_1 + 2,8x_2 + 1,9x_3 = 0,7;$ $2,1x_1 + 3,4x_2 + 1,8x_3 = 1,1;$ $4,2x_1 - 3,3x_2 + 1,3x_3 = 2,1$	4	$3,75x_1 - 0,28x_2 + 0,17x_3 = 0,75;$ $2,11x_1 - 0,11x_2 - 0,12x_3 = 1,11;$ $0,22x_1 - 3,17x_2 + 1,81x_3 = 0,05$
5	$3,1x_1 + 2,8x_2 + 1,9x_3 = 0,2;$ $1,9x_1 + 3,1x_2 + 2,1x_3 = 2,1;$ $7,5x_1 + 3,8x_2 + 4,8x_3 = 5,6$	6	$0,21x_1 - 0,18x_2 + 0,75x_3 = 0,11;$ $0,13x_1 + 0,75x_2 - 0,11x_3 = 2,00;$ $3,01x_1 - 0,33x_2 + 0,11x_3 = 0,13$
7	$9,1x_1 + 5,6x_2 + 7,8x_3 = 9,8;$ $3,8x_1 + 5,1x_2 + 2,8x_3 = 6,7;$ $4,1x_1 + 5,7x_2 + 1,2x_3 = 5,8$	8	$0,13x_1 - 0,14x_2 - 2,00x_3 = 0,15;$ $0,75x_1 + 0,18x_2 - 0,77x_3 = 0,11;$ $0,28x_1 - 0,17x_2 + 0,39x_3 = 0,12$
9	$3,3x_1 + 2,1x_2 + 2,8x_3 = 0,8;$ $4,1x_1 + 3,7x_2 + 4,8x_3 = 5,7;$ $2,7x_1 + 1,8x_2 + 1,1x_3 = 3,3$	10	$3,01x_1 - 0,14x_2 - 0,15x_3 = 1,00;$ $1,11x_1 + 0,13x_2 - 0,75x_3 = 0,13;$ $0,17x_1 - 2,11x_2 + 0,71x_3 = 0,17$
11	$7,6x_1 + 5,8x_2 + 4,7x_3 = 10,1;$ $3,8x_1 + 4,1x_2 + 2,7x_3 = 9,7;$ $2,9x_1 + 2,1x_2 + 3,8x_3 = 7,8$	12	$0,92x_1 - 0,83x_2 + 0,62x_3 = 2,15;$ $0,24x_1 - 0,54x_2 + 0,43x_3 = 0,62;$ $0,73x_1 - 0,81x_2 - 0,67x_3 = 0,88$
13	$3,2x_1 - 2,5x_2 + 3,7x_3 = 6,5;$ $0,5x_1 + 0,34x_2 + 1,7x_3 = -0,2;$ $1,6x_1 + 2,3x_2 - 1,5x_3 = 4,3$	14	$1,24x_1 - 0,87x_2 - 3,17x_3 = 0,46;$ $2,11x_1 - 0,45x_2 + 1,44x_3 = 1,50;$ $0,48x_1 + 1,25x_2 - 0,63x_3 = 0,35$

Окончание табл. 2.3

Вариант	СЛАУ	Вариант	СЛАУ
15	$5,4x_1 - 2,3x_2 + 3,4x_3 = -3;$ $4,2x_1 + 1,7x_2 - 2,3x_3 = 2,7;$ $3,4x_1 + 2,4x_2 + 7,4x_3 = 1,9$	16	$0,64x_1 - 0,83x_2 + 4,2x_3 = 2,23;$ $0,58x_1 - 0,83x_2 + 1,43x_3 = 1,71;$ $0,86x_1 + 0,77x_2 + 0,88x_3 = -0,54$
17	$3,6x_1 + 1,8x_2 - 4,7x_3 = 3,83;$ $2,7x_1 - 3,6x_2 + 1,9x_3 = 0,4;$ $1,5x_1 + 4,5x_2 + 3,3x_3 = -1,6$	18	$0,32x_1 - 0,42x_2 + 0,85x_3 = 1,32;$ $0,63x_1 - 1,43x_2 - 0,58x_3 = -0,44;$ $0,84x_1 - 2,23x_2 - 0,52x_3 = 0,64$
19	$5,6x_1 + 2,7x_2 - 1,7x_3 = 1,9;$ $3,4x_1 - 3,6x_2 - 6,7x_3 = -2,4;$ $0,8x_1 + 1,3x_2 + 3,7x_3 = 1,2$	20	$0,73x_1 + 1,24x_2 - 0,38x_3 = 0,58;$ $1,25x_1 + 0,66x_2 - 0,78x_3 = 0,66;$ $0,75x_1 + 1,22x_2 - 0,83x_3 = 0,92$

2.4 Решение систем нелинейных уравнений

В общем случае система нелинейных уравнений имеет вид:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0; \\ f_2(x_1, x_2, \dots, x_n) = 0; \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0. \end{cases} \quad (2.6)$$

Системы нелинейных уравнений в Excel можно решать с помощью средства анализа **Поиск решения** (**Данные** | группа **Анализ** | **Поиск решения**).

Если на ленте **Данные** отсутствует вкладка **Анализ**, то для ее установки нужно выполнить последовательность действий: **Файл** | **Параметры** | **Надстройки** | **Пакет анализа** | **Перейти** (рис. 2.3) | **ОК**.

Составим новую функцию (назовем ее **целевой функцией**) $F(x_1, x_2, \dots, x_n)$, представляющую собой сумму квадратов левых частей уравнений данной системы (2.6):

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i^2(x_1, x_2, \dots, x_n). \quad (2.7)$$

Очевидно, переменные $x_1, x_2, \dots, x_n$, являющиеся решением системы (2.6), с необходимостью и достаточностью являются также решением уравнения:

$$\sum_{i=1}^n f_i^2(x_1, x_2, \dots, x_n) = 0. \quad (2.8)$$

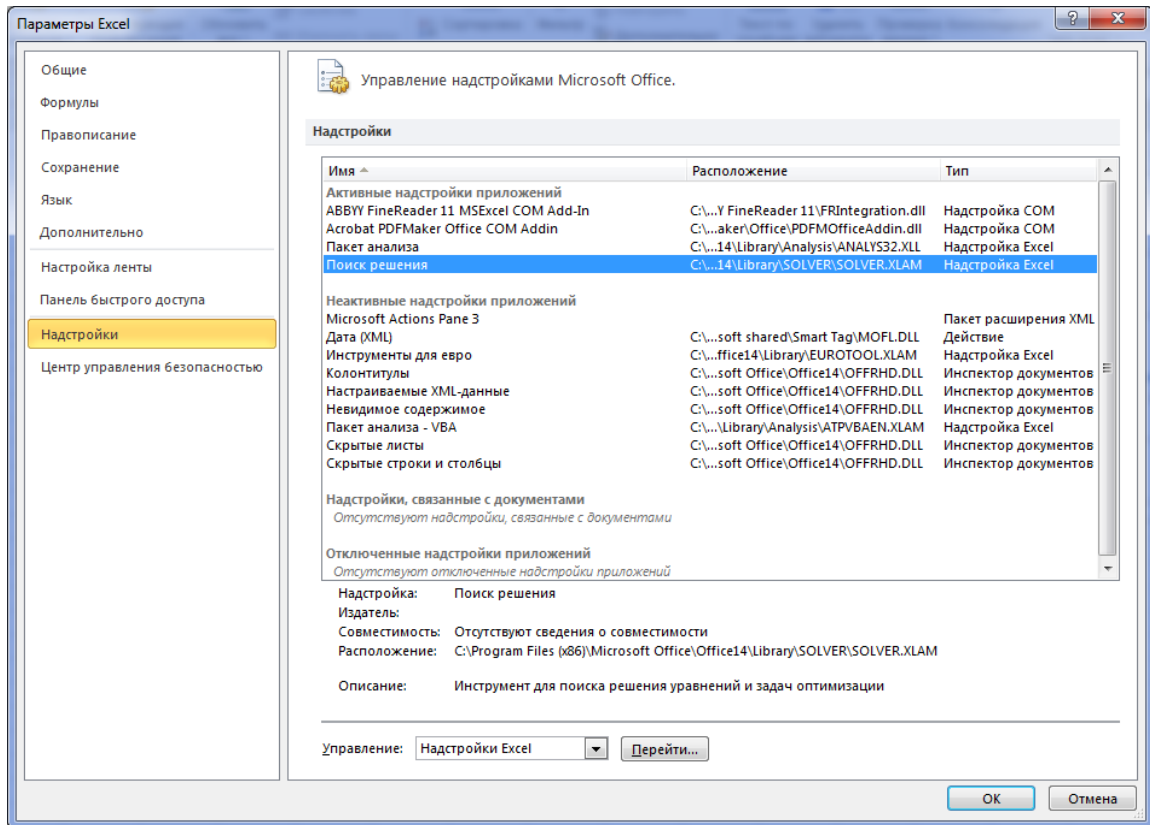


Рис. 2.3 — Установка Поиска решения

Таким образом, найдя с помощью Поиска решения вектор $\{x_1, x_2, \dots, x_n\}$, доставляющий целевой функции нулевое значение, мы также получим решение системы нелинейных уравнений (2.6).

Задание 2.6. Метод целевой функции

Найти все корни системы нелинейных уравнений, взяв данные из таблицы 2.4. Проверить найденное решение.

Построить поверхность, описываемую функцией $F(x, y)$ в окрестности всех найденных корней, пользуясь описанием, приведенным в п.1.7.3.

Путь решения следующий.

На листе Excel отводим ячейки для неизвестных заданной системы уравнений, например с A1 по A5 (если пять переменных), и вводим туда начальные приближения. В ячейку B2 вводим формулу, вычисляющую функцию (2.7).

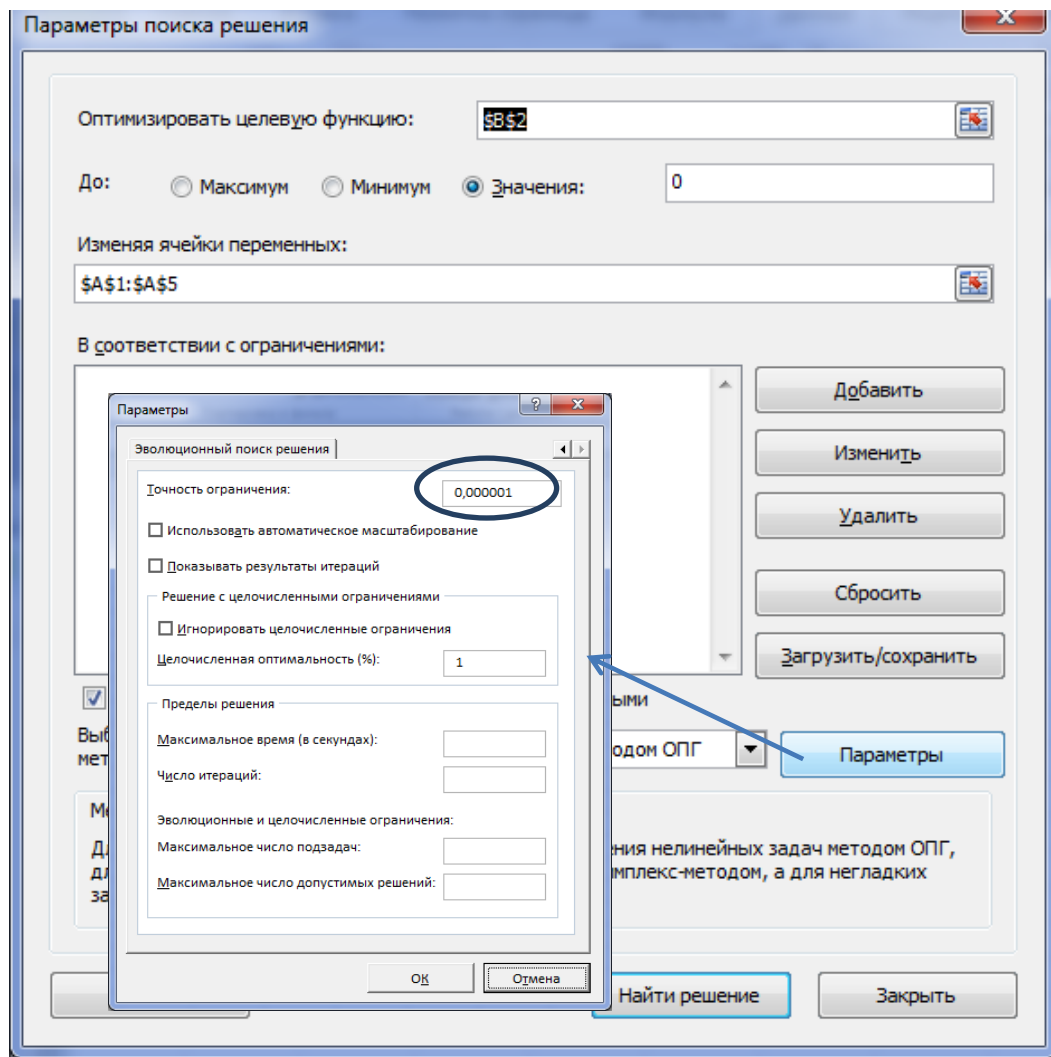


Рис. 2.4 — Диалоговое окно Параметры поиска решения при решении

Таблица 2.4

Вариант	СНУ	Вариант	СНУ
1	$\begin{cases} \operatorname{tg}(xy + 0,4) = x^2 \\ 0,6x^2 + 2y^2 = 1 \end{cases}$	2	$\begin{cases} 2x^2 + 5y^2 = 3 \\ 5x + 9y = 3 \end{cases}$
3	$\begin{cases} \sin(x + y) - 1,6x = 0 \\ x^2 + y^2 = 1 \end{cases}$	4	$\begin{cases} 3x^2 + 4y^2 = 4 \\ 3x + 4y = 2 \end{cases}$
5	$\begin{cases} \sin(x + 1) - y = 1,2 \\ 2x + \cos(y) = 2 \end{cases}$	6	$\begin{cases} 5x^2 + 2y^2 = 4 \\ 2x + 7y = 1 \end{cases}$
7	$\begin{cases} \cos(x + 1) + y = 0,5 \\ x - \cos(y) = 3 \end{cases}$	8	$\begin{cases} 4x^2 + 5y^2 = 3 \\ 5x + 3y = 1 \end{cases}$
9	$\begin{cases} \cos(x) + y = 1,5 \\ 2x - \sin(y - 0,5) = 1 \end{cases}$	10	$\begin{cases} 5x^2 + 6y^2 = 3 \\ 7x + 3y = 1 \end{cases}$

Окончание табл. 2.4

Вариант	СНУ	Вариант	СНУ
11	$\begin{cases} \operatorname{tg}(xy + 0,1) = x^2 \\ x^2 + y^2 = 1 \end{cases}$	12	$\begin{cases} 3x^2 + 5y^2 = 3 \\ 5x + 2y = 2 \end{cases}$
13	$\begin{cases} \sin(x + y) - 1,2x = 0,2 \\ x^2 + y^2 = 1 \end{cases}$	14	$\begin{cases} 7x^2 + 6y^2 = 3 \\ 5x + 3y = 2 \end{cases}$
15	$\begin{cases} \operatorname{tg}(xy) = x^2 \\ 0,8x^2 + 2y^2 = 1 \end{cases}$	16	$\begin{cases} 5x^2 + 6y^2 = 3 \\ 3x + 2y = 2 \end{cases}$
17	$\begin{cases} \sin(y + 1) - x = 1 \\ 2y + \cos(x) = 2 \end{cases}$	18	$\begin{cases} 3x^2 + 2y^2 = 2 \\ 2x + 7y = 3 \end{cases}$
19	$\begin{cases} \exp(x + y) - x^2 + y = 1 \\ (x + 0,5)^2 + y^2 = 2 \end{cases}$	20	$\begin{cases} 5x^2 + y^2 = 3 \\ 3x + 5y = 2 \end{cases}$

Примечание. При неудачном выборе вектора начального приближения решение может быть не найдено. Поэтому необходим предварительный анализ системы уравнений с целью определения лучшего (более близкого к корню) начального приближения.

Задавая разные начальные приближения, можно получить разные решения системы.

Раздел 3. Обработка данных

Цель раздела: познакомиться со способами сортировки и фильтрации данных в Excel, научиться создавать сводные таблицы, изучить способы консолидации данных.

Область электронной таблицы можно рассматривать как *базу данных*. При этом столбцы называются *полями*, а строки — *записями*. Столбцам присваиваются имена, которые будут использоваться как *имена полей записей*.

При формировании структуры базы данных необходимо учитывать следующие ограничения:

- первый ряд базы данных должен содержать неповторяющиеся имена полей и располагаться в одной строке;
- для имен полей следует использовать шрифт, тип данных, формат, рамку, отличные от тех, которые используются для данных в записях;
- таблицу следует отделить от других данных рабочего листа пустым столбцом и пустой строкой;
- информация по полям должна быть однородной, т. е. только цифры или только текст.

Работа с любой базой данных заключается в поиске информации по определенным критериям, перегруппировке и обработке информации.

Примечание. Прежде чем начинать работу с базой данных, необходимо «встать в таблицу», т. е. щелкнуть мышью по какой-либо ее ячейке (*но не выделять отдельные столбцы!*).

3.1 Сортировка и фильтрация

Поместить на листе Excel табл. 3.1 и, используя ее данные, создать на этом же листе новую (рабочую) таблицу с относительными величинами, разделив все параметры на соответствующий параметр Земли. Не забудьте изменить названия полей в новой таблице!

Для проведения дальнейших операций с полученной таблицей необходимо, чтобы в ее ячейках находились числовые значения, а не формулы, их вычисляющие¹. Замена производится с помощью опции **Специальная вставка** (см. рис. 2.2).

¹ Можно использовать и формулы, но тогда адреса всех ячеек, встречающихся в них, должны иметь абсолютную адресацию.

Таблица 3.1

Планета	Период обращения, земной год	Расстояние от Солнца, млн км	Диаметр, тыс. км	Масса, Тт	Количество спутников
Меркурий	0,241	58	4,9	$0,32 \cdot 10^9$	0
Венера	0,615	108	12,1	$4,86 \cdot 10^9$	0
Земля	1	150	12,8	$6,0 \cdot 10^9$	1
Марс	1,881	288	6,8	$6,1 \cdot 10^8$	2
Юпитер	11,86	778	142,6	$19,07 \cdot 10^{11}$	16
Сатурн	29,46	1426	120,2	$57,09 \cdot 10^{10}$	17
Уран	84,01	2869	49,0	$87,24 \cdot 10^9$	14
Нептун	164,8	4496	50,2	$10,34 \cdot 10^{10}$	2
Плутон	247,7	5900	2,8	$0,1 \cdot 10^9$	1

Задание 3.1. Сортировка

В полученной таблице, используя вкладку **Сортировка и фильтр** ленты **Главная** или ленты **Данные** (рис. 3.1), выполнить следующие операции:

- 1) отсортировать данные в порядке убывания количества спутников;
- 2) отсортировать данные в алфавитном порядке названий планет;
- 3) отсортировать данные в порядке возрастания массы.

Разместить все результаты сортировок на одном листе рабочей книги.

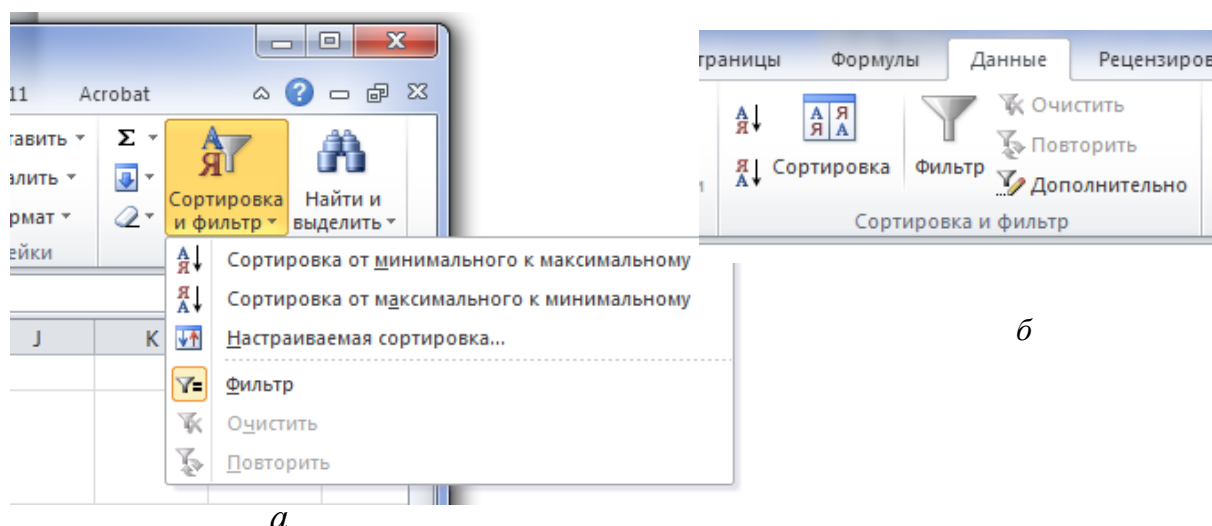


Рис. 3.1 — Вкладки **Сортировка и фильтр** ленты **Главная** (а) и ленты **Данные** (б)

Задание 3.2. Фильтрация

Фильтрация позволяет быстро сделать выборку из таблицы, используя различные критерии выборки.

В Excel эту операцию можно выполнять различными способами.

Способ 1. Фильтр

Вызов **Фильтра** осуществляется через вкладку **Сортировка и фильтр** ленты **Главная** или ленты **Данные** кнопкой с изображением воронки (рис. 3.1). При этом в заголовках таблицы появляются кнопки с треугольниками. При нажатии на них появляется меню с условиями отбора автофильтра (рис. 3.2) для выбранного столбца.

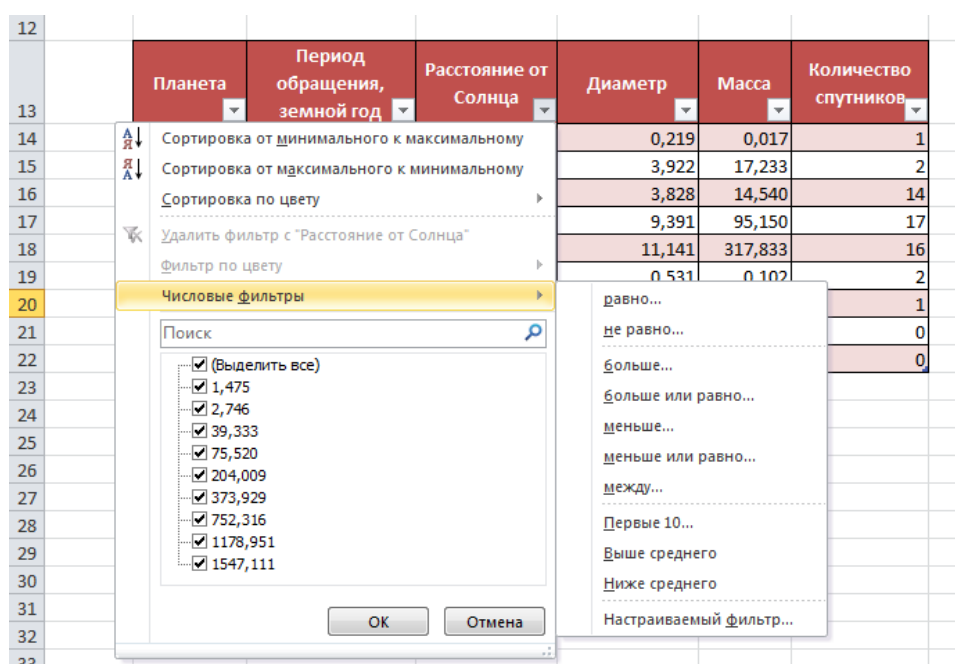


Рис. 3.2 — Условия фильтрации

Щелчок по условию вызывает появление пользовательского автофильтра (рис. 3.3), где задаются критериальные значения.

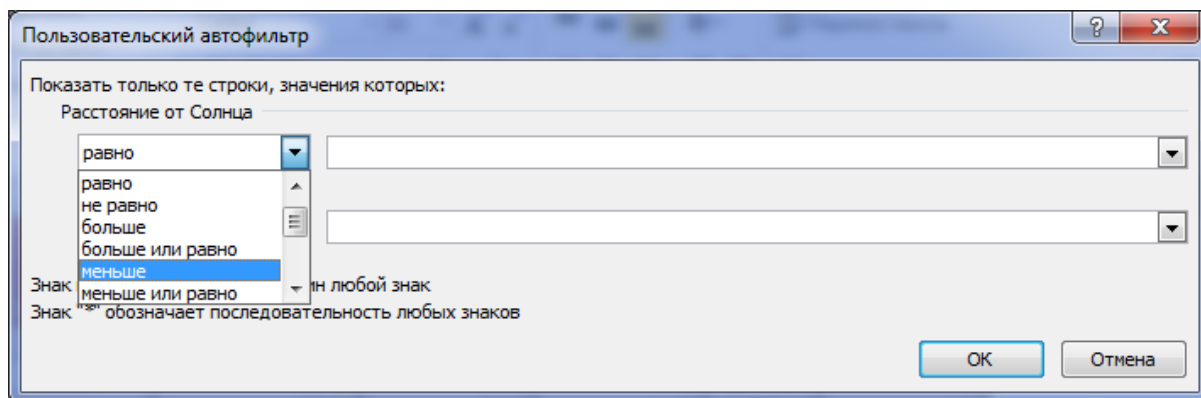


Рис. 3.3 — Диалоговое окно Пользовательский автофильтр

В левом поле этого окна под надписью **Показать только те строки, значения которых**, выбирается необходимый оператор сравнения (равно, меньше, больше и т. п.), а в правом поле задается значение сравнения. Нижние поля используются для второго условия отбора, соединяющегося с первым знаками логических операций **И** или **ИЛИ**.

После работы автофильтра в таблице оказываются только те записи, которые удовлетворяют критериям фильтрации.

Параметр **Первые 10...** определяет строки с максимальным или минимальным значением ячеек текущего столбца. Эта строка открывает диалоговое окно **Наложение условия по списку** (рис. 3.4).

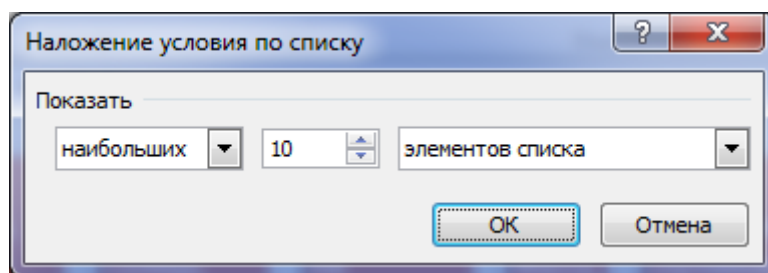


Рис. 3.4 — Диалоговое окно Наложение условия по списку

Задание 3.3. Автофильтр

С помощью **Автофильтра** найти:

- 1) планеты, имеющие диаметр менее 4-х диаметров Земли и период обращения более 80 земных лет;

2) планеты, находящиеся от Солнца на расстоянии не менее 0,5 расстояния от Земли, имеющие массу от одной до 100 масс Земли и не более 2-х спутников;

3) три планеты, имеющие самый большой диаметр.

Разместить результаты фильтрации на различных листах рабочей книги.

Способ 2. Расширенный фильтр

В отличие от **Автофильтра** **Расширенный фильтр** (рис. 3.5) позволяет проводить фильтрацию не только по месту расположения таблицы, но и выносить результат в другое место листа.

Чтобы выполнить фильтрацию, необходимо воспользоваться командами ленты **Данные** | группа **Сортировка и фильтр** | **Дополнительно**. В открывшемся диалоговом окне указать адреса трех диапазонов, которые *предварительно должны быть определены*.

Первый диапазон — это исходная таблица (область базы данных). Пусть, например, она располагается в ячейках B2:G11. Эта область указывается в поле **Исходный диапазон** (рис. 3.5).

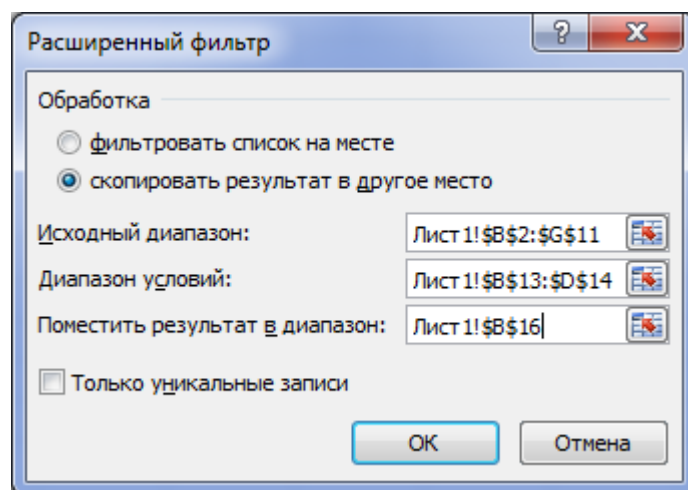


Рис. 3.5 — Диалоговое окно
Расширенный фильтр

Второй диапазон — это область, где задаются критерии фильтрации. Его формируют из заголовков столбцов таблицы (т. е. имен полей базы данных), по которым ведется отбор,

и строк, где в соответствующих ячейках записываются условия фильтрации. Эта область указывается в поле **Диапазон условий**.

Если все условия отбора разместить в одной строке сразу под заголовками, то этим самым реализуется их взаимодействие по схеме «И» (требуется одновременное их выполнение):

Заголовок 1	Заголовок 2	Заголовок 3
≥ 5	< 110	$= 2$

Если по одному полю необходимо задать два условия отбора («двойное неравенство»), то они располагаются рядом, а в следующей строке записываются ограничение снизу и ограничение сверху:

Заголовок 1	Заголовок 1
≥ 5	≤ 10

Для соединения условий отбора по схеме «ИЛИ» необходимо каждое из них разместить в отдельной строке:

Заголовок 1	Заголовок 2	Заголовок 3
≥ 5		
	< 110	
		$= 2$

Для того чтобы найти записи, отвечающие одному из двух наборов условий, каждый из которых содержит условия более чем для одного столбца, необходимо ввести эти условия отбора в отдельные строки:

Заголовок 1	Заголовок 2
$= 5$	> 110
$= 8$	≤ 1000

Третий диапазон нужен, если результат фильтрации собираются расположить не на месте исходного диапазона, а в другом месте. Для этого достаточно указать в поле **Поместить результат в диапазон** адрес только одной ячейки. Она станет *левой верхней ячейкой диапазона результатов*.

Все три диапазона на листе должны быть отделены друг от друга хотя бы одной пустой строкой или столбцом.

Опция **Только уникальные записи** позволяет исключить из таблицы все повторяющиеся строки, оставив только уникальные.

Чтобы восстановить таблицу после работы **Автофильтра** или **Расширенного фильтра**, следует снова щелкнуть по значку воронки.

Задание 3.4. Расширенный фильтр

С помощью **Расширенного фильтра** найти:

- 1) планеты с периодом обращения от 10 до 100 земных лет и количеством спутников не более 15;
- 2) планеты, у которых либо диаметр не менее 4-х земных, либо масса более 100 земных масс;
- 3) среди планет без спутников тех, которые находятся от Солнца на расстоянии менее половины земного, а среди планет с не менее чем 14-ю спутниками тех, которые находятся от Солнца не ближе чем 10 земных расстояний.

Результаты поместить на одном листе в последовательности: исходная таблица, условия, результат, условия, результат и т. д.

3.2 Сводные таблицы

Сводная таблица — это еще один инструмент Excel для обработки больших списков с данными, позволяющий подводить итоги, выполнять сортировку и фильтрацию списков.

Построение сводной таблицы (**Вставка | группа Таблицы | Сводная таблица**) осуществляется в несколько шагов. Сначала указывается источник данных и указывается, где поместить сводную таблицу (рис. 3.6). Если было указано **На новый лист**, то при нажатии **ОК** открывается новый лист, если **На существующий лист**, то для указания места размещения достаточно задать адрес левого верхнего угла сводной таблицы.

При этом в правой части листа появляются элементы управления сводной таблицей (рис. 3.7).

