

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

# ОФИСНОЕ ПРОГРАММИРОВАНИЕ

*Методические рекомендации к практическим занятиям  
для студентов специальности 1-53 01 02 «Автоматизированные  
системы обработки информации»  
дневной формы обучения*



Могилев 2024

УДК 004.42  
ББК 32.973-018  
О91

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»  
«24» сентября 2024 г., протокол № 2

Составители: канд. техн. наук, доц. В. М. Ковальчук;  
канд. техн. наук, доц. В. А. Широченко

Рецензент канд. техн. наук, доц. С. К. Крутолевич

В методических рекомендациях к практическим занятиям по дисциплине  
«Офисное программирование» приведены основные теоретические сведения,  
практические задания, контрольные вопросы и список литературы.

Учебное издание

## ОФИСНОЕ ПРОГРАММИРОВАНИЕ

Ответственный за выпуск	А. И. Якимов
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2024

## Содержание

1 Практическое занятие № 1. Создание простого офисного приложения .....	4
2 Практическое занятие № 2. Автоматизация создания отчетов в текстовых документах.....	7
3 Практическое занятие № 3. Автоматизация обработки данных в электронных таблицах .....	9
4 Практическое занятие № 4. Автоматизация создания презентаций для презентаций и отчетов .....	12
5 Практическое занятие № 5. Автоматизация работы с данными из базы данных .....	18
6 Практическое занятие № 6. Интеграция пользовательских форм в офисные приложения .....	23
7 Практическое занятие № 7. Автоматизация обмена данными между различными формами файлов.....	28
8 Практическое занятие № 8. Практические примеры оптимизации работы офисных приложений .....	30
Список литературы .....	33

# 1 Практическое занятие № 1. Создание простого офисного приложения

**Цель занятия:** изучение *Visual Basic for Application* создания простого офисного приложения.

## Основные теоретические положения

Корпорация Microsoft интегрировала в свои офисные продукты язык программирования Visual Basic for Applications (VBA). С помощью этого языка каждый пользователь может автоматизировать работу приложения и максимально приспособить его работу для решения текущих задач.

Код VBA набирается в редакторе Visual Basic. Для того чтобы попасть в этот редактор, выберите в MS Excel команду *Сервис | Макрос | Редактор Visual Basic* или нажмите комбинацию клавиш *Alt + F11*. В результате откроется интегрированная среда разработки приложений (рисунок 1.1.).

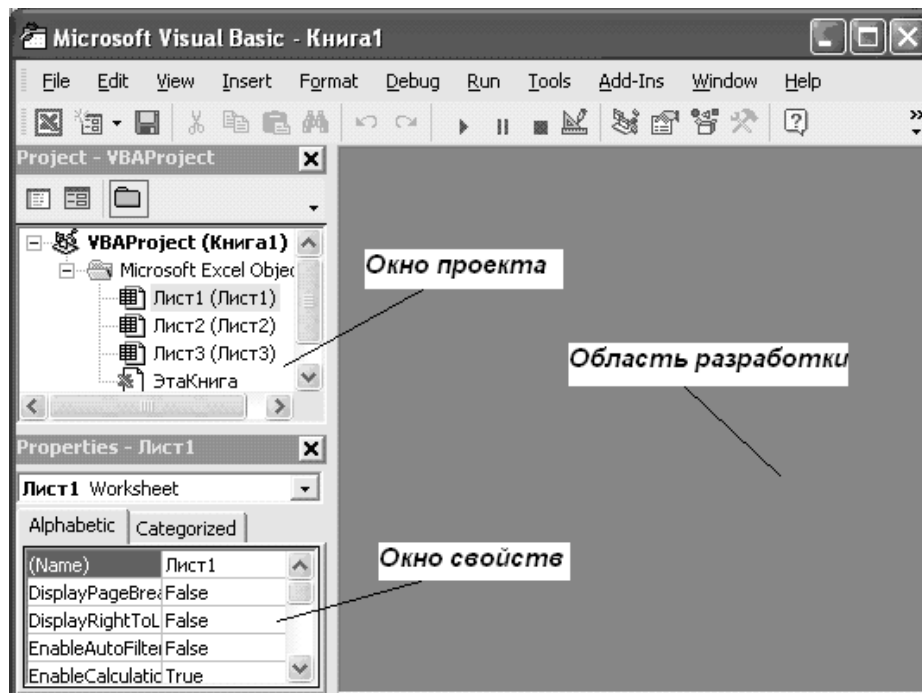


Рисунок 1.1 – Интегрированная среда разработки приложений

Возвратиться из редактора Visual Basic в рабочую книгу можно, нажав кнопку с пиктограммой Excel.

Все приложения, написанные на VBA, создаются как проекты. В проект входят несколько модулей: код программы, параметры формы, конфигурация интегрированной среды разработки и др.

Среда разработки приложений имеет стандартный для Windows-приложений вид: строка меню, панель инструментов и еще несколько открытых окон. На рисунке 1.1 открыты два окна: *Project – VBA Project* и *Properties*.

Окно *Project – VBA Project* активизируется выбором команды *View | Project Explorer (1)* или нажатием кнопки *Project Explorer(2)* панели инструментов. В окне *Project – VBAProject* представлена иерархическая структура форм и модулей текущего проекта. В этом окне отображается реестр модулей и форм, входящих в создаваемый проект. Двойным щелчком на значке модуля в окне *Project – VBA Project* можно открыть соответствующий модуль.

В проекте автоматически создается по модулю для каждого рабочего листа и для всей книги. Кроме того, можно создавать и другие модули.

При написании кода программы необходимо придерживаться следующей структуры:

```
SubИмяПрограммногоМодуля ()
Объявление переменных и констант
Тело программы (последовательность исполняемых операторов)
EndSub
```

Переменные в программе используются для хранения данных, которые могут изменяться в процессе выполнения процедуры. Объявление типа переменной означает, что пользователь устанавливает определённые границы, в которых может изменяться переменная. Тип переменной можно вообще не определять. Если тип переменной не объявляется, по умолчанию он принимается как тип *Variant*. В таблице 1.1 содержится информация о размере данных, т. е. об объёме памяти, выделяемом для хранения данных.

Таблица 1.1 – Основные типы переменных языка VBA

Тип	Содержание	Объем	Диапазон значений
Byte	Короткое неотрицательное число	1 байт	0...255
Integer	Целое число	2 байта	-32768...+32767
Long	Длинное целое число	4 байта	-2147483648... +2147483647
Single	Десятичное число обычной точности	4 байта	$\pm(1,4011298e - 45... 3,402823e + 38)$
Double	Десятичное число двойной точности	8 байт	$\pm(1e - 324...1e + 308)$
String	Набор символов (строка)	Зависит от числа символов в строке	
Boolean	Логическая величина	2 байта	Истина (True) или Ложь (False)
Date	Дата	8 байт	Информация о дате
Object	Указатель на объект	4 байта	Значением является ссылка на объект
Variant	Произвольное значение	Не менее 16 байт	Может быть переменной любого типа

Для объявления переменной используется оператор *Dim*. Этот оператор имеет следующий синтаксис:

### *DimИмяПеременнойAsТипДанных*

Имена переменных должны начинаться с буквы, могут содержать также цифры и знаки подчёркивания. Имя не может содержать пробелы, точки, запятые, восклицательные знаки и символы @, &, \$, # и не должно иметь более 255 символов.

Если в вычислениях нужна величина, которая бы не меняла своего значения, то применяются константы. Для их объявления используется оператор Const, имеющий следующий синтаксис:

*ConstИмяКонстантыAsТипДанных = Значение*

Например:

*ConstGruppaAsInteger = 25*

Для задания переменным исходных значений используется функция InputBox, а для вывода значений переменных на экран – процедура MsgBox. Синтаксис их написания можно посмотреть в справочной системе при работе с кодом программы с помощью клавиши *F1*.

Для вычисления значений переменных в программе используется оператор присваивания, имеющий следующий синтаксис:

*Переменная = Выражение*

Если *Выражение* математическое, то в нем используются знаки арифметических действий, математические функции и круглые скобки, например:  $A = 2 * \sin(x + 2) / \cos(x - 2)$ .

### **Задание**

1 Составить необходимый код программы с обеспечением ввода исходных данных и вывода результата на экран.

2 Запустить программу на выполнение и сравнить полученный результат со значением, вычисленным другим способом, например с помощью калькулятора.

### **Контрольные вопросы**

1 Как объявить переменную, которая может иметь целые значения в диапазоне от 0 до 100?

2 Как в программе присвоить значение переменной?

3 Как показать результат вычислений?

