

*МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ*  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Курганский государственный университет»

Кафедра экономической теории и моделирования экономических процессов

**ЭКОНОМИЧЕСКАЯ ИНФОРМАТИКА  
(ЧАСТЬ 2)**

Методические указания  
к выполнению практических и самостоятельных заданий  
для студентов направлений 080100.62, 080200.62 и 100762.62 и  
специальностей 080101.65 и 036401.65  
очной формы обучения

Курган 2014

Кафедра: «Экономическая теория и моделирование экономических процессов»

Дисциплина: «Экономическая информатика»

(направление 080100.62, 080200.62, 100762.62,  
специальность 080101.65, 036401.65).

Составили: ассистент Е.А. Студентова, ст. преподаватель С.М. Филимонов.

Утверждены на заседании кафедры «23» декабря 2013 г.

Рекомендованы методическим советом университета «31» декабря 2013 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 ОСНОВЫ VBA (VISUAL BASIC FOR APPLICATION).....	5
2 ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ В MS ACCESS .....	15
3 БЛОК-СХЕМЫ И АЛГОРИТМИЗАЦИЯ.....	29
СПИСОК ЛИТЕРАТУРЫ.....	38

## ВВЕДЕНИЕ

Информатика – это техническая дисциплина, систематизирующая приемы работы с данными средствами вычислительной техники, принципы функционирования этих средств и методы управления ими.

Методические рекомендации по дисциплине «Экономическая информатика» составлены в соответствии с рабочей программой, содержат рекомендуемую последовательность изучения дисциплины и варианты практических заданий.

В результате изучения дисциплины студенты должны иметь представление:

- об основных этапах решения задач с помощью ПК, методах и средствах сбора, обработки, хранения, передачи и накопления информации;
- о программном и аппаратном обеспечении вычислительной техники, о компьютерных сетях и сетевых технологиях обработки информации, о методах защиты информации.

Знать:

- основные понятия автоматизированной обработки информации;
- общий состав и структуру персональных компьютеров и вычислительных систем;
- базовые системные программные продукты и пакеты прикладных программ.

Уметь:

- использовать изученные прикладные программные средства в профессиональной деятельности.

Практический курс «Экономическая информатика» разделен на две части. Ко второй части курса относится изучение основ VBA (Visual Basic for Application), основ работы с базами данных в Microsoft Access, а также закрепление теоретических знаний и получение практических навыков по теме «Блок-схемы и алгоритмизация».

# 1 ОСНОВЫ VBA (VISUAL BASIC FOR APPLICATION)

## Практическое задание 1.1

*Цель – научиться создавать пользовательские функции*

**Примечание:** Одной из возможностей VBA является создание новой функции *MS Excel*, которую впоследствии можно использовать аналогично встроенным функциям (СУММ, МАКС, ЕСЛИ и др.). Это целесообразно в тех случаях, если необходимой функции нет в стандартном наборе встроенных функций *MS Excel*, например формулы Пифагора, а ею приходится часто пользоваться.

Необходимо создать пользовательскую функцию, вычисляющую по теореме Пифагора длину гипотенузы прямоугольного треугольника по двум заданным катетам.

Для этого:

1 Создайте новую книгу *MS Excel* и перейдите в редактор VB. Для открытия редактора выберите в Меню *Разработчик / Код / Visual Basic* или нажмите комбинацию клавиш *[Alt+F11]*.

2 Создайте новый модуль, выполнив команду *Insert→Module*.

3 В новом модуле выполните команду *Insert→Procedure*, чтобы открыть диалоговое окно «*Add Procedure*» для создания новой процедуры (рисунок 1).

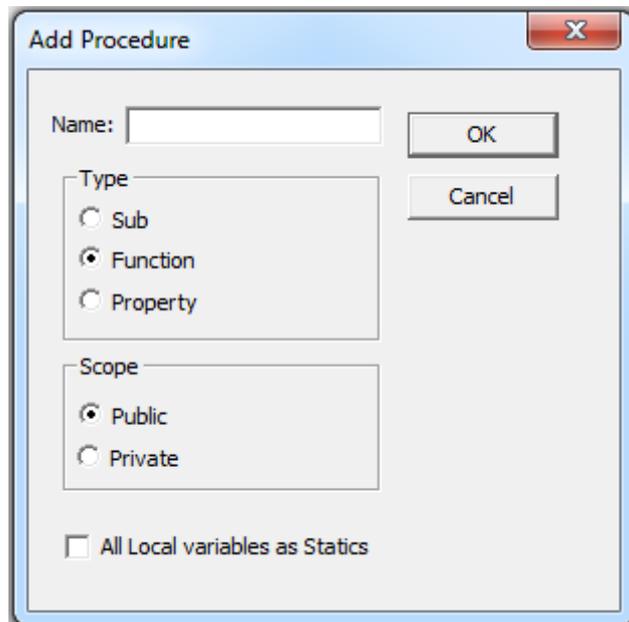


Рисунок 1 – Диалоговое окно «Add Procedure»

4 В диалоговом окне «*Add Procedure*» задайте имя «*Пифагор*» и выберите тип «*Function*». Нажмите кнопку *<OK>*.