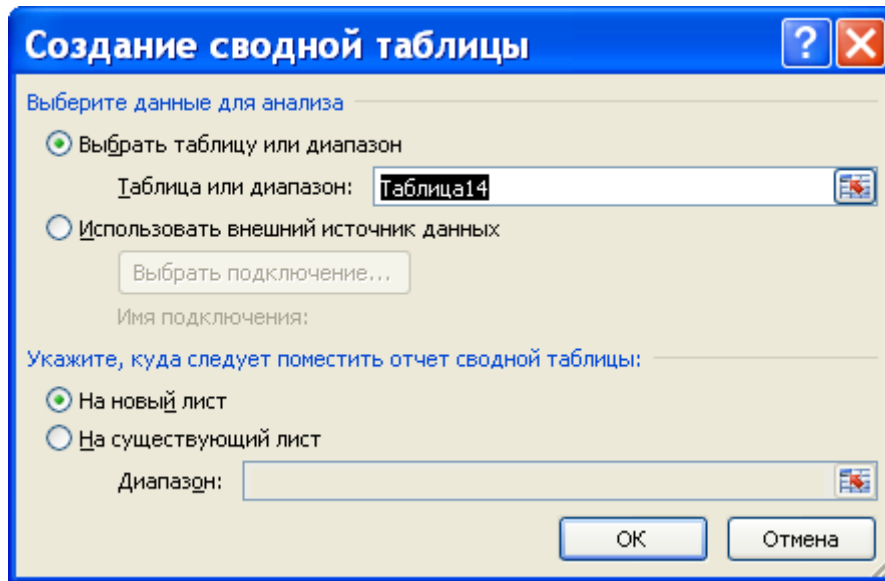


Рис. 3.6 — Начало создания сводной таблицы

В окне **Список полей...** (рис. 3.7, *а*) отмечаем те поля, которые должны быть отражены в сводной таблице. В окно **Фильтр** перетаскиваем из предыдущего окна те поля, по которым должна проводиться фильтрация сводной таблицы. Названия строк и столбцов сводной таблицы задаются в соответствующих окнах (если значения из окна **Названия столбцов** перенести в **Название строк**, получим вертикальное расположение сводной таблицы).

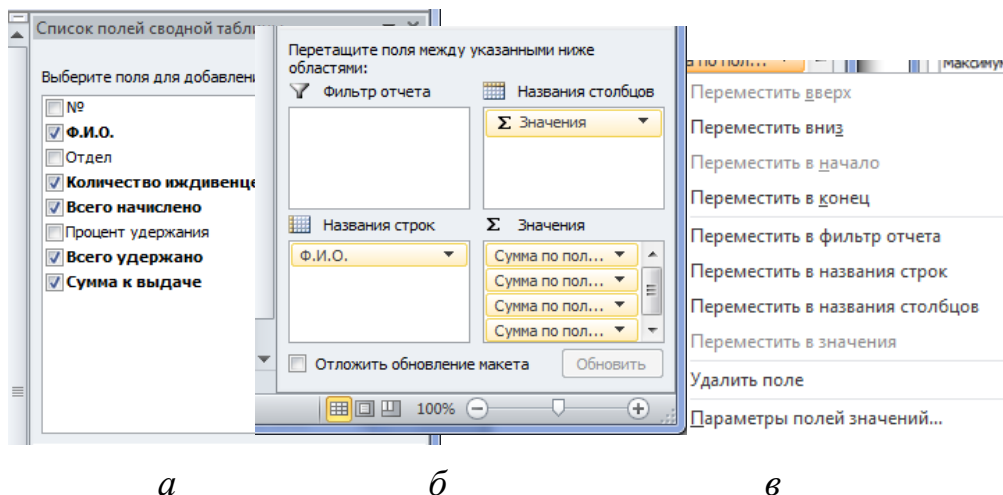


Рис. 3.7 — Параметры сводной таблицы:
а — выбор полей; *б* — итоги и фильтры;
в — меню поля в окне **Значения**

В последнем окне **Значения** задают те поля, по которым нужно провести нужную операцию сведения данных. По умолчанию это суммирование. Изменить операцию можно щелчком мыши по соответствующему полю. В появившемся меню (рис. 3.7, в) выбрать **Параметры полей значений** (рис. 3.8) и задать нужную операцию.

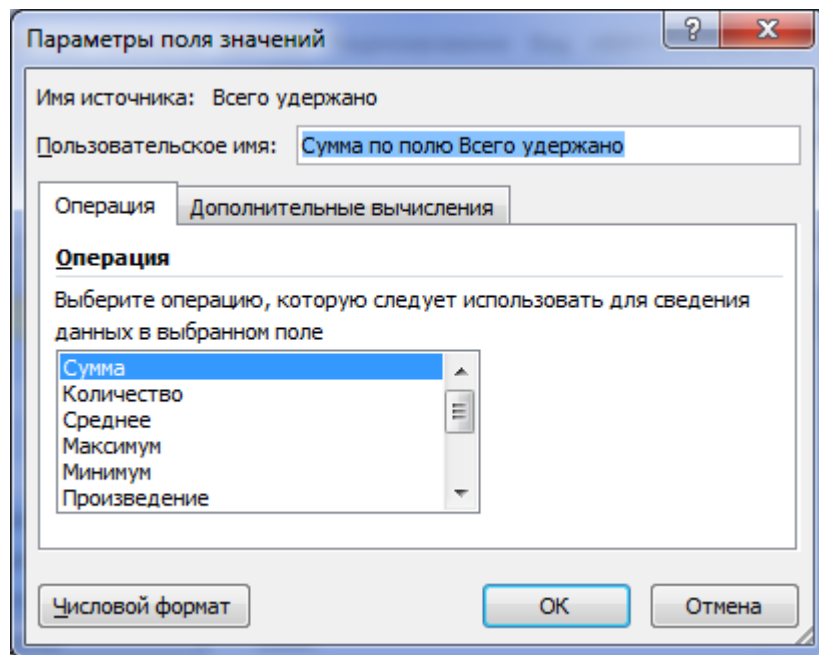


Рис. 3.8 — Параметры поля значений

Задание 3.5. Создание сводной таблицы

Заполнить табл. 3.2, занеся недостающие данные и рассчитав процент удержания с начисленной суммы по следующему правилу: при количестве иждивенцев более трех — 0 %, при трех — 5 %, при двух — 10 %, при одном — 12 %, если нет — 14 %. Расчет оформить с помощью вложенных функций ЕСЛИ.

На основе построенной таблицы создать сводную таблицу, найдя итоги (суммы) по полям **Всего начислено**, **Всего удержано**, **Сумма к выдаче** по каждой фамилии, расположив фамилии в алфавитном порядке, и провести фильтрацию отдельно по категории **Количество иждивенцев** и по категории **Отдел**. Поместить сводную таблицу на отдельном листе. Постарайтесь создать

компактную и наглядную сводную таблицу, избежав излишнего текста и пустых ячеек.

На основе сводной таблицы сформировать отчет по лицам, работающим в третьем отделе и имеющим по одному иждивенцу.

Таблица 3.2

Ф.И.О.	Отдел	Количество иждивенцев	Всего начислено, руб.	Процент удержания	Всего удержано, руб.	Сумма к выдаче, руб.
1. Петухова К.И.	1	1	1260			
2. Безенчук П.Ф.	3	2	1100			
3. Воробьянинов И.М.	2	3	800			
4. Востриков Ф.О.	2	2	750			
5. Коробейников В.А.	3	1	715			
6. Грицацук В.С.	1	3	630			
7. Гаврилин З.С.	1	4	620			
8. Треухов Т.И.	3	1	560			
9. Изнуренков А.В.	1	0	420			
10. Щукина Э.Е.	3	1	250			

3.3 Промежуточные итоги

Промежуточные итоги по различным группам можно подводить и без создания сводных таблиц, используя команду Excel **Промежуточный итог**.

Эту команду можно применить к таблице, если она не подвергалась форматированию с помощью команды **Форматировать как таблицу** из вкладки **Стили** ленты **Главная**. Необходимо сначала вернуть ее в обычный диапазон данных. Для этого, выделив таблицу, нужно выполнить команду: **Работа с таблицами | Конструктор | Преобразовать в диапазон**.

Сначала таблица сортируется по столбцу, по которому необходимо подвести промежуточные итоги. Затем выбирается команда **Данные | группа Структура | Промежуточный итог**. В появившемся диалоговом окне (рис. 3.9) в поле **При каждом изменении в:** задается классификационная категория; в поле **Операция:** задается функция, необходимая для подведения итогов; в поле **Добавить итоги по:** выделяются названия столбцов, где находятся итожимые данные.

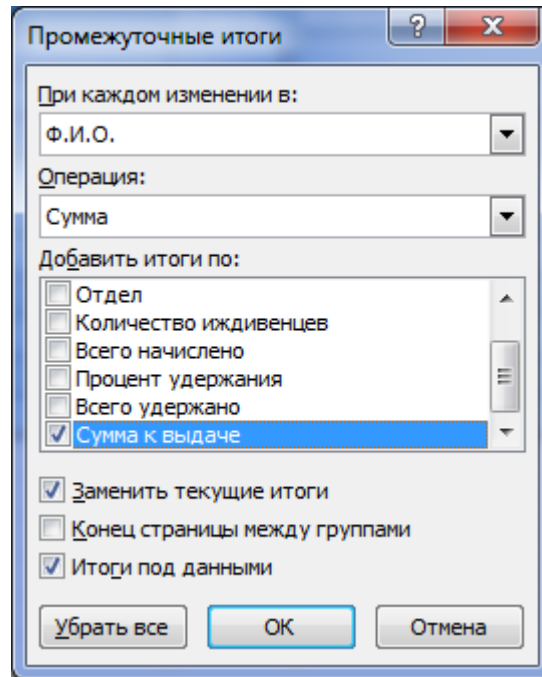


Рис. 3.9 — Диалоговое окно
Промежуточные итоги

Для отображения только промежуточных и общих итогов используйте обозначения уровней структуры рядом с номерами строк. Кнопки + и – позволяют отобразить и скрыть строки подробных данных для отдельных итогов.

Задание 3.6. Получение промежуточных итогов

На таблице 3.2 получить промежуточные итоги во всех отделах по позициям **Количество иждивенцев**, **Всего начислено**, **Всего удержано** и **Сумма к выдаче**. Изучить и описать структуру полученной таблицы.

3.4 Связь таблиц (консолидация данных)

Excel позволяет связывать ячейки таблиц, находящихся на различных рабочих листах одного файла, в разных файлах, на различных логических дисках таким образом, чтобы в результате изменения значения ячейки в одной таблице изменялось бы содержание другой.

Связь табличных данных можно осуществлять различными способами.

Способ 1. Формула связи. Связь между листами можно задать путем использования в ячейке какого-либо листа формулы связи со ссылкой на ячейку другого листа, например:

=НазвЛиста!АдрЯч.

Связь между файлами можно задать путем использования в ячейке листа какого-либо файла Excel формулы связи со ссылкой на ячейку в другом файле Excel, например:

='[ИмяФайла.xls]НазвЛиста'!АдрЯч.

Путь заключен в одинарные кавычки, имя файла заключено в квадратные скобки. Если файлы находятся в одном каталоге, то формула немного упрощается:

=[ИмяФайла]НазвЛиста!АдрЯч.

Если файлы Excel находятся на разных логических дисках, то формула связи выглядит следующим образом:

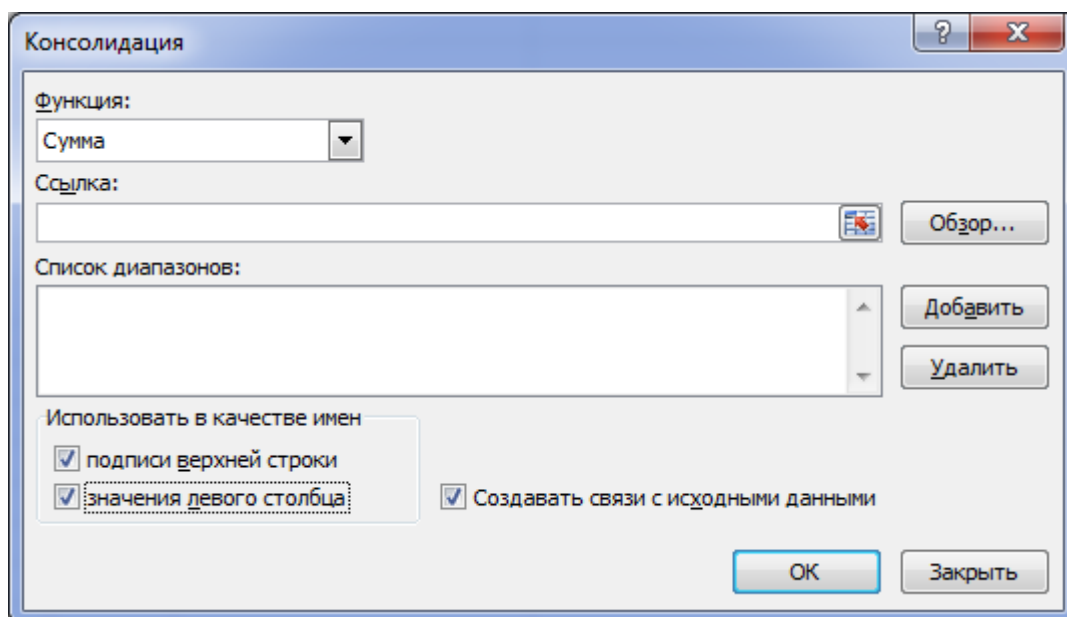
= 'А:\Папка\[ИмяФайла.xls] НазвЛиста'!АдрЯч.

Используя формулы связи, можно в ячейках итоговой таблицы совершать любые операции над ячейками исходных таблиц. Для упрощения набора таких формул рекомендуется задавать адрес нужной ячейки не набором его с клавиатуры, а щелчком указателя мыши в соответствующем месте исходной таблицы, выбрав нужный путь **Файл | Открыть** и т. д.

Способ 2. Консолидация данных. Консолидация позволяет объединять данные из областей-источников и выводить их в области назначения. При этом могут использоваться различные функции: суммирования, расчета среднего арифметического, подсчетов минимальных и максимальных значений и т. д.

Для проведения консолидации необходимо выбрать **Данные | группа Работа с данными | Консолидация** и в поле **Ссылка** (рис. 3.10) определить по порядку области-источники (массивы ячеек), данные из которых нужно консолидировать, добавляя их в поле **Список диапазонов**. Не забудьте поставить флажок напротив опции **Создавать связи с исходными данными!**

Для переноса текстовой информации (имена строк и столбцов) используйте опцию **Использовать в качестве имен**.

Рис. 3.10 — Диалоговое окно **Консолидация**

Задание 3.7. Формула связи

Занести табл. 3.3 на три листа (не забудьте о возможности, описанной в п. 1.4!). Рассчитать выручку и изменить название листов на **Январь**, **Февраль**, **Март**. Подкорректировать заголовки таблиц и изменить данные второго и третьего столбцов (по вашему усмотрению).

Таблица 3.3

Наименование продукции	Исходные данные за январь		
	Цена в USD	Продано	Выручка
Телевизоры	350	20	
Ноутбуки	520	35	
Планшетные ПК	750	65	
Цифровые камеры	250	30	
Электронные книги	150	15	
Смартфоны	400	27	
Медиаплееры	170	11	
Blu-ray плееры	110	14	
Итого:			

На четвертом листе (**Квартал. Способ 1**) создать таблицу итоговых показателей (**Продано** и **Выручка за квартал**) по всему ассортименту продукции за квартал и заполнить эту таблицу,

суммируя данные, находящиеся в соответствующих ячейках листов показателей за январь-март.

На пятом листе (**Квартал. Способ 2**) получить аналогичную таблицу, но с помощью консолидации.

Изучить структуру полученной консолидированной таблицы. Она содержит скрытые строки, в которых находится информация из других листов. Скрытые строки можно открывать и снова скрывать, нажимая соответственно кнопки «+» и «-». Проверьте, изменяются ли значения консолидированной таблицы при изменениях в исходных таблицах.

Сравнить оба способа.

Задание 3.8. Консолидация

Выбрав данные из табл. 3.4, разнести их *поквартально* по разным листам одного файла. Определить среднюю цену производителей по каждой позиции в каждом квартале (в тыс. руб.) и соотношение цен на отдельные виды энергоресурсов с ценой на нефть (в процентах). *В другом файле* с помощью консолидации сформировать таблицу со среднегодовыми данными по ценам и соотношению цен.

Таблица 3.4

Период (1997 г.)	Цены, тыс. руб. за тонну						
	Нефть	Бензин автомобильный	Топливо дизельное	Мазут топочный	Газ естественный*	Уголь	
						энергетический	для коксования
январь	370	933	943	443	37,2	144	133
февраль	373	886	944	433	39,0	143	130
март	376	907	947	434	39,0	146	138
апрель	371	932	970	440	39,0	141	129
май	372	949	980	437	39,3	141	136
июнь	375	968	984	436	39,3	141	129
июль	375	968	981	441	39,5	139	127
август	375	978	981	440	39,6	137	128
сентябрь	376	1011	1013	440	39,4	138	124
октябрь	384	1007	1012	440	40,2	140	125
ноябрь	379	1001	1011	445	40,3	148	122
декабрь	353	1054	985	438	44,6	146	123

* Цена за 1000 м³.

3.5 Форма данных

Форма данных предоставляет удобный способ ввода или отображения одной строки данных из диапазона или таблицы, не требующий применения горизонтальной прокрутки.

Кнопки **Форма** нет на ленте, но в Excel 2010 кнопку **Форма** можно добавить на панель быстрого доступа. Для этого необходимо:

- 1) щелкнуть стрелку, расположенную рядом с панелью быстрого доступа (левый верхний угол листа), и выбрать пункт **Другие команды**;
- 2) в поле **Выбрать команды** из выбрать пункт **Все команды**;
- 3) в списке выбрать кнопку **Форма** и нажать **Добавить**.

Форма данных обычно используется для перемещения между столбцами, которых слишком много для одновременного отображения на экране. Форму данных можно применять, когда достаточно простой формы с текстовыми полями, в качестве подписей которых используются заголовки столбцов, и когда не требуются сложные или пользовательские возможности, например списки и счетчики.

Рис. 3.11 — Форма, созданная на основе таблицы (рис. 1.1).
Строка заголовков преобразована в одну строку листа Excel

Excel автоматически создает встроенные формы данных для диапазона или таблицы. Такая форма представляет собой диалоговое окно, в котором все заголовки столбцов отображаются

в виде подписей. Каждой подписи соответствует текстовое поле, в которое можно вводить данные для столбца (максимальное количество столбцов — 32). В форме данных можно вводить новые строки, искать строки с помощью навигации, а также (в зависимости от содержимого ячеек) обновлять строки или удалять их. Если ячейка содержит формулу, то в форме данных отображается только результат ее вычисления. Саму формулу невозможно изменить с помощью формы данных.

Задание 3.9. Создание формы данных

Создать форму данных для таблицы 3.1.

ЛАБОРАТОРНАЯ РАБОТА № 2. ОСНОВЫ VBA

Раздел 4. Создание макросов с помощью макрорекодера

Цель раздела: используя сервисное средство MacroRecorder среды программирования Visual Basic for Application (VBA) в Excel, предназначенное для автоматического создания макросов, записать простой макрос и обеспечить его вызов с помощью объекта управления **Кнопка**.

Microsoft Excel — мощный инструмент, используемый для управления данными, их анализа и представления. Но иногда, несмотря на богатый набор возможностей стандартного интерфейса пользователя, может понадобиться найти несложный способ выполнять повседневные повторяющиеся задачи или какие-то задачи, которые не удастся решить с помощью интерфейса пользователя. Для этих целей в приложения MS Office интегрирован Visual Basic для приложений (VBA) — язык и среда программирования, позволяющие расширять возможности этих приложений. VBA работает, выполняя макросы (пошаговые процедуры), написанные на языке Visual Basic. Из всех приложений Office чаще всего используемой платформой для применения макросов является именно Excel

Для работы с VBA необходимо вывести на ленту вкладку **Разработчик** (по умолчанию она отсутствует), выполнив следующие действия (в Excel 2010): **Файл** | **Параметры** | **Настройка ленты** | **Настройка ленты** | **Основные вкладки** | флажок **Разработчик** | **ОК**.

На ленте Excel появится вкладка **Разработчик** (рис. 4.1).

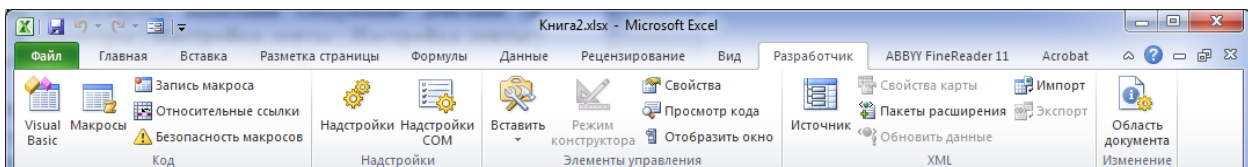


Рис. 4.1 — Вкладка **Разработчик**

Кнопка **Безопасность макросов**, находящаяся в группе **Код**, открывает окно, в котором можно определить, какие макросы выполняются и при каких условиях. Рекомендуется установить

опцию **Отключить все макросы с уведомлением**. Тогда при открытии книги, содержащей макрос, между лентой и листом появляется строка **Предупреждение системы безопасности: запуск макросов отключен**. Чтобы включить макрос, нужно нажать кнопку **Включить содержимое**. Также в качестве мер безопасности не сохраняйте макрос в формате файлов Excel, используемом по умолчанию (xlsx-файлы); вместо этого макрос должен быть сохранен в файле со специальным расширением xlsxm-файл.

Кнопка **Запись макроса** включает макрорекодер (автоматическую запись макроса).

Кстати, на вкладке **Вид** имеется кнопка **Макросы**, также содержащая **Запись макроса** (рис. 4.2), т. е. макрорекодер можно запускать и без установки вкладки **Разработчик**.

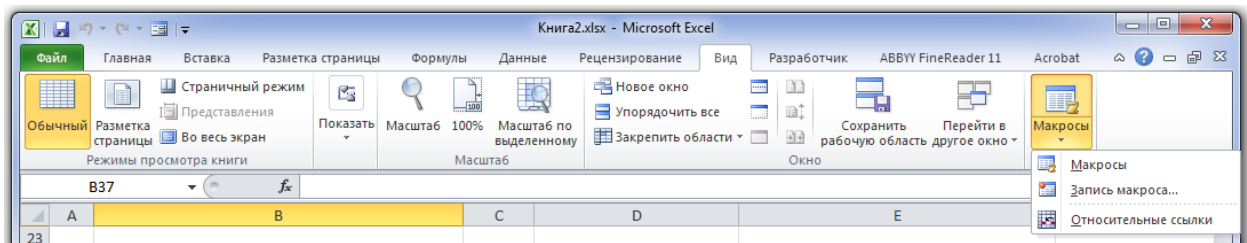


Рис. 4.2 — Вкладка Вид | Макросы

При выборе этой команды открывается окно (рис. 4.3), в котором вводится имя будущего макроса, сочетание клавиш, с помощью которых его можно будет запускать (не обязательно), функциональное назначение макроса.

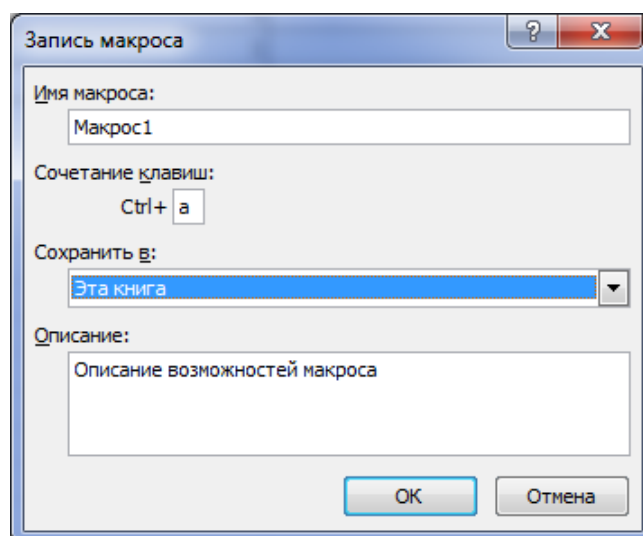


Рис. 4.3 — Запуск Макрорекодера

После нажатия на **ОК** все действия пользователя на листах Excel рабочего файла («книжки») будут записываться транслятором Макрорекодер на языке Visual Basic — от момента запуска до окончания записи макроса.

Автоматическая запись макроса позволяет, даже не зная самого языка Visual Basic, создавать на нем простейшие макросы для автоматизации рутинных действий по обработке данных и по выполнению группы команд.

Остановка записи происходит при вызове команды **Остановить запись** (рис. 4.4).

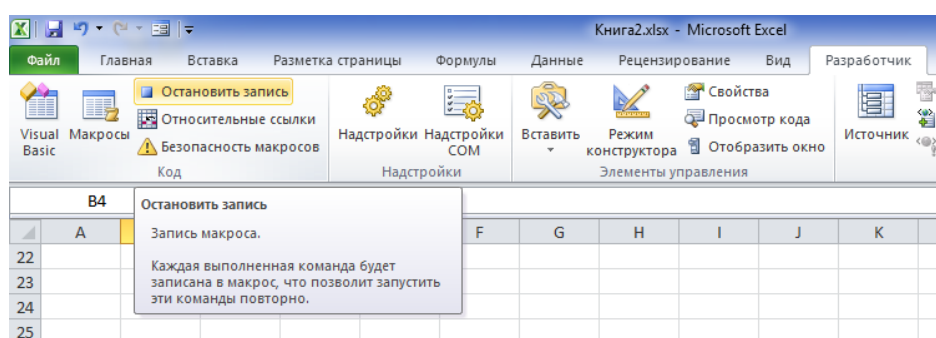


Рис. 4.4 — Остановка Макрорекодера

Чтобы посмотреть на код макроса, нужно нажать на кнопку **Макросы** (рис. 4.4). Появится диалоговое окно **Макрос** (рис. 4.5). Отметим, что этот же результат достигается нажатием сочетания клавиш **Alt+F8**.

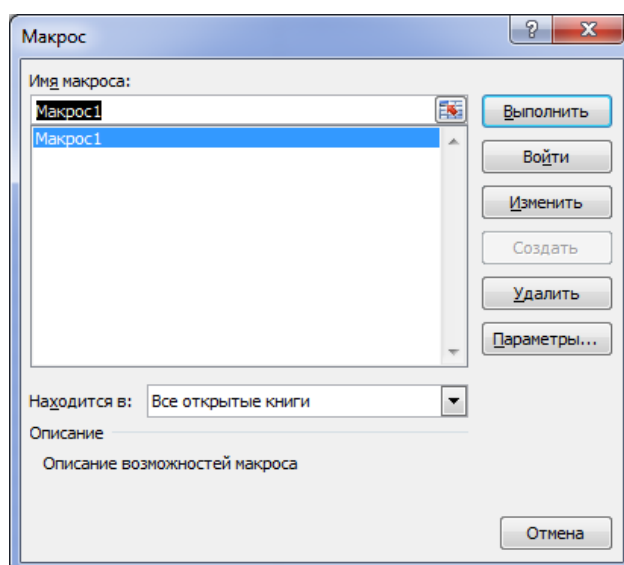


Рис. 4.5 — Диалоговое окно «Макрос»

Правое меню обеспечивает запуск выбранного макроса на выполнение (**Выполнить**), переход к коду макроса (**Выполнить** или **Изменить** — при этом открывается диалоговое окно Microsoft Visual Basic for Application (рис. 4.6)), удаление макроса (**Удалить**), вызов параметров макроса (то же окно, что и на рис. 4.3).

Можно упростить запуск макроса, создав на листе кнопку и назначив ей макрос. В Excel существует несколько вариантов кнопок.

- 1) кнопка-автофигура;
- 2) кнопка-картинка;
- 3) кнопка как элемент управления формы;
- 4) кнопка как элемент ActiveX¹.

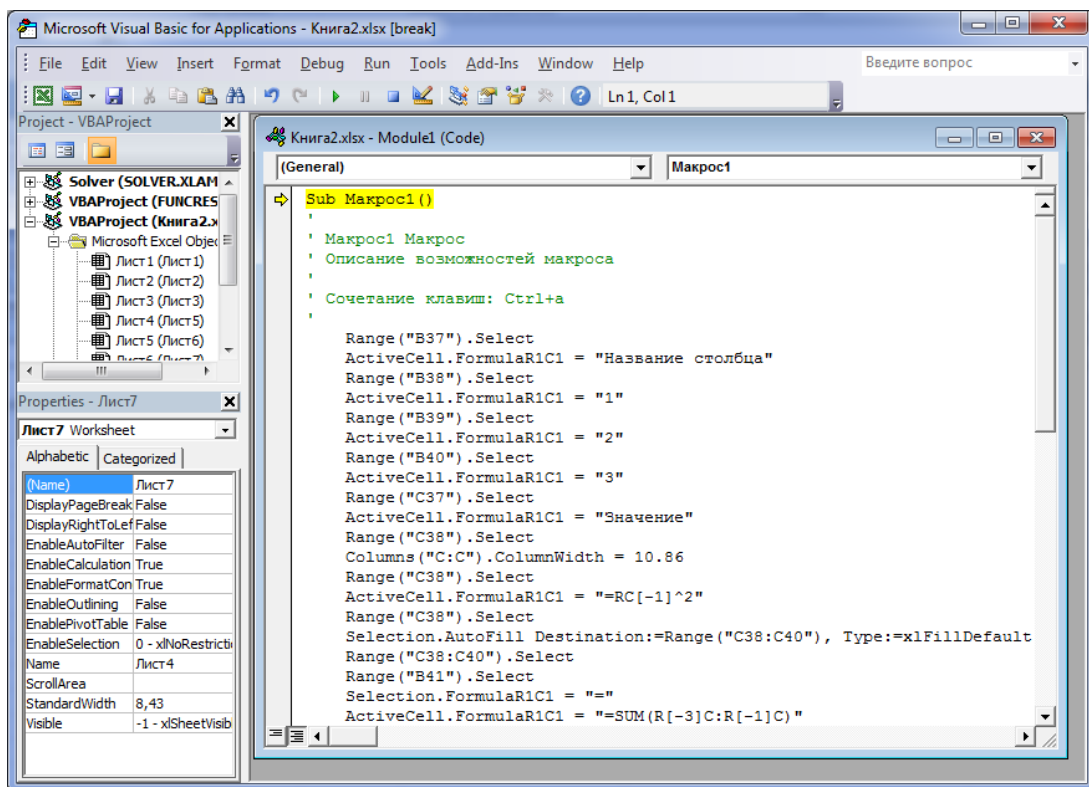


Рис. 4.6 — Диалоговое окно Microsoft Visual Basic for Application

В первых двух вариантах на лист вставляется фигура или рисунок (**Вставка** | группа **Иллюстрации**). Затем вызывается контекстно-зависимое меню, выбирается **Назначить макрос**.

¹ ActiveX (ранее OLE — Object Linking and Embedding) — технология Microsoft, позволяющая встраивать управляющие элементы в приложения.

В появившемся окне **Назначить макрос объекту** выбирается нужный макрос.

В третьем варианте вызывается объект **Элементы управления формы (Разработчик | Вставить | группа Элементы управления формы)** (рис. 4.7). На появившейся панели щелчком мыши выбираем форму **Кнопка**. При этом указатель мыши превращается в тонкий крестик. Щелкаем им по листу. На нем появляется кнопка с именем **Кнопка1** и одновременно открывается окно **Назначить макрос объекту**, в котором выбирается имя макроса.

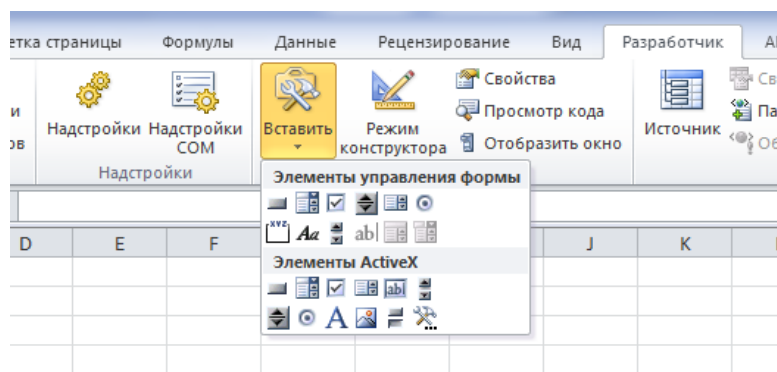


Рис. 4.7 — Элементы управления формы и элементы ActiveX

Для четвертого варианта требуется знание некоторых конструкций VBA и об этом будет рассказано чуть позже.

Можно изменить формат кнопки (шрифт надписи, размер и т. п.). Для этого следует вызвать контекстно-зависимое меню и выполнить необходимые операции.

Задание 4.1. Штатное расписание

Составить штатное расписание небольшой частной больницы, то есть определить, сколько сотрудников, с каким окладом и на какие должности необходимо принять на работу. Общий месячный фонд зарплаты (ФЗП) составляет \$20000.