## 2 Практическое занятие № 2. Автоматизация создания отчетов в текстовых документах

**Цель занятия:** изучение методов создания отчетов в текстовых документах.

### *Основные теоретические положения*

Существуют задачи, требующие подготовки большого количества экземпляров документа, в каждом из которых отличаются лишь некоторые сведения, например: фамилия, должность, адрес, номер телефона и т. д. В качестве примеров таких документов можно привести: приглашение на конференцию, различные повестки, справки и т. д.

В редакторе MS Word для этих целей используется вкладка *Рассылки*, включающая группы *Создать*, *Начать слияние*, *Просмотр результатов*, *Составление документа и вставка полей* и *Завершить*.

Документ с источником данных состоит из двух файлов: основного документа (рисунок 2.1) и источника данных (списка получателей) (таблица 2.1). Основной документ представляет собой обычный текстовый документ Word, содержащий поля для вставки значений, обычно текстовых, из источника данных.

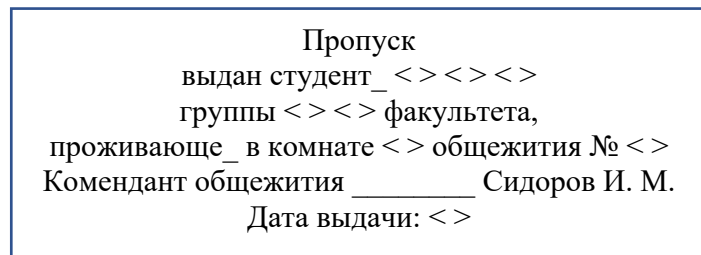


Рисунок 2.1 – Вид основного документа

Символы "< >" указывают места вставки в документ элементов текста из источника данных, а "\_" обозначают окончания слов. Вводить в документ эти символы не нужно.

Источник данных – это файл таблицы, содержащей строку заголовка, и соответствующие ей столбцы со значениями. Источник данных может быть подготовлен в виде таблицы Word, Excel или Access. Например, для приведенного выше основного документа источник данных может иметь следующий вид (см. таблицу 2.1).

Чтобы создать источник данных (список получателей), необходимо на вкладке *Рассылки* в группе *Начать слияние* выбрать элемент *Выбрать получателей* и далее – команду *Ввести новый список*. Затем, используя кнопку *Настройка столбцов*, отредактировать структуру предложенной таблицы, ввести в нее данные из таблицы 2.1, выполнить команду *Использовать существующий список ...*, найти его и нажать *Ok*.

Таблица 2.1 – Источник данных

Фамилия	Имя	Отчество	Пол	Группа	Факультета	Номер общежития	Комната	Дата
Ивановой	Инне	Сергеевне	м	КР-061	Экономического	3	617	01.09.07
Петровой	Анне	Ивановне	м	КД-052	Экономического	1	235	05.09.07
Крылову	Олегу	Петровичу	ж	ТМ-051	Механического	2	221	02.09.07
Сергееву	Ивану	Ильичу	м	КР-051	Экономического	3	442	01.09.07
Вавилову	Петру	Петровичу	м	СД-042	Строительного	1	524	02.09.07

Если источник данных уже подготовлен, то следует выбрать команду *Открыть источник данных*, затем найти его, открыть и, если необходимо, отредактировать.

Следующим шагом слияния является вставка *полей Слияния* и *полей Word* в основной документ. Для вставки поля слияния следует поместить курсор в место вставки, в группе *Составление документа и вставка полей* выбрать элемент *Вставить поле слияния*, выбрать необходимое поля вставки и нажать кнопку *Ок*. Аналогично ввести все поля слияния.

Чтобы вставить в основной документ правильные окончания в словах *студент* (студенту или студентке) и *проживающе* (проживающему или проживающей), следует установить курсор после буквы *т* в слове *студент*, воспользоваться элементом *Правила* в группе *Составление документа и вставка полей*, выбрать команду *IF .. THEN .. ELSE*, затем – поле *пол*, оператор *равно* и значение (*м* или *ж*), а в полях *Вставить следующий текст* и *В противном случае вставить следующий текст* ввести соответствующие окончания слов: "у" и "е" соответственно.

После этого можно просмотреть результаты слияния, щелкнув на элементе *Просмотр результатов*.

Следующим этапом слияния документа является сохранение результатов слияния в новый документ. Для этого на вкладке *Завершить* группы *Рассылки* следует выбрать команду *Изменить отдельные документы*.

### **Задание**

- 1 По указанию преподавателя получить задание.
- 2 Построить документ с источником данных, вставить в него необходимые поля слияния, построить окончания слов и выполнить операцию объединения.

### **Контрольные вопросы**

- 1 Что понимается под слиянием документов в редакторе MS Word?
- 2 Как создать основной документ для слияния?



3 Что такое источник данных и в среде какого приложения его можно подготовить?

4 Как подключить источник данных к основному документу?

5 Как просмотреть результаты слияния в основном документе?

6 Как сохранить результаты слияния документа и что является результатом слияния?

### 3 Практическое занятие № 3. Автоматизация обработки данных в электронных таблицах

**Цель занятия:** изучение методов автоматизации обработки данных в электронных таблицах

#### *Основные теоретические положения*

В качестве примера попытаемся решить следующую задачу. Допустим, вы решили вести учет своих расходов и с этой целью в конце каждого месяца намерены составлять таблицу (рисунок 3.1) и строить диаграмму для более наглядного отображения доли каждой статьи расходов вашего бюджета. Составлять ежемесячно одну и ту же таблицу с одновременным построением диаграммы – довольно непроизводительная трата времени. Более разумно один раз научить компьютер создавать таблицу, а потом по мере необходимости лишь отдавать команду подготовки таблицы, чтобы осталось только внести в нее данные.

Для обучения компьютера отлично подходит Macro Recorder – транслятор, создающий программу (макрос) на языке VBA, которая является результатом перевода на язык VBA действий пользователя с момента запуска Macro Recorder до окончания записи макроса.

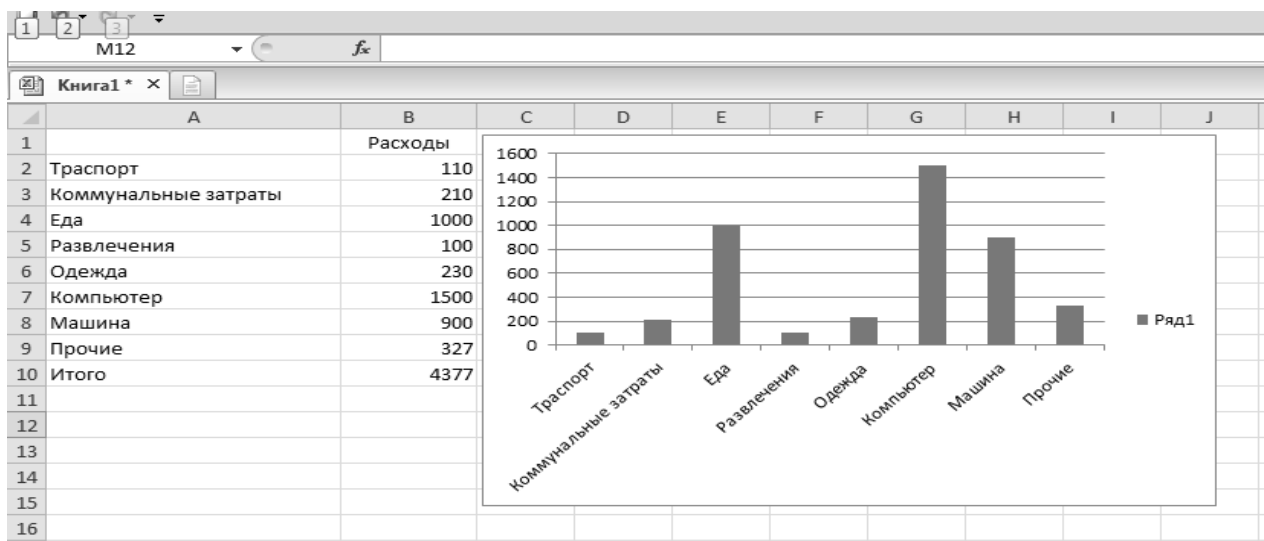


Рисунок 3.1 – Таблица ежемесячных расходов

Для активизации Macro Recorder выберите закладку *Вид* → *Макросы* → *Запись макроса*. Появится диалоговое окно *Запись макроса*. Это диалоговое окно позволяет задать параметры макроса. В диалоговом окне *Запись макроса* в поле *Имя макроса* введем *Расходы*, а в поле *Описание* – *Расчет месячных расходов*.

Присвоим макросу комбинацию клавиш быстрого вызова:

- *сочетание клавиш* – в расположенном рядом поле введем букву, например, *r* (вызов макроса будет осуществляться при нажатии клавиш *Ctrl + r*);
- сохраним макрос в текущей рабочей книге, установим в разделе *Сохранить в Эта книга*.