5 В окне кода между двумя появившимися строчками напишите программный код для данной функции, учитывая, что для нахождения длины гипотенузы по формуле Пифагора нужно знать значения длин двух катетов

$a$  и  $b$ :

```
Public Function Пифагор(a, b)
Пифагор = (a^2 + b^2) ^ (1 / 2)
End Function
```

Оператор « $\wedge$ » означает возведение числа в степень.

1 Закройте редактор VB и воспользуйтесь созданной функцией. В ячейки A1, B1 и C1 введите соответственно символы  $a$ ,  $b$ , и  $c$ ; в ячейки A2 и B2 – значения длин катетов, а в ячейку C2 вставьте функцию, воспользовавшись командой *Вставка*→*Функция* и выбрав созданную функцию в категории «*Определенные пользователем*» диалогового окна «*Мастер функций*» (рисунок 2).

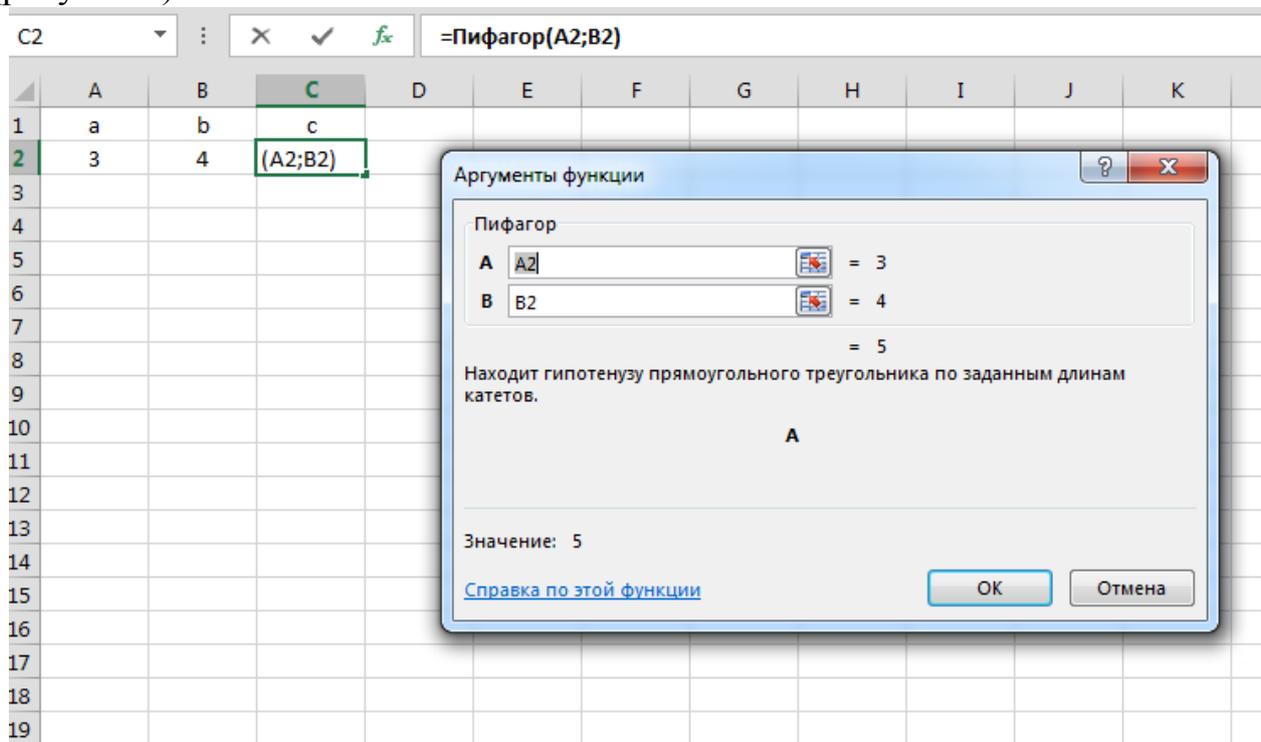


Рисунок 2 – Результат выполнения функции

2 Добавьте к созданной функции описание, поясняющее ее назначение. Для этого выполните команду Меню Разработчик / Код / Макросы и, набрав в поле «*Имя макроса*» диалогового окна «*Макрос*» название данной функции, введите описание, нажав кнопку «*Параметры*» (рисунок 3).