Предположим, что для нормальной работы больницы нужно: 1 заведующий больницей, 1 главный врач, 3 заведующих отделениями, 10–12 врачей, 8–10 медсестер, 5–7 санитарок, 1 заведующий аптекой, 1 заведующий хозяйством.

Предлагается следующая модель решения задачи. За основу берется оклад санитарки. Размер оклада остальных сотрудников определяется по формуле

$$\text{Оклад} = A \times (\text{Оклад санитарки}) + B,$$

где A — должностной коэффициент оклада;

B — величина надбавки, \$.

Значения A и B назначаются исходя из следующих соображений:

- медсестра должна получать в 1,5 раза больше санитарки;
- врач — в 3 раза больше санитарки;
- заведующий отделением — на \$300 больше, чем врач;
- заведующий аптекой — в 2 раза больше санитарки;
- заведующий хозяйством — на \$180 больше медсестры;
- главный врач — в 4 раза больше санитарки;
- заведующий больницей — на \$200 больше главного врача.

1. На листе Excel оформить таблицу, используя следующие столбцы: **Должность, Количество сотрудников, Должностной коэффициент оклада, Надбавка, Оклад, Итого.**

При решении задачи используйте сервисную функцию Excel **Подбор параметра** (см. задание 2.1). В поле **Установить в ячейке** ввести адрес ячейки, где вычисляется общая месячная зарплата всех сотрудников больницы. В поле **Значение** ввести сумму ФЗП. В поле **Изменяя значение ячейки** ввести адрес ячейки, где находится оклад санитарки. После нажатия **ОК** произойдет автоматический подбор значения оклада санитарки таким образом, чтобы ФЗП составил \$20000.

2. Рассчитать оклады для нескольких вариантов штата, изменяя количество штатных единиц в соответствии с заданными условиями.

3. Включить запись макроса и выполнить описанные выше действия по расчету штатного расписания «под запись». Посмотреть код созданного макроса.

4. Изменить макрос таким образом, чтобы можно было в некоторой ячейке задавать произвольное значение ФЗП и в соответствии с этим значением рассчитывать оклады сотрудников. Проверить его работу.

5. Создать на листе кнопку вызова макроса. Теперь описанная выше процедура расчета штатного расписания будет выполняться простым нажатием кнопки.

Примечание. Иногда требуется скрыть (не уничтожить!) несколько столбцов или строк в таблице, для того, например, чтобы при печати они не отображались или с целью спрятать конфиденциальную информацию.

Для этого щелкаем мышью по имени столбца (строки) и, удерживая нажатой левую кнопку, проводим по тем столбцам (строкам), которые необходимо скрыть. Затем вызываем контекстно-зависимое меню и выбираем **Скрыть**. Чтобы снова их показать, необходимо выделить столбцы (строки), между которыми находится скрытое, и в контекстно-зависимом меню выбрать **Показать**.

Чтобы наложить запрет на изменение данных в созданном вами листе Excel, необходимо выполнить последовательность действий **Рецензирование** | группа **Изменения** | **Защитить лист**. В появившемся окне следует отметить, что именно защищается: содержимое, объекты, сценарии. Можно указать пароль. Снимается защита по команде **Снять защиту листа** (там же — в группе **Изменения**) и паролю.

Задание 4.2. Шаблоны таблицы и диаграммы

1. Записать в виде макроса процесс создания шаблона таблицы и построения диаграммы по табличным данным. Варианты выбрать из лабораторной работы 1 (задание 1.1). При вызове макроса на активном рабочем листе должен автоматически создаваться шаблон таблицы. При занесении исходных данных расчет результатных данных (которые помечены знаком вопроса) и построение диаграммы будут происходить автоматически.

При наборе формул предусмотреть возможность возникновения аварийных ситуаций (например, деление на нуль) и обойти их, используя функцию ЕСЛИ.

2. Открыть код записанного макроса и добавить комментарии, описав, каким вашим действиям соответствуют те или иные строки программного кода. Комментарии начинаются с кавычки (рис. 4.6) и не влияют на выполнение кода.

Раздел 5. Среда программирования VBA в Excel: процедуры и функции

Цель раздела: освоить программирование в среде Visual Basic for Application пользовательских процедур и работу с массивами.

Итак, после выполнения предыдущей работы на ленте осталась вкладка **Разработчик**. Нажатие на кнопку **Visual Basic** (или сочетание клавиш **Alt+F11**) приводит к появлению окна проектов редактора Visual Basic for Application (рис. 5.1). Рассмотрим на примерах некоторые часто используемые возможности языка программирования Visual Basic (в связке со средой Excel) и визуальной среды программирования VBA.

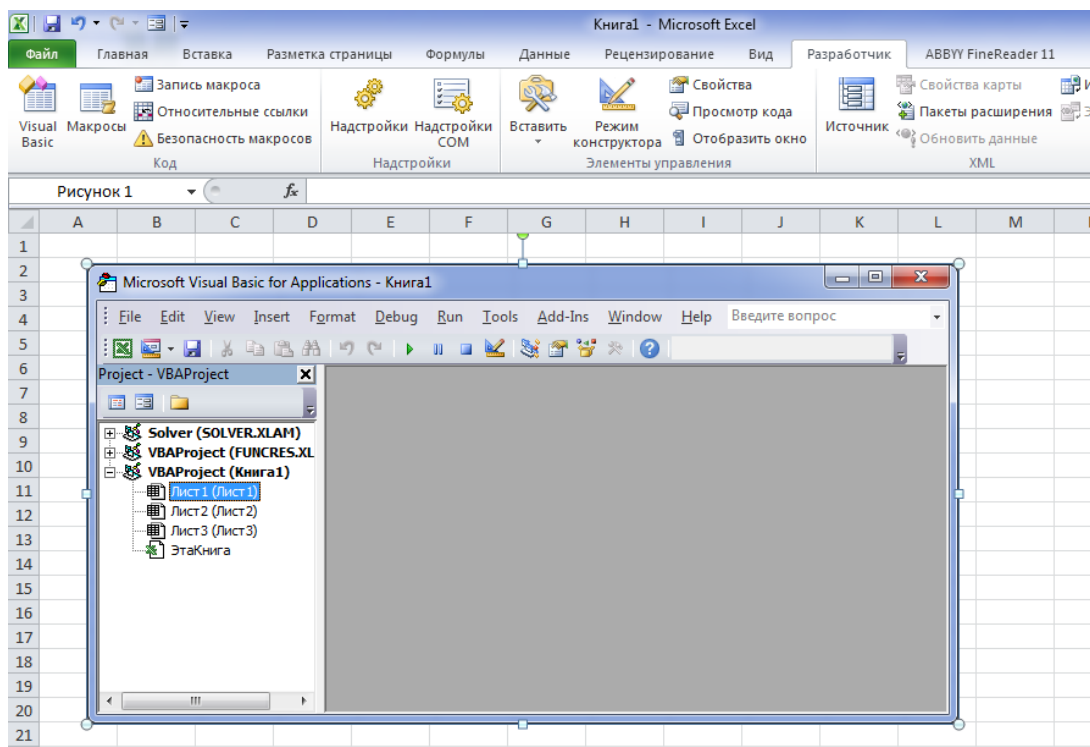


Рис. 5.1 — Окно редактора VBA

5.1 Структура программ на VBA

Программы на VBA хранятся в проектах (**VBAProject**). Проект содержит модули различных типов, а модули включают различные процедуры.

Имеется три типа модулей:

1) стандартные модули (**Module**) — это модули, в которых можно описать доступные во всем проекте процедуры;

2) модули класса (**Class Module**) содержат описание объекта, который является членом класса. Процедуры, написанные в модуле класса, используются только в этом модуле;

3) модули форм и отчетов (**UserForm**), которые связаны с конкретной формой или отчетом. Модули форм и отчетов часто содержат процедуры обработки событий, которые срабатывают в ответ на событие в форме или отчете. Процедуры обработки событий используются для управления поведением форм и отчетов и их реакцией на действия пользователя типа щелчка мыши на кнопке.

Модули содержат описания и процедуры — наборы описаний и операторов, сгруппированных для выполнения. Существует три типа процедур:

1) процедура **Sub** (подпрограмма) — набор команд, с помощью которого можно решить определенную задачу. При ее запуске выполняются команды процедуры, а затем управление передается в приложение пакета MS Office или процедуру, которая вызвала данную процедуру.

2) процедура **Function** (функция) также представляет собой набор команд, который решает определенную задачу. Различие заключается в том, что такие процедуры обязательно возвращают значение, тип которого можно описать при создании функции.

3) процедура **Property** используется для ссылки на свойство объекта. Данный тип процедур применяется для установки или получения значения пользовательских свойств форм и модулей.

Начало модуля называется *общей областью*, в которой располагаются общие описания, например, типа данных, используемого по умолчанию (DefТип), инструкция **Option Explicit**, требующая явного описания всех используемых в модуле переменных, а также описания общих (глобальных) для всех модулей и для данного модуля переменных.

Процесс создания проекта программы можно разделить на следующие этапы:

— постановка задачи (подготовительный этап) — составление точного и полного словесного описания работы программы,

определение исходных и выходных данных, будет ли использоваться экранная форма и как она будет выглядеть, т. е. какие объекты должны быть на ней и какими должны быть их свойства (объекты могут располагаться непосредственно на рабочем листе);

- разработка пользовательского интерфейса — создание экранной формы в среде VBA со всеми находящимися на этой форме объектами и их свойствами;

- программирование — определение событий, происходящих в процессе работы программы, составление процедур этих событий и написание программных кодов этих процедур;

- отладка программы — устранение логических ошибок в процедурах и достижение правильности работы программы;

- сохранение проекта.

Рассмотрим основные элементы языка программирования VBA.

5.2 Скалярные типы данных

Все данные, которыми оперирует VBA, относятся к определенному типу.

Тип данных определяет:

- область возможных значений переменной;
- структуру организации данных;
- операции, определенные над данными этого типа.

Типы данных подразделяются на простые (скалярные) и сложные (структурированные). У простых типов данных возможные значения данных едины и неделимы. Сложные же типы имеют структуру, в которую входят различные простые типы данных. Скалярные типы данных представлены в таблице 5.1.

Таблица 5.1 — Скалярные типы VBA

Имя типа	Название типа	Возможные значения
Boolean	Логический	True, False
Byte	Байтовый	0...255
Integer	Целое	–32768...+32767
Long	Длинное целое	–2147483648...+2147483647
Single	Число с плавающей точкой	–3,4E38...–1,4E–45 для отрицательных значений. 1,4E–45...3,4E38 для положительных значений

Окончание табл. 5.1

Имя типа	Название типа	Возможные значения
Double	Число с плавающей точкой двойной точности	–1,7E308...–4,9E–324 для отрицательных значений. 4,9E–324...1,7E308 для положительных значений
Currency	Денежный	Десятичные числа с фиксированной позицией запятой. Возможны 15 цифр до запятой и 4 после
String	Строковый	Есть два вида строк: строки фиксированной длины (до 216 символов) и строки переменной длины (до 231 символов). Данные записываются в кавычках
Date	Дата	Даты изменяются в диапазоне от 1.01.100 г. до 31.12.9999г.
Object	Объект	Ссылка на объект (указатель)
Variant	Вариант	Универсальный тип, значением которого могут быть данные любого из перечисленных выше типов, объекты, значения NULL и значения ошибок ERROR

Переменные в программе можно описывать или не описывать. В последнем случае переменной будет присвоен тип Variant. Явно описывать переменную можно как в начале блока, так и в любом месте, где возникла необходимость использовать новую переменную. Как правило, переменные описывают явно и в начале блока. Для запрета использования переменных, которые не были описаны явно, в начало программы необходимо вставить оператор **Option Explicit**.

Описание простых переменных имеет следующий синтаксис:

Dim ИМЯ_ПЕРЕМЕННОЙ As ИМЯ_ТИПА

Одним оператором Dim можно описать произвольное число переменных, но конструкция As должна быть указана для каждой из них, иначе переменным без As будет присвоен тип Variant.

Например,

Dim X As Byte, Z As Integer, C, Слово As String

Здесь переменная X — это переменная байтового типа, переменная Z — целого типа, переменная C — типа вариант (по умолчанию), переменная Слово — строкового типа.

Данные, не изменяющиеся внутри программы, можно считать константами. Их описание:

Const ИМЯ_КОНСТАНТЫ As ИМЯ_ТИПА=ПОСТОЯННОЕ ВЫРАЖЕНИЕ

Например.

Const Pi As Double = 3.141593

5.3 Структурированные типы данных

Для хранения векторов, матриц и т. д. используют массивы.

Массив — это структурированный тип данных, который представляет собой последовательность ячеек памяти, имеющих общее имя и хранящих данные одного типа. Каждый элемент массива определяется индексом (номером). Количество элементов в массиве называется размерностью массива. Массив описывается следующей конструкцией:

**Dim ИМЯ_МАССИВА (СПИСОК_РАЗМЕРНОСТЕЙ) As
ИМЯ_ТИПА**

В списке размерностей массива задаются нижняя и верхняя границы изменения индексов. Каждый элемент списка отделяется запятой.

Например.

Dim X(1 To 5) As Integer, Y(1 To 10, 1 To 20) As Single

Здесь X — одномерный массив, состоящий из 5 элементов целого типа, Y — двумерный массив, у которого 10 строк и 20 столбцов с элементами числового типа одинарной точности.

5.4 Выражения

Выражения устанавливают порядок выполнения действий над элементами данных. Выражения состоят из операндов и знаков операций. Операндами являются константы, переменные, указатели функций, выражения, взятые в скобки.

Операции бывают арифметические, отношения и логические.

Арифметические операции:

- ^ возведение в степень,
- * умножение,
- / деление,
- \ деление нацело,

mod остаток от деления,
 + плюс,
 – минус.

Операции отношения:

< меньше,
 > больше,
 <= меньше или равно,
 >= больше или равно,
 = равно,
 <> не равно.

Логические операции:

Not логическое отрицание,
 And логическое «И»,
 Or логическое «ИЛИ».
 Xor логическое исключение,
 Imp логическая импликация,
 Eqv логическая эквивалентность.

Результатом логической операции может быть одно из двух значений: True («истина») или False («ложь»).

Выражения бывают арифметические, отношения и логические.

Арифметические выражения записываются с помощью операндов числовых типов, арифметических операций, стандартных математических функций, а результатом является числовое значение.

Выражения отношения определяют истинность или ложность результата при сравнении двух операндов. Сравнить можно данные любого одинакового типа. Результат операции отношения только логический: True — «истина» или False — «ложь».

Результатом логического выражения является логическое значение True или False. Простейшими видами логических выражений являются: логическая константа, логическая переменная, логическая функция, выражение отношения. Логические операции выполняются только над операндами логического типа.

Например, выражение $1 \leq X \leq 5$ в VBA будет выглядеть так:

$X \geq 1 \text{ And } X \leq 5$

Его значение при $X=3.1$ будет True.

5.5 Операторы

Оператор — это наименьшая способная выполняться единица кода VBA.

Основные операторы программирования — это операторы присваивания, условного перехода и цикла.

При записи операторов необходимо следовать следующим правилам:

- каждый новый оператор записывается с новой строки;
- для записи нескольких операторов на одной строке их разделяют между собой двоеточием (:);
- если оператор не помещается в одной строке, то необходимо поставить в конце строки пробел и знак подчеркивания (_), а затем продолжить его на следующей строке.

5.5.1 Оператор присваивания

Оператор присваивания используется, если какой-то переменной нужно присвоить новое значение. Он имеет следующий синтаксис:

ИМЯ_ПЕРЕМЕННОЙ = ВЫРАЖЕНИЕ

Сначала вычисляется выражение в правой части (см. п.5.4), а затем результат присваивается переменной, стоящей в левой части.

X=3.1

Q = X>=1 And X<=5

5.5.2 Условные операторы

Оператор условного перехода (условный оператор) имеет несколько форм записи.

1. Оператор **If УСЛОВИЕ Then**

 БЛОК ОПЕРАТОРОВ1

Else

 БЛОК ОПЕРАТОРОВ2

End If

УСЛОВИЕ — это отношение или выражение логического типа. Их результатами является логическое значение True или False.

БЛОК ОПЕРАТОРОВ — один или несколько операторов. Первый блок выполняется, если УСЛОВИЕ истинное, второй — если ложное.

Если блоки содержат по одному оператору, то условный оператор записывается на одной строке и имеет вид:

If УСЛОВИЕ Then ОПЕРАТОР1 Else ОПЕРАТОР2

В условном операторе могут проверяться и несколько условий:

If УСЛОВИЕ1 Then
 БЛОК ОПЕРАТОРОВ1
ElseIf УСЛОВИЕ2 Then
 БЛОК ОПЕРАТОРОВ2
Else

....

End If

Для обхода по условию используется укороченный условный оператор:

If УСЛОВИЕ Then
 БЛОК ОПЕРАТОРОВ
End If

2. Оператор выбора **Select Case**.

Его удобно использовать, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия.

Select Case ПРОВЕРЯЕМОЕ ВЫРАЖЕНИЕ
 Case ЗНАЧЕНИЯ 1
 ОПЕРАТОРЫ 1
 Case ЗНАЧЕНИЯ 2
 ОПЕРАТОРЫ 2
 ...
 Case ЗНАЧЕНИЯ N
 ОПЕРАТОРЫ N
 ...
 Case Else
 ОПЕРАТОРЫ ПРИ НЕСОВПАДЕНИИ

End Select

ПРОВЕРЯЕМОЕ ВЫРАЖЕНИЕ имеет любой скалярный тип, кроме вещественного. ЗНАЧЕНИЯ состоят из произвольного количества значений или диапазонов (n **to** m), отделенных друг от друга запятыми.

Тип ЗНАЧЕНИЙ должен совпадать с типом ПРОВЕРЯЕМОГО ВЫРАЖЕНИЯ.

Сначала вычисляется ПРОВЕРЯЕМОЕ ВЫРАЖЕНИЕ. Если его значение совпадает с одним из значений ЗНАЧЕНИЯ I , то выполняются ОПЕРАТОРЫ I , и управление передается оператору, стоящему после **End Select**. Если его значение не совпадает ни с одним из значений ЗНАЧЕНИЯ I , то выполняются ОПЕРАТОРЫ ПРИ НЕСОВПАДЕНИИ и управление передается оператору, стоящему после **End Select**.

5.5.3 Операторы цикла

Операторы цикла тоже имеют несколько форм записи.

1. **Оператор For...Next**. Используется для выполнения набора инструкций указанное число раз. Циклы For используют в качестве счетчика переменную, значение которой увеличивается или уменьшается при каждом выполнении цикла.

```
For СЧЕТЧИК = НАЧАЛО To КОНЕЦ [Step ШАГ]
БЛОК_ОПЕРАТОРОВ
[Exit For]
БЛОК_ОПЕРАТОРОВ
Next [СЧЕТЧИК]
```

Повторяет выполнение блока операторов, пока переменная СЧЕТЧИК изменяет свое значение от начального до конечного с указанным шагом. Если ШАГ не указан, то по умолчанию он предполагается равным 1. Немедленный выход из цикла осуществляется оператором Exit For. Квадратные скобки обозначают необязательность конструкции в синтаксисе оператора.

2. **Оператор While...Wend**

```
While УСЛОВИЕ
БЛОК_ОПЕРАТОРОВ
Wend
```

Выполняет блок операторов, пока заданное условие имеет значение True (истина). Условие проверяется перед выполнением операторов, расположенных внутри цикла.

3. Оператор **Do While...Loop**

Do [**While** УСЛОВИЕ]
 БЛОК_ОПЕРАТОРОВ
[Exit Do]
 БЛОК_ОПЕРАТОРОВ
Loop

Выполняет блок операторов, пока заданное условие имеет значение True (истина). Условие проверяется перед выполнением операторов, расположенных внутри цикла. Если условие отсутствует, то получаем бесконечный цикл. Оператор Exit Do позволяет немедленно выйти из цикла.

4. Оператор **Do...Loop While**

Do
 БЛОК_ОПЕРАТОРОВ
[Exit Do]
 БЛОК_ОПЕРАТОРОВ
Loop [**While** УСЛОВИЕ]

Выполняет блок операторов, пока заданное условие имеет значение True (истина). Условие проверяется после выполнения операторов, расположенных внутри цикла.

5. Оператор **Do Until...Loop**

Do [**Until** УСЛОВИЕ]
 БЛОК_ОПЕРАТОРОВ
[Exit Do]
 БЛОК_ОПЕРАТОРОВ
Loop

Выполняет блок операторов, пока заданное условие имеет значение False (ложь). Условие проверяется перед выполнением операторов, расположенных внутри цикла.

6. Оператор **Do...Loop Until**

Do
 БЛОК_ОПЕРАТОРОВ
[Exit Do]
 БЛОК_ОПЕРАТОРОВ
Loop [**Until** УСЛОВИЕ]

Выполняет блок операторов, пока заданное условие имеет значение False (ложь). Условие проверяется после выполнения операторов, расположенных внутри цикла.

7. Оператор **For Each...Next**

For Each ЭЛЕМЕНТ **In** ГРУППА

БЛОК_ОПЕРАТОРОВ

[Exit For]

БЛОК_ОПЕРАТОРОВ

Next [ЭЛЕМЕНТ]

Повторяет выполнение блока операторов для элементов группы. Группой может быть как массив, так и набор объектов (ячейки, диаграммы, формы и т. д.). VBA автоматически задает значение переменной ЭЛЕМЕНТ во время каждой итерации цикла.

Пример 5.1. Составить таблицу начисления премиальных по итогам работы сети 4-х магазинов за три месяца по следующему правилу:

- если продукции продано не меньше чем на 60000 рублей, то премиальные составляют 2 % от суммарной выручки магазина;
- за первое место дополнительно начисляется 4 % премиальных, за второе — 2 %, за третье — 1 % от суммарной выручки магазина.

Сформируем таблицу в Excel и заполним значениями, например так, как показано на рис. 5.2.

	A	B	C	D	E	F	G	H	I	J	K
1	№ Магазины	Июнь	Июль	Август	Выручка	% за перевып.	Начисление % за место	Итого %	Премия	C=	60000
2	Магазин 1	10000	20000	30000							
3	Магазин 2	15000	20000	31000							
4	Магазин 3	18000	21000	20000							
5	Магазин 4	10000	10000	19000							
6											

Рис. 5.2 — Исходная таблица примера 1

Для удобства будущей работы в VBA можно сменить стиль ссылок на ячейки: буквы столбцов заменить на цифры (см. рис. 5.5). Для этого исполните команду: **Файл | Параметры | Формулы | Работа с формулами | Стиль ссылок R1C1** (поставить галочку)). Теперь адрес ячейки D5 будет выглядеть как

R5C4, а в формулах адрес используемой ячейки, будет задаваться сдвигом относительно той ячейки, где вводится формула. Удобство этого вы оцените, когда будете работать с объектом **Cells**.

Создадим в редакторе VBA процедуру обработки события **Click** (щелчок мыши по кнопке), которая проводит вычисления по заданному алгоритму и заполняет пустые ячейки таблицы.

Сначала создадим кнопку (в группе **Элементы управления** выполнить **Вставить | Кнопка (Элементы ActiveX)**). На листе Excel рисуем мышкой кнопку с именем по умолчанию **CommandButton1**.

Изменить имя можно несколькими способами. Оба способа работают при нажатой кнопке **Режим конструктора** в **Элементах управления**.

Способ 1. Вызвать на кнопке контекстно-зависимое меню и выбрать опцию **Свойства** (рис. 5.3). В поле **Caption** заменить имя **CommandButton1** на **Заполнение таблицы**.

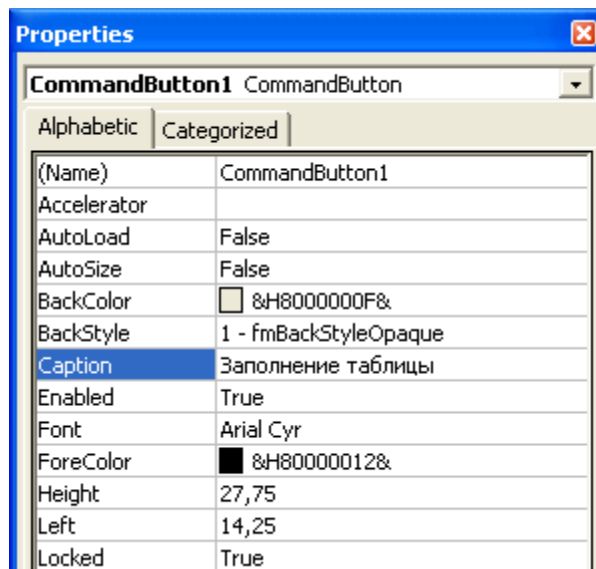


Рис. 5.3 — Окно **Свойства кнопки**

Способ 2. Вызвать контекстно-зависимое меню и выбрать опцию **Объект CommandButton | Edit**. В поле кнопки появляется маркер — можно изменить текст.

Двойной щелчок по созданной кнопке, или выбор опции **Исходный текст** в контекстно-зависимом меню, или щелчок по кнопке **Visual Basic** осуществляет переход в редактор VBA

(рис. 5.4), в правом окне которого уже появился шаблон процедуры обработки события **Click**.

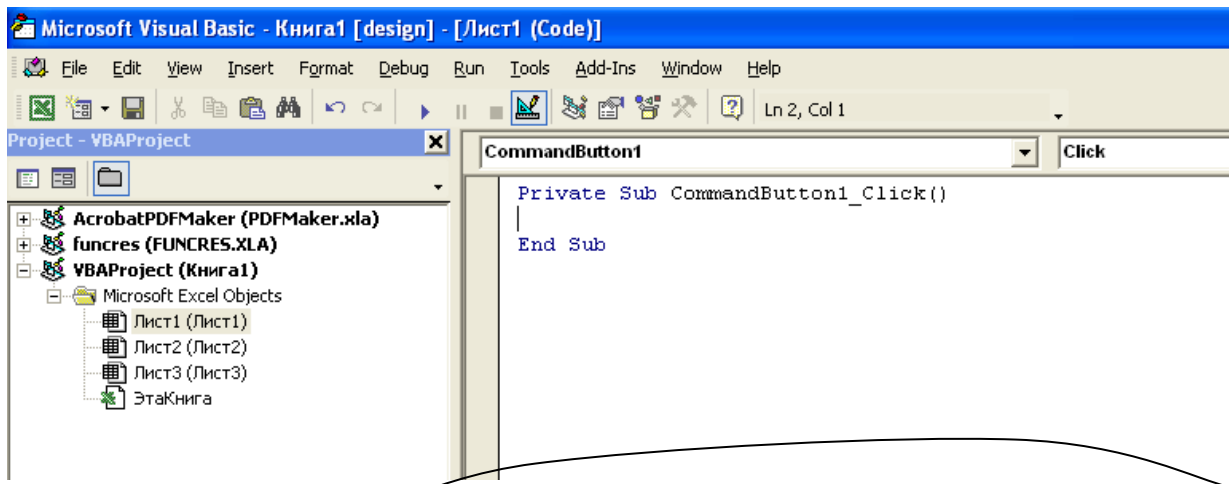


Рис. 5.4 — Интерфейс редактора VBA

Напишем следующий код для этой процедуры с использованием циклических структур. Текст, начинающийся с кавычки, — это примечание.

Private Sub CommandButton1_Click()

'Цикл суммирования выручки за 3 месяца

For i = 2 To 5

Cells(i, 5).Value = Cells(i, 2).Value + Cells(i, 3).Value + Cells(i, 4).Value

Next

'Создание вспомогательного массива box

'и заполнение его значениями выручки

Dim box(4) As Double

box(1) = Cells(2, 5)

box(2) = Cells(3, 5)

box(3) = Cells(4, 5)

box(4) = Cells(5, 5)

'Сортировка выручки за 3 месяца по убыванию методом «пузырька».

'При этом в box(1) окажется максимальное значение выручки

For i = 1 To 4

For j = i + 1 To 4

If box(i) < box(j) Then

q = box(i)

box(i) = box(j)

box(j) = q

End If

Next

Next

'Начисление процентов в зависимости от занятого места

For i = 2 To 5

Cells(i, 7) = 0

If Cells(i, 5) = box(1) Then Cells(i, 7) = 4

If Cells(i, 5) = box(2) Then Cells(i, 7) = 2

If Cells(i, 5) = box(3) Then Cells(i, 7) = 1

Next

'Начисление процентов, если выручка за 3 месяца больше плановой выручки

For i = 2 To 5

If Cells(i, 5).Value >= Cells(1, 11).Value Then

Cells(i, 6) = 2

Else: Cells(i, 6) = 0

End If

Next

'Суммирование процентов

For i = 2 To 5

Cells(i, 8).Value = Cells(i, 6).Value + Cells(i, 7).Value

Next

'Расчет итоговой премии

For i = 2 To 5

Cells(i, 9).Value = Cells(i, 5).Value * Cells(i, 8).Value / 100

Next

End Sub

Закроем редактор и нажмем на кнопку «Заполнение таблицы», отключив предварительно **Режим конструктора**. Получим заполненную таблицу (рис. 5.5).

	1	2	3	4	5	6	7	8	9	10	11
1	№ магазина	Июнь	Июль	Август	Выручка	% за перевып.	Начисление % за место	Итого %	Премия		C= 60000
2	Магазин 1	10000	20000	30000	60000	2	2	4	2400		
3	Магазин 2	15000	20000	31000	66000	2	4	6	3960		
4	Магазин 3	18000	21000	20000	59000	0	1	1	590		
5	Магазин 4	10000	10000	19000	39000	0	0	0	0		

Рис. 5.5 — Результирующая таблица: значение премии в зависимости от выручки магазина

В результате в столбце Е окажется сумма выручек за 3 месяца, в столбце F — процент, назначенный за перевыполнение плана, в столбце G — процент, назначенный в зависимости от занятого места, в столбце H — итоговый процент, в столбце I — величина премии.

Пример 5.2. Составить таблицу начисления премии по итогам работы сети 4-х магазинов за три месяца по следующему правилу:

- если продукции продано меньше чем на 20 тыс. руб., то премия не начисляется;
- если продукции продано на сумму от 20 до 40 тыс. руб., премия составляет 3 % от выручки;
- если продукции продано на сумму от 40 до 80 тыс. руб., премия составляет 4,5 % от выручки;
- если продукции продано больше чем на 80 тыс. руб., то премия составляет 6,5 %.

Аналогично примеру 1 составим исходную таблицу, введя значение выручки каждого магазина за три месяца (рис. 5.6).

9						
10	№ магазина	Январь	Февраль	Март	Выручка	% за перевып.
11	Магазин 1	5000	50000	10000		
12	Магазин 2	10000	10000	30000		
13	Магазин 3	20000	20000	20000		
14	Магазин 4	40000	40000	10000		
15						
16		CommandButton2				
17						
18						

Рис. 5.6 — Исходная таблица примера 2

Создадим кнопку и назначим ей процедуру обработки события **Click** (по щелчку), позволяющую вычислять и заносить в соответствующие ячейки выручку и премию. Для расчета премии используем функцию **Премия**, код которой приведен ниже процедуры.

```

Private Sub CommandButton1_Click()
'Цикл суммирования выручки за 3 месяца
For i = 11 To 14
    Cells(i, 5).Value = Cells(i, 2).Value + Cells(i, 3).Value_
        + Cells(i, 4).Value
Next
'Цикл начисления премии в зависимости от выручки за 3 месяца
For i = 11 To 14
    Cells(i, 6).Value = Премия(Cells(i, 5).Value)
Next
End Sub

'Функция начисления премии в зависимости от объема продаж
Function Премия(Продажа As Double) As Double
Select Case Продажа
Case 0 To 20000
    Премия = 0
Case 20001 To 40000
    Премия = 0.03 * Продажа
Case 40001 To 80000
    Премия = 0.045 * Продажа
Case Is>80001
    Премия = 0.065 * Продажа
End Select
End Function

```

Закроем редактор и нажмем на кнопку. В столбце Е появится доход от продаж, в столбце F — премия, которая начисляется в зависимости от того, на какую сумму продан товар (рис. 5.7).

9						
10	№ магазина	Январь	Февраль	Март	Выручка	% за перевып.
11	Магазин 1	5000	50000	10000	65000	2925
12	Магазин 2	10000	10000	30000	50000	2250
13	Магазин 3	20000	20000	20000	60000	2700
14	Магазин 4	40000	40000	10000	90000	5850
15						
16						
17						
18						

Рис. 5.7 — Результирующая таблица

Пример 5.3. Вычислить значение
$$S = \frac{2 \sum_{i=1}^m a_i + \left(\sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)}{\left(1 + \sum_{i=1}^m a_i \right) \left(1 + \sum_{i=1}^m a_i^2 \right)},$$

где a — вектор из m компонентов, c — матрица размерности

$n \times n$, при $a = [3; 1; 2; 3]$, $\tilde{c} = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{bmatrix}$.