Нажмем кнопку *ОК*. Появится кнопка *Остановить запись*. Теперь все производимые действия будут записываться до тех пор, пока не будет нажата эта кнопка. Построим шаблон таблицы расходов по следующему алгоритму:

- активизируем ячейку A2 и введем в нее *Транспорт*;
- активизируем ячейку A3 и введем в нее *Коммунальные*;
- активизируем ячейку A4 и введем в нее *Еда*;
- активизируем ячейку A5 и введем в нее *Развлечения*;
- активизируем ячейку A6 и введем в нее *Одежда*;
- активизируем ячейку A7 и введем в нее *Компьютер*;
- активизируем ячейку A8 и введем в нее *Машина*;
- активизируем ячейку A9 и введем в нее *Прочие*;
- активизируем ячейку A10 и введем в нее *Итого*;
- активизируем ячейку B10 и введем в нее формулу  $=\text{СУММ}(B2:B9)$ , вычисляющую суммарные расходы;
- выберем диапазон B2:B9 и при помощи вкладки *Границы* панели инструментов *Главная* создадим рамку, окаймляющую этот диапазон;
- выберем столбец A, изменим его ширину так, чтобы введенный в диапазон A2:A9 текст помещался в этом столбце;
- выберем диапазон A2:B9 и при помощи мастера диаграмм, вызываемого кнопкой *Диаграмма* панели инструментов *Вставка*, создадим диаграмму.

Рабочий лист теперь будет выглядеть так, как показано на рисунке 3.1. Остановим запись макроса, нажав кнопку *Остановить запись* (Stop Recording). Заполним ячейки таблицы исходными данными. Расчет суммарных расходов и построение диаграммы теперь будет происходить автоматически.

Для просмотра записанной процедуры необходимо выбрать закладку *Вид* → *Макросы*, которая вызовет диалоговое окно *Макрос*.

В этом диалоговом окне в списке выделим макрос и нажмем кнопку *Изменить*. Это вызовет появление главного окна редактора VBA.

Для запуска созданного макроса необходимо перейти на новый рабочий лист или очистить текущий, затем выбрать закладку *Вид* → *Макросы*, которая вызовет диалоговое окно *Макрос*. В этом окне в списке выделим исходный макрос и нажмем кнопку *Выполнить*. Диалоговое окно закроется и выполнится процедура, создающая на активном рабочем листе шаблон таблицы. Теперь в нее остается ввести новые данные, а расчет суммарных расходов и построение

диаграммы будет происходить автоматически.

Также для запуска созданного макроса можно нажать комбинацию клавиш *Ctrl + r*. Excel запустит макрос, который последовательно выполнит все записанные действия.

Пользователю предоставляется возможность отредактировать существующий макрос. Для изменения макроса используют закладку *Вид* → *Макросы* → *Макросы* → *Изменить* непосредственно на листе модуля.

### **Задание**

- 1 Выполнить разобранный выше пример.
- 2 Проанализировать поставленную задачу (см. варианты заданий) и определить, данные какого типа содержит данная таблица.
- 3 Разработать макрос для создания заголовка таблицы и для задания формата ячеек каждого типа данных. Текстовая информация должна вводиться в ячейку в несколько строк, стоимость или цена – с указанием единиц.
- 4 Рассмотреть различные варианты возможностей вызова макроса.

### **Варианты заданий**

- 1 Название фирмы, дата создания, годовой доход в рублях, число сотрудников, телефон с кодом города.
- 2 Фамилия студента, номер зачетной книжки, дата рождения, адрес, телефон.
- 3 Заболевание, код заболевания, количество заболевших в текущем году, количество заболевших в прошедшем году, процент роста.
- 4 Фамилия абонента, телефон, дата разговора, тариф, сумма в рублях.
- 5 Наименование товара, код товара (страна – код), дата использования, цена в рублях.
- 6 Город, номер телефона, дата разговора, время разговора (часы и минуты).
- 7 Наименование товара, дата поставки, стоимость в рублях, стоимость в рублях.
- 8 Наименование детали, код детали, код цеха-производителя, количество.
- 9 Наименование строительных работ, объем, расценка в рублях, стоимость, телефон подрядчика.
- 10 Марка автомобиля, регистрационный номер, стоимость в рублях, телефон владельца.
- 11 Фамилия сотрудника, табельный номер, дата поступления, телефон.
- 12 Фамилия студента, название предметов и оценка по предмету, средний балл, процент хороших и отличных оценок.
- 13 Наименование товара, цена в долларах, дата поставки, курс доллара, цена в рублях.

### **Контрольные вопросы**

- 1 Что такое макрос?
- 2 Как начать запись макроса?

3 Можно ли просмотреть или изменить записанный макрос?

4 Перечислите способы запуска макроса.

5 Какие параметры макроса позволяет задать диалоговое окно *Запись макроса*?

## 4 Практическое занятие № 4. Автоматизация создания презентаций для презентаций и отчетов

**Цель занятия:** изучение методов автоматизации создания презентаций.

### *Основные теоретические положения*

В бизнесе, обучении, представлении научных достижений все более широкое применение получают ясные, привлекающие внимание и убедительные *презентации*. Они создаются средствами специальных программ, которые называются *пакетами для создания динамических презентаций*.

На современном рынке пакетов для создания динамических презентаций представлены средства, которые можно разделить на две группы: средства создания презентаций непрофессиональным пользователем (например, PowerPoint из офисного пакета Microsoft Office фирмы Microsoft); средства, ориентированные в основном на профессионалов и предоставляющие развитые возможности (например, Astound фирмы Gold Disk).

Пакет динамических презентаций PowerPoint (в дальнейшем – PowerPoint) – один из лучших пакетов для подготовки и проведения презентаций. PowerPoint имеет следующие возможности:

- создание и отображение электронных презентаций, в которых текст сочетается с графическими объектами, картинками, фотографиями, звуком, видео и мультипликационными эффектами;
- настройка показа презентации (проведение показа пользователем, в автоматическом режиме и др.);
- перенос презентации на 35-миллиметровые слайды, прозрачные пленки или печатные материалы для выдачи слушателям;
- распространение презентаций через сеть Интернет, создание сайтов для Всемирной паутины (World Wide Web);
- тесная интеграция с другими приложениями офисного пакета Microsoft Office (тексты Word, листы Excel).

*Презентация PowerPoint* – это набор слайдов по определенной тематике, собранных в слайд-фильм, обычно оформленных в едином стиле, с соответствующим сопроводительным материалом (план презентации, замечания докладчика, материалы для раздачи слушателям).

*Слайд* представляет собой сложный объект, который может включать заголовки, текст, таблицы, графические объекты, схемы организации, звуковые фрагменты, видеоклипы и гиперссылки. Каждый слайд сопровождается

страницей заметок, на которую можно заносить поясняющий текст как во время создания, так и при его демонстрации.

*Гиперссылка* – фрагмент текста, который является указателем на другой слайд, файл или объект. Гиперссылки необходимы для того, чтобы обеспечить возможность перехода от одного документа к другому (документы могут находиться на различных компьютерах в сети). Гиперссылка – это выделенный фрагмент текста, при наведении на который указателя мыши указатель принимает вид «руки».

Слайды оформляются на основе образцов. В каждой презентации можно использовать три вида образцов – образец слайдов, образец выдач, образец заметок.

*Образец слайдов* задает расположение и некоторые параметры слайда: шрифт, размер и цвет, цвет фона, заливку, стиль маркера списка и др. Он содержит пустые рамки для текста и колонтитулов, включающих дату, время и номер слайда, и управляет оформлением всех слайдов, кроме титульного. Любое изменение образца отражается на каждом слайде презентации, например при добавлении графического рисунка или текста в образец они появятся на всех слайдах презентации. С другой стороны, если созданы уникальные слайды (например, использованы цвета или заливки, отличные от цветовой схемы образца слайдов), они сохраняют свое оформление, несмотря на изменения образца слайдов.

*Образец выдач* задает параметры распечатки слайдов, *образец заметок* содержит параметры оформления заметок к слайдам.

*Демонстрация презентации* представляет собой процесс показа слайдов в некотором порядке, при этом смена слайдов может осуществляться вручную или автоматически.

Общая технология подготовки и редактирования презентации:

- 1) создать или открыть презентацию;
- 2) сформировать содержимое слайда;
- 3) выбрать макет слайда (*макет* используется для автоматизации размещения и формирования объектов в слайде);
- 4) в метки-заполнители ввести необходимую информацию.