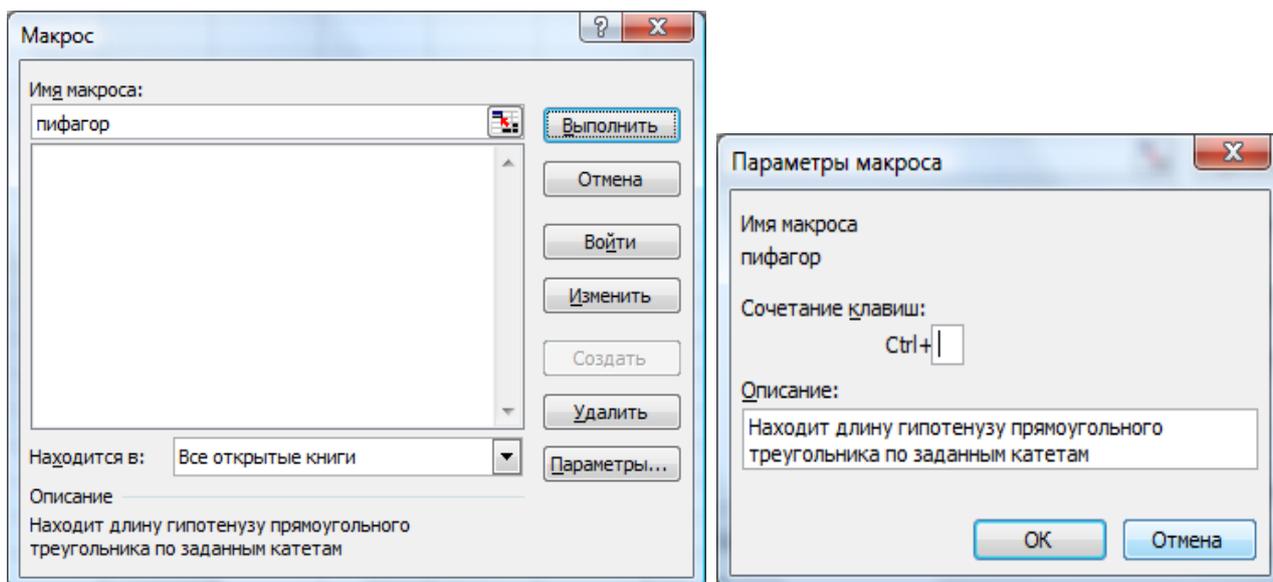


Рисунок 3 – Диалоговые окна «Макрос» и «Параметры макроса»

Аналогичным образом создайте функцию, математически определенную как  $y = \sin(\pi x)e^{-2x}$  и постройте ее график.

Для этого:

1 В редакторе *VB* в новом модуле создайте функцию с именем «*Y*» и напишите для нее программный код:

```
Public Function Y(x)
    Y = Sin(Application.Pi*x)*Exp(-2*x)
End Function
```

Здесь воспользовались стандартной функцией *Pi*, которая возвращает значение постоянной  $\pi$ . Так как она не является внутренней функцией *VBA*, то ее необходимо записать в виде *Application.Pi*.

2 Введите в ячейки A1 и B1 соответственно «*x*» и «*y*», в ячейки A2 и A3 – значения *x*, например, –0,5 и –0,4 соответственно, и с помощью *маркера автозаполнения* скопируйте значения в ячейки A4:A12.

3 В ячейку B2 вставьте формулу «=Y(A2)» и также с помощью *маркера автозаполнения* скопируйте ее в ячейки B3:B12.

4 Выделите диапазон ячеек B2:B12 и с помощью *Мастера диаграмм* постройте график данной функции (подписи по оси X должны быть – значения *x*) (рисунок 4).