Откроем лист Excel и введем исходные данные. В диапазон (A1:D1) введем элементы вектора a , в диапазон (A3:C5) — элементы матрицы c .

Способ 1. Непосредственный ввод формулы в ячейку с использованием стандартных функций рабочего листа Excel.

В ячейку E3 введем формулу:

$$=((2*\text{СУММ}(A1:D1)+\text{СУММ}(A3:C5))/((1+\text{СУММ}(A1:D1))*(1+\text{СУММКВ}(A1:D1))))$$

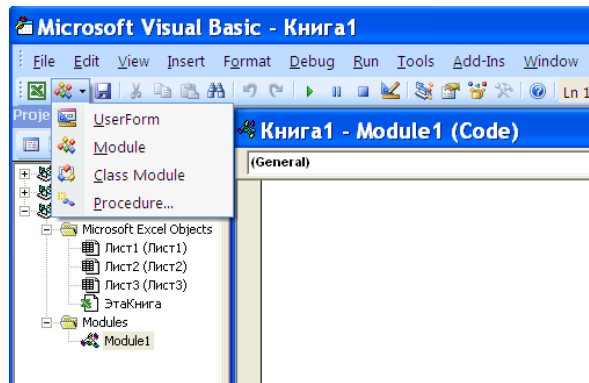
и нажмем клавишу Enter. В результате в ячейке E3 получим искомое значение S (рис. 5.8).

	A	B	C	D	E	F	G	H	I	J	K
1	3	1	2	3							
2											
3	2	2	4		0,2						
4	2	4	6		0,2						
5	2	5	3		0,2						
6											

Рис. 5.8 — Результирующая таблица примера 3

Способ 2. Программирование в VBA с использованием стандартных функций для работы с массивами.

Для этого в редакторе VBA добавляем модуль с помощью команды **Insert | Module** (рис. 5.9).

Рис. 5.9 — Выбор команды **Insert | Module**

Запишем в модуле код:

Public Function FirstFunction(a, c As Variant) As Variant
Dim s1, s2, s3 As Variant

'Сумма элементов вектора *a*

s1 = WorksheetFunction.Sum(a)

'Сумма элементов матрицы *c*

s2 = WorksheetFunction.Sum(c)

'Сумма квадратов элементов вектора *a*

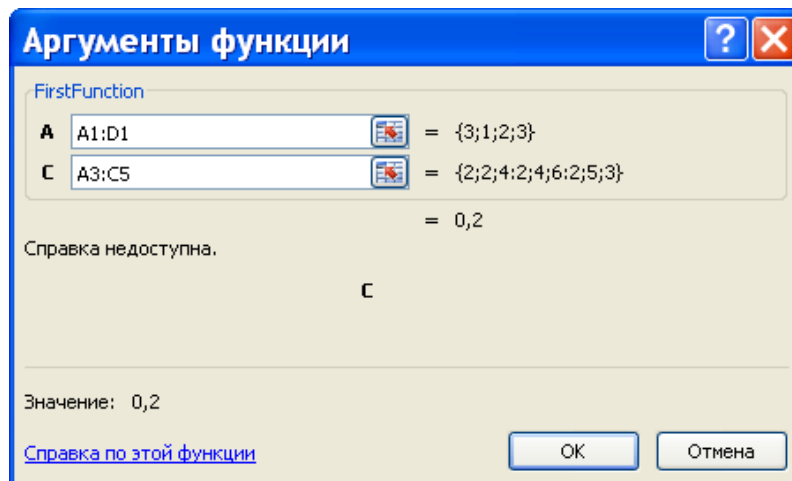
s3 = WorksheetFunction.SumSq(a)

'Расчет результата

FirstFunction = (2 * s1 + s2) / ((1 + s1) * (1 + s3))

End Function

Теперь в списке функций Excel в категории **Определенные пользователем** появилась функция **FirstFunction**. Ее вызов на лист Excel (**Формулы | группа Библиотека функций | Вставить функцию** или щелчок по соответствующей кнопке на строке формул) приводит к появлению диалогового окна **Аргументы функции** (рис. 5.10), где вводятся адреса массивов *a* и *c*.

Рис. 5.10 — Диалоговое окно **Аргументы функции**

После нажатия кнопки **ОК** в той ячейке, из которой вызвали функцию (здесь E4), получаем результат (рис. 5.8).

Способ 3. Программирование в VBA с использованием циклических конструкций.

Для этого в том же модуле запишем новую функцию, где расчет вычисления искомого значения S оформим без использования стандартных функций, используя циклы (для разнообразия — трех разных типов).

```

Public Function SecondFunction(a, c As Variant) As Double
Dim s1, s2, s3 As Double, i, j, n, m As Integer
'Нахождение количества элементов вектора  $a$ 
    m = a.Columns.Count
'Нахождение количества строк матрицы  $c$  (оно же — количество столбцов,
'если матрица квадратная)
    n = c.Rows.Count
'Найдем сумму элементов вектора  $a$  (цикл типа While... Wend)
    s1 = 0
    i = 1
    While i <= m
        s1 = s1 + a(i)
        i = i + 1
    Wend
'Найдем сумму элементов матрицы  $c$  (цикл типа For... Next)
    s2 = 0
    For i = 1 To n
        For j = 1 To n
            s2 = s2 + c(i, j)
        Next j
    Next i
'Найдем сумму квадратов элементов вектора  $a$  (цикл типа Do... Loop)
    s3 = 0
    i = 1
    Do Until i > m
        s3 = s3 + a(i) ^ 2
        i = i + 1
    Loop
'Расчет результата
    SecondFunction = (2 * s1 + s2) / ((1 + s1) * (1 + s3))
End Function

```

Теперь в списке функций Excel в категории **Определенные пользователем** появилась еще одна функция **SecondFunction**. Ее вызов в ячейке E5 приводит к появлению третьего ответа (рис. 5.8). Все три решения совпали, что и следовало ожидать.

Пример 5.4. Построить квадратную матрицу G размером $n \times n$, элементы которой определяются по формуле:

$$G(i, j) = \begin{cases} \sin^2(C(i)), & i \leq j \\ C(i - j) + \cos(C(i)), & i > j \end{cases}, \text{ для } C=[2;4;1;7], n=4.$$

Сформируем на новом листе массив вектора C в ячейках A2:D2, в ячейку B4 занесем значение n (рис.5.11).

	A	B	C	D	E
1		Вектор C			
2	2	4	1	7	
3					
4	n=	4	Старт!		
5					
6		Матрица G			
7	0,8268	0,8268	0,8268	0,8268	
8	1,3464	0,5728	0,5728	0,5728	
9	4,5403	2,5403	0,7081	0,7081	
10	1,7539	4,7539	2,7539	0,4316	
11					

Рис. 5.11 — Таблицы примера 4

Способ 1. Создание процедуры построения матрицы, запускаемой кнопкой.

Создадим кнопку «Старт!», как это было сделано в предыдущей работе. Затем щелкнем два раза по кнопке и в окне редактирования кода модуля запишем процедуру обработки щелчка по кнопке — процедуру построения матрицы.

Private Sub CommandButton1_Click()

'Определяем переменную, в которой будет храниться размер матрицы

Dim n As Integer

'Определяем массив C для исходных данных и массив g

'для результирующей квадратной матрицы

```

Dim C(), g()
'Задаем значение переменной n, выбирая его из ячейки B4
n = Cells(4, 2)
'Выделяем память для массивов C и g
ReDim C(1 To n), g(1 To n, 1 To n)
'Вводим в массив C значений из ячеек A2:D2
For k = 1 To n
    C(k) = Cells(2, k)
Next
'Проводим расчет элементов матрицы 'по заданной формуле,
'организуя двойной цикл (по строкам и столбцам)
For i = 1 To n
    For j = 1 To n
        If i <= j Then
            g(i, j) = Sin(C(i)) ^ 2
        Else
            g(i, j) = C(i - j) + Cos(C(i))
        End If
    Next j
Next i
End Sub

```

Затем, нажав на кнопку «Старт!», в ячейках A7:D10 получим искомую матрицу G (рис. 5.11).

Задание 5.1. Исключить лишнее входное данные

Измените код так, чтобы в примере, рассмотренном в способе 1 примера 5.4, можно было обойтись без использования ячейки со значением n , т. е. убрать его из списка входных данных.

Способ 2. Создание пользовательской функции.

По аналогии с примером 3 создадим функцию, аргументом которой является массив и которая после работы возвращает также массив.

Function T(C As Variant) As Variant

'Определяем локальные переменные

```

Dim n, i, j As Integer
'Определяем массив R для исходных данных
Dim R() As Variant
'Определяем число элементов вектора, расположенного в строке
n = C.Columns.Count
'Выделяем память для вспомогательного двумерного массива R
'размером  $n \times n$ 
ReDim R(1 To n, 1 To n)
'Вычисляем элементы матрицы
For i = 1 To n
    For j = 1 To n
        If i <= j Then R(i, j) = Sin(C(i)) ^ 2
        If i > j Then R(i, j) = C(i - j) + Cos(C(i))
    Next
Next
'Возвращаем элементы матрицы
T = R
End Function

```

Теперь в списке пользовательских функций появилась функция T.

Вызов этой функции отличается от вызова функции, результатом которой является одно число. Сначала надо выделить на листе Excel массив ячеек, куда будет занесен ответ. В нашем случае это массив ячеек A7:D10 размером 4×4 . Затем после вызова функции и занесения данных (массив ячеек A2:D2) следует выйти из диалогового окна функции, нажав сочетание клавиш **Shift+Ctrl+Enter** (а не ОК или Enter, как обычно).

Задание 5.2. Процедуры и функции

Пусть C — одномерный массив, состоящий из n элементов, а G — квадратная матрица $n \times n$. В задании (а) необходимо вычислить квадратную матрицу G и вывести значения ее элементов на рабочий лист. В задании (б) нужно вычислить значение s тремя способами:

- 1) написать функцию, используя различные операторы цикла;
- 2) написать функцию, используя стандартные функции VBA;
- 3) ввести формулу в ячейку с помощью стандартных функций рабочего листа Excel.

Вариант 1

а) построить матрицу ($n = 3$ и $C = [3 \ 4 \ 1]$)

$$G(i, j) = \begin{cases} C^2(i), & i \leq j; \\ \sin(C(i) \cdot C(i - j)) + C^2(i), & i > j; \end{cases}$$

б) вычислить значение

$$S = \frac{2 \cdot \sum_{i=1}^n x_i y_i + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^2}{3 + \sum_{i=1}^n x_i},$$

где x, y — векторы из n компонентов, b — матрица размерности $m \times m$, причем $n = 4, m = 2$,

$$x = [3 \ 1 \ 2 \ 3], y = [1 \ 7 \ 2 \ 3], b = \begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}.$$

Вариант 2

а) построить матрицу ($n = 4$ и $C = [2 \ 4 \ 1 \ 7]$)

$$G(i, j) = \begin{cases} \sin^2(C(i)), & i \leq j; \\ C(i - j) + \cos(C(i)), & i > j; \end{cases}$$

б) вычислить значение

$$S = \frac{2 \cdot \sum_{i=1}^m a_i + \left(\sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)}{\left(1 + \sum_{i=1}^m a_i \right) \cdot \left(1 + \sum_{i=1}^m a_i^2 \right)},$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3, m = 4$

$$a = [3 \ 1 \ 2 \ 3], c = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{bmatrix}.$$

Вариант 3

а) построить матрицу ($n = 5$ и $C = [6 \ 4 \ 2 \ 7 \ 6]$)

$$G(i, j) = \begin{cases} C(i) + i + j^2, & i - j \leq 3; \\ C(i - j) + C(i)^3, & i - j > 3; \end{cases}$$

б) вычислить значение

$$s = \frac{\sum_{i=1}^n x_i + 2 \cdot \sum_{i=1}^n y_i^2 + 5 \cdot \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^3}{3 + \sum_{i=1}^n y_i},$$

где x, y — векторы из n компонентов, b — матрица размерности $m \times m$, причем $n = 4, m = 2$,

$$x = [1 \ 2 \ 7 \ 4], y = [1 \ 7 \ 2 \ 3], b = \begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}.$$

Вариант 4

а) построить матрицу ($n = 4$ и $C = [1 \ 2 \ 1 \ 7]$)

$$G(i, j) = \begin{cases} C(i) \cdot C(j), & i \leq j + 2; \\ C(i - j)^2 \cdot C(i), & i > j + 2; \end{cases}$$

б) вычислить значение

$$s = 3 \cdot \sum_{i=1}^m a_i^2 + 7 \cdot \sum_{i=1}^n \sum_{j=1}^n c_{ij} - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)^2,$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3, m = 4$,

$$a = [3 \ 1 \ 2 \ 3], c = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{bmatrix}.$$

Вариант 5

а) построить матрицу ($n = 6$ и $C = [1 \ 3 \ 1 \ 7 \ 2 \ 2]$)

$$G(i, j) = \begin{cases} \sin(C(i) \cdot C(j)), & i \leq j + 2; \\ |C(i - j) - C(i)|, & i > j + 2; \end{cases}$$

б) вычислить значение

$$s = \sum_{i=1}^n x_i + 2 \cdot \sum_{i=1}^n y_i^2 + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right) \cdot \left(2 + \sum_{i=1}^n x_i \right) - 2 \cdot \left(1 + \sum_{i=1}^n x_i y_i \right),$$

где x, y — векторы из n компонентов, b — матрица размерности $m \times m$, причем $n = 4, m = 2$,

$$x = [1 \ 2 \ 7 \ 4], y = [1 \ 7 \ 2 \ 3], b = \begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}.$$

Вариант 6

а) построить матрицу ($n = 7$ и $C = [1 \ 4 \ 1 \ 3 \ 4 \ 5 \ 7]$)

$$G(i, j) = \begin{cases} 3 \cdot C(i) - 2 \cdot C(j), & i \leq j + 2; \\ |C(i - j) - C(i)^3|, & i > j + 2; \end{cases}$$

б) вычислить значение

$$s = \left(\sum_{i=1}^m a_i \right)^2 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 - \left(3 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \cdot \left(1 + \sum_{i=1}^m a_i^2 \right),$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3, m = 4$,

$$a = [3 \ 3 \ 1 \ 3], c = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{bmatrix}.$$

Вариант 7

а) построить матрицу ($n = 4$ и $C = [1 \ 4 \ 8 \ 7]$)

$$G(i, j) = \begin{cases} C(i) \cdot \cos^2(C(j)), & i \leq j + 1; \\ |C(i - j)^3 - C(i)|, & i > j + 1; \end{cases}$$

б) вычислить значение

$$s = \left(2 \cdot \sum_{i=1}^n x_i + \sum_{i=1}^n x_i y_i \right) \cdot \left(2 - \sum_{i=1}^n x_i \right) + 3 + \sum_{i=1}^n x_i^2,$$

где x, y — векторы из n компонентов, причем $n = 4$,

$$x = [1 \quad 2 \quad 7 \quad 4], \quad y = [1 \quad 7 \quad 2 \quad 3].$$

Вариант 8

а) построить матрицу ($n = 5$ и $C = [2 \quad 9 \quad 1 \quad 7 \quad 3]$)

$$G(i, j) = \begin{cases} \sin(C(i) + C(j)), & i \leq j + 1; \\ C(i - j) + 3 \cdot C(i) - 7 \cdot C(j), & i > j + 1; \end{cases}$$

б) вычислить значение

$$s = \left(1 + \sum_{i=1}^m a_i \right)^2 \cdot \left(1 + \sum_{i=1}^m a_i^2 \right) - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \cdot \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 \right),$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 2$, $m = 4$,

$$a = [1 \quad 4 \quad 1 \quad 3], \quad c = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}.$$

Вариант 9

а) построить матрицу ($n = 6$ и $C = [1 \quad 2 \quad 1 \quad 7 \quad 8 \quad 2]$)

$$G(i, j) = \begin{cases} 3 \cdot C(i) + 2 \cdot C(j), & i \leq j + 2; \\ \sin(C(i - j)) + |3 \cdot C(i - j) - 7 \cdot C(j)|, & i > j + 2; \end{cases}$$

б) вычислить значение

$$s = \left(\sum_{i=1}^n x_i^2 + 5 \cdot \sum_{i=1}^n x_i y_i \right) \cdot \left(1 + \sum_{i=1}^n x_i + \sum_{i=1}^n y_i \right) - 3 + \sum_{i=1}^n x_i^2,$$

где x, y — векторы из n компонентов, причем $n = 4$,

$$x = [7 \quad 5 \quad 7 \quad 4], \quad y = [2 \quad 4 \quad 2 \quad 3].$$

Вариант 10

а) построить матрицу ($n = 7$ и $C = [5 \ 1 \ 1 \ 7 \ 1 \ 2 \ 1]$)

$$G(i, j) = \begin{cases} |C(i) - 5 \cdot C(j)|, & i \leq j + 1; \\ C(i - j) + 4 \cdot \sin(C(i)) - 7 \cdot C(j), & i > j + 1; \end{cases}$$

б) вычислить значение

$$s = \left(1 + \sum_{i=1}^m a_i\right)^2 \cdot \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2\right) - 1 - \sum_{i=1}^m a_i^2 + 4 \cdot \sum_{i=1}^n \sum_{j=1}^n c_{ij},$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3$, $m = 4$,

$$a = [2 \ 1 \ 1 \ 3], \quad c = \begin{bmatrix} 5 & 2 & 4 \\ 1 & 4 & 2 \\ 7 & 3 & 5 \end{bmatrix}.$$

Раздел 6. Объекты пользовательского интерфейса

Цель раздела: изучить объекты Excel, используемые в VBA при разработке интерфейса пользователя, и научиться работать с оператором With.

В предыдущем разделе были созданы простейшие VBA-приложения, или макросы (процедуры и функции), работающие в Excel.

Но если требуется создать сколько-нибудь продвинутое приложение с содержательной обработкой данных, то необходимо понять структуру VBA как технологию визуального программирования.

VBA использует идеи объектно-ориентированного программирования, т. е. *методику анализа, проектирования и написания приложений с помощью объектов*.

К основным понятиям VBA относятся объект, метод, свойство, событие.

6.1 Объект

Объект — это конструкция, позволяющая инкапсулировать данные вместе с кодом, предназначенным для их обработки, т. е. объединить их в единое целое. Все визуальные объекты VBA, такие как само приложение Excel (**Application**), рабочий лист (**Worksheet**), диапазон (**Range**), диаграмма (**Chart**), форма (**UserForm**) и др., являются объектами. В VBA имеется более 100 встроенных объектов.

Объекты могут объединяться в семейства. Это тоже объект, но содержащий несколько других объектов, как правило, одного и того же типа. Например, объект **Workbooks** (рабочие книги) содержит все открытые объекты **Workbook** (рабочая книга). Каждый элемент семейства нумеруется и может быть идентифицирован либо по номеру, либо по имени. Например, **Worksheets (1)** обозначает первый рабочий лист активной книги, а **Worksheets ("Лист1")** — рабочий лист с именем Лист1.

Полная ссылка на объект состоит из ряда имен вложенных последовательно друг в друга объектов. Разделителями имен

объектов в этом ряду являются точки. Ряд начинается с объекта **Application** и заканчивается именем самого объекта. Например, полная ссылка на ячейку A1 рабочего листа Лист1 рабочей книги (файла Excel) с именем Книга1 имеет вид:

Application.Workbooks("Книга1").Worksheets("Лист1").Range("A1")

Приводить каждый раз полную ссылку на объект совершенно не обязательно. Обычно достаточно ограничиться только *неявной, или относительной, ссылкой* на объект. В неявной ссылке, в отличие от полной, объекты, которые активны в данный момент, как правило, можно опускать. Например, если ссылка на ячейку A1 дана в программе, выполняемой в среде Excel, то ссылка на объект **Application** может быть опущена, т. е. достаточно привести относительную ссылку:

Workbooks("Книга1").Worksheets("Лист1").Range("A1")

Если рабочая книга Книга1 является активной, то ссылку можно записать еще короче:

Worksheets("Лист1").Range("A1")

Если рабочий лист Лист1 активен, то в относительной ссылке можно ограничиться упоминанием только диапазона A1:

Range("A1")

Примечание. Характеристика некоторых объектов:

Application — представляет собой среду, в которой выполняется приложение Excel. Любое обращение к свойству или методу этого объекта в программе на языке VBA влияет на все приложение Excel.

Workbooks — семейство объектов, в котором находятся объекты **Workbook**. **Workbook** — это открытая рабочая книга Excel, обращение к которой осуществляется через семейство **Workbooks**.

Worksheets — семейство объектов, в котором находятся объекты **Worksheet**. **Worksheet** — это открытые рабочие листы Excel. Обращение к листам осуществляется через семейство **Worksheets**.

Range — содержится в объекте **Worksheet** и представляет собой диапазон (последовательный набор) ячеек рабочего листа.

Borders — семейство объектов, в котором находятся объекты **Border**. **Border** — это объект, который применяется для изменения стиля, цвета и толщины рамок ячейки.

Font — содержится в объекте **Range** и применяется для изменения формата шрифта ячейки.

Interior — содержится в объекте **Range** и применяется для изменения цвета и узора ячейки.

Style — содержится в объекте Range и представляет собой стиль ячейки. Используется для хранения наборов значений свойств объектов Font, Interior и Border. Заданные стили рабочей книги доступны через семейство Styles.

Windows — семейство объектов, в котором находятся объекты Window. Объект Application в семействе Windows содержит все окна приложения, а объект Workbook — только окна рабочей книги. Объект Window используется для изменения параметров окон приложения или рабочих листов.

Объект Application

Объект Application располагается на самой верхней ступени иерархии и представляет само приложение Excel. Объект можно считать средой, в которой выполняются приложения на языке VBA. Объект Application имеет 120 свойств и 40 методов. Основные свойства объекта перечислены ниже, причем свойство только для чтения обозначается буквой R, а для записи — W.

Caption — заголовок для главного окна Excel (R/W);

Path — путь к папке, где установлена программа Excel (R/W);

ScreenUpdating — если True, то обновление экрана при выполнении программы разрешено (R/W);

WindowState — состояние главного окна Excel, принимает три значения: xlNormal, xlMaximized, xlMinimized (R/W);

DisplayStatusBar — если True, то строка состояния видна (R/W);

DisplayFormulaBar — если True, то строка формул видна (R/W).

Основные методы объекта Application следующие:

Calculate — перерасчет всех формул на всех листах всех книг;

Quit — закрывает Excel;

Run — выполняет подпрограмму (макрос) на языке VBA.

Объект Workbook

Объект Workbook расположен на второй ступени иерархии объектов Excel и представляет файл рабочей книги. Обычно одиночное приложение располагается только в одной рабочей книге. Любое обращение к методу или свойству объекта Workbook влияет на все приложение VBA.

Основные свойства объекта следующие:

Name — имя рабочей книги (R);

Path — путь к файлу рабочей книги (R);

Saved — если True, то изменений в книге не было (R/W).

Основные методы объекта Application следующие:

Activate — активизирует рабочую книгу;

Close — закрывает рабочую книгу;

Protect — защищает книгу от редактирования;
 Save — сохраняет рабочую книгу;
 SaveCopyAs — сохраняет рабочую книгу в новом файле.

Объект Worksheet

Объект Worksheet содержится в объекте Workbook и представляет рабочий лист Excel с табличной структурой.

Основные свойства рабочего листа следующие:

Index — номер листа в рабочей книге (R);

Name — имя рабочего листа (R/W);

Visible — если False, то лист является скрытым (R/W);

Основные методы рабочего листа следующие (все R/W):

Activate — активизирует рабочий лист;

Select — активизирует (выбирает) рабочий лист;

Calculate — пересчитывает все формулы на листе;

Delete — удаляет лист из рабочей книги;

Protect — защищает лист от редактирования;

SetBackgroundPicture — загружает указанный графический файл как фоновый рисунок для рабочего листа.

Объект Window

Свойства объект Window позволяют изменять вид окна рабочего листа или всей книги.

Список основных свойств объекта Window, влияющих на отдельный рабочий лист, следующий (все R/W):

DisplayFormulas — если True, то в ячейках отображаются введенные формулы;

DisplayGridlines — если True, то линии сетки ячеек видны;

DisplayHeadings — если True, то показываются заголовки строк и столбцов вдоль левого и верхнего краев листа;

DisplayZeros — если True, то в пустых ячейках видны нули;

ScrollColumn — устанавливает номер самого левого столбца рабочего листа;

ScrollRow — устанавливает номер самой верхней строки рабочего листа.

Список свойств, влияющих на всю рабочую книгу, имеет следующий вид:

Caption — название объекта Window, которое отображается в заголовке окна. По умолчанию хранит название рабочей книги;

DisplayHorizontalScrollBar — если True, то отображается горизонтальная полоса прокрутки;

DisplayVerticalScrollBar — если True, то отображается вертикальная полоса прокрутки;

DisplayWorkbookTabs — если True, то вдоль нижнего края окна видны ярлычки рабочих листов;

Height — высота окна книги в пунктах (1/72 дюйма);

Left, Top — координата верхнего левого угла окна относительно родительского окна Excel;

Width — ширина окна книги в пунктах;

WindowState — состояние окна рабочей книги, принимает три значения: xlNormal, xlMaximized, xlMinimized.

Объект Range

Объект Range возвращает группу указанных ячеек (диапазон) рабочего листа Excel. Ячейки можно форматировать средствами самого объекта Range и объектов, входящих в него. Основные свойства объекта Range перечислены ниже:

Column — номер первого столбца диапазона (R);

ColumnWidth — ширина столбца (R/W);

EntireColumn — возвращает все столбцы диапазона (R);

EntireRow — возвращает все строки диапазона (R);

Font — шрифт диапазона (R);

Height — высота диапазона в пунктах (1/72 дюйма) (R);

Hidden — если True, то скрывает столбец или строку (R/W);

HorizontalAlignment — горизонтальное выравнивание содержимого ячеек (R/W);

Interior — объект форматирования фона ячеек (R);

NumberFormat — формат чисел ячеек (R/W);

Orientation — ориентация содержимого ячеек (R/W);

Row — номер первой строки диапазона (R);

RowHeight — высота всех строк в диапазоне (R/W);

ShrinkToFit — если True, то автоматически настраивает размер шрифта текста под размер ячейки (R/W);

Style — объект, содержащий стиль диапазона ячеек (R);

UseStandartHeight — если True, то высота строк диапазона становится равна стандартной высоте строк всего листа (R/W);

UseStandartWidth — если True, то ширина столбцов диапазона становится равна стандартной ширине столбцов всего листа (R/W);

VerticalAlignment — вертикальное выравнивание содержимого ячеек (R/W);

Width — ширина диапазона в пунктах (R);

WrapText — если True, то текст в ячейках разбивается на несколько строк в каждой ячейке, при этом высота ячейки автоматически увеличивается.

Основные методы объекта Range следующие:

AutoFormat — позволяет применять к диапазону 16 встроенных табличных форматов;

BorderAround — окружает заданный диапазон рамкой;

Borders — семейство рамок диапазона;

Columns — семейство столбцов диапазона;

Justify — равномерно распределяет текст по вертикали (при необходимости переносит его в нижние ячейки);

Merge — соединяет ячейки;

UnMerge — разъединяет ячейки;

Rows — семейство строк диапазона.

Объект Font

Для форматирования шрифта содержимого диапазона ячеек используется объект Font, который входит в состав объекта Range.

Основные свойства объекта Font следующие (все R/W):

Bold — если True, то шрифт имеет полужирный стиль;

Color — цвет шрифта в системе RGB;

ColorIndex — индекс цвета для шрифта, который принимает значения встроенных цветов Excel в интервале от 0 до 56;

FontStyle — задает шрифт четырьмя значениями: Regular, Bold, Italic и Bold Italic;

Italic — если True, то шрифт имеет наклонный стиль;

Name — название шрифта;

Size — размер шрифта в пунктах;

Strikethrough — если True, то текст перечеркивается линией;

Subscript — если True, то текст имеет вид нижнего индекса;

Superscript — если True, то текст имеет вид верхнего индекса;

Underline — подчеркивает текст линией, принимает пять значений: xlNone, xlSingle, xlDouble, xlSingleAccounting, xlDoubleAccounting.

Объект Interior

Объект Interior задает цвет и фон ячеек рабочего листа и имеет следующие свойства (все R/W):

Color — цвет фона ячеек в системе RGB;

ColorIndex — индекс цвета для фона ячеек, который принимает значения встроенных цветов Excel в интервале от 0 до 56;

Pattern — задает один из 20 встроенных фоновых узоров Excel;

PatternColor — цвет фонового узора;

PatternColorIndex — индекс цвета для фонового узора.

Объект Border

Доступен через семейство Borders и применяется для изменения формата границ ячеек. Основные свойства объекта Borders следующие (все R/W):

Color — цвет рамки ячеек в системе RGB;
 ColorIndex — индекс цвета для рамки ячеек;
 LineStyle — задает стиль границы из 8 встроенных стилей Excel;
 Weight — толщина границы, принимает 4 значения: xlHairline, xlThin, xlMedium, xlThick.

Объект Style

Объект Style содержит именованный стиль ячеек и включает в себя следующие свойства: HorizontalAlignment, Orientation, ShrinkToFit, Style, UseStandardHeight, UseStandardWidth, VerticalAlignment, Width, WrapText и др., а также объекты: Font, Interior, Name. Свойство Name позволяет присвоить стилю имя и затем в программе использовать заданный стиль повторно.

6.2 Метод

Объект содержит список методов, которые к нему применимы.

Методы — это то, что вы можете делать с объектом.

Синтаксис применения метода: Объект.Метод

Например,

Application.Quit — закрыть приложение. Применен метод **Quit** (выйти, завершить).

UserForm1.Hide — спрятать (убрать с экрана, но не выгружать из памяти) форму. Применен метод **Hide** (спрятать).

Метод можно применять и ко всем объектам семейства.

Например,

Worksheets("Лист1").ChartObjects.Delete — удалить все диаграммы с рабочего листа Лист1. Применен метод **Delete** (удалить).

6.3 Свойство

Каждый объект обладает некоторыми свойствами.

Свойства определяют его характеристики, такие как размер, цвет, положение на экране и состояние объекта, например доступность или видимость.

Синтаксис установки значения свойства:

Объект.Свойство = ЗначениеСвойства

Чтобы изменить характеристику объекта, надо просто изменить значение его свойства. Например,

Worksheets(2).Range("A1").Font.Name = "Times New Roman" — назначить шрифт Times New Roman для отображения информации в ячейке A1 второго рабочего листа.

Среди свойств особое место занимают *свойства, возвращающие объект*. Их синтаксис иной. Наиболее часто употребляемые подобные объектные свойства приведены в таблице 6.1.

Таблица 6.1 — Свойства, возвращающие объект

Название свойства	Описание объекта
ActiveWindow	Активное окно Excel
ActiveWorkbook	Активная рабочая книга активного окна Excel
ActiveSheet	Активный лист активной книги
ActiveDialog	Активное диалоговое окно активного листа
ActiveChart	Активная диаграмма активного листа
ActiveCell	Активная ячейка активного листа

Например, **Application.ActiveWindow.DisplayGridlines = False** — сделать невидимыми линии сетки ячеек в активном окне Excel. Работает только на том листе (активном), с которого запущен макрос.

Обратите внимание, когда вы набираете этот текст в макросе, то при нажатии каждой точки выпадает подсказка — список возможных методов и свойств, которые можно применить в данном состоянии (рис. 6.1).

Это помогает правильно и быстро записать программную конструкцию.

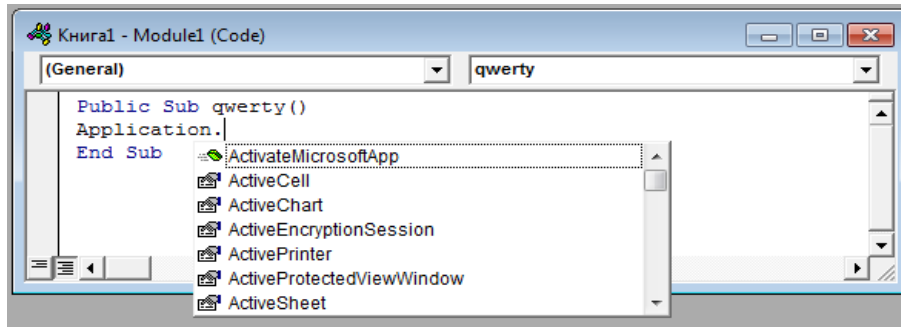


Рис. 6.1 — «Подсказки» при наборе кода VBA

Примечание. Если курсор расположить на ключевом слове языка VBA, имени процедуры, функции, свойства или метода и нажать клавишу F1, то на экране появится окно со справочной информацией об этой функции. Обычно в справке имеется пример использования кода, что позволяет быстрее разобраться в проблемной ситуации.