Выбрать шаблон оформления, цветовую схему. *Шаблон оформления* содержит цветовое оформление, образцы слайдов и заголовков с настраиваемыми форматами и стилизованные шрифты, определяющие вид презентации. Выбранный шаблон по умолчанию применяется ко всем слайдам презентации. *Цветовая схема* состоит из восьми гармонично подобранных цветов, используемых в качестве основных для текста, фона, заливки, акцентов и т. п. Каждый цвет схемы используется автоматически для различных элементов слайда.

Настроить анимацию объектов на слайде. *Анимация* – это порядок и эффект появления объектов на слайде или смены слайда во время демонстрации.

В режиме заметок задать комментарии к слайдам.

Настроить переходы слайдов.

Настроить демонстрацию готовой презентации.

Просмотреть презентацию.

Если необходимы изменения, вернуться к одному из предыдущих пунктов.

Программа PowerPoint позволяет открыть уже готовую презентацию или создать новую с помощью команд *Файл / Открыть*, *Файл / Создать* или с использованием команд *Открыть*, *Создать* на панели *Область задач*.

Существует два способа создания новой презентации:

1) *с помощью мастера автосодержания* (поэтапное создание новой презентации пользователем-новичком);

2) *новая презентация* (формирование презентации «с нуля», способ предназначен для работы профессионального пользователя).

С помощью мастера автосодержания можно выбрать тип создаваемой презентации, стиль ее оформления, продолжительность и тип выдачи презентации. По указанным пользователем параметрам мастер автосодержания построит предварительную схему презентации с размеченными слайдами. Разметка слайда представляет собой схему размещения меток-заполнителей (заголовка, текста, таблицы, графического объекта). Пользователю необходимо изменить содержимое слайдов в соответствии со своими потребностями. Метки-заполнители после щелчка (для ввода текста) или двойного щелчка (для ввода графических объектов) открываются для редактирования, и пользователь вводит в них свой текст или вставляет нужный ему объект. Постепенно заменяя содержимое меток-заполнителей на всех слайдах сформированной последовательности, пользователь получит готовую презентацию.

Для обеспечения удобства работы пользователя в программе PowerPoint существуют следующие режимы работы с презентациями:

– *режим слайда* (просмотр или изменение текущего слайда);

– *режим структуры* (показ общей структуры – плана, сценария – презентации, что позволяет быстро изменить текстовую ее часть);

– *режим страницы заметок* (для заполнения или просмотра страниц заметок для каждого слайда);

– *режим сортировщика слайдов* (на экране отображаются миниатюры всех слайдов презентации).

Переход из одного режима в другой осуществляется с помощью команды *Вид*.

Оформление слайда удобнее всего осуществлять в режиме слайдов.

Порядок оформления слайда:

1) выбрать макет слайда на панели *Разметка слайда* (команда *Формат / Разметка слайда*);

2) выбрать шаблон оформления, цветовую схему и эффект анимации при переходе к следующему слайду на панели *Дизайн слайда* (команда *Формат / Оформление слайда*);

3) в метки-заполнители ввести необходимую информацию;

4) настроить анимацию объектов на слайде.

С помощью команды *Показ слайдов / Настройка анимации* можно установить все анимационные эффекты для слайда, например обеспечить появление текста по буквам, словам или абзацам. Графические изображения и другие объекты (диаграммы, кино) могут появляться постепенно; также возможна

анимация элементов диаграммы. Существует возможность изменять порядок возникновения объектов на слайде и устанавливать время показа каждого объекта.

Режим сортировщика слайдов позволяет:

- копировать, перемещать слайды, перетаскивая их с помощью мыши;
- удалять слайды, выделяя их и нажимая клавишу *Delete*;

– скрывать слайды на время демонстрации, выбирая команду *Скрыть слайд* контекстного меню слайда (вызов контекстного меню объекта – щелкнуть правой кнопкой мыши по объекту); чтобы показать скрытый слайд, это действие надо повторить.

Настройка демонстрации презентации осуществляется командой *Показ слайдов / Настройка демонстрации*.

Если при показе слайдов их смена осуществляется по времени, то время должно быть установлено с помощью команды *Показ слайдов / Настройка времен*.

Можно установить интервалы показа слайдов вручную. Для этого необходимо в режиме сортировщика выделить слайд, для которого устанавливается интервал показа, выполнить команду *Показ слайдов / Смена слайдов*, отменить переход слайдов по щелчку мыши (снять флажок *По щелчку* в разделе *Смена слайда*), в разделе *Смена слайда* установить флажок *Автоматически после*, указать интервал показа слайда на экране в секундах, повторить процедуру для каждого слайда или для применения интервала ко всем слайдам нажать кнопку [*Применить ко всем*].

Когда настройка демонстрации презентации завершена, просмотр готовой презентации осуществляется с помощью команды *Показ слайдов / Начать показ*.

Демонстрация готовой презентации – очень важный режим, особенно при выступлении перед аудиторией. Для того чтобы провести выступление более эффектно, можно использовать команды контекстного меню, которое вызывается правым щелчком по слайду в момент демонстрации. Демонстрацию слайдов удобнее осуществлять не с помощью команд контекстного меню, а с помощью сочетания клавиш. Описание этих команд приведено в таблицах 4.1 и 4.2.

Дополнительные возможности программы PowerPoint представлены в таблице 4.3.

### **Задание**

Подготовить презентацию. В презентации к каждому слайду оформить заметки, настроить переходы слайдов, анимацию объектов на слайдах, создать итоговый слайд из заголовков других слайдов, вставить гиперссылки из итогового слайда на все остальные слайды и кнопки возврата со слайдов на итоговый слайд. Настроить демонстрацию готовой презентации.

Таблица 4.1 – Команды контекстного меню для управления демонстрацией презентации

Команда	Функция
<i>Вперед</i>	Переход к следующему объекту на слайде
<i>Назад</i>	Переход к предыдущему объекту на слайде
<i>Перейти</i>	Вызывает навигатор слайдов, с помощью которого можно указать следующий слайд, на который надо перейти
<i>Записная книжка или Заметки докладчика</i>	Вызывает соответствующее окно, в котором можно прочитать заметки или замечания, которые приготовлены заранее, или вспомнить о тех действиях, которые надо предпринять
<i>Указатель</i>	Позволяет выбирать вид указателя мыши. Указателя мыши вообще может не быть на экране, чтобы не портить вид презентации, или он может быть в виде <i>стрелки</i> и им можно пользоваться вместо указки, или он может быть в виде <i>пера</i> («карандаша»), которым можно делать пометки: что-то подчеркивать, обводить и т. д. После окончания демонстрации след «карандаша» пропадает
<i>Экран</i>	Позволяет остановить демонстрацию, вызвав <i>паузу</i> (тогда на экране зафиксировается текущий слайд) или <i>черный экран</i> (чтобы информация на экране не отвлекала аудиторию от выступления докладчика). Продолжить демонстрацию можно щелчком мыши на поле экрана. Здесь можно также убрать след «карандаша» на слайде
<i>Завершить демонстрацию</i>	Демонстрация завершается

Таблица 4.2 – Управление демонстрацией презентации с помощью клавиш

Клавиша	Функция
<i>N, Enter, Page Down, Стрелка вправо, Стрелка вниз, Пробел</i> или щелчок мышью	Показ следующего эффекта анимации или перехода к следующему слайду
<i>P, Page Up, Стрелка влево, Стрелка вверх</i>	Показ предыдущего эффекта анимации или перехода к предыдущему слайду
<i>S</i> или <i>Плюс</i>	Остановка или возобновление автоматического показа
<i>Esc, Ctrl + Break</i> или <i>минус</i>	Завершение показа
<i>E</i>	Удаление примечаний с экрана, сделанных «пером»
<i>Ctrl + P</i>	Вывод скрытого указателя и/или установка «пера» в качестве указателя
<i>Ctrl + A</i>	Вывод скрытого указателя и/или установка стрелки в качестве указателя
<i>Shift + F10</i> или щелчок правой кнопкой мыши	Вызов контекстного меню