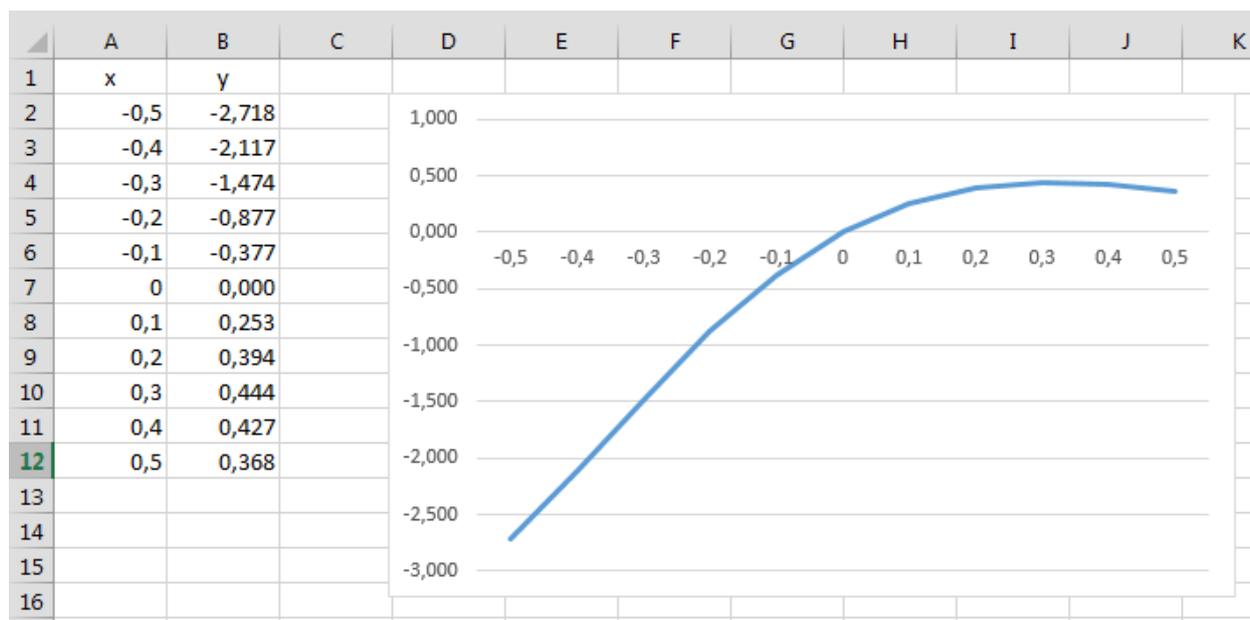


Рисунок 4 – Результат выполнения функции и ее график

## Практическое задание 1.2

Цель – научиться создавать диалоговые окна.

**Примечание:** В VBA имеется два стандартных диалоговых окна для взаимодействия с пользователем:

1 **InputBox** используется для ввода информации в отдельном диалоговом окне и имеет следующий синтаксис (в квадратных скобках указаны необязательные параметры):

*InputBox* («Текст сообщения», [*Текст заголовка диалогового окна*]).

2 **MsgBox** используется в качестве диалогового окна вывода сообщений и имеет синтаксис:

*MsgBox* «Текст сообщения», [*Кнопки + Иконки*], [*Текст заголовка диалогового окна*],

где *Кнопки + Иконки* – параметры, задающие отображаемые кнопки и значки диалогового окна.

Необходимо создать процедуру, вычисляющую длину гипотенузы треугольника по заданным катетам.

Для этого:

1 Создайте новую книгу *MS Excel* и перейдите в редактор *VB*.

2 Создайте новый модуль, выполнив команду *Insert*→*Module*.

3 В новом модуле выполните команду *Insert*→*Procedure*, чтобы открыть диалоговое окно «*Add Procedure*» для создания новой процедуры.

4 В диалоговом окне «*Add Procedure*» задайте имя «*Pythagor*» и выберите тип «*Sub*» (рисунок 5). Нажмите кнопку <*OK*>. Процедуры, в отличие от рассмотренных ранее функций, не возвращают значений, а только выполняют последовательность действий.

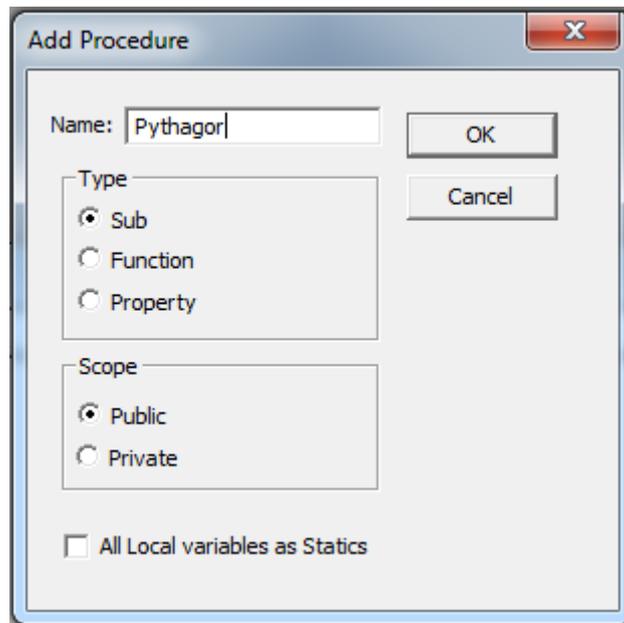


Рисунок 5 – Диалоговое окно «Add