В окне **Object Browser** (рис. 6.2), которое открывается в Редакторе VBA кнопкой F2 клавиатуры или выбором опции **Object Browser** в меню **View**, можно увидеть список доступных объектов (поле **Classes**) и соответствующие выбранному объекту методы и свойства (поле **Member of...**).

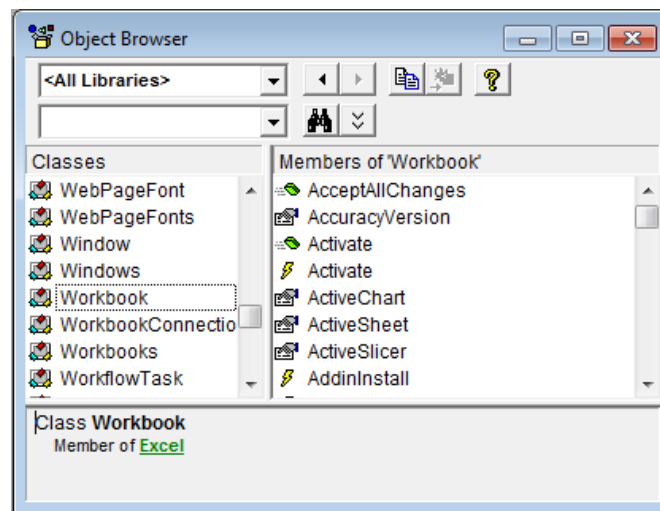


Рис. 6.2 — Окно просмотра объектов

6.4 Событие

Событие представляет собой действие, распознаваемое объектом (например, щелчок мышью или нажатие клавиши), для которого можно запрограммировать процедуру отклика. События возникают в результате действий пользователя или программы. Также они могут быть вызваны системой.

Суть программирования на VBA как раз и заключается в создании процедур обработки различных событий. Если пользователь производит какое-то воздействие на систему (нажимает кнопку), тогда в качестве отклика выполняется код созданной пользователем процедуры. Если такой отклик не создан, т. е. не написана соответствующая процедура, то система никак не реагирует на данное событие.

В разделах 4 и 5 мы уже создавали процедуры отклика нажатия на кнопку — как с помощью макрорекодера, так и самостоятельно.

6.5 Оператор With...End With

Оператор **With...End With** применяется тогда, когда необходимо задать много свойств или выполнить большое количество методов одного объекта. Он позволяет существенно упростить обращение к объекту, заметно сокращает объем кода, делает его понятнее и производительнее.

Рассмотрим две подпрограммы, решающие одну и ту же задачу.

```
'----- Макрос без оператора With -----
Sub NonUseWith()
Worksheets(2).Select
Worksheets(2).Unprotect
Worksheets(2).Range("A1").Font.Bold = True
Worksheets(2).Range("A1").Font.Italic = True
Worksheets(2).Range("A1").Font.Size = 22
Worksheets(2).Range("A1").Font.Name = _
    "Times New Roman"
Worksheets(2).Range("A1").Font.ColorIndex = 3
End Sub
```

```
'----- Макрос с оператором With -----
Sub UseWith()
  With Worksheets(2)
    .Select
    .Unprotect
  With .Range("A1").Font
    .Bold = True
    .Italic = True
    .Size = 22
    .Name = "Times New Roman"
    .ColorIndex = 3
  End With
  End With
End Sub
(проверьте, что выполняют эти макросы).
```

Во втором макросе объем кода значительно меньше, и программа гораздо понятнее. Видно также, что язык VBA допускает применение вложенных операторов With. Понятно, что в случае, когда происходит всего лишь единственное обращение к объекту, нет смысла применять оператор With.

6.6 Создание вспомогательных процедур

Создадим две вспомогательные процедуры без параметров: очистки листа **AllClear** и восстановления начальных параметров **Reset**, которые потребуются в дальнейшем. Для этого в редакторе VBA добавляем модуль с помощью команды **Insert | Module** (см. предыдущий раздел).

1. Процедура очистки листа. Выполним команду **Insert | Procedure**. В открывшемся диалоговом окне укажем: имя макроса «**AllClear**», тип макроса **Подпрограмма (Sub)**, область определения **Общая (Public)**.

Затем в окне редактирования кода модуля запишем следующий код:

```
Public Sub AllClear()
'Обращаемся ко всем ячейкам листа (если надо было бы обратиться
'к диапазону ячеек, следовало бы написать: Range("A1:F40"))
With Cells
'Очищаем содержимое ячеек
```

```

.Clear
'Восстанавливаем стандартную высоту строк
.UseStandardHeight = True
' Восстанавливаем стандартную ширину столбцов
.UseStandardWidth = True
' Восстанавливаем разъединенные строки
.UnMerge
End With
'В цикле очищаем лист от всех объектов
For Each i In ActiveSheet.Shapes
i.Delete
Next i
End Sub

```

2. Процедура восстановления начальных параметров листа Excel.

В уже открытом окне редактирования кода модуля запишем следующий код:

```

Public Sub Reset()
'Выбираем первый рабочий лист
Worksheets(1).Select
With Application
'Показать строку состояния
.DisplayStatusBar = True
'Показать строку формул
.DisplayFormulaBar = True
End With
With ActiveWindow
'Показать заголовки строк и столбцов
.DisplayHeadings = True
'Показать горизонтальную полосу прокрутки
.DisplayHorizontalScrollBar = True
'Показать вертикальную полосу прокрутки
.DisplayVerticalScrollBar = True
'Показать ярлычки рабочих листов
.DisplayWorkbookTabs = True
End With
End Sub

```

Пример 6.1. Создать макрос заливки всех листов рабочей книги Excel сплошным цветом.

Как и ранее, выполним команду **Insert | Procedure**. В открывшемся диалоговом окне укажем имя макроса «Цвет_книги», тип макроса Подпрограмма (**Sub**), область определения **Общая (Public)**. Наберем следующий код:

```
Public Sub Цвет_книги()
'Цикл изменения номера листа рабочей книги от первого до последнего
For i = 1 To Worksheets.Count
'Выбираем рабочий лист под номером «i»
Worksheets(i).Select
'Обращаемся к ячейкам листа
With Cells
'Закрашиваем ячейки цветом под номером 11
.Interior.ColorIndex = 11
'Выдаем информационное сообщение о номере закрашиваемого листа
MsgBox "Лист № " & i
End With
Next i
End Sub
```

Затем выйдем из окна редактирования кода и запустим макрос. Все листы рабочей книги закрасятся одним цветом.

Запуском макроса AllClear цвет листов возвращается в первоначальное состояние.

Параметр ColorIndex может принимать значения от 1 до 56. Базовым цветам соответствуют номера с 1 по 8:

- 1 — черный;
- 2 — белый;
- 3 — красный;
- 4 — зеленый;
- 5 — синий;
- 6 — желтый;
- 7 — фиолетовый;
- 8 — бирюзовый.

Остальные номера соответствуют различным оттенкам базовых цветов.

6.7 Функции MsgBox и InputBox

В коде макроса использована функция вывода MsgBox, которая выводит информационное сообщение в диалоговом окне и устанавливает режим ожидания нажатия кнопки пользователем.

Она имеет следующий синтаксис:

MsgBox Сообщение, Кнопки, Заголовок

Аргументы:

Сообщение — обязательный аргумент, задающий в окне выводимое информационное сообщение. Может состоять из нескольких текстовых строк, переменных, функций, объединенных знаком *конкатенации* (сцепки) **&**. Использование в этом аргументе функции **Chr(13)** приводит к переходу на новую строку при выводе информации.

Кнопки — необязательный аргумент; определяет категории появляющихся в окне кнопок. По умолчанию соответствует появлению кнопки ОК. В табл. 6.1 приведены возможные значения аргумента.

Заголовок — необязательный аргумент; задает заголовок окна (по умолчанию «Microsoft Excel»).

Например, выполнение оператора

MsgBox "Сегодня у нас: " & Chr(13) & "Дата и время (функция Now): " _

& Now & Chr(13) & "Дата (функция Date): " & Date & Chr(13) & _

"Системное время (функция Time): " & Time, vbMsgBoxHelpButton, _ "Внимание"

приведет к появлению сообщения:

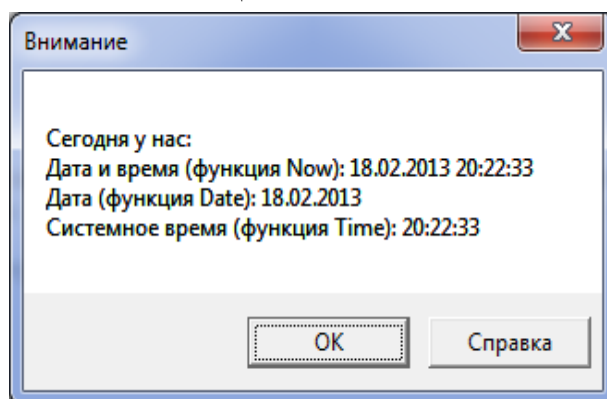


Рис. 6.3 — Пример сообщения MsgBox

Таблица 6.1 — Допустимые значения аргумента **Кнопки**

Отображение	Аргумент
Кнопка ОК	VbOKOnly
Кнопки ОК и Отмена	VbOKCancel
Кнопки Да и Нет	VbYesNo
Кнопки Да, Нет и Отмена	VbYesNoCancel
Кнопки Да и Справка	VbMsgBoxHelpButton
Кнопки Прекратить, Повторить и Пропустить	VbAbortRetryIgnore
Кнопки Повторить и Отмена.	VbRetryCancel
Информационный знак	VbInformation
Знак запрета (косой крест)	VbCritical
Знак вопроса	VbQuestion
Знак восклицания	VbExclamation

При таком синтаксисе функции важно соблюсти порядок следования аргументов, запятыми указывая место аргумента при его отсутствии.

Но при использовании метода именованных аргументов допускается произвольный порядок:

MsgBox (Prompt [, Buttons] [, Title] [, HelpFile, Context])

Единственным обязательным аргументом является аргумент **Prompt**, который может быть любым строковым выражением и содержит информацию, которая выводится в диалоговом окне. Все остальные аргументы — необязательные:

Title — строка для заголовка диалогового окна.

HelpFile — строковое выражение, содержащее имя справочного файла Windows (который уже должен быть создан).

Context — численное выражение, указывающее раздел в справочном файле, относящийся к отображаемому диалоговому окну.

Buttons — численное выражение, которое задает количество и тип кнопок в диалоговом окне.

Например, **MsgBox Buttons:=VbYesNo, Title := "Заголовок", Prompt := "Сообщение"**

Обратите внимание, если пропущены круглые скобки, то функция MsgBox получает значение (см. табл. 6.2), которое может быть использовано для обработки события нажатия на ту или иную кнопку.

Каждой кнопке, помещенной в поле информационного сообщения, можно поставить в соответствие макрос, обрабатывающий событие (нажатие на «Да», «Нет» и т. п.).

Таблица 6.2 — Возвращаемые значения-константы функции MsgBox

Действие	Константа	Числовое значение
Пользователь выбирает кнопку ОК	vbOK	1
Пользователь выбирает кнопку Отмена	vbCancel	2
Пользователь выбирает кнопку Прервать	vbAbort	3
Пользователь выбирает кнопку Повтор	vbRetry	4
Пользователь выбирает кнопку Пропустить	vbIgnore	5
Пользователь выбирает кнопку Да	vbYes	6
Пользователь выбирает кнопку Нет	vbNo	7

Функция **InputBox** используется для ввода значений переменных с помощью окна ввода.

Она имеет следующий синтаксис:

Имя_Переменной = InputBox(Сообщение, Заголовок, Значение по умолчанию)

Ее аргументы:

Сообщение — обязательный аргумент; задает в окне информационное сообщение, обычно поясняющее смысл вводимой величины.

Заголовок — необязательный аргумент; задает заголовок окна;

Значение по умолчанию — необязательный аргумент.

Например, ввести значение переменной N с клавиатуры, предусмотрев значение по умолчанию равное 1024:

N = InputBox("Введите N", "Ввод исходных данных", 10)

В результате появится окно для ввода значения переменной N (рис. 6.4).

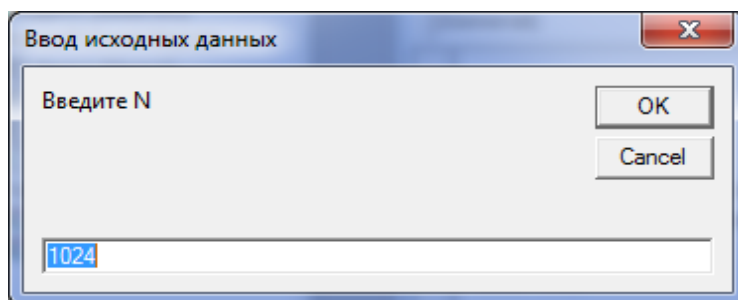


Рис. 6.4 — Окно функции InputBox

Если значение по умолчанию подходит пользователю, то после щелчка кнопки ОК окно ввода закроется, переменной N присвоится значение 1024 и выполнение программы возобновится с оператора, стоящего непосредственно за вызовом InputBox.

Если же значение по умолчанию не подходит пользователю, то перед щелчком по кнопке ОК необходимо ввести нужное значение переменной N.

Задание 6.1. Закраска листов

1. Открыть новый файл Excel.
2. Написать макрос, добавляющий к имеющимся трем листам еще 5 листов и закрашивающий все 8 листов в различные базовые цвета, выдавая перед каждой окраской информационное сообщение с названием цвета листа.

6.8 Создание кнопок

Выберем лист под номером 1 и изменим его название на «Кнопка». Для этого нужно кликнуть по закладке листа внизу экрана правой клавишей и в появившемся меню выбрать пункт «Переименовать».

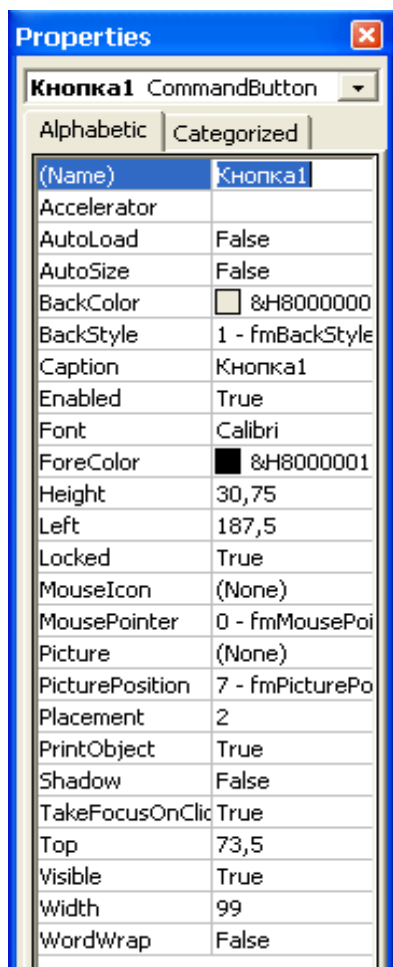


Рис. 6.5 — Окно свойств кнопки

В меню **Разработчик** вызвать опцию **Вставить**, выбрать **Элементы ActiveX**, щелкнуть по элементу **Кнопки** одновременно рядом с кнопкой **Вставить**.

Создадим кнопку, для этого щелкнем на рабочем листе левой кнопкой мыши и потянем указатель так, чтобы рамка кнопки занимала площадь четырех ячеек.

Вызвав контекстно-зависимое меню, откроем окно свойств (опция **Свойства**) для кнопки **CommandButton1**. Изменим свойства **Name** и **Caption**, присвоив значение **Кнопка1** (рис. 6.5).

Выберем в контекстно-зависимом меню опцию **Исходный текст**. Появится окно модуля рабочего листа и одновременно сгенерируется тело процедуры отклика для события **Click** (щелчок левой кнопкой мыши).

Внесем в тело процедуры код, выводящий на экран соответствующее информационное сообщение:

```
Private Sub Кнопка1_Click()
```

```
    'Выводим на экран подтверждающее сообщение
```

```
    MsgBox "Щелчок левой кнопкой мыши"
```

```
End Sub
```

Аналогично создадим элемент управления **Кнопка2**.

Поместим на **Кнопку2** рисунок, для этого выберем свойство **Picture** (рис. 6.5). Щелчок по полю справа вызывает появление кнопки с тремя точками, нажатие на которую, в свою очередь, вызывает появление окна загрузки рисунка. Следует подбирать небольшие рисунки, например, «иконки» (файлы с расширением .ico).

Свойству **PicturePosition** (размещение рисунка на кнопке) присвоим значение **1-fmPicturePositionLeftCenter** (слева по центру). На кнопке появится рисунок и надпись (рис. 6.6).

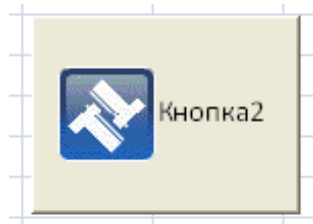


Рис. 6.6 — Кнопка с рисунком

Для этой (второй) кнопки запрограммируем отклик на событие **DbClick** (двойной щелчок). Для этого перейдем в окно редактирования кода (**Исходный текст**). В левом верхнем списке окна выберем объект Кнопка2, а в правом списке — событие DbClick (рис. 6.7).

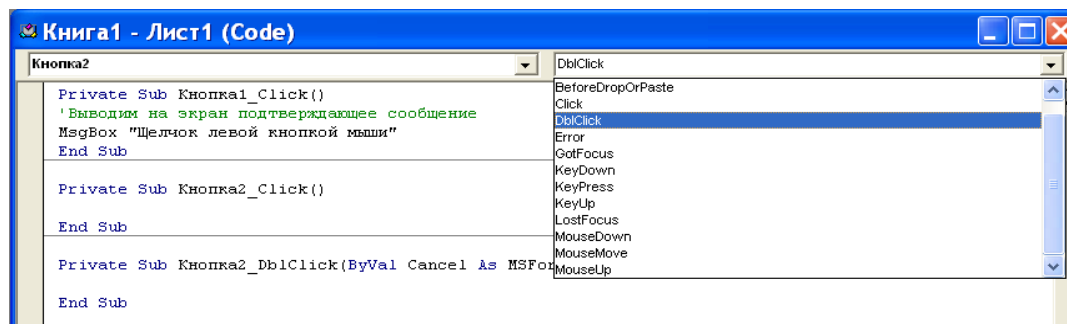


Рис. 6.7 — Левое и правое поле заголовка окна кода кнопок

Заголовок процедуры приобретет вид:

Private Sub Кнопка2_DbClick(ByVal Cancel As MSForms.ReturnBoolean)

Запишем в тело процедуры код:

'Выводим на экран подтверждающее сообщение

MsgBox "Двойной щелчок"

Закрыв Режим конструктора, проверим работу кнопок.

6.9 Форматирование листов

Пример 6.1. Написать *НастроитьРазмерЯчеек*, который в цикле изменяет одновременно ширину и высоту столбца *B* и строки 5 от нуля до размера 40 пикселей с шагом 5 пикселей. Выдать информационное сообщение о каждом изменении.

Откроем новый файл и создадим макрос с помощью команды **Insert | Procedure**. В открывшемся диалоговом окне укажем тип макроса **Подпрограмма (Sub)** и область определения **Общая (Public)**.

Затем в окне редактирования кода модуля запишем следующий код:

```
Public Sub НастроитьРазмерЯчеек()
'Объявляем переменную «Размер_Ячейки» целого типа для исполь-
зования ее как счетчика оператора цикла
Dim Размер_Ячейки As Integer
'Выбираем первый рабочий лист
Worksheets(1).Select
'Выделяем ячейку B5
Range("B5").Select
With Selection
'Запускаем цикл изменения свойств
For Размер_Ячейки = 0 To 40 Step 5
'Задаем ширину столбца
.ColumnWidth = Размер_Ячейки
'Задаем высоту строки
.RowHeight = Размер_Ячейки
'Выводим на экран сообщение о ширине и высоте ячейки
MsgBox "Ширина и высота ячейки B5 = " & Размер_Ячейки
Next
End With
'Выводим на экран сообщение о завершении работы
MsgBox "Действие закончено"
'Вызываем процедуру очистки листа и восстановления начальных
значений ячейки
AllClear
End Sub
```

Пример 6.2. Написать два макроса *СкрытьЯчейки* и *ВосстановитьЯчейки*, которые позволяют прятать и показывать соответственно нужный столбец для заданной ячейки **B1**.

Аналогично первому заданию создадим макрос **Скрыть-Ячейки**.

```
Public Sub СкрытьЯчейки()
'Выбираем первый рабочий лист
Worksheets(1).Select
```

```

'Вызываем процедуру очистки листа
AllClear
'Обращаемся к столбцу B
Range("B:B").Activate
'Обращаемся к ячейке B1
With Range("B1")
    'Вводим в ячейку B1 значение «Текст»
    .Value = "Текст"
    'Копируем значение в диапазон ячеек «B1:B10»
    .AutoFill Range ("B1:B10")
    'Выводим на экран подтверждающее сообщение
    MsgBox "Данные введены в первые 10 ячеек столбца B"
    'Скрываем столбец
    .EntireColumn.Hidden = True
    'Выводим на экран подтверждающее сообщение
    MsgBox "Столбец скрыт"
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
End Sub

```

Теперь создадим макрос **ВосстановитьЯчейки**.

```

Public Sub ВосстановитьЯчейки()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Обращаемся к ячейке B1 столбца B
With Range("B1")
    'Проверяем, показан ли столбец, в котором мы находимся
    If .EntireColumn.Hidden = False Then
        'Выводим на экран подтверждающее сообщение
        MsgBox ("Столбец уже показан")
    Else
        'Показываем столбец
        .EntireColumn.Hidden = False
        'Выводим на экран подтверждающее сообщение
        MsgBox "Столбец показан"
    End If
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
End Sub

```

Пример 6.3. Написать макрос *УстановитьГоризонтальное Выравнивание*, в котором последовательно перебрать все типы горизонтального выравнивания текста в заданной ячейке **B1**.

```
Public Sub УстановитьГоризонтальноеВыравнивание()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Вызываем процедуру очистки листа
AllClear
'Обращаемся к ячейке B1
With Range("B1")
    'Задаем выделенному диапазону значение
    .Value = "Кашу маслом не испортишь"
    'Используем выравнивание xlGeneral
    .HorizontalAlignment = xlGeneral
    MsgBox "Используется выравнивание xlGeneral"
    'Устанавливаем ширину столбца
    .ColumnWidth = 50
    MsgBox "Установлена ширина столбца=50"
    'Используем выравнивание xlCenter
    .HorizontalAlignment = xlCenter
    MsgBox "Используется выравнивание xlCenter"
    'Используем выравнивание xlRight
    .HorizontalAlignment = xlRight
    MsgBox "Используется выравнивание xlRight"
    'Используем выравнивание xlLeft
    .HorizontalAlignment = xlLeft
    MsgBox "Используется выравнивание xlLeft"
    'Устанавливаем стандартную ширину столбца
    .UseStandardWidth = True
    'Используем выравнивание xlJustify
    .HorizontalAlignment = xlJustify
    MsgBox "Используется выравнивание xlJustify"
    'Используем выравнивание xlCenterAcrossSelection
    Range("B1").HorizontalAlignment = xlCenterAcrossSelection
    MsgBox "Используется выравнивание xlCenterAcrossSelection"
    'Устанавливаем ширину столбца
    .ColumnWidth = 100
    MsgBox "Установлена ширина столбца=100"
    'Используем выравнивание xlFill
    .HorizontalAlignment = xlFill
    MsgBox "Выравнивание xlFill"
```

```

End With
MsgBox "Действие закончено"
'Вызываем процедуру очистки листа
AllClear
End Sub

```

Пример 6.4. Написать макрос **УстановитьОриентацию**, в котором последовательно перебрать все типы ориентации текста в заданной ячейке **B1**.

```

Public Sub УстановитьОриентацию ()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Обращаемся к ячейке B1
With Range("B1")
    'Задаем ширину ячейки (и столбца в целом)
    .ColumnWidth = 30
    'Присваиваем диапазону значение
    .Value = "Долг платежом красен"
    'Используем выравнивание xlHorizontal
    .Orientation = xlHorizontal
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlHorizontal"
    'Задаем высоту строки
    .RowHeight = 150
    'Используем выравнивание xlVertical
    .Orientation = xlVertical
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlVertical"
    'Используем выравнивание xlUpward
    .Orientation = xlUpward
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlUpward"
    'Используем выравнивание xlDownward
    .Orientation = xlDownward
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlDownward"
    'Используем ориентацию 30 градусов
    .Orientation = 30
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание 30 град."
    'Используем ориентацию -78 градусов
    .Orientation = -78
    'Выводим на экран подтверждающее сообщение

```

```

    MsgBox "Используется выравнивание –78 град."
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
'Вызываем процедуру очистки листа
AllClear
End Sub

```

Пример 6.5. Написать макрос *НастроитьРазмерТекста*, в котором последовательно присваивать заданной ячейке B1 строковые предложения разной длины при установленной автоматической настройке размера шрифта.

```

Public Sub НастроитьРазмерТекста ()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Настраиваем размер шрифта
Worksheets(1).Range("B1").Font.Size = 12
'Обращаемся к столбцу B
With Range("B1")
    'Устанавливаем ширину столбца
    .ColumnWidth = 20
    'Автоматически настраиваем размер шрифта текста под размер
ячейки
    .ShrinkToFit = True
    'Присваиваем диапазону значение
    .Value = "aaaaaaaaaaaaaaaaaaaaa"
    MsgBox "Сообщение из 20 букв"
    'Присваиваем диапазону значение
    .Value = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
    MsgBox "Сообщение из 30 букв"
    'Присваиваем диапазону значение
    .Value = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
    MsgBox "Сообщение из 40 букв"
    'Присваиваем диапазону значение
    .Value = "aaaaaaaaaa"
    MsgBox "Сообщение из 10 букв"
    'Выводим на экран подтверждающее сообщение
    MsgBox "Вывод текста окончен"
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
'Вызываем функцию очистки листа

```

AllClear
End Sub

Задание 6.2. Работа с *InputBox*

Изменить код макроса примера 5 таким образом, чтобы:

- 1) текст сообщения предлагалось ввести в окне ввода (**InputBox**),
- 2) число букв в тексте сообщения определялось программно,
- 3) это число появлялось в информационном сообщении **MsgBox** как переменная величина.

Пример 6.6. *Написать макрос **УстановитьВертикальноеВыравнивание**, в котором последовательно перебрать все типы вертикального выравнивания текста в заданной ячейке **B1**.*

```
Public Sub УстановитьВертикальноеВыравнивание ()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Обращаемся к столбцу В
With Range("B1")
    'Ширина столбца
    .ColumnWidth = 20
    'Высота строк в диапазоне
    .RowHeight = 50
    'Присваиваем диапазону значение
    .Value = "Кашу маслом не испортишь"
    'Используем выравнивание xlCenter
    .VerticalAlignment = xlCenter
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlCenter"
    'Используем выравнивание xlBottom
    .VerticalAlignment = xlBottom
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlBottom"
    'Используем выравнивание xlTop
    .VerticalAlignment = xlTop
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlTop"
    'Используем выравнивание xlJustify
    .VerticalAlignment = xlJustify
    'Выводим на экран подтверждающее сообщение
    MsgBox "Используется выравнивание xlJustify"
```

End With

'Выводим на экран подтверждающее сообщение

MsgBox "Действие закончено"

'Вызываем функцию очистки листа

AllClear

End Sub

Пример 6.7. *Написать два макроса **СоединениеЯчеек** и **РазъединениеЯчеек** для заданного диапазона **B1:B3**.*

Public Sub СоединениеЯчеек ()

'Выбираем первый рабочий лист

Worksheets(1).Select

'Определяем диапазон

Range("B1:B3").Select

'Обращаемся к диапазону

With Selection

'Задаем ширину столбца «В»

.ColumnWidth = 20

'Соединяем ячейки

.Merge

'Используем выравнивание xlCenter

.HorizontalAlignment = xlCenter

'Присваиваем диапазону значение

.Value = "Ячейки объединены"

'Выводим на экран подтверждающее сообщение

MsgBox "Объединено три ячейки столбца В"

End With

'Выводим на экран подтверждающее сообщение

MsgBox "Действие закончено"

End Sub

Теперь запишем код макроса **Разъединение ячеек**.

Public Sub РазъединениеЯчеек ()

'Выбираем первый рабочий лист

Worksheets(1).Select

'Выделяем диапазон с объединенными ячейками

Range("B1:B3").Select

'Обращаемся к диапазону

With Selection

'Разъединяем ячейки

.UnMerge

'Используем выравнивание xlCenter

```

.HorizontalAlignment = xlCenter
'Присваиваем диапазону значение
.Value = "Ячейки разъединены"
'Выводим на экран подтверждающее сообщение
MsgBox "Разъединение закончено"
'Очищаем содержимое ячеек диапазона
.Clear
End With
Range(«A1»).Select
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
AllClear
End Sub

```

Пример 6.8: *Написать макрос УстановитьЦветТекста, в котором последовательно перебрать 10 цветов, созданных с помощью функции RGB и генератора случайных чисел.*

```

Public Sub УстановитьЦветШрифта ()
'Объявляем переменные целого типа
Dim i, Красный, Синий, Зеленый As Integer
'Выбираем первый рабочий лист
Worksheets(1).Select
'Выделяем четыре ячейки первой строки
Range("B1:E1").Select
'Обращаемся к ячейке B1
With Range("B1")
'Присваиваем ячейке значение
.Value = "Кашу маслом не испортишь"
'Задаем размер шрифта
.Font.Size = 12
'Задаем полужирное начертание
.Font.Bold = True
'Цикл случайного изменения цвета текста
For i = 1 To 10
Красный = Int(255 * Rnd)
Зеленый = Int(255 * Rnd)
Синий = Int(255 * Rnd)
'Определяем значение цвета
.Font.Color = RGB(Красный, Зеленый, Синий)
MsgBox "Случайное изменение цвета шрифта"
Next
'Выводим на экран подтверждающее сообщение

```

```

MsgBox "Изменение цвета шрифта окончено"
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
'Вызываем функцию очистки листа
AllClear
End Sub

```

Пример 6.9. Написать макрос *УстановитьШрифт*, в котором последовательно перебрать разные форматы шрифта.

```

Public Sub УстановитьШрифт ()
'Выбираем первый рабочий лист
Worksheets(1).Select
'Выделяем четыре ячейки первой строки
Range("B1:E1").Select
'Обращаемся к ячейке B1
With Range("B1")
'Присваиваем ячейке значение
.Value = "Кашу маслом не испортишь"
With .Font
'Задаем название шрифта
.Name = "Times New Roman"
'Задаем полужирное начертание
.Bold = True
'Задаем размер шрифта
.Size = 26
'Задаем наклонное начертание
.Italic = True
'Выводим на экран подтверждающее сообщение
MsgBox «"Использование шрифта Times New Roman
полужирного курсивного размером 26 п. "
'Нижний индекс
.Subscript = True
'Выводим на экран подтверждающее сообщение
MsgBox "Написание нижним индексом"
'Верхний индекс
.Superscript = True
'Выводим на экран подтверждающее сообщение
MsgBox "Написание верхним индексом"
'Отключение верхнего индекса
.Superscript = False
'Текст перечеркивается линией
.Strikethrough = True

```

```

'Выводим на экран подтверждающее сообщение
MsgBox "Написание с перечеркиванием"
'Текст не перечеркивается линией
.Strikethrough = False
'Подчеркивает текст линией
.Underline = xlSingle
'Выводим на экран подтверждающее сообщение
MsgBox "Написание с одинарным подчеркиванием"
'Подчеркиваем текст двойной линией
.Underline = xlDouble
'Выводим на экран подтверждающее сообщение
MsgBox "Написание с двойным подчеркиванием"
End With
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
'Вызываем функцию очистки листа
AllClear
End Sub

```

Пример 6.10. Написать макрос преобразования чистого рабочего листа в титульный лист рабочей книги Вашей компании. В макросе применить операции по изменению цвета фона, формата текста и границ ячеек. Скройте ярлычки рабочих листов, полосы прокрутки листа, строки состояния и формул, сетку ячеек и заголовки строк и столбцов.

```

Public Sub Задание10 ()
'Выбираем второй рабочий лист
Worksheets(2).Select
With Application
'Не видна строка состояния
.DisplayStatusBar = False
'Не видна строка формул
.DisplayFormulaBar = False
End With
With ActiveWindow
'Не видны линии сетки ячеек
.DisplayGridlines = False
'Не показываются заголовки строк и столбцов
.DisplayHeadings = False
'Не отображается горизонтальная полоса прокрутки
.DisplayHorizontalScrollBar = False

```

```

'Не отображается вертикальная полоса прокрутки
.DisplayVerticalScrollBar = False
'Не видны ярлычки рабочих листов
.DisplayWorkbookTabs = False
'Очищаем содержимое ячеек
Cells.Clear
'В цикле очищаем лист от всех объектов
For Each i In ActiveSheet.Shapes
  i.Delete
Next i
'Размещаем кнопку на листе
ActiveSheet.Buttons.Add(200, 200, 80, 15).Characters.Text =
"Вернуться на Лист 1"
'По нажатию на кнопку вызывается макрос «reset»
ActiveSheet.Shapes(1).OnAction = "reset"
'Обращаемся к ячейке «E5»
With Range("E5")
  'Равномерное распределение текста по вертикали
  .Justify
  'Присваиваем диапазону значение
  .Value = "ТРЕКОМ"
  With .Font
    'Размер шрифта
    .Size = 30
    'Полужирное начертание
    .Bold = True
    'Наклонное начертание
    .Italic = True
    'Подчеркивание двойной линией
    .Underline = xlDouble
  End With
End With
'Обращаемся к ячейке «D10»
With Range("D10")
  'Равномерное распределение текста по вертикали
  .Justify
  'Присваиваем диапазону значение
  .Value = "Трейдинговая компания"
  With .Font
    'Размер шрифта
    .Size = 15
    'Полужирное начертание
    .Bold = True

```

```

'Наклонное начертание
.Italic = True
'Подчеркивание линией
.Underline = xlDouble
End With
End With
With Cells
    'Цвет шрифта
    .Font.ColorIndex = 5
    .Interior.ColorIndex = 25
End With
'Выводим на экран подтверждающее сообщение
MsgBox "Действие закончено"
End With
End Sub

```

Затем поместим на листе 2 Excel элемент **Группа (Вставить | Элементы управления формы | Группа)** и внутри нее расположим 12 кнопок (рис. 6.8). Названия кнопок дать в соответствии с названиям макросов.