Таблица 4.3 – Дополнительные возможности программы PowerPoint

Операция	Выполнение
Копирование слайдов из другой презентации	Выделить слайд, после которого следует вставить другие слайды, выполнить команду <i>Вставка / Слайды из файлов</i> ⇒ выбрать презентацию, из которой копируются слайды ⇒ выделить копируемые слайды, затем нажать кнопку <i>Вставить</i> (чтобы скопировать презентацию полностью, нажать кнопку <i>Вставить все</i> )
Создание нового слайда (итогового) из заголовков других слайдов	В режиме сортировщика или в режиме структуры необходимо выделить слайды, заголовки которых будут использоваться ⇒ на панели инструментов <i>Сортировщик слайдов</i> или <i>Структура</i> нажать кнопку <i>Итоговый слайд</i> или использовать «горячие клавиши» <i>Alt + Shift + S</i>
Выбор цвета фона, способа заливки и текстуры фона	Выбрать команду <i>Формат / Фон</i> ⇒ в группе <i>Заливка фона</i> из поля со списком выбрать: – для изменения цвета фона на цвет, входящий в цветовую схему, выбрать один из восьми цветов в группе <i>Авто</i> ; на цвет, не входящий в цветовую схему, выбрать <i>Другие цвета</i> ; – для установки способа заливки и текстуры фона выбрать <i>Способы заливки</i> ; перейти на одну из страниц и выбрать нужные параметры
Применение гиперссылок	Выделить текст или объект, представляющий гиперссылку, выполнить команду <i>Показ слайдов / Настройка действия</i> , выбрать способ перехода: страницу <i>По щелчку мыши</i> или страницу <i>По наведению указателя мыши</i> (чтобы назначить объекту более одного действия, например, переход по гиперссылке и звук, следует установить гиперссылку на странице <i>По щелчку мыши</i> , а звук на странице <i>По наведению указателя мыши</i> ), установить флажок <i>Перейти по гиперссылке</i> , выбрать место назначения гиперссылки, установить другие необходимые параметры
Создание кнопки действия	Выбрать на панели инструментов <i>Рисование</i> команду <i>Автофигуры / Управляющие кнопки</i> ⇒ выбрать нужную кнопку, щелчком мыши в поле слайда установить ее местоположение, в результате чего в слайд будет вставлена кнопка стандартного размера, принимаемого по умолчанию. В диалоговом окне <i>Настройки действия</i> определить назначение кнопки (для каждой кнопки предусмотрено действие по умолчанию: при нажатии кнопки <i>Отмена</i> все связанные с данной кнопкой действия отменяются). Определив порядок использования кнопки, можно ее перенести, а также изменить местоположение или размеры

### **Контрольные вопросы**

- 1 Назовите функциональные возможности программы PowerPoint.
- 2 Что такое слайд? Какие компоненты в нем можно выделить?
- 3 Какие способы создания презентации существуют в программе PowerPoint? Какие возможности они предоставляют?
- 4 Как осуществляется настройка переходов слайдов?
- 5 Каким образом настроить анимацию объектов слайда?
- 6 Как создать итоговый слайд из заголовков других слайдов?
- 7 Каким образом создать гиперссылку?
- 8 Как создаются активные кнопки на слайдах?

9 Как настроить демонстрацию готовой презентации?

10 Какими средствами располагает программа PowerPoint для управления показом презентации?

## **5 Практическое занятие № 5. Автоматизация работы с данными из базы данных**

**Цель занятия:** изучение методов автоматизации работ с базами данных.

### ***Основные теоретические положения***

Объектами базы данных в СУБД Access являются:

- таблицы;
- запросы;
- формы;
- отчеты;
- макросы;
- модули.

*Таблицы* предназначены для хранения информации. Таблицы состоят из полей (столбцов) и записей (строк). С помощью таблиц можно вводить и редактировать данные, сортировать и фильтровать, выполнять вычисления в полях, изменять внешний вид отображения данных.

*Запросы* используются для отбора данных из таблиц по определенным критериям и выполнения операций над данными в таблицах. Существует несколько типов запросов, каждый из которых предназначен для определенной цели. Например, запрос на выборку служит для отбора данных или для создания новой таблицы. Запрос на изменение служит для изменения данных в источнике данных. Способы создания запросов зависят от их типа.

*Формы* как объект базы данных используются для ввода, просмотра и редактирования данных в удобном для пользователя виде. Формы могут применяться для управления доступом к данным: с их помощью можно определять, какие поля или строки данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только нужные им поля, это облегчит для них использование базы данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы. Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных.

*Отчеты* позволяют выводить на экран или на печать информацию из базы данных в виде, удобном для восприятия и анализа. В отчетах можно использовать такие способы анализа данных, как сортировка, группировка, расчет итогов по группам записей.

*Макросы* и *модули* предназначены для автоматизации часто выполняемых операций, но делают это по-разному: макросы – без написания программ,

а модули – с использованием языка программирования VBA. Макросы позволяют добавлять функциональные возможности в формы, отчеты и элементы управления без необходимости написания программного кода на языке VBA (Visual Basic for Applications). Для большинства пользователей создание макроса оказывается проще, нежели написание кода VBA, которое требуется для создания модуля.

### Задание

1 Создать макрос для открытия одной из ранее созданных форм (рисунок 5.1).


	<ol style="list-style-type: none"> <li>1. На вкладке <b>[Создание]</b> в группе Макросы и код выбрать команду <b>Макрос</b>. СУБД Access перейдет в режим Конструктора макроса.</li> <li>2. Из списка <b>Добавить новую макрокоманду</b>  выбрать макрокоманду <b>ОткрытьФорму</b>. В списке <b>Имя формы</b> выбрать одну из ранее созданных форм. Остальные аргументы оставить без изменения. Сохранить макрос, дать ему имя <b>Макрос_1</b>.</li> <li>3. На вкладке <b>[Файл]</b> в группе Сервис выбрать <b>Выполнить</b>. При этом должна открыться форма. <b>Закрыть форму, закрыть макрос</b>.</li> </ol>
--	--

Рисунок 5.1 – Пример создания макроса 1

2 Создать макрос для открытия одного из ранее созданных отчетов. Назначить макрос клавишам, с помощью которых он будет выполняться (например, *Ctrl* и *A*) (рисунки 5.2 и 5.3).

	<ol style="list-style-type: none"> <li>1. По аналогии с заданием 1.</li> <li>2. В группе Показывать или Скрыть включить <b>Каталог макрокоманд</b>.</li> <li>3. В <b>Каталоге макрокоманд</b> выбрать <b>Вложенный макрос</b>. В поле <b>Вложенный макрос</b> указать условное обозначение клавиш, с помощью которых он будет запускаться (если это клавиши <b>Ctrl</b> и <b>A</b>, то условное обозначение будет <b>^A</b>).</li> <li>4. Из списка <b>Добавить новую макрокоманду</b> выбрать макрокоманду <b>Открыть Отчет</b>. В списке <b>Имя отчета</b> выбрать один из ранее созданных отчетов. Остальные аргументы оставить без изменения (см. рисунок ниже). Сохранить макрос, дать ему имя <b>AutoKeys</b>. <b>Закрыть макрос</b>.</li> <li>5. Нажать клавиши <b>Ctrl</b> и <b>A</b>. При этом должен открыться отчет. <b>Закрыть отчет</b>.</li> </ol>
--	--

Рисунок 5.2 – Пример создание макроса 2

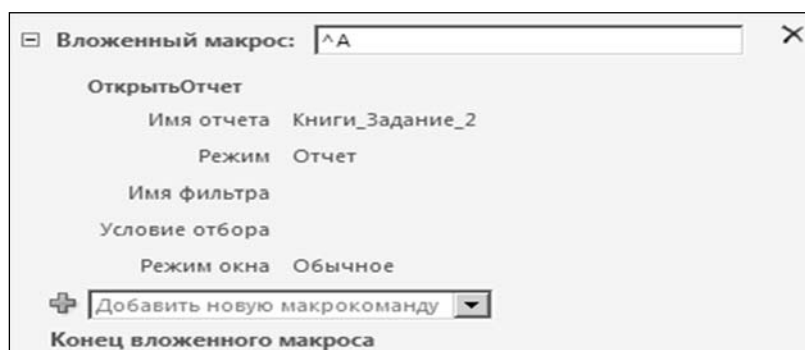


Рисунок 5.3 – Пример создания отчета макросом 2

3 Создать макрос, открывающий одну форму в момент открытия другой формы (формы будут предварительно созданы и будут называться *Книги\_для\_макроса* и *Продавцы-консультанты\_для\_макроса*) (рисунок 5.4).

1. Создать 2 простые формы: по таблице **Книги** и по таблице **Продавцы-консультанты** (назвать их **Книги\_для\_макроса** и **Продавцы-консультанты\_для\_макроса**).
2. На вкладке [**Создание**] в группе Макросы и код выбрать команду **Макрос**.
3. Из списка **Добавить новую макрокоманду** выбрать макрокоманду **ОткрытьФорму**. В списке **Имя формы** выбрать форму **Книги\_для\_макроса**. Остальные аргументы оставить без изменения. Сохранить макрос, дать ему имя **Макрос\_3**.
4. Открыть форму **Продавцы-консультанты\_для\_макроса** в режиме **Конструктора**.
5. В **Окне свойств** в списке выделенных элементов выбрать **Форму**. Перейти на страницу **События**. Для события **Открытие** выбрать **Макрос\_3**. Сохранить и закрыть форму.
6. Открыть форму **Продавцы-консультанты\_для\_макроса**. При этом должна открыться и форма **Книги\_для\_макроса**. Закрыть обе формы.

Рисунок 5.4 – Пример создания макроса 3

4 Создать форму ленточного вида для следующих полей таблицы *Книги*: *Название*, *Код автора*, *Цена*, *В наличии*, *Продано*. Создать составной макрос для фильтрации книг в этой форме по следующим условиям отбора:

- код автора 2 или 4;
- цена больше 0,5 и меньше 1,5.

Расположить на форме две кнопки, запускающие макросы для фильтрации по перечисленным выше условиям отбора, и одну кнопку для снятия фильтрации (рисунки 5.5–5.7).

5 Создать макрос для фильтрации записей в момент открытия отчета *Книги\_Задание\_2* по следующему условию отбора: в наличии книг больше 50 (рисунок 5.8).

6 Создать макрос, осуществляющий по форме *Форма\_8* поиск сведений о книге по введенному названию. При отсутствии такой книги макрос должен выдавать сообщение об этом (рисунки 5.9 и 5.10).



1. На вкладке **[Создание]** в группе Макросы и код выбрать команду **Макрос**.
2. Из списка **Добавить новую макрокоманду** выбрать макрокоманду **ОткрытьОтчет**. В списке **Имя отчета** выбрать **Книги Задание 2**. Аргумент **Условие отбора** заполнить следующим образом: **[В наличии]>50**. Сохранить макрос, дать ему имя **Макрос\_5**. Закрывать макрос.
3. В области навигации выбрать **Макрос\_5**, выполнить по нему двойной щелчок мышью для запуска. При этом должен открыться отчет **Книги\_Задание\_2**, отфильтрованный по наличию книг. Закрывать отчет.

Рисунок 5.8 – Создание макроса фильтра

1. Открыть форму **Форма\_8** в режиме **Конструктор**. В области данных добавить поле. В свойствах поля на вкладке **Другие** задать ему имя **Найти книгу**, а для свойства **Всплывающая подсказка** ввести текст *Введите название интересующей Вас книги*. В свойствах надписи добавленного поля для свойства **Подпись** ввести **Найти книгу**.
2. На вкладке **[Создание]** в группе Макросы и код выбрать команду **Макрос**.
3. В группе **Показать или Скрыть** включить **Каталог макрокоманд**.
4. Из списка **Добавить новую макрокоманду** выбрать макрокоманду **КЭлементуУправления**. Имя элемента – поле **Название**.
5. Из списка **Добавить новую макрокоманду** выбрать макрокоманду **НайтиЗапись**. Аргумент **Образец поиска** равен значению в поле **Найти книгу**, т.е. заполнить следующим образом: **=[Найти книгу]**.
6. В **Каталоге макрокоманд** выбрать команду **Если**. Ввести условие **[Название]<>[Найти книгу]**. В блоке **Если** из списка **Добавить новую макрокоманду** выбрать макрокоманду **ОкноСообщения**. Для аргумента **Сообщение** ввести текст *Указанная книга отсутствует* (см. рисунок ниже).
7. Сохранить макрос, дать ему имя **Поиск**.
8. Открыть форму **Форма\_8** в режиме **Конструктора**.
9. Открыть окно свойств для поля **Найти книгу**. На вкладке **События** для события **После обновления** выбрать макрос **Поиск**.
10. Проверить работоспособность формы (в поле **Найти книгу** ввести название книги, которая присутствует в таблице **Книги** – получить запись об этой книге; в поле **Найти книгу** ввести название книги, которая отсутствует в таблице **Книги** – получить сообщение). Сохранить и закрыть форму.

Рисунок 5.9 – Создание макроса 6

КЭлементуУправления	
Имя элемента	Название
<b>НайтиЗапись</b>	
Образец поиска	=[Найти книгу]
Совпадение	Поля целиком
С учетом регистра	Нет
Область поиска	Все
С учетом формата поля	Нет
Только в текущем поле	Да
Первое вхождение	Да
<input type="checkbox"/> <b>Если</b> [Название]<>[Найти книгу] <b>то</b>	
<b>ОкноСообщения</b>	
Сообщение	Указанная книга отсутствует
Сигнал	Да
Тип	Отсутствует
Заголовок	
<b>Конец блока "Если"</b>	

Рисунок 5.10 – Создание формы макроса 6

## ***Контрольные вопросы***



- 1 Назовите объекты базы данных в СУБД Access.
- 2 Для чего предназначены формы?
- 3 Для чего предназначены запросы?
- 4 Для чего предназначены формы?
- 5 Для чего предназначены отчеты?
- 6 Для чего предназначены макросы?

## **6 Практическое занятие № 6. Интеграция пользовательских форм в офисные приложения**

**Цель занятия:** ознакомление с возможностями создания пользовательских форм в VBA, некоторыми его элементами, их свойствами и методами.

### ***Основные теоретические положения***

По своей сути форма (или пользовательская форма) представляет собой диалоговое окно, в котором можно размещать различные элементы управления. В приложении может быть как одна, так и несколько форм. Новая форма добавляется в проект выбором команды *Вставка (Insert) → UserForm*.

В VBA имеется обширный набор встроенных элементов управления. Используя этот набор и редактор форм, нетрудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows. Элементы управления являются объектами. Как любые объекты, они обладают свойствами, методами и событиями. Элементы управления создаются при помощи Панели элементов, которая отображается на экране либо выбором команды *Вид (View) → Панель элементов (Toolbox)*, либо нажатием кнопки  панели инструментов *Standard*. На этой панели представлены кнопки, позволяющие конструировать элементы управления. Для создания элементов управления служат все кнопки панели инструментов, за исключением кнопки *Выбор объекта* . Щелкнув по кнопке *Выбор объекта*, можно выбрать уже созданный в форме элемент управления для последующего его редактирования (изменения размеров или редактирования).

В таблице 6.1 представлен список основных элементов управления и соответствующих кнопок панели элементов.

Для размещения элемента управления на лист или в форму необходимо нажать соответствующую кнопку на панели элементов и с помощью мыши перетащить рамку элемента управления в нужное место. После этого элемент управления можно перемещать, изменять его размеры, копировать в буфер обмена, вставлять из буфера обмена и удалять из формы.

В таблице 6.2 представлены основные общие свойства элементов управления.

Таблица 6.1 – Список основных элементов управления

Элемент управления	Имя	Кнопка, его создающая	Элемент управления	Имя	Кнопка, его создающая
Поле	TextBox		Переключатель	OptionButton	
Надпись	Label		Флажок	CheckBox	
Кнопка	CommandButton		Выключатель	ToggleButton	
Список	ListBox		Рамка	Frame	
Поле со списком	ComboBox		Рисунок	Image	
Полоса прокрутки	ScrolBar		Набор страниц	MultiPage	
Элемент управления	Имя	Кнопка, его создающая	Элемент управления	Имя	Кнопка, его создающая
Счетчик	SpinButton		Набор вкладок	TabStrip	

Таблица 6.2 – Основные общие свойства элементов управления

Свойство	Описание
Caption	Надпись, отображаемая при элементе управления
AutoSize	Допустимые значения: True (устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption) и False (в противном случае)
Visible	Допустимые значения: True (элемент управления отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь вручную может управлять элементом управления) и False (в противном случае)
Height и Width	Устанавливают геометрические размеры объекта (высоту и ширину)
Left и Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме
ControlTipText	Устанавливает текст в окне всплывающей подсказки, связанной с элементом управления. В следующем примере элементу управления CommandButton назначен текст всплывающей подсказки «Это кнопка» CommandButton1.ControlTipText = «Это кнопка»
BackColor, ForeColor и BorderColor	Устанавливают цвет заднего и переднего плана элемента управления, также его границы
BackStyle	Устанавливает тип заднего фона



## Окончание таблицы 6.2

Свойство	Описание
BorderStyle	Устанавливает тип границы. Допустимые значения: fmBorderStyleSingle (граница в виде контура); fmBorderStyleNone (граница невидима)
SpecialEffect	Устанавливает тип границы. Отличается от свойства BorderStyle тем, что позволяет установить несколько типов, но одного цвета. BorderStyle позволяет установить только один тип, но различных цветов
Picture (создание картинки)	Внедряет картинку на элемент управления

В таблице 6.3 представлены наиболее часто используемые свойства элементов управления.