Протестировать работу каждого макроса.

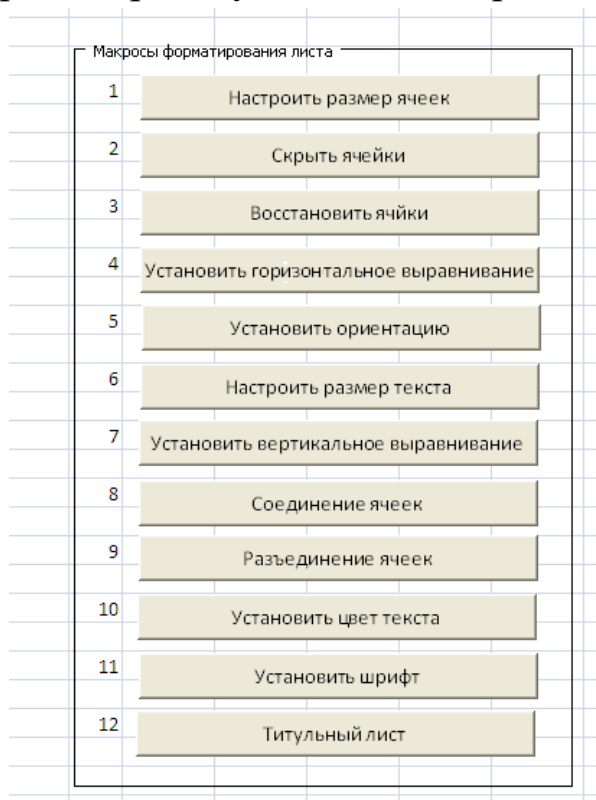


Рис. 6.8 — Сгруппированные кнопки запуска макросов

Задание 6.3. Формирование титульного листа

1. Создать макрос, оформляющий титульный лист для лабораторной работы по разделу 6.

2. Поместить на титульном листе кнопку перехода к содержанию, в котором происходит вызов всех вышеописанных макросов (примеры 1—9). Пример 5 должен быть с измененным кодом.

Раздел 7. Элементы управления

Цель раздела: изучить элементы управления VBA Excel, их свойства и методы; научиться создавать рабочие листы с использованием элементов управления.

7.1 Формы

В Microsoft Excel с использованием VBA (или встроенных возможностей) можно создавать замечательные формы и добавлять на них многочисленные элементы управления и объекты, значительно расширяющие возможности ввода данных и их представления.

Форма — это документ, имеющий стандартную структуру и формат, упрощающий сбор, упорядочение и изменение данных.

В Excel можно создавать формы нескольких типов:

- формы данных (см. п.3.9),
- листы с элементами управления формы и элементами ActiveX (то есть сам лист будет являться формой),
- пользовательские формы VBA.

Каждый из этих типов формы можно использовать по отдельности или сочетать с другими типами для создания нужного решения.

Сам лист Excel можно считать формой в виде сетки, позволяющей вводить и просматривать данные. В листы Excel по умолчанию встроены возможности, аналогичные элементам управления, например примечания и средства проверки данных. Ячейки, как и текстовые поля, поддерживают различные способы ввода и форматирования данных. Ячейки часто используются в качестве подписей, а путем настройки высоты и ширины ячеек, а также их объединения можно сделать лист похожим на простую форму для ввода данных. Другие возможности, например примечания к ячейкам, гиперссылки, фоновые изображения, проверка данных, условное форматирование, встроенные диаграммы и автофильтр, рассмотренные ранее, позволяют использовать лист в качестве формы с расширенными функциями.

Для обеспечения гибкости управления на полотно листа добавляют элементы управления и другие графические объекты. В Excel поддерживаются **два типа элементов управления**: элементы управления форм и элементы ActiveX. Кроме того, можно добавлять объекты из средств рисования, например автофигуры, графические элементы SmartArt и надписи.

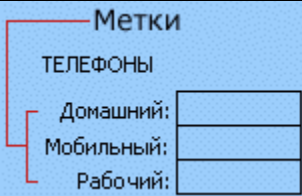
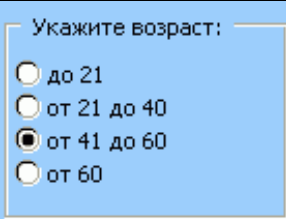
7.2 Элементы управления формы

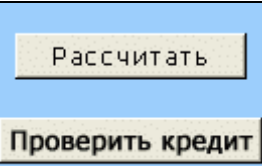
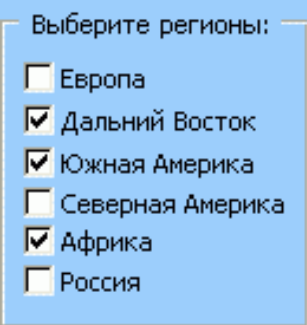
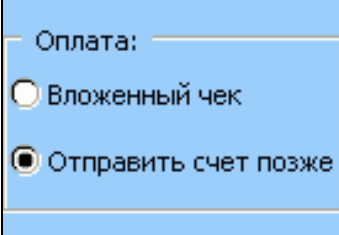
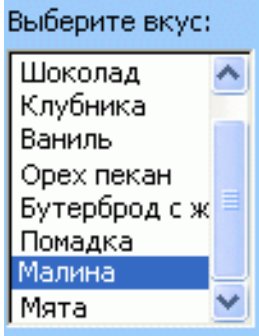
Элементы управления формы (табл. 7.1) позволяют ссылаться на данные ячеек и взаимодействовать с ними **без использования программного кода VBA**. Кроме того, их можно добавлять на листы диаграмм. Например, после добавления списка на лист и его связи с ячейкой можно возвращать числовое значение текущей позиции выбранного элемента. Затем можно использовать это значение с функцией **ИНДЕКС** для выбора других элементов из списка.

С помощью элементов управления формы можно также выполнять макросы. Можно назначить элементу управления существующий макрос либо создать или записать новый. Когда пользователь формы щелкает элемент управления, запускается макрос.

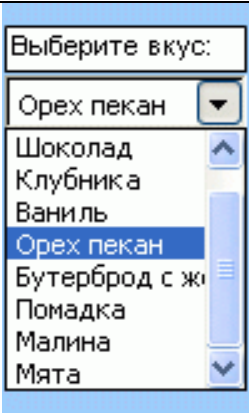
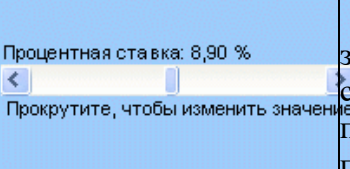
Но такие элементы управления невозможно добавлять в пользовательские формы, а также применять для выполнения веб-скриптов на веб-страницах.

Таблица 7.1 — Элементы управления формы

Элемент	Название элемента	Пример	Описание
	Надпись		Указывает назначение ячейки или текстового поля либо содержит описание (заголовки, подписи, рисунки) или краткие инструкции
	Рамка		Группирует связанные элементы управления в виде прямоугольника с необязательной подписью. Обычно группируются переключатели, флажки или тесно связанные данные

Элемент	Название элемента	Пример	Описание
	Кнопка		Запуск макроса, который выполняет действие при нажатии кнопки
	Флажок		Включает или отключает значение, что соответствует выбору противоположных действий. На листе или в группе можно установить несколько флажков. Флажок может иметь одно из трех состояний: установлен (значение включено), снят (значение отключено) и смешенное состояние, то есть сочетание состояний «включено» и «отключено» (если поддерживается выбор нескольких элементов)
	Переключатель		Позволяет выбрать одно из ограниченного набора взаимоисключающих значений. Переключатели обычно находятся в группе или рамке. Переключатель может иметь одно из трех состояний: установлен (значение включено), снят (значение отключено) и смешенное состояние, т. е. сочетание состояний «включено» и «отключено» (если поддерживается выбор нескольких элементов)
	Список		Выводит список из одного или нескольких текстовых элементов, которые может выбрать пользователь. Список можно применять для представления большого числа вариантов, количество и содержимое которых может быть различным. Существует три типа списков: • Простой список поддерживает выбор только одного элемента. Такой список напоминает группу переключателей, но позволяет более эффективно работать с большим числом элементов. • Список связанного выбора позволяет выбрать один элемент, а также несколько расположенных рядом элементов. • Список, разрешающий несвязный выбор нескольких строк, позволяет выбрать один элемент, расположенные рядом элементы, разрозненные элементы

Окончание табл. 7.1

Элемент	Название элемента	Пример	Описание
	Поле со списком		Представляет собой сочетание текстового поля и раскрывающегося списка. Поле со списком компактнее обычного списка, но чтобы отобразить список элементов, пользователь должен щелкнуть стрелку вниз. Поле со списком следует использовать, когда требуется обеспечить возможность ввода элементов в список и выбора из него одного элемента. В этом элементе управления отображается текущее значение независимо от того, каким образом оно было введено
	Полоса прокрутки		Используется для прокрутки диапазона значений с помощью кнопок со стрелками или путем перетаскивания ползунка полосы прокрутки. Можно перемещаться по странице значений (с заранее заданным интервалом), щелкая область между ползунком и одной из кнопок со стрелками для прокрутки. Как правило, пользователь также может вводить текст непосредственно в связанную ячейку или текстовое поле
	Счетчик		Позволяет увеличивать и уменьшать значение, например числовое значение, время или дату. Чтобы увеличить значение, щелкните стрелку вверх, чтобы уменьшить — стрелку вниз. Как правило, пользователь также может вводить текст непосредственно в связанную ячейку или текстовое поле

Примечание. Элементы управления  (Текстовое поле),  (Поле со списком) и  (Поле с раскрывающимся списком) недоступны в книгах Excel 2010 (см. рис 4.7).

7.3 Элементы ActiveX

Элементы ActiveX (табл. 7.2) можно использовать на листе Excel (без кода VBA или с ним), а также на пользовательских формах VBA. Их рекомендуется применять, если требуются более гибкие возможности, чем у элементов управления формы.

Многочисленные свойства элементов ActiveX позволяют настраивать их внешний вид, поведение, шрифты и другие характеристики.

Можно также управлять событиями, которые происходят при взаимодействии с элементом ActiveX. Например, можно выполнять различные действия в зависимости от того, какой элемент выбирается из списка, или отправлять запрос базе данных для заполнения поля со списком элементами при нажатии кнопки. Можно также создавать макросы, которые реагируют на события, связанные с элементами ActiveX. При выборе элемента управления будет выполняться код VBA, обрабатывающий все назначенные ему события.

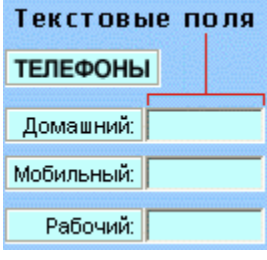
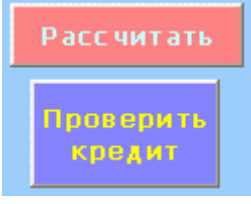
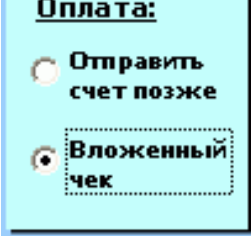
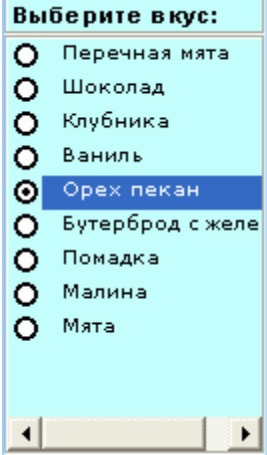
Примечание. Не все элементы ActiveX можно использовать непосредственно на листах; некоторые из них можно применять только в пользовательских формах Visual Basic для приложений (VBA). При попытке добавить такие элементы управления на лист приложение Excel отображает сообщение «Вставка объекта неосуществима».

Элементы ActiveX невозможно добавить на листы диаграмм (с помощью пользовательского интерфейса) или на листы макросов XLM. Кроме того, невозможно назначить макрос, который будет выполняться непосредственно элементом ActiveX, как для элемента управления формы.

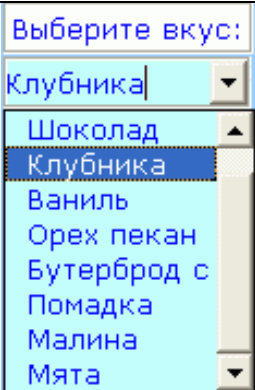
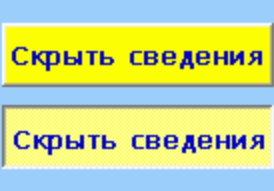
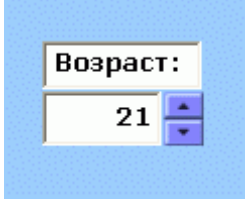
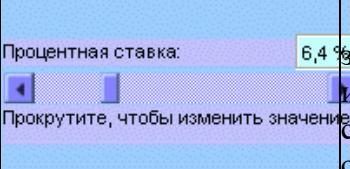
Таблица 7.2 — Обзор элементов ActiveX

Кнопки	Название	Пример	Описание
	Флажок		Включает или отключает значение, что соответствует выбору противоположных действий. На листе или в группе можно одновременно установить несколько флажков. Флажок может иметь одно из трех состояний: установлен (значение включено), снят (значение отключено) и смешенное состояние, то есть сочетание состояний «включено» и «отключено» (если поддерживается выбор нескольких элементов)

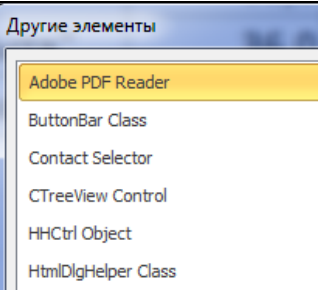
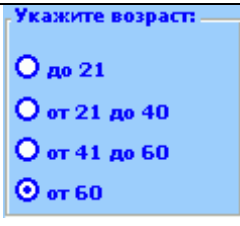
Продолжение табл. 7.2

Кноп-ки	Назва-ние	Пример	Описание
	Текстовое поле		Представляет собой прямоугольник, в котором можно просматривать, вводить и изменять текст или данные, связанные с ячейкой. Текстовое поле также может быть статическим и содержать данные, предназначенные только для чтения
	Кнопка		Запуск макроса, который выполняет действие при нажатии кнопки
	Переключатель		Позволяет выбрать одно из ограниченного набора взаимоисключающих значений. Переключатели обычно находятся в группе или рамке. Переключатель может иметь одно из трех состояний: установлен (значение включено), снят (значение отключено) и смешенное состояние, то есть сочетание состояний «включено» и «отключено» (если поддерживается выбор нескольких элементов)
	Список		Выводит список из одного или нескольких текстовых элементов, которые может выбрать пользователь. Список можно применять для представления большого числа вариантов, количество и содержимое которых может быть различным. Существует три типа списков: • Простой список поддерживает выбор только одного элемента. Такой список напоминает группу переключателей, но позволяет более эффективно работать с большим числом элементов. • Список связанного выбора позволяет выбрать один элемент, а также несколько расположенных рядом элементов. • Список, разрешающий несвязный выбор нескольких строк, позволяет выбрать один элемент, расположенные рядом элементы, а также разрозненные элементы

Продолжение табл. 7.2

Кноп-ки	Назва-ние	Пример	Описание
	Поле со списком		Представляет собой сочетание текстового поля и раскрывающегося списка. Поле со списком компактнее обычного списка, но чтобы отобразить список элементов, пользователь должен щелкнуть стрелку вниз. Поле со списком следует использовать, когда требуется обеспечить возможность ввода элементов в список и выбора из него одного элемента. В этом элементе управления отображается текущее значение независимо от того, каким образом оно было введено
	Выключатель		Указывает на состояние (да/нет) или режим (вкл./вкл.). При нажатии кнопки она меняет свое состояние на противоположное
	Счетчик		Позволяет увеличивать и уменьшать значение, например числовое значение, время или дату. Чтобы увеличить значение, щелкните стрелку вверх, чтобы уменьшить — стрелку вниз. Как правило, пользователь также может вводить текст в связанную ячейку или текстовое поле
	Полоса прокрутки		Используется для прокрутки диапазона значений с помощью кнопок со стрелками или путем перетаскивания ползунка полосы прокрутки. Можно перемещаться по странице значений (с заранее заданным интервалом), щелкая область между ползунком и одной из кнопок со стрелками для прокрутки. Пользователь также может вводить текст непосредственно в связанную ячейку или текстовое поле
	Надпись		Указывает назначение ячейки или текстового поля либо содержит описание (заголовки, подписи, рисунки) или краткие инструкции
	Изображение		Выводит изображение, например точечный рисунок (BMP), JPEG или GIF

Окончание табл. 7.2

Кноп-ки	Назва-ние	Пример	Описание
	Другие элементы		Выводит список доступных на компьютере элементов ActiveX, которые можно добавить в настраиваемую форму (например, проигрывателя Windows Media). В этом диалоговом окне также можно зарегистрировать пользовательский элемент управления
	Рамка		Группирует связанные элементы управления в виде прямоугольника с необязательной подписью. Обычно группируются переключатели, флажки или тесно связанные данные. ПРИМЕЧАНИЕ. Элемент ActiveX «Рамка» недоступен в разделе Элементы ActiveX на вкладке Вставка. Но его можно добавить из диалогового окна Другие элементы, выбрав пункт Рамка (Microsoft Forms 2.0 Frame)

7.4 Объекты средств рисования

В форму также можно включать графические элементы SmartArt, фигуры, WordArt и надписи. Можно изменять размер и цвет таких объектов, поворачивать, отражать и объединять их, чтобы получать еще более сложные фигуры. При вводе текста непосредственно в фигуру или надпись он становится частью объекта — при повороте или отражении объекта текст также меняет свое положение. В отличие от элементов ActiveX отдельным словам и знакам в объекте можно назначать различные атрибуты, например размер или начертание шрифта. Кроме того, объектам можно назначать макросы или гиперссылки. Можно даже связать текст в фигуре или надписи с ячейкой листа и динамически выводить обновленные значения.

Примечание. По внешнему виду иногда трудно определить тип элемента управления. Для этого щелкните его правой кнопкой мыши по периметру объекта и просмотрите контекстное меню.

- Если контекстное меню содержит команду **Свойства**, элемент управления является элементом ActiveX и сейчас используется режим конструктора.
- Если контекстное меню содержит команду **Назначить макрос**, это элемент управления формы.
- Если контекстное меню содержит команду **Изменить текст**, это объект-рисунок.

Работа с **Кнопкой** — и как с элементом управления формы, и как с элементом ActiveX была рассмотрена выше. Поработаем с другими элементами.

Пример 7.1. *Использование элементов управления «Флажок» и «Переключатель».*

Флажки (**CheckBox**) используются для выбора не взаимоисключающих вариантов (если этих вариантов немного), а переключатели (**OptionButton**), наоборот, — для выбора варианта в ситуации или/или.

Разместим на новом рабочем листе вертикально в ряд три элемента ActiveX **CheckBox**. Назовем их «Два состояния», «Три состояния» и «Связан с D6» (рис. 7.1).

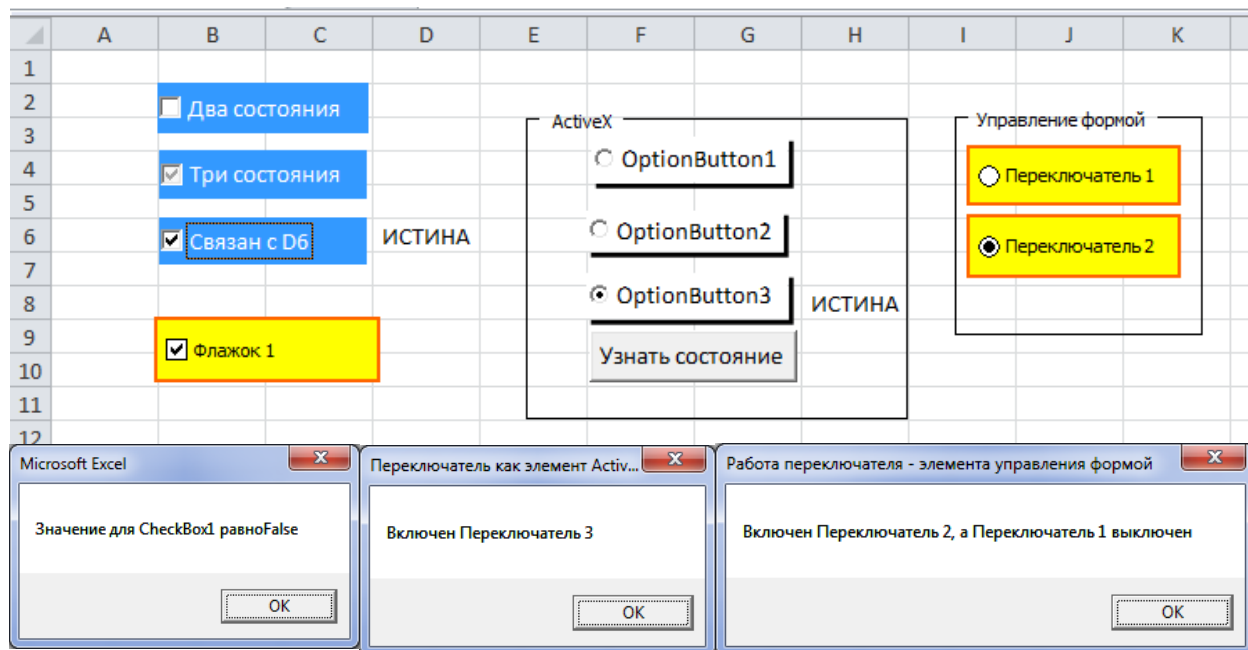


Рис. 7.1 — Часть листа Excel с элементами управления **Флажок** и **Переключатель**

Создадим для них процедуры отклика на событие **Click**. В теле процедур поместим соответствующую инструкцию *MsgBox*. Затем в окне редактирования кода модуля запишем следующее:

```
Private Sub CheckBox1_Click()
    MsgBox "Значение для CheckBox1 равно" & CheckBox1.Value
End Sub
Private Sub CheckBox2_Click()
    MsgBox "Значение для CheckBox2 равно" & CheckBox2.Value
End Sub
Private Sub CheckBox3_Click()
    MsgBox "Значение для CheckBox3 равно" & CheckBox3.Value
End Sub
```

Для второго флажка установим свойство **TriState** равным *True*.

Для третьего флажка установим свойство **LinkedCell** (связанная ячейка) равным соответственно D6.

Ниже поместим элемент управления формы **Флажок** и назначим ему макрос:

```
Sub Флажок1_Щелчок()
    MsgBox "Работает флажок"
End Sub
```

Затем установим три элемента ActiveX **CheckBox OptionButton**.

Для каждого переключателя установим в свойствах различные детали оформления (**BackColor**, **Shadow** и др.).

Создадим кнопку **Узнать состояние**. В окне редактирования кода модуля запишем:

```
Private Sub CommandButton1_Click()
    If OptionButton1.Value = True Then MsgBox "Включен Переключатель 1"
    If OptionButton2.Value = True Then MsgBox "Включен Переключатель 2"
    If OptionButton3.Value = True Then MsgBox "Включен Переключатель 3"
End Sub
```

Рядом поместим переключатели как элементы управления формы.

Назначим им макросы

```
Sub Перекл1_Щелчок()
MsgBox "Включен Переключатель 2, а Переключатель 1 вы-
ключен", , "Работа переключателя - элемента управления формой"
End Sub
Sub Перекл2_Щелчок()
MsgBox "Включен Переключатель 1, а Переключатель 2 вы-
ключен", , "Работа переключателя - элемента управления формой"
End Sub
```

Задание 7.1. Работа с элементами управления CheckBox, OptionButton и ToggleButton

Протестировать работу всех элементов, расположенных на листе (рис. 7.1) и описать отличия в работе элементов управления формы и элементов ActiveX. Поместить элемент ActiveX **ToggleButton** (Выключатель), при нажатии на который появляется сообщение «Включено!», при отжатии — «Выключено!». Описать особенности его работы по сравнению с другими кнопками («button»).

Пример 7.2. *Использование элементов управления «Список», «Счетчик» и «Полоса прокрутки».*

Откроем новый рабочий лист и поместим элемент ActiveX **Список (ListBox)**.

Выделим рядом со списком двенадцать ячеек и присвоим этому диапазону имя **Данные1**, выбрав в контекстном меню команду **Присвоить имя...** или вызвав **Диспетчер имен** (Ctrl+F3).

Затем введем значения в эти ячейки, например названия месяцев. Для заполнения списка данными из этого диапазона укажем в свойстве **ListFillRange** значение **Данные1** (можно указать и массив ячеек).

Установим связь списка с какой-либо ячейкой на листе (например, G2), изменив свойство **LinkedCell**.

На этом же рабочем листе поместим второй элемент управления **Список**, три кнопки, три переключателя и один флажок. Переименуем все элементы управления согласно рис. 7.2.

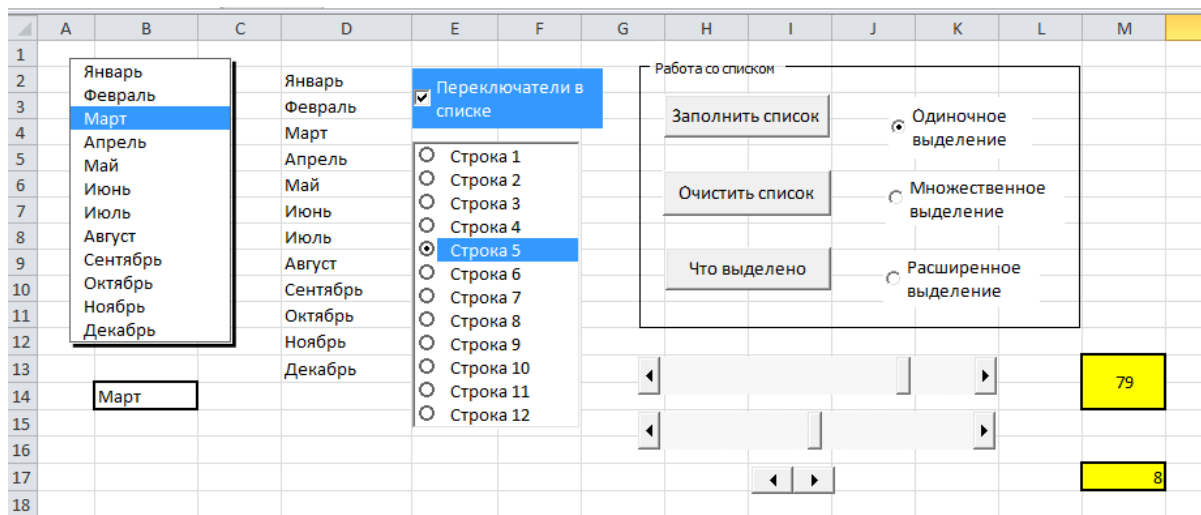


Рис. 7.2 — Часть листа Excel с элементами управления **ListBox**, **ScrollBar** и **SpinButton**

Затем в окне редактирования кода флажка **CheckBox1** запишем:

```
Private Sub CheckBox1_Click()
If CheckBox1.Value Then
    ListBox2.ListStyle = fmListStyleOption
Else
    ListBox2.ListStyle = fmListStylePlain
End If
ListBox2.Height = 170
End Sub
```

Эта процедура проверяет значение флажка **CheckBox1**. Если флажок установлен, то стиль списка **ListBox2** устанавливается равным **fmListStyleOption** (отображаются переключатели). Иначе устанавливается стиль **fmListStylePlain** (переключателей нет).

Первым двум кнопкам (**Заполнить список** и **Очистить список**) присвоим следующие процедуры:

```
Private Sub CommandButton1_Click()
'Обращаемся к объекту ListBox2
With ListBox2
```

```

'Очищаем содержимое
.Clear
'Добавляем строки
For i = 1 To 12
    .AddItem "Строка " & i
Next i
End With
End Sub

Private Sub CommandButton2_Click()
'Очищаем содержимое списка
ListBox2.Clear
End Sub

```

Для переключателей напомним следующие процедуры, которые определяют способ выделения строк в списке:

```

Private Sub OptionButton1_Click()
'Выделение одной строки
ListBox2.MultiSelect = fmMultiSelectSingle
End Sub

```

```

Private Sub OptionButton2_Click()
'Выделение нескольких строк в списке
ListBox2.MultiSelect = fmMultiSelectMulti
End Sub

```

```

Private Sub OptionButton3_Click()
'Выделение нескольких строк в списке с помощью Ctrl или Shift
ListBox2.MultiSelect = fmMultiSelectExtended
End Sub

```

Третьей кнопке (**Что выделено?**) назначим процедуру проверки режима выделения.

Если переключателем включен режим выделения одной строки **fmMultiSelectSingle**, то на экран выводится значение выделенной строки или сообщение «Ничего не выделено» при отсутствии выделения. Если включен другой режим (множественное выделение), то осуществляется перебор всех строк списка и в случае обнаружения выделенной строки переменной **ListFlag** присваивается значение **True**. Затем идет проверка выделения:

если значение **True**, то на экран выдается сообщение — значение выделенной строки, если **False**, то сообщение, что ничего не выделено.

Соответствующий код выглядит следующим образом:

```

Private Sub CommandButton3_Click()
'Проверка выделения
Dim ListFlag As Boolean
Dim x As Integer
ListFlag = False
With ListBox2
    If .MultiSelect = fmMultiSelectSingle Then
        If .Text = "" Then
            MsgBox "Ничего не выделено"
        Else
            MsgBox .Value
        End If
    Else
        For x = 0 To .ListCount - 1
            If .Selected(x) = True Then
                ListFlag = True
            End If
        Next x
        Select Case ListFlag
        Case Is = False
            MsgBox "Ничего не выделено"
        Case Is = True
            For x = 0 To .ListCount - 1
                If .Selected(x) Then
                    MsgBox .List(x)
                End If
            Next x
        End Select
    End If
End With
End Sub

```

Поместим на этот же лист два элемента ActiveX **Полоса прокрутки (ScrollBar)** и один элемент **Счетчик (SpinButton)**. Для всех элементов зададим максимальное и минимальное значение (**Min = 0** и **Max = 100**).

Для первой полосы прокрутки задаем связанную ячейку M13, изменив свойство **LinkedCell**. (рис. 7.2).

Для второй полосы прокрутки создадим процедуру отклика на событие **Change**. Запишем в окне редактирования кода модуля следующий код:

```
Private Sub ScrollBar2_Change()  
  MsgBox ScrollBar2.Value  
End Sub
```

Для счетчика также зададим связанную ячейку M17.

При движении бегунка первой полосы прокрутки в связанной ячейке F3 должно появляться соответствующее значение бегунка из заданного диапазона.