Таблица 6.3 – Свойства элементов управления

Свойство	Описание
TextBox (поле)	используется для ввода текста пользователем или для вывода из него результатов расчетов программ
Text	Возвращает текст, содержащийся в поле
Multiline	Допустимые значения: True (устанавливает многострочный режим ввода текста в поле) и False (однотрочный режим)
WordWrap	Допустимые значения: True (устанавливает режим автоматического переноса) и False (в противном случае)
Label (надпись)	используется для отображения надписей, например, заголовков элементов управления, не имеющих свойства Caption
Caption	Возвращает текст, отображаемый в надписи
Multiline	Допустимые значения: True (устанавливает многострочный режим ввода) и False (однотрочный режим)
WordWrap	Допустимые значения: True (устанавливает режим автоматического переноса) и False (в противном случае)
CommandButton (кнопка)	используется для инициирования выполнения некоторых действий, вызываемых нажатием кнопки, например, запуск программы или остановка ее выполнения, печать и т. д.
Caption	Возвращает текст, отображаемый на кнопке
Default	Задаёт кнопку по умолчанию, т. е. устанавливает ту кнопку, для которой действия, связанные с ней, будут выполняться при нажатии клавиши Enter
Cancel	Допустимые значения: True (устанавливаются отменяющие функции для кнопки, т. е. нажатие клавиши Esc приводит к тем же результатам, что и нажатие кнопки) и False (в противном случае)
Accelerator	Назначает клавишу, при нажатии на которую одновременно с клавишей Alt происходит запуск действий, связанных с кнопкой. Например, CommandButton1.Accelerator=«C»

**Пример** – В качестве примера работы с формой сконструируем простое приложение, вычисляющее значение функции, например,  $\cos(x)$ .

Перейдем в VBA и, выполнив команду *Insert (Вставка) → UserForm*, добавим в проект форму. Расположим на форме следующие элементы управления: Label, TextBox, CommandButton (рисунок 6.1).

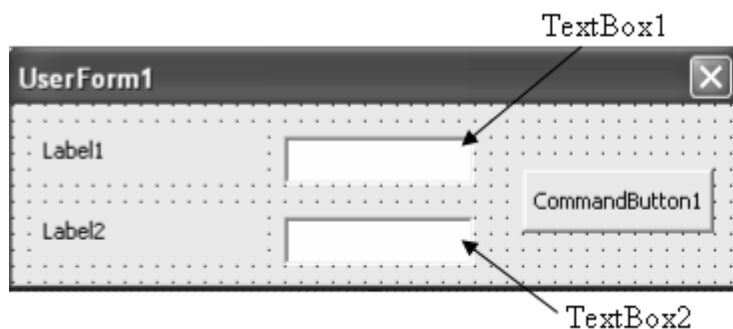


Рисунок 6.1 – Вид формы в режиме конструктора

В таблице 6.4 приведено описание создаваемой формы.

Таблица 6.4 – Описание создаваемой формы

Элемент управления	Предназначение
CommandButton1 (кнопка)	При нажатии на кнопку запускается процедура обработки события (PrivateSub CommandButton1_Click()), которая считывает значение аргумента из поля TextBox1. Проверяется, введено ли в это поле число. Если введено не число, то на экране отображается соответствующее сообщение, прерывается выполнение процедуры и фокус (курсор) устанавливается на поле TextBox1, предлагая исправить вводимые данные. Если введено число, то находится значение функции при введенном значении аргумента, результат выводится в TextBox2
Label1 (надпись)	Пояснительная надпись для поля ввода
TextBox1 (поле)	Поле для ввода пользователем значения аргумента
TextBox2 (поле)	В это поле будет выводиться значение функции. Поле сделаем недоступным для пользователя, т. е. пользователь не сможет ни ввести, ни скорректировать данные в этом поле

Форма создана, функция каждого элемента управления известна. Для написания кода программы, связанного с пользовательской формой, достаточно дважды щелкнуть, например, кнопку CommandButton1. Откроется редактор кода на листе модуля UserForm1. Более того, он откроется на том месте, где программируются действия, связанные с элементом управления, который был дважды нажат. Если код еще не набран, то при открытии редактора кода появятся инструкции заголовка и окончания процедуры, которая будет связана с элементом управления. Редактор кода представлен в таблице 6.5.

После конструирования формы и написания кода в модуле формы выберем команду *Run* → *RunSub/UserForm*, либо нажмем клавишу <F5>, либо кнопку панели инструментов *Standard*, и форма отобразится поверх активного рабочего листа Excel. Введем значение аргумента и нажмем кнопку *OK*. Вид полученной пользовательской формы представлен на рисунке 6.2.

Таблица 6.5 – Пример программы с применением элементов управления

Текст программы	Описание
<pre>Private Sub CommandButton1_Click()   If Not IsNumeric(TextBox1.Text) Then     MsgBox «Аргумент должен быть     числом», _vbExclamation     TextBox1.SetFocus      Exit Sub   End If   x = CDb1(TextBox1.Text)    y = Cos(x)    TextBox2.Text = CStr(y)  End Sub Private Sub UserForm_Initialize()  UserForm1.Caption = «Значение функции Cos(x)» Label1.Caption = «Аргумент» Label2.Caption = «Значение функции» CommandButton1.Caption = «OK» TextBox2.Enabled = False EndSub</pre>	<p>Проверка, является ли введённое значение числом.</p> <p>Вывод окна сообщения.</p> <p>Фокус (курсор) устанавливается на поле TextBox1.</p> <p>Досрочный выход из процедуры.</p> <p>При считывании числа из поля ввода при помощи функции CDb1 строковый тип, возвращаемый свойством Text, преобразуется в числовой.</p> <p>Чтобы вывести результат в поле, переводим число в строковый формат при помощи функции CStr.</p> <p>Процедура UserForm_Initialize конструирует форму до ее загрузки.</p> <p>Инструкция устанавливает текст, отображаемый в строке заголовка формы.</p> <p>Инструкции задают видимые надписи для объектов.</p> <p>Инструкция делает TextBox2 недоступным для пользователя</p>



Рисунок 6.2 – Вид пользовательской формы

### Задание

Выполнить примеры из методических рекомендаций.

### Контрольные вопросы

- 1 Как создать графический интерфейс своего приложения с помощью VBA?
- 2 Назовите этапы создания форм.
- 3 Что такое элемент управления?
- 4 Какое назначение элементов управления Label, TextBox и CommandButton?

## 7 Практическое занятие № 7. Автоматизация обмена данными между различными формами файлов

**Цель занятия:** создание пользовательских диалоговых окон с применением различных элементов управления.

### *Основные теоретические положения*

Инициализировать и отобразить диалоговое окно на экране очень просто. Инициализация производится при помощи процедуры обработки события `Initialize` формы `UserForm`. Отображение диалогового окна на экране осуществляется методом `Show`. Инструкцию с методом `Show` обычно помещают в процедуру, которая связана с командой пользовательского меню, кнопкой панели инструментов или элементом управления, как правило, кнопкой диалогового окна.

Простой инициализации или обычного отображения диалогового окна часто бывает недостаточно, т. к. это приводит к появлению на экране функционально ненастроенного диалогового окна. Такое диалоговое окно можно сравнить с каркасом дома. В таком доме жить неприятно и в него совсем не хочется въезжать. Для того чтобы жить в доме было приятно и удобно, прежде чем в него вселяться, надо сделать много отделочных работ. Также и при инициализации диалогового окна необходимо предусмотреть огромное количество на первый взгляд мелочей, но без которых работать с диалоговым окном неудобно. В частности, при отображении диалогового окна на экране нужно установить значения полей, применяемые по умолчанию, задать функции кнопок, назначить им комбинации клавиш, связать с элементами управления всплывающие подсказки, вывести в списках первоначально выводимые элементы списков, задать первоначальную установку флажков, вывести в элементы управления формы требуемые рисунки и т. д.

В VBA диалоговые окна работают в режиме модального диалога. Это означает, что пользователь, прежде чем перейти к выполнению действий, не связанных с текущим активным диалоговым окном, должен его закрыть. Закрытие диалогового окна производится методом `Hide`. Следующая процедура является примером процедуры закрытия диалогового окна. Эта процедура активизируется при нажатии кнопки `CommandButton2` диалогового окна `UserForm1` и выполняет только одну инструкцию, осуществляющую закрытие этого диалогового окна.

```
PrivateSub CommandButton2_Click()  
    UserForm1.Hide ' Процедура закрытия диалогового окна  
EndSub
```

Закрыть диалоговое окно также, конечно, можно, нажав системную кнопку, расположенную в правом верхнем углу любого диалогового окна. Если при закрытии диалогового окна необходимо произвести какие-то действия,

например, считать информацию из окна в файл на диске и т. д., во избежание потери информации действия, производимые программой при закрытии окна, разумно также продублировать в процедуре обработки события *Terminate* (закрытие) пользовательской формы.

### **Задание**

Необходимо создать форму (диалоговое окно пользователя) для расчета параметра (по вариантам). Все расчеты производятся при нажатии кнопки *Расчет*. Данная кнопка должна срабатывать при нажатии на клавишу *Enter* (независимо от того, какой элемент управления является активным в момент нажатия данной клавиши). Перед проведением расчета программа должна выполнить проверку корректности ввода данных. Результат расчета выводится в соответствующее текстовое поле *TextBox* и на страницу *Excel*. Кнопка *Отмена* должна закрывать текущую форму и срабатывать при нажатии на клавишу *Esc*.