При движении бегунка второй полосы прокрутки соответствующее значение бегунка из заданного диапазона будет появляться в окне сообщения.

При нажатии кнопок со стрелками «влево» или «вправо» счетчика в связанной ячейке B11 должно появляться соответствующее значение счетчика из заданного диапазона.

Задание 7.2. Работа с элементами управления **ListBox, **ScrollBar** и **SpinButton****

1. Протестировать созданные программные коды.
2. Вставить **Полосу прокрутки** и **Счетчик** как элементы формы управления. Сравнить их работу с соответствующими элементами ActiveX.
3. Изменить макрос **CommandButton1_Click** таким образом, чтобы в список **ListBox2** добавлялось содержимое из **ListBox1** с номером строки.

Раздел 8. Программирование пользовательских форм

Цель раздела: приобрести навыки формирования и программирования диалоговых форм для создания баз данных и управления ими средствами Excel и VBA.

Чтобы добиться максимальной гибкости использования элементов управления, создают пользовательские формы — настраиваемые диалоговые окна, которые обычно содержат один или несколько элементов ActiveX. Пользовательские формы вызываются из программного кода VBA, который создается в редакторе Visual Basic.

Общая схема работы по созданию пользовательских форм следующая:

1. Вставить пользовательскую форму в проект VBA книги.
2. Создать процедуру для отображения пользовательской формы.
3. Добавить элементы ActiveX.
4. Изменить свойства элементов ActiveX.
5. Создать процедуры обработчика событий для элементов ActiveX.

Самые важные методы форм:

Show() — запуск формы (UserForm1.Show — если форма уже была загружена в память, она просто станет видимой, если еще нет — то будет автоматически загружена (произойдет событие Load);

Hide() — спрятать форму (UserForm1.Hide — форма будет убрана с экрана, но останется в памяти. Потом при помощи метода Show() можно будет опять ее вызвать в том же состоянии, в каком она была на момент «прятанья», а можно, например, пока она спрятана, программно изменять ее и расположенные на ней элементы управления. Окончательно форма удалится из памяти при закрытии документа;

Unload — удалить из памяти, если форма больше точно не потребуется (Unload UserForm1).

Самое важное событие формы — **Initialize** происходит при подготовке формы к открытию (появлению перед пользователем). Обычно в обработчик этого события помещается код, связанный с открытием соединений базы данных, настройкой элементов управления на форме, присвоением им значений по умолчанию и т. п.

8.1 Создание формы с элементами **Label** и **TextBox**

Откроем новую рабочую книгу Excel и перейдем в редактор Visual Basic (Alt+F11). Выберем из меню **Вставка (Insert)** команду **UserForm**. На экране появится заготовка диалоговой формы. Вызовем панель инструментов **ToolBox** (меню **View** или соответствующая кнопка на панели управления **Standart**).

Поместим в поле формы объект **Label** (надпись), под надписью поместим текстовое поле (**TextBox**), а справа добавим две кнопки (**CommandButton**), взяв все это с панели инструментов **ToolBox**. Сменим надписи на кнопках на «OK» и «Cancel», для этого выберем в контекстном меню команду **Properties** и изменим в свойстве **Caption** имена кнопок на соответствующие. Измените название формы и надпись (рис 8.1).

Нажав на клавишу F5, запустим форму на выполнение.

Так как кнопкам, расположенным на форме, не присвоено никаких процедур, то при щелчке по ним ничего происходить не будет.

Создадим новый модуль, для этого в меню **Вставка** выберем команду **Модуль**.

Создадим макрос с помощью команды **Insert | Procedure**. В открывшемся диалоговом окне укажем: имя макроса *DisplayDialog*, тип макроса **Подпрограмма (Sub)**, область определения **Общая (Public)**.

Затем в окне редактирования кода модуля запишем код активизации формы:

```
Public Sub DisplayDialog()  
    UserForm1.Show  
    If UserForm1.Tag = vbOK Then  
        MsgBox "OK clicked"
```

```

Else
    MsgBox "Cancel clicked"
End If
End Sub

```

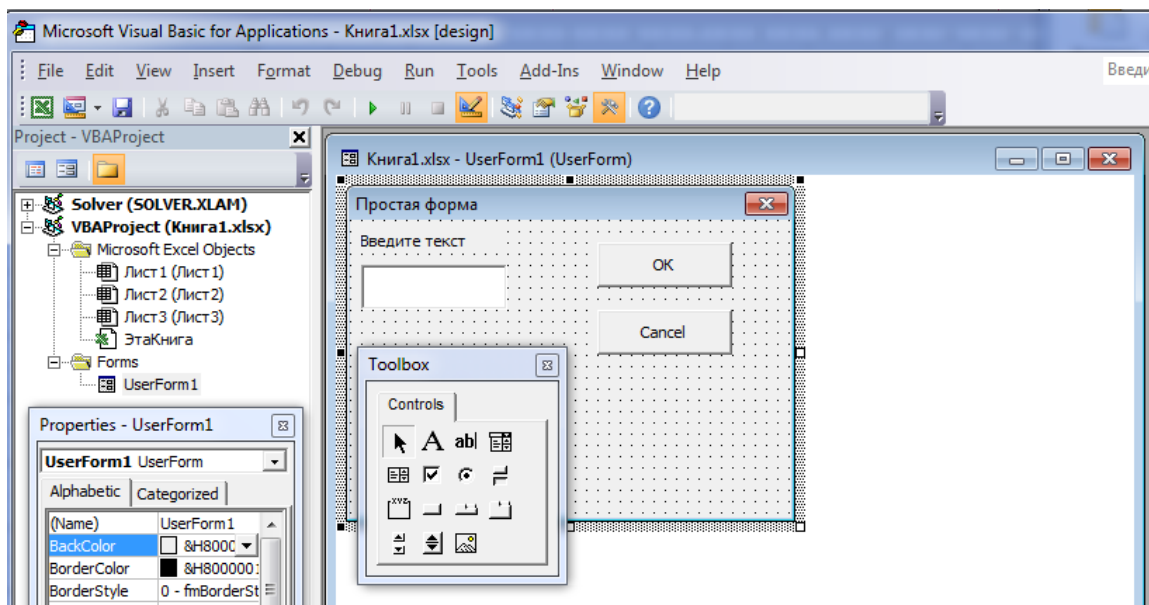


Рис. 8.1 — Форма с размещенными на ней объектами управления

Свойство **Tag** (англ. *ярлык*), использованное в макросе, сохраняет дополнительную информацию о каждом элементе управления на UserForm. Синтаксис его применения:

объект.**Tag** [=строка]

Строка идентифицирует объект (не обязательное).

Создадим процедуры отклика на нажатие кнопок диалоговой формы. В процедурах поместим метод сокрытия формы **Hide** и установим код завершения выполнения формы в свойстве **Tag**. Для этого в окошке **Project — VBAProject** два раза щелкнем по **UserForm1**. Затем в появившемся окошке **UserForm1** два раза щелкнем по кнопке **OK**, тем самым перейдем в окно редактирования кода процедуры отклика кнопки **OK** на щелчок по ней. В теле процедуры запишем следующее:

```
Private Sub CommandButton1_Click()
```

```
    'Занести в ячейку A1 значение, набранное в текстовом поле
```

```
    ActiveSheet.Cells(1, 1).Value = TextBox1.Value
```

```
    'Скрыть форму
```

```
    Me.Hide
```

```
'Свойству Tag зададим значение vbOK
Me.Tag = vbOK
End Sub
```

В теле процедуры отклика кнопки **Cancel** на щелчок по ней запишем следующее:

```
Private Sub CommandButton2_Click()
'Удалить из ячейки A1 значение
ActiveSheet.Cells(1, 1).Value = ""
'Скрыть форму
Me.Hide
'Свойству Tag зададим значение vbCancel
Me.Tag = vbCancel
End Sub
```

Здесь ключевое слово **Me** VBA заменяет объект UserForm1, который требуется спрятать (убрать с экрана), но не удалить (метод **Hide**). Оно обозначает тот объект, в котором находится другой объект, код которого сейчас исполняется (кнопка находится в форме).

Затем организуем запуск макроса на выполнение. Сделать это можно разными способами, например создав на листе Excel кнопку как элемент управления и назначить ей макрос DisplayDialog. Теперь при нажатии на кнопку **Запуск формы** будет запускаться этот макрос (рис. 8.2).

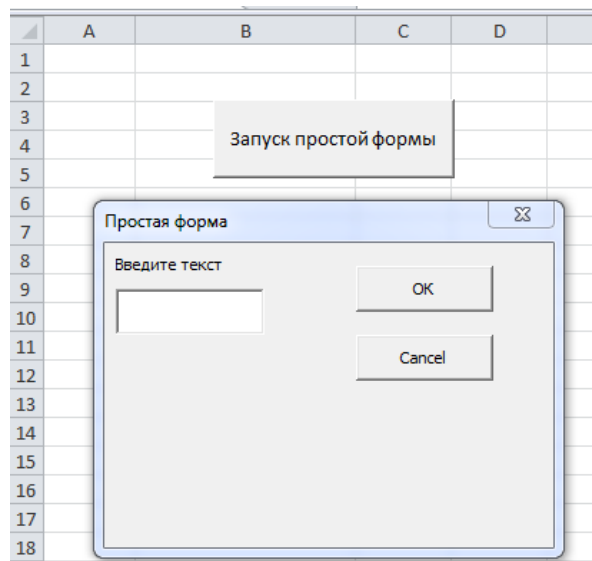


Рис. 8.2 — Вызов пользовательской формы

Протестируем работу кнопок. Введем в текстовое поле значение «Привет!» и нажмем на кнопку **OK**. В ячейке A1 появится введенный текст, а на листе — окно с надписью «OK clicked» (рис. 8.3).

При нажатии на кнопку **Cancel** текст в ячейке A1 исчезает и появляется окно с надписью «Cancel clicked».

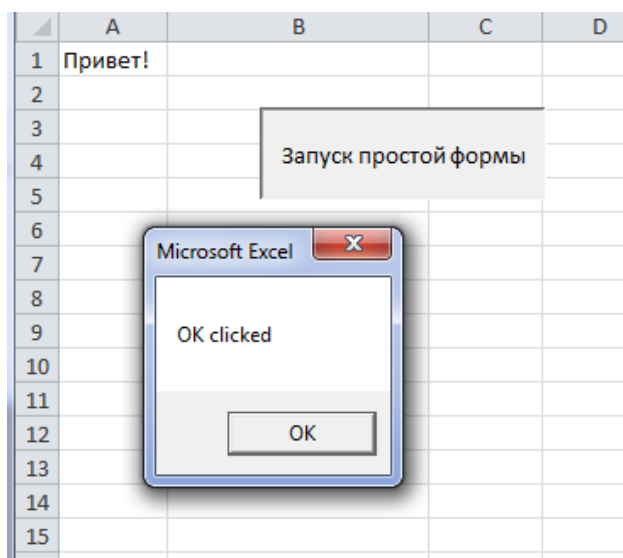


Рис. 8.3 — Тестирование кнопки **OK**

Примечание. Щелчок по крестику в правом верхнем углу формы приводит к появлению информационного сообщения об ошибке несовпадения типов в строке кода

If UserForm1.Tag = vbOK Then

Если нажатие на крестик считать аналогом работы кнопки **Cancel**, то имея в виду числовые значения формы (см. п.6.7), избежать ошибки можно, заменив эту строку на:

If UserForm1.Tag = "1" Then

8.2 Создание формы с RefEdit

Откроем редактор Visual Basic и выберем из меню **Вставка** команду **UserForm**. На экране появится заготовка диалоговой формы и панель инструментов **ToolBox**.

В диалоговой форме поместим элементы управления **RefEdit** (редактирование ссылок) и **TextBox** (текстовое поле).

Если на панели **ToolBox** отсутствует кнопка **RefEdit**, то ее следует вызвать, выполнив команду **Tools | Additional Controls | RefEdit.Ctrl**.

Добавим к размещенным элементам управления надписи (**Label**) «Выбор диапазона ячеек» и «Значение для заполнения» соответственно. Как и в предыдущем пункте создадим две кнопки **OK** и **Cancel** (рис. 8.4) и назначим им те же процедуры отклика.

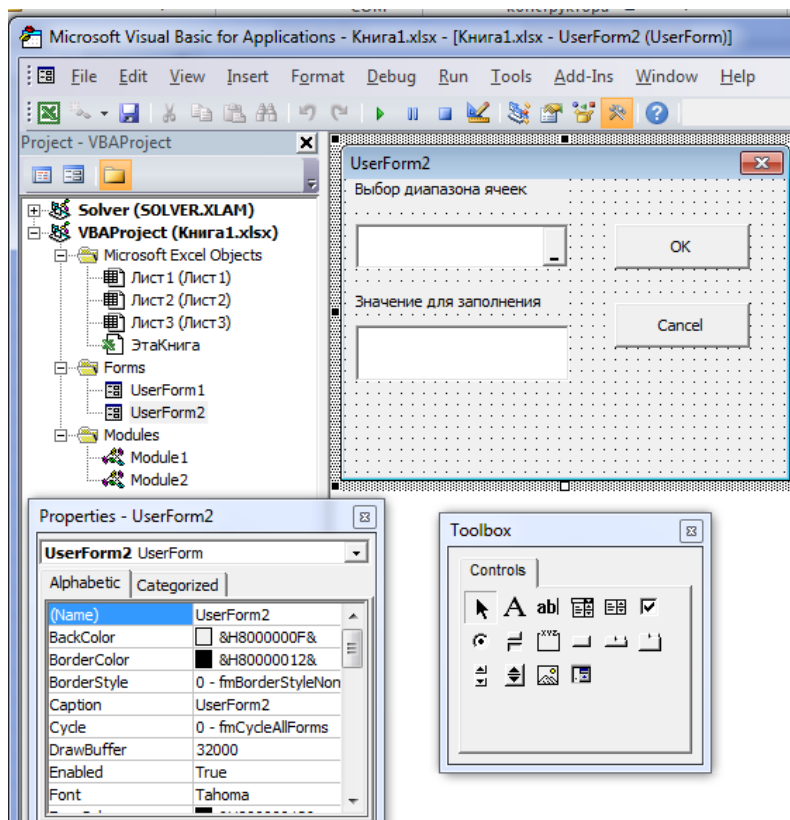


Рис. 8.4 — Форма **UserForm2** с размещенными на ней элементами управления

Создадим новый модуль, для этого в меню **Вставка** выберем команду **Модуль**. Создадим макрос с помощью команды **Insert | Procedure**. В открывшемся диалоговом окне укажем: имя макроса *ShowRefEdit*, тип макроса **Подпрограмма (Sub)**, область определения **Общая (Public)**.

Затем в окне редактирования кода модуля запишем код активизации формы:

```
Public Sub ShowRefEdit()  
With UserForm2
```

```

.RefEdit1.Text = Selection.Address
.Show
If .Tag = vbOK Then
    Range(.RefEdit1.Text) = .TextBox1.Text
End If
End With
End Sub

```

Первая инструкция передает элементу **RefEdit** адрес выбранного диапазона на рабочем листе. Затем запускается форма **UsesForm2** с помощью метода **Show**. После того как форма завершит работу, проверяется возвращаемое значение. Если завершение было успешным, то выбранный диапазон заполняется указанным значением, если завершение было неуспешным, то выбранный диапазон не заполняется.

Для запуска макроса на выполнение так же, как и в предыдущем задании, создадим кнопку **Работа с RefEdit** и протестируем работу макроса (рис. 8.5).

В поле **Выбор диапазона ячеек** задаем массив ячеек на листе, а в поле **Значение для заполнения** вводим какой либо текст. При нажатии на кнопку **ОК** выделенный массив ячеек (A1:A10) заполнится текстом, при нажатии на кнопку **Cancel** заполнения происходить не будет.

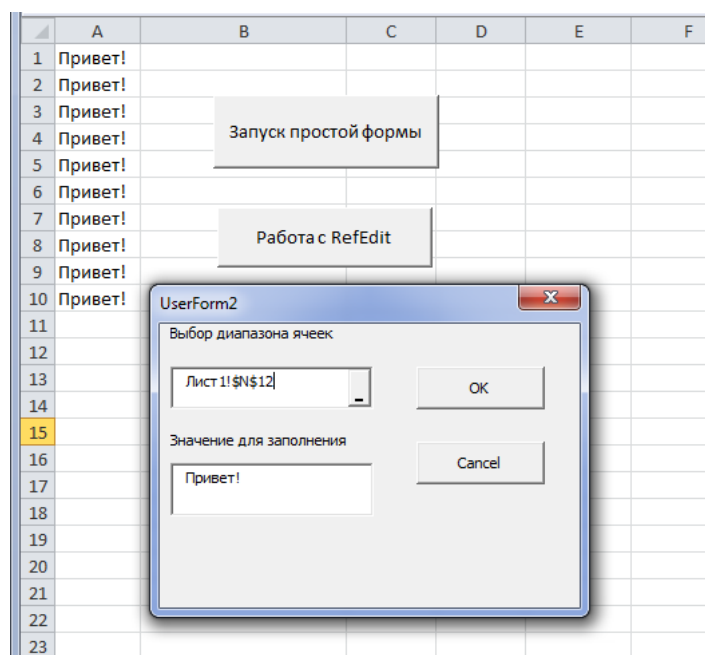


Рис. 8.5 — Тестирование формы **UserForm2**

8.3 Создание формы с ComboBox и ListBox

Откроем новый рабочий лист Excel. Перейдем в редактор Visual Basic (**Alt+F11**) и выберем в меню «Вставка» команду **UserForm**. В появившейся диалоговой форме разместим два элемента управления **ComboBox** (Поле со списком) и один элемента управления **ListBox** (Список). Добавим к этим элементам надписи (*Label*) «Компания», «Адрес» и «Контактный телефон» соответственно. Также добавим кнопки с названиями **OK** и **Close** (рис. 8.6).

Создадим процедуру отклика кнопки **Close**, которая закрывала бы уже открытую форму. Для этого два раза щелкнем по кнопке и запишем в окне редактирования кода следующее:

```
Private Sub CommandButton2_Click()  
    Me.Hide  
End Sub
```

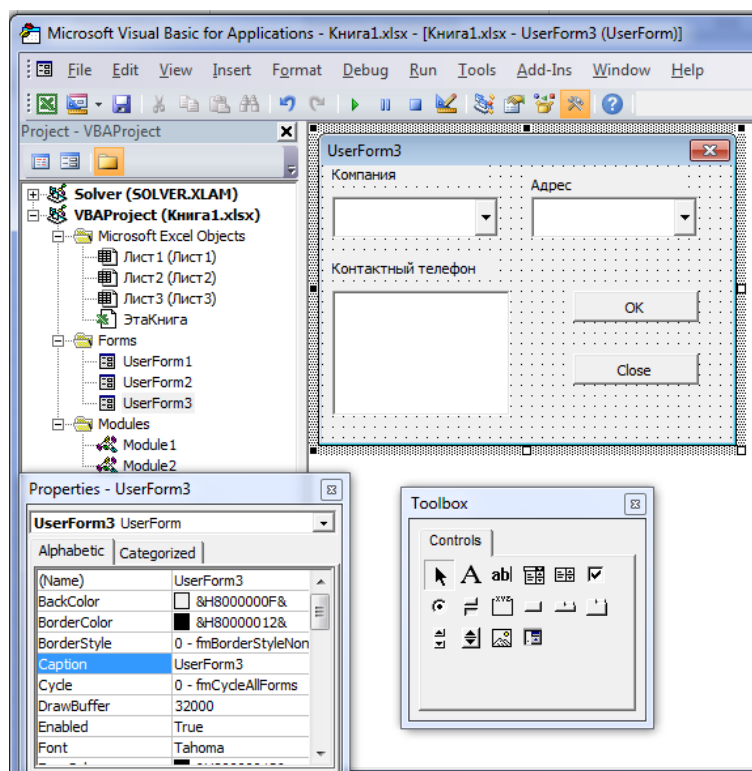


Рис. 8.6 — Форма **UserForm3** с размещенными на ней элементами управления

На рабочем листе в ячейках A1, B1 и C1 напишем заголовки «Компания», «Город» и «Телефон», а ниже введем 9 соответствующих значений (название компании, местонахождение и контактный телефон).

Затем выделив поочередно массивы этих значений и выбрав в контекстном меню команду **Присвоить имя...** или вызвав **Диспетчер имен** (Ctrl+F3), зададим массивам имена соответственно «Company», «City» и «Tel» (рис. 8.7).

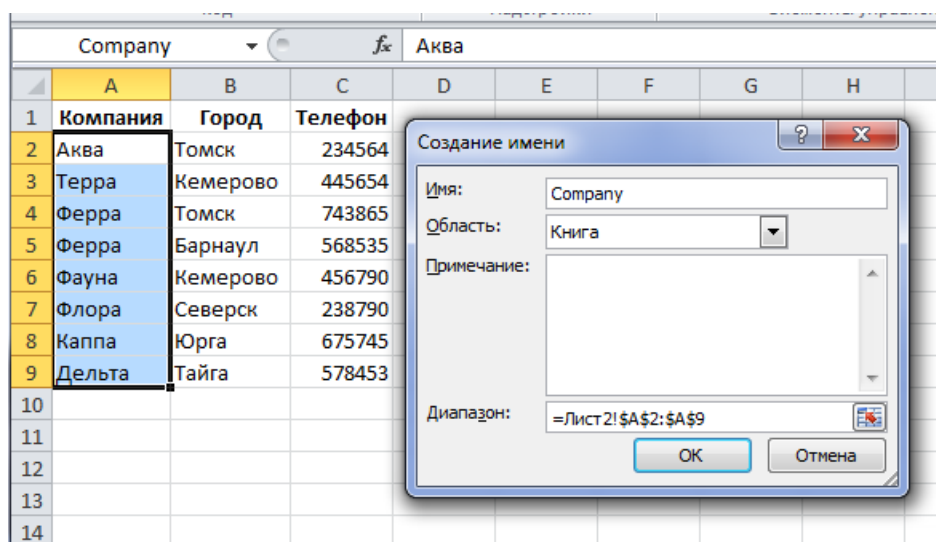


Рис. 8.7 — Присвоение имен группам ячеек на листе Excel

Свяжем эти ячейки с элементами управления **ComboBox1** и **ComboBox2** формы **UserForm3**. Для этого в их **Свойствах** укажем в поле **RowSource** имя соответствующего массива ячеек («Company» и «City»).

Теперь создадим процедуру отклика при нажатии на кнопку ОК. Алгоритм предлагается следующий. Пользователь выбирает в окнах *Компания* и *Адрес* соответствующие значения и нажимает на ОК. Если компания находится по этому адресу, то в окне *Контактный телефон* появляется ее номер телефона, если нет, то появляется информационное сообщение о том, что такой компании не существует.

Для этого два раза щелкнем по кнопке ОК в форме **UserForm3** в редакторе VBA и запишем в окне редактирования следующий код

Private Sub CommandButton1_Click()

'Очистить поле ListBox

Me.ListBox1.Clear**flag = 0**

' Цикл по строкам столбцов Компания и Адрес и проверка того,
 ' соответствуют ли введенные (или выбранные) компании и их адреса
 ' без учета пробелов в именах (функция Trim) тому, что записано в таблице.
 ' Если совпадение, то в ListBox заносится соответствующий номер телефона

For i = 1 To 9**If Trim(ComboBox1.Text) = Cells(i + 1, 1) And _****Trim(ComboBox2.Text) = Cells(i + 1, 2) Then****Me.ListBox1.AddItem Cells(i + 1, 3)****flag = 1****End If****Next i**

'Если соответствия названия компании и адреса не найдено,

'то вывод предупреждающего сообщения

If flag = 0 Then**MsgBox "Компании " + Trim(ComboBox1.Text) + _****" в городе " + Trim(ComboBox2.Text) + " нет! "****End If****End Sub**

Теперь создадим макрос запуска нашей формы (переименуем ее в *Поиск контактного телефона*). Для этого в новом модуле «Module3» вставим процедуру:

Public Sub Find()**With UserForm3****.Show****End With****End Sub**

Затем создадим на листе Excel кнопку **Найти** и назначим этой кнопке макрос *Find*.

Протестируем работу макросов (рис. 8.8, 8.9).

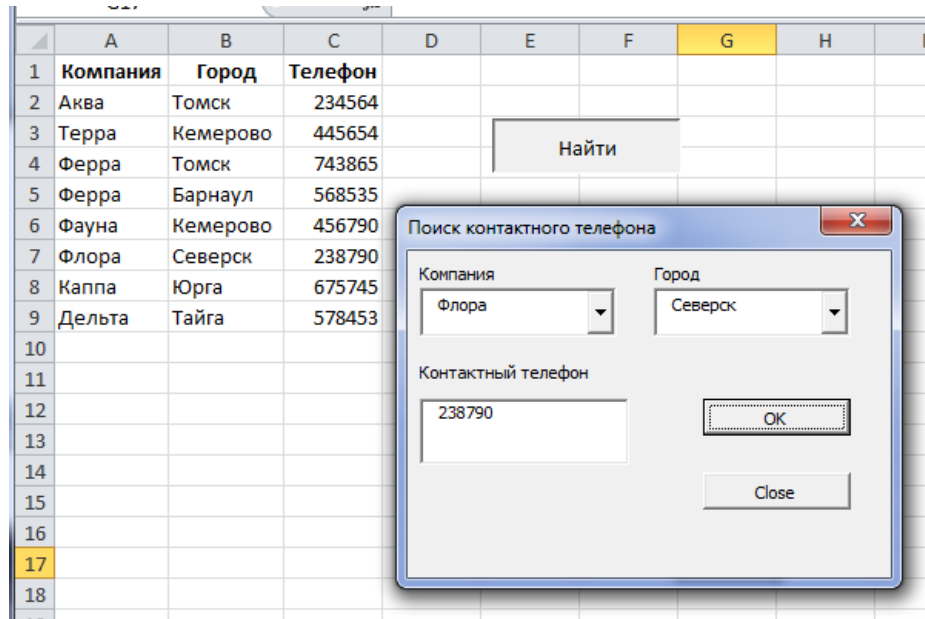


Рис. 8.8 — Положительный результат поиска

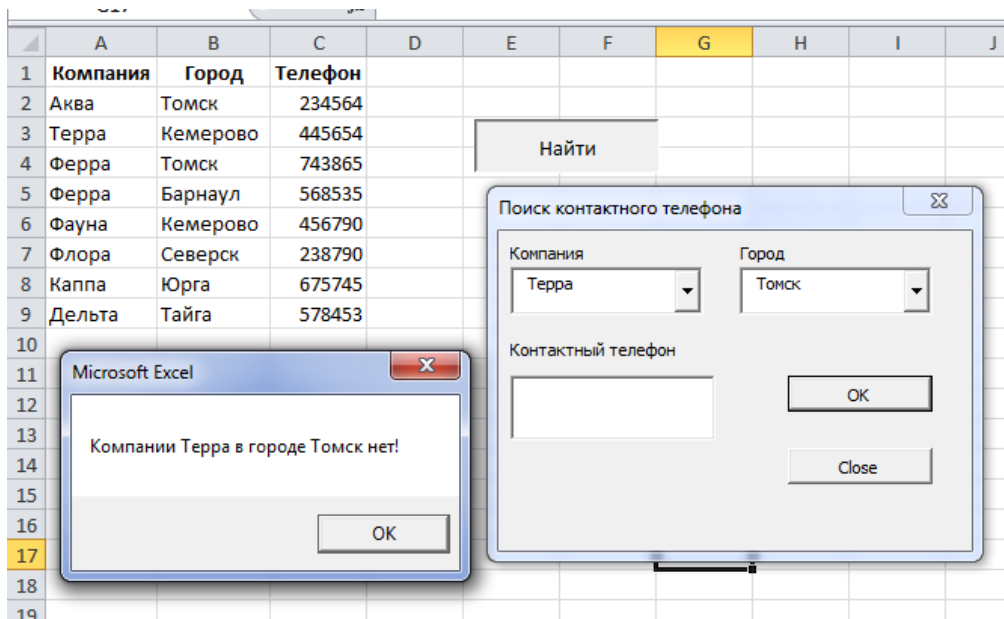


Рис. 8.9 — Отрицательный результат поиска

8.4 Создание диалоговой формы «Мастер Путешествий»

Откроем новый рабочий лист Excel. Перейдем в редактор Visual Basic (**Alt+F11**) и выберем в меню «Вставка» команду **UserForm**. В свойствах формы изменим имя формы (**Name**) и заголовков (**Caption**) на **TripWizard**.

В нижней левой части формы разместим четыре кнопки (**CommandButton**). Изменим их заголовки на **Назад**, **Вперед**, **Отмена** и **Готово**, а имена — на *cmdPrevious*, *cmdNext*, *cmdCancel*, *cmdFinish* соответственно.

Примечание.

Вы уже, наверно, обратили внимание, что при создании форм или элементов управления, VBA устанавливает их имена (свойство **Name**) по умолчанию. Но часто возникает необходимость в их смысловом переименовании (как сейчас, например). Для переименования объектов и переменных в среде Windows существует *соглашение об именах*.

Его суть: имя начинается с короткого префикса, одинакового для всех объектов одного и того же рода (табл. 8.1). За ним следует собственно имя — идентификатор объекта. Будем в дальнейшем придерживаться этого соглашения при работе с элементами управления.

Таблица 8.1 — Примеры префиксов имен объектов

Имя	Управляющий элемент	Префикс
CheckBox	Флажок	chk
ComboBox	Поле со списком	cbo
CommandButton	Кнопка	cmd
Frame	Рамка	fra
Image	Изображение	img
Label	Надпись	lbl
ListBox	Список	lst
MultiPage	Набор страниц	mlt
OptionButton	Переключатель	opt
RefEdit	Редактирование ссылок	ref
ScrollBar	Полоса прокрутки	scr
SpinButton	Счетчик	spn
TextBox	Текстовое поле	txt
ToggleButton	Выключатель	tgl
UserForm	Форма	ftm

Поместим на форму четыре элемента управления Рамки (*Frame*) (рис. 8.10). Пронумеруем рамки, задав заголовки 1, 2, 3, 4 в свойстве **Caption**. Свойству **SpecialEffect** установим значение

2-fmSpecialEffectSunken (верхняя и левая границы объекта затенены, а правая и нижняя подсвечены; элемент углублен в окружающее окно).

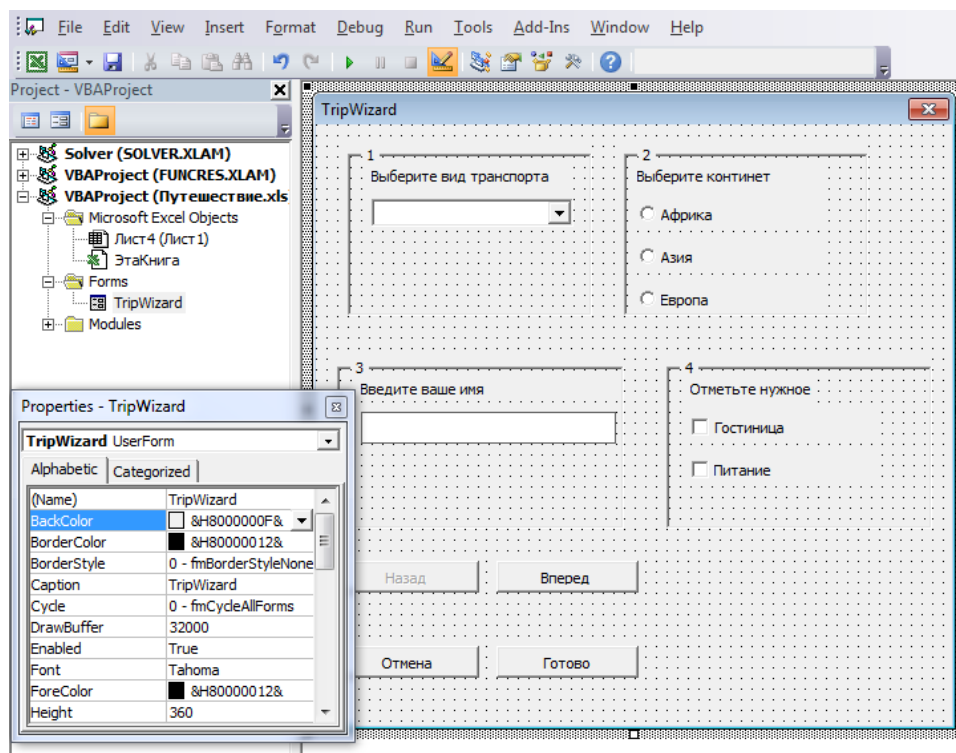


Рис. 8.10 — Форма **TripWizard** с размещенными на ней элементами управления

Разместим в рамках элементы управления. На первой рамке поместим **Label** и **ComboBox** (рис. 8.10). Дадим полю со списком имя *cboTransport*.

На второй рамке разместим **Label** и три элемента управления **OptionButton** друг под другом (рис. 8.10). Дадим имена *optAfrica*, *optAsia*, *optEurope* соответственно.

На третьей рамке разместим элементы **Label** и **TextBox** (рис. 8.10). Дадим текстовому полю имя *txtName*.

На четвертой рамке разместим элемент **Label** и два элемента управления **CheckBox** (рис. 8.10). Дадим флажкам имена *chkHotel* и *chkMeals* соответственно.

Наложим по очереди все рамки точно друг на друга.

Примечание.

Если после этого понадобится перейти к нужной рамке (например, для внесения изменений), то необходимо сначала выбрать ее в списке объектов окна свойств формы **TripWizard**, а затем щелкнуть на контуре рамки правой кнопкой и в появившемся контекстном меню выбрать команду **Переместить (Bring Forward)**.

Справа поместим кнопку с картинкой и назначим ей по клику мышкой переход на сайт отзывов <http://www.otzyv.ru>. Для этого в процедуру обработки щелчка по кнопке надо поместить следующий код:

```
Private Sub CommandButton1_Click()
ActiveWorkbook.FollowHyperlink Address:="http://www.otzyv.ru",
_ NewWindow:=True
End Sub
```

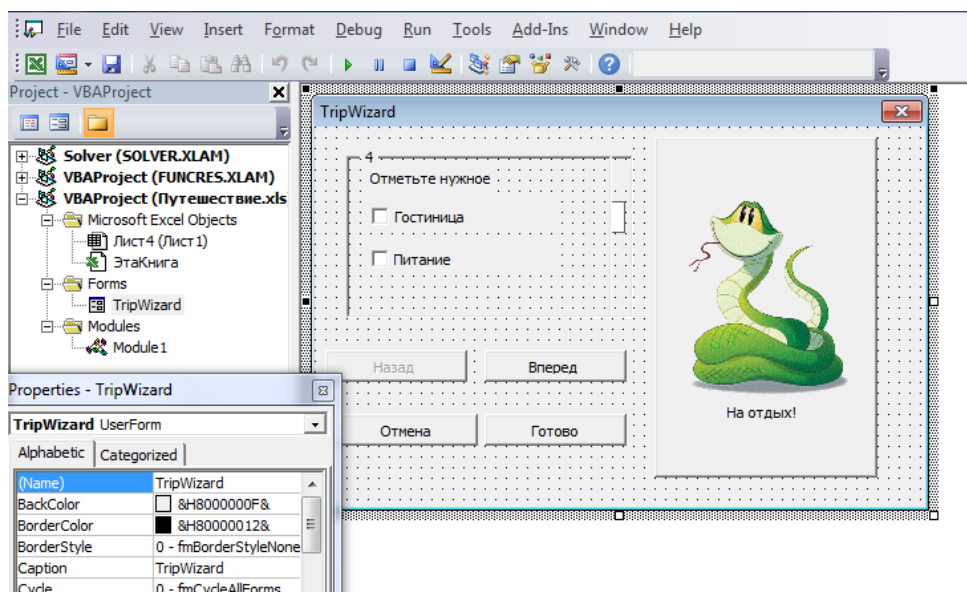


Рис. 8.11 — Окончательный вид формы **TripWizard**

Итак, форма с необходимыми объектами готова (рис. 8.11).

Создадим в редакторе модуль и поместим туда макрос запуска формы **StartWizard**:

```

Public Sub StartWizard()
With TripWizard
.Show
End With
End Sub

```

С помощью элементов управления создадим на листе Excel кнопку **Путешествие** и назначим кнопке созданный макрос. Но отображение рамок не произошло, кнопки тоже не работают (рис. 8.12):

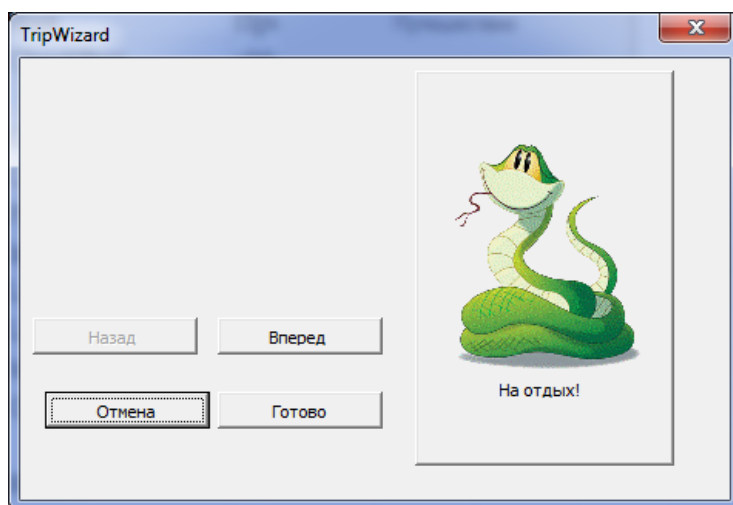


Рис. 8.12 — Неудачный запуск формы **TripWizard**

Дело в том, что не обеспечена возможность перехода от рамки к рамке и не заданы начальные значения в списках (инициализация формы). Ну, и нажатие кнопок пока не обработано.

Введем переменную *CurrentPanel* (счетчик рамок (панелей)) и создадим подпрограмму *ShowPanel*, с помощью которых сможем в нужный момент вызывать требуемую рамку.

Щелкнем два раза по свободному месту на форме **TripWizard**, тем самым попадем в окно редактирования кода формы.

Сначала объявим переменные и константы, общие для всех макросов:

```

Dim CurrentPanel As Integer      'Это счетчик рамок
Const FirstPanel = 1
Const LastPanel = 4

```

Затем создадим макрос *ShowPanel* и поместим в него следующий код:

```
Public Sub ShowPanel()
'Сначала делаем все рамки невидимыми
frame1.Visible = False
frame2.Visible = False
frame3.Visible = False
frame4.Visible = False
'Выбор рамки в зависимости от значения счетчика
Select Case CurrentPanel
  Case 1
    frame1.Visible = True 'если 1, то рамка 1 видима
  Case 2
    frame2.Visible = True 'если 2, то рамка 2 видима
  Case 3
    frame3.Visible = True 'если 3, то рамка 3 видима
  Case 4
    frame4.Visible = True 'если 4, то рамка 4 видима
  Case Else
  End Select 'Окончание выбора
'Даем название заголовку формы
Me.Caption = "TripWizard: Step " & CurrentPanel & " of " & LastPanel
End Sub
```

Проинициализируем начальные значения мастера при его вызове. Для этого в коде формы создадим процедуру отклика на событие диалогового окна *Initialize* и поместим в тело процедуры следующий код:

```
Private Sub UserForm_Initialize()
'Задаем начальное значение счетчика
CurrentPanel = 1
'Запускаем процедуру показа первой рамки
ShowPanel
'Введем список для ComboBox, находящегося в этой рамке
With cboTransport
  .AddItem "Самолет"
  .AddItem "Поезд"
  .AddItem "Корабль"
End With
End Sub
```

Создадим процедуры отклика на щелчок для кнопок *Отмена*, *Назад*, *Вперед* и *Готово*.

Два раза щелкнем по кнопке *Готово*, тем самым мы создадим процедуру отклика кнопки на щелчок по ней. Запишем в появившемся окне редактирования кода следующее:

```
Private Sub cmdFinish_Click()  
  Me.Hide  
  Me.Tag = vbOK  
End Sub
```

Затем, не выходя из окна редактирования кода, создадим еще 3 процедуры отклика.

Процедура отклика кнопки *Отмена*:

```
Private Sub cmdCancel_Click()  
  Me.Tag = vbCancel  
  End  
End Sub
```

Процедура отклика кнопки *Назад*:

```
Private Sub cmdPrevious_Click()  
  'Движение к предыдущей рамке  
  CurrentPanel = CurrentPanel - 1  
  ShowPanel  
  cmdNext.Enabled = True  
  If CurrentPanel = FirstPanel Then  
    cmdPrevious.Enabled = False  
  Else  
    cmdPrevious.Enabled = True  
  End If  
End Sub
```

Процедура отклика кнопки *Вперед*:

```
Private Sub cmdNext_Click()  
  'Движение к следующей рамке  
  CurrentPanel = CurrentPanel + 1  
  ShowPanel  
  cmdPrevious.Enabled = True
```

```

If CurrentPanel = LastPanel Then
cmdNext.Enabled = False
Else
cmdNext.Enabled = True
End If
End Sub

```

Для того чтобы на первом шаге кнопка **Назад** была не активна (ниже первого шага нам двигаться некуда), нужно для этой кнопки в свойстве *Enabled* поставить значение **False**.

Теперь, запустив макрос **StartWizard**, увидим, как происходит переход от одного состояния формы к другой. В каждом состоянии обеспечивается прием от пользователя определенной информации. Но в итоге не формируется никакого отчетного документа.

Чтобы это исправить, внесем изменения в макрос **StartWizard**:

```

Public Sub StartWizard()
Dim msg As String
With TripWizard
.Show
    'Последовательно обрабатываем щелчки мыши,
    'выбирая данные с элементов управления и
    'формируя информационную строку msg
    msg = "Имя: " & .txtName.Text & Chr(13)
    msg = msg & "Континент: "
If .optAfrica.Value Then
    msg = msg & "Африка" & Chr(13)
Else
If .optAsia.Value Then
    msg = msg & "Азия" & Chr(13)
Else
If .optEuropa.Value Then
    msg = msg & "Европа" & Chr(13)
End If
End If
End If
    msg = msg & "Оплата включает: "
If .chkHotel.Value Then
    msg = msg & "Гостиница"

```

```

End If
If .chkMeals.Value Then
    msg = msg & "Питание"
End If
msg = msg & Chr(13)
msg = msg & "Транспорт: " & .cboTransport.Value
MsgBox msg, , "Приятного отдыха!"
End
End With
End Sub

```

Теперь можно протестировать работу созданного «Мастера путешествий» — от нажатия на кнопку «Путешествие» до нажатия на кнопку «Готово».

Появится результирующий отчет, составленный по данным, полученным на всех шагах «Мастера путешествий» (рис. 8.13).

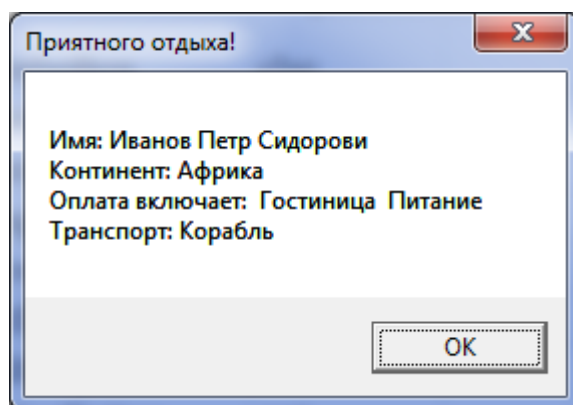


Рис. 8.13 — Результирующий отчет

Задание 8.1. Создание формы туристического ваучера

Измените макрос таким образом, чтобы в результате работы «Мастера» формировался ваучер на туристическую поездку, где были бы указаны следующие данные: ФИО путешественника, страна и город посещения, название отеля, срок пребывания, тип номера, вид питания, стоимость тура. Выразите сожаление, если клиент вышел, ничего не заполнив.

Для формирования многостраничности использовать элемент управления MultiPage.

8.5 Создание интерфейса работы с базой данных

Возможности Excel и VBA можно использовать для формирования простых баз данных и пользовательского интерфейса управления ими.

Создадим базу данных по учету оплаты за электроэнергию. Таблица базы данных должна содержать: фамилию, имя, адрес плательщика, дату платежа, тариф, показания счетчика (текущее и предыдущее), подсчитанный расход электроэнергии, сумму к оплате.

Откроем новую рабочую книгу Excel и перейдем в редактор VBA. Выберем из меню **Вставка** команду **UserForm**. В редакторе появится заготовка диалоговой формы. Изменим в свойстве **Caption** заголовок формы **UserForm1** на **Учет оплаты за электроэнергию**.

Создадим в форме две рамки (**Frame**).

Первой рамке в свойстве **Caption** изменим заголовок на **Информация о клиенте**, а второй рамке — на **Информация о платеже**. В этих рамках разместим элементы управления **Label** и **TextBox**.

Изменим имена всем семи элементам **TextBox**. В их соответствии можно разобраться по коду макроса **CommandButton1_Click()**, приведенного ниже.

Также добавим три элемента управления **CommandButton** с заголовками: **Принять**, **Отмена**, **Выход**, **Диаграмма** (рис. 8.14).

Затем вернемся к нашей рабочей книге. Листу с именем **Лист2** присвоим имя **База**, а листу **Лист1** присвоим имя **Меню**. На листе с именем **Меню** поместим элемент ActiveX **Кнопка**. Изменим заголовок кнопки на **Прием платежа** (свойство **Caption**), а имя кнопки (свойство **Name**) — на **ПриемПлатежа**.

Рис. 8.14 — Форма с размещенными на ней объектами интерфейса

Перейдем в редакторе на лист 1 (Меню) и создадим процедуру отклика кнопки на щелчок:

Private Sub ПриемПлатежа_Click()

'Вызываем процедуру формирования заголовков базы данных

ЗаголовокРабочегоЛиста

End Sub

Для формирования заголовков столбцов базы данных создадим процедуру **ЗаголовокРабочегоЛиста**, которая будет вызываться из процедуры обработки кнопки рабочего листа «Прием платежа». Для этого выполним команду меню «Вставка | Процедура». Присвоим ей имя.

Процедура выполняет следующие действия:

- проверяет, заполнена ли первая строка — строка заголовков столбцов (по значению ячейки *A1*); если заполнена, то не выполняет никаких действий и завершает работу, передавая управление в точку вызова;

– если первая строка не заполнена, то в ячейки первой строки рабочего листа записывает название граф баз данных, комментарии к ним, закрепляет первую строку и завершает работу, передавая управление в точку своего вызова.

В окне редактирования кода введем текст этой процедуры:

Private Sub ЗаголовокРабочегоЛиста()

'Активизируем рабочий лист

Application.Worksheets("База").Activate

'Проверяем, есть ли названия столбцов. В противном случае вносим их

With ActiveSheet

If .Range("A1").Value = "Фамилия" Then

.Range("A2").Select

Else

'Очищаем рабочий лист

ActiveSheet.Cells.Clear

'Записываем названия столбцов

Application.Worksheets("База").Range("A1:I1").Select

With Selection

.Value= Array("Фамилия", "Имя", "Адрес", "Текущее показание счетчика", "Предыдущее показание счетчика", "Тариф", "Дата платежа", "Расход электроэнергии", "Сумма")

.Interior.ColorIndex = 8

.Font.Bold = True

End With

'Вставляем комментарии

.Range("A1").AddComment

.Range("A1").Comment.Visible = False

.Range("A1").Comment.Text Text:= "Фамилия клиента"

.Range("B1").AddComment

.Range("B1").Comment.Visible = False

.Range("B1").Comment.Text Text:= "Имя клиента"

.Range("C1").AddComment

.Range("C1").Comment.Visible = False

.Range("C1").Comment.Text Text:= "Адрес клиента"

.Range("D1").AddComment

.Range("D1").Comment.Visible = False

.Range("D1").Comment.Text Text:= "Текущее показание счетчика"

.Range("E1").AddComment

.Range("E1").Comment.Visible = False

```

        .Range("E1").Comment.Text Text:= "Предыдущее показание
счетчика"
        .Range("F1").AddComment
        .Range("F1").Comment.Visible = False
        .Range("F1").Comment.Text Text:= "Тариф"
        .Range("G1").AddComment
        .Range("G1").Comment.Visible = False
        .Range("G1").Comment.Text Text:= "Дата платежа"
        .Range("H1").AddComment
        .Range("H1").Comment.Visible = False
        .Range("H1").Comment.Text Text:= "Расход электроэнергии"
        .Range("I1").AddComment
        .Range("I1").Comment.Visible = False
        .Range("I1").Comment.Text Text:= "Сумма"
    End If
End With
'Форматирование табличных ячеек
Worksheets("База").Range("A:I").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = True
    .Orientation = 0
    .AddIndent = False
    .ShrinkToFit = False
End With
'Вызвать на экран форму
UserForm1.Show
End Sub

```

Напишем процедуры для кнопок **Принять**, **Отмена** и **Выход**.

Перейдем к форме, щелкнув два раза по **UserForm1** в окне **Project-VBAProject**. В появившейся форме два раза щелкнем по кнопке «Принять», перейдя в процедуру обработки события **Click**.

Запишем следующий код (здесь, как и в предыдущей работе, изменены имена элементов управления на форме:

```

Private Sub CommandButton1_Click()
'Декларация переменных
Dim fam, nam, adr As String
Dim tarif, prpok, tekpok, rashod, summa As Single

```

Dim nomer As Integer

Dim data As Date

'Вычисление номера первой свободной строки в таблице

nomer = Application.CountA(ActiveSheet.Columns(1)) + 1

With UserForm1

'Проверяем, введена ли фамилия

If .txtFamil.Text = "" Then

MsgBox "Вы забыли указать фамилию", vbExclamation

Exit Sub 'Выход из процедуры до ее естественного окончания

End If

'Проверяем, введено ли имя

If .txtName.Text = "" Then

MsgBox "Вы забыли указать имя", vbExclamation

Exit Sub

End If

'Проверяем, введен ли адрес

If .TxtAdres.Text = "" Then

MsgBox "Вы забыли указать адрес", vbExclamation

Exit Sub

End If

'Присваиваем значения переменным в элементах TextBox

fam = .txtFamil.Text

nam = .txtName.Text

adr = .TxtAdres.Text

'Проверяем, введено ли текущее показание счетчика

If IsNumeric(.txttekpok.Text) = False Then

MsgBox "Введено неверное показание счетчика",
vbExclamation

Exit Sub

End If

tekpok = CSng(.txttekpok.Text)

'Проверяем, введено ли предыдущее показание счетчика

If IsNumeric(.txtprpok.Text) = False Then

MsgBox "Введено неверное показание счетчика",
vbExclamation

Exit Sub

End If

prpok = CSng(.txtprpok.Text)

'Проверяем, введен ли тариф

If IsNumeric(.txttarif.Text) = False Then

MsgBox "Введен неверный тариф", vbExclamation

Exit Sub

End If

```

tarif = CSng(.txttarif.Text)
If IsDate(.txtdata) = False Then
    MsgBox "Дата введена не верно", vbExclamation
Exit Sub
End If
data = .txtdata
If Val(txtprpok.Text) > Val(txttekpok.Text) Then
    MsgBox "Предыдущее показание счетчика больше текущего", vbExclamation
Exit Sub
End If
End With
'Вычисляем расход электроэнергии и сумму оплаты
rashod = tekpok - prpok
summa = rashod * tarif
'Записываем данные в ячейки рабочего листа
With ActiveSheet
    .Cells(nomer, 1).Value = fam
    .Cells(nomer, 2).Value = nam
    .Cells(nomer, 3).Value = adr
    .Cells(nomer, 4).Value = tekpok
    .Cells(nomer, 5).Value = prpok
    .Cells(nomer, 6).Value = tarif
    .Cells(nomer, 7).Value = data
    .Cells(nomer, 8).Value = rashod
    .Cells(nomer, 9).Value = summa
End With
ClearForm
End Sub

```

Примечание.

В данном коде функция **CSng** преобразует выражение в числовой тип данных **Single** для того, чтобы можно было провести арифметические операции над данными, внесенными в **Text-Box**. Обратное действие совершает функция **Str**.

Функция **Val** читает цифры символьного выражения слева направо до тех пор, пока не встретится нецифровой символ, и возвращает число.

Функция **IsNumeric** проверяет, является ли значение данного выражения числом.

Функция **IsDate** проверяет, является ли данное выражение корректной датой или временем

В данном коде использована процедура **ClearForm**, необходимая для очистки формы после добавления записи в базу данных. Вот ее код:

```
Private Sub ClearForm()
    Unload UserForm1
    UserForm1.Show
End Sub
```

Теперь обработаем нажатие на кнопку **Отмена**.

Так же перейдем в процедуру обработки события **Click** и запишем следующий код:

```
Private Sub CommandButton2_Click()
    Dim nomer As Integer
    'Вычисляем номер последней строки
    nomer = Application.CountA(ActiveSheet.Columns(1))
    'Удаляем содержимое ячеек строки
    With ActiveSheet
        If nomer > 1 Then
            .Cells(nomer, 1).Value = ""
            .Cells(nomer, 2).Value = ""
            .Cells(nomer, 3).Value = ""
            .Cells(nomer, 4).Value = ""
            .Cells(nomer, 5).Value = ""
            .Cells(nomer, 6).Value = ""
            .Cells(nomer, 7).Value = ""
            .Cells(nomer, 8).Value = ""
        End If
    End With
End Sub
```

Теперь обработаем нажатие на кнопку **Выход**.

В процедуре обработки события **Click** запишем следующий код:

```
Private Sub CommandButton3_Click()
    'Активизируем рабочий лист с именем Меню
    Sheets("Меню").Activate
    'Завершаем выполнение программы
    End
End Sub
```

Перейдем в нашу рабочую книгу. Выберем лист **Меню**. Щелкнем по кнопке «Прием платежа», появится пустая таблица с заголовками и форма для заполнения. Введем в нее значения (рис. 8.15).

Если они введены полностью и правильно, то при нажатии на кнопку «Принять» они появятся в таблице, а форма очистится и будет готова к приему новых данных.

	A	B	C	D	E	F	G	H	I	J
	Фамилия	Имя	Адрес	Текущее показание счетчика	Предыдущее показание счетчика	Тариф	Дата платежа	Расход электроэнергии	Сумма	
1										
2	Иванов	Петр	Томск, пр. Ленина, 36/3, кв. 1	12345	12000	2,15	12.03.2013	345	741,75	
3	Петров	Иван	Томск, пр. Ленина, 36/8, кв. 2	10000	8000	2	13.03.2013	2000	4000	
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										

Информация о клиенте

Фамилия:

Имя:

Адрес:

Информация о платеже

Текущее показание счетчика:

Предыдущее показание счетчика:

Тариф:

Дата платежа:

Рис. 8.15 — Заполненные форма и таблица

8.6 Построение диаграммы

Построить диаграмму на основе данных, находящихся в созданной выше табличной базе.

Вызовем в редакторе VBA созданную форму **UserForm1** и напишем процедуру отклика кнопки **Диаграмма** на щелчок по ней. Процедура должна создавать на отдельном листе диаграмму на основе данных листа **База**.

Для создания процедуры два раза щелкнем по кнопке **Диаграмма**, тем самым мы попадем в окно редактирования кода процедуры. Запишем следующий код:

Private Sub CommandButton4_Click()

' Активируем рабочий лист с именем Диаграмма

Sheets("Диаграмма").Activate

' Очищаем лист от всех объектов

For Each i In ActiveSheet.Shapes

i.Delete

Next i

' Создаем новую диаграмму

ActiveSheet.ChartObjects.Add(25, 25, 500, 300).Select

With ActiveChart

' Задаем тип диаграммы (объемная гистограмма)

.ChartType = xl3DBarClustered

' Находим, сколько записей в таблице

M = 2

Do

If Sheets("База").Cells(M, 1).Value = "" Then Exit Do

M = M + 1

Loop

M = M - 1

' Определяем источник данных для построения диаграммы:

' с листа «База» от ячейки I2 до ячейки IM

.SetSourceData

Source:=Sheets("База").Range("I2:I" + Trim(Str(M))),

PlotBy:=xlRows

' Выбираем подписи к данным из первого столбца таблицы

For i = 2 To M

.SeriesCollection(i - 1).Name = "=База!R" + Trim(Str(i)) + "C1"

Next

' Размещение диаграммы на отдельном листе

.Location Where:=xlLocationAsObject, Name:= "Диаграмма"

With ActiveChart

' Заголовок

.HasTitle = True

**.ChartTitle.Characters.Text = "Сумма оплаты _
за электроэнергию"**

' Легенда

.HasLegend = True

.Legend.Select

Selection.Position = xlLeft

.HasDataTable = False

.Axes(xlCategory).MajorTickMark = xlNone

.Axes(xlCategory).MinorTickMark = xlNone

.Axes(xlCategory).TickLabelPosition = xlNone

End With
End With
End Sub

Затем нам нужно изменить процедуру кнопки **Выход** так, чтобы при выходе активным оставался лист с диаграммой. Для этого в окне **Project-VBAProject** два раза щелкнем по **UserForm1**, затем, в появившейся форме, два раза щелкнем по кнопке **Выход**. Запишем там следующее:

```
Private Sub CommandButton3_Click()  

    'Активизируем рабочий лист с именем «Меню»  

    Sheets("Диаграмма").Activate  

    'Завершаем выполнение программы  

End  

End Sub
```

Затем перейдем в рабочую книгу и протестируем наш макрос. Запустим форму, нажмем на кнопку **Диаграмма**. На листе **Диаграмма** появится диаграмма, построенная по табличным данным (рис. 8.16).

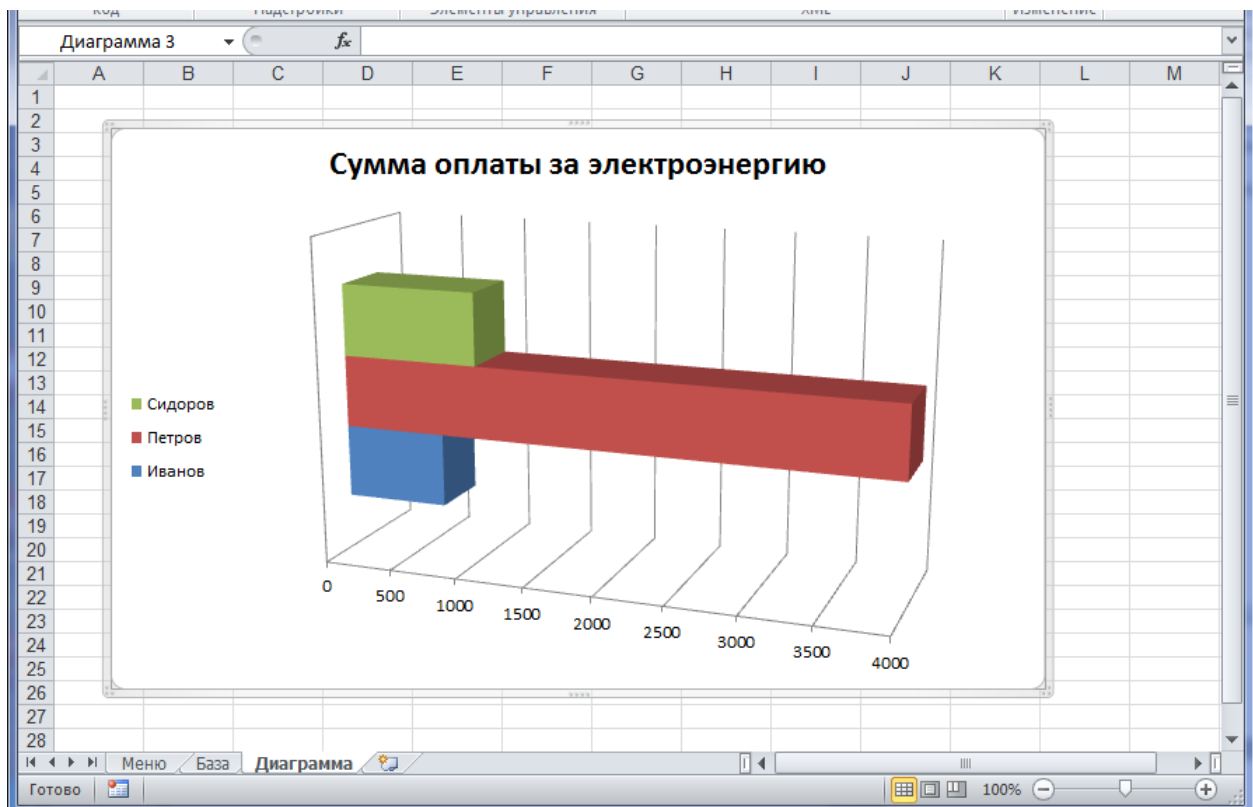


Рис. 8.16 — Лист Excel с диаграммой

Задание 8.2. Создание базы данных

Разработайте программу с удобным диалоговым окном для создания на рабочем листе базы данных в соответствии с выбранным вариантом. Выбор элементов управления, соответствующего каждому полю базы данных, оставляется на ваше усмотрение. Написать макрос формирования диаграммы данных.

Вариант 1. База данных по учету выданных кредитов.

Таблица базы данных должна содержать: фамилию, имя, отчество, адрес заемщика, срок кредита, сумму кредита, пол заемщика, сведения о залоге и его сумме (или о поручителях), дату совершения операции.

Вариант 2. База данных по учету банковских вкладов физических лиц.

Таблица базы данных должна содержать: фамилию, имя, отчество, адрес вкладчика, дату вклада, вид вклада (до востребования, срочный, выигрышный), отделение банка (названия или номера), текущую сумму на счете.

Вариант 3. База данных по учету оплаты за услуги водоснабжения и водоотведения.

Таблица базы данных должна содержать следующие поля: фамилия, имя, отчество, адрес плательщика, дата платежа, показания счетчиков холодной воды, показания счетчиков горячей воды, объем водопотребления, платежные тарифы (холодная вода, водоотведение), долг (либо переплата), сумма.

Вариант 4. База данных по учету оплаты за услуги отопления и горячего водоснабжения.

Таблица базы данных должна содержать следующие поля: фамилия, имя, отчество, адрес плательщика, дата платежа, показания счетчиков горячей воды, объем потребления горячей воды, количество кв. м жилой площади, норматив потребления тепловой энергии (Гкал/кв.м), платежные тарифы (горячая вода (руб./куб.м), тепловая энергия (руб./Гкал)), долг (либо переплата), сумма платежа.

Вариант 5. База данных по учету оплаты за жилищно-коммунальные услуги.

Таблица базы данных должна содержать поля: фамилия, имя, отчество адрес плательщика, дата платежа, площадь квартиры, количество жильцов, вид платежа (содержание имущества, текущий ремонт, уборка подъезда, лифт, домофон), стоимость за единицу услуги, долг (переплата), сумма, дата операции.

Вариант 6. База данных по учету оплаты телефонных услуг.

Таблица базы данных должна содержать поля: фамилия, имя, отчество, адрес плательщика, дата платежа, вид платежа (абонентская плата за телефон, абонентская плата за радио, услуги Интернет, межгород), стоимость за единицу услуги, долг (переплата), сумма.

Вариант 7. База данных по реализации товаров со склада.

Таблица базы данных должна содержать поля: наименование товара, артикул товара, количество товара, цена, тип продаж (оптом, в розницу), покупатель, отпущено товара, остаток, дата операции.

Вариант 8. База данных по продаже железнодорожных билетов.

Таблица базы данных должна содержать поля: тип поезда (скорый, пассажирский), наименование поезда, номер поезда, тип вагона (плацкартный, купейный, мягкий), номер вагона, место, фамилия, имя пассажира, стоимость проезда в зависимости от типа вагона и купе, дата продажи.

Вариант 9. База данных по учету комплектующих на складе компании, торгующей вычислительной техникой.

Таблица базы данных должна содержать поля: группа товаров (мониторы, процессоры, материнские платы, прочее), фирма-производитель, наименование, цена, количество, дата поступления.

Вариант 10. База данных по результатам экзаменационной сессии в университете.

Таблица базы данных должна содержать поля: номер группы, предметы (Информатика, Высшая математика, Физика, История, Дискретная математика, Физкультура, ...), фамилия, имя, отчество студента, оценка (или зачет), результат (переведен на следующий курс, назначен на стипендию, назначен на повышенную стипендию, отчислен).

ЛИТЕРАТУРА

1. Миньков С. Л. Лабораторный практикум по информатике : учеб.пособие / С. Л. Миньков, М. Н. Еременко. — Томск : ТУСУР, 2003. — 249 с.
2. Миньков С. Л. Excel:лабораторный практикум / С. Л. Миньков. — Томск : ТУСУР, 2004. — 145 с.
3. Практикум по экономической информатике : в 3 ч. : учеб. пособие / под ред. П. П. Мельникова. — М. : Финансы и статистика ; Перспектива, 2002. — Ч. 3. — 160 с.
4. Гарнаев А. Ю. Excel 2010: разработка приложений / А. Ю. Гарнаев, Л. В. Рудикова. — СПб. : БХВ-Петербург, 2011. — 528 с.
5. Моделирование в экономике : учеб. пособие / под ред. В. В. Сизова, С. Л. Минькова. — 2-е изд. — Томск : Изд-во Томск. гос. педагог. ун-та, 2004. — 336 с.