Варианты индивидуальных заданий для выполнения лабораторной работы выдаются преподавателем.

### **Контрольные вопросы**

- 1 Как создать графический интерфейс своего приложения с помощью VBA?
- 2 Назовите самые важные свойства и методы форм.
- 3 Назовите основные свойства элементов управления.
- 4 Понятие события. Как происходит обработка событий формы?
- 5 Как обеспечивается связывание помещенных на форму элементов управления?
- 6 Как назначить горячие клавиши элементам управления?
- 7 Как назначить последовательность перехода фокуса по элементам управления?
- 8 Для чего применяется свойство *Visible* элементов управления?
- 9 Для чего применяется свойство *Enable* элементов управления?
- 10 Для чего применяется свойство *Cancel* элементов управления?
- 11 Как расширить набор стандартных элементов управления?

## **8 Практическое занятие № 8. Практические примеры оптимизации работы офисных приложений**

**Цель занятия:** ознакомление с примером оптимизации работы офисных приложений.

*Business Intelligence (BI)* или Бизнес-аналитика – это набор IT-технологий для сбора, хранения и анализа данных, позволяющих предоставлять пользователям достоверную аналитику в удобном формате, на основе которой можно принимать эффективные решения для управления бизнес-процессами

компании. Все уровни пользователей, от сотрудников до учредителей, получают гибкий доступ к необходимой им управленческой отчетности, не прибегая к помощи IT-специалистов.

Сегодня на рынке существует несколько платформ Бизнес-аналитики (BI), концептуально (рисунок 8.1) они представляют собой следующее.

ETL-инструменты: программы, позволяющие выполнять загрузку данных в DWH из различных учетных систем.

DWH-хранилище: полноценная база данных SQL для подготовки и хранения данных для аналитики.

OLAP-кубы: технология, позволяющая делать в реальном времени (1...5 с) любые отчеты и проводить полноценный анализ данных.

Клиентские приложения: как правило, для детального анализа данных и построения динамических отчетов пользователи используют Сводные таблицы Microsoft Excel, подключенные к OLAP-кубам.

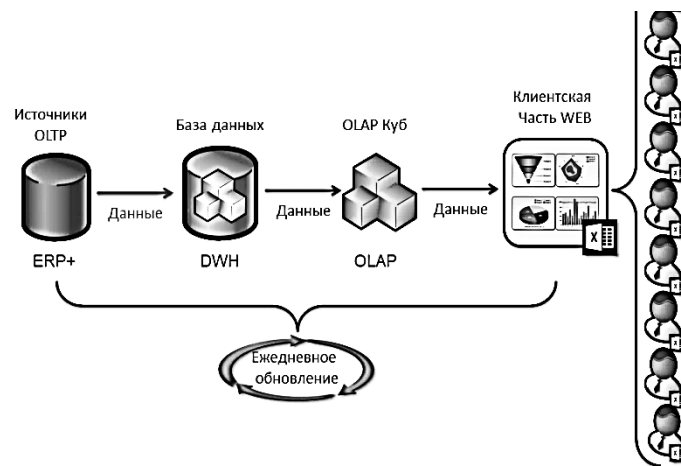


Рисунок 8.1 – Концепция платформы BI

На сегодняшний день платформа Microsoft BI – это лучшее решение на рынке BI-систем, в первую очередь, по соотношению «цена – качество – современность». И что также очень важно – основным и «родным» приложением для этой платформы является уже известный всем Microsoft Excel.

Более того, все компоненты этой BI-платформы доступны в одном программном обеспечении – Microsoft SQL Server, которое приобретается один раз без необходимости последующих платежей за использование.

OLAP (от англ. online analytical processing) – аналитическая технология обработки данных в реальном времени. Простым языком – хранилище с многомерными данными (Куб), еще проще – просто база данных, из которой можно получить данные в Excel и проанализировать с помощью инструмента Excel – Сводные таблицы. Сводные таблицы – это пользовательский интерфейс для отображения многомерных данных. Иными словами, это специальный вид таблиц, с помощью которых можно сделать практически любой отчет.

Сравним Обычную таблицу (рисунок 8.2) со Сводной таблицей (рисунок 8.3).



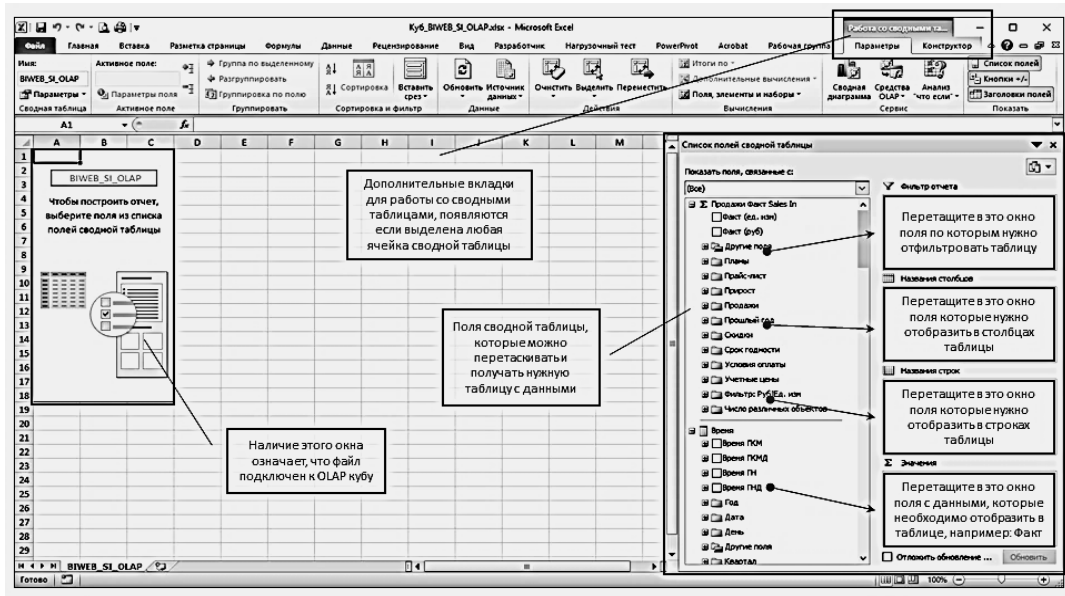


Рисунок 8.4 – Файл Excel, подключен к OLAP-кубу

	Квартал 1, 2015	Квартал 2, 2015	Квартал 3, 2015	Квартал 4, 2015	Общий итог
Бренд №5020186731486	307 880 609	415 930 621	416 833 968	1 055 547 944	2 190 202 162
Бренд №5020006115451	306 496 902	348 075 855	449 363 663	465 952 027	1 569 888 447
Бренд №5020006093550	270 523 521	81 816 233	925 982 207	127 518 168	1 405 839 129
Бренд №5020127708034	-139 155 607	44 945 161	690 737 528	408 144 102	1 003 671 184
Бренд №5020006093537	283 147 703	116 651 948	408 662 955	185 960 617	994 423 223
Бренд №5020006093560	147 860 844	115 884 269	320 998 328	287 629 292	871 772 733
Бренд №5020006115675	148 987 445	248 176 009	156 561 955	250 107 530	803 832 939
Бренд №5020006085901	119 401 210	134 298 836	123 112 389	267 736 797	644 549 232
Бренд №5020393961644	13 038 418	178 614 423	195 952 550	231 183 546	618 784 937
Бренд №5020006093564	125 716 377	86 797 159	212 224 370	170 127 113	594 855 019
Общий итог	1 583 897 421	1 771 190 513	3 900 429 692	3 449 907 137	10 705 424 763

Рисунок 8.5 – Сводная таблица с выбранными полями

### Контрольные вопросы

- 1 Что понимается под Business Intelligence (BI)?
- 2 Что включает концепция платформы BI?
- 3 Что такое OLAP?
- 4 Чем отличается Обычная таблица Excel от Сводной таблицы?



## Список литературы

1 **Гуриков, С. Р.** Введение в программирование на языке Visual Basic for Applications (VBA) : учеб. пособие / С. Р. Гуриков. – М.: ИНФРА-М, 2020. – 317 с.

2 **Гуриков, С. Р.** Интернет-технологии: учеб. пособие / С. Р. Гуриков. – М. : ФОРУМ; ИНФРА-М, 2017. – 184 с.

3 **Коваленко, В. В.** Проектирование информационных систем: учеб. пособие / В. В. Коваленко. – 2-е изд., перераб. и доп. – М.: ИНФРА-М, 2021. – 357 с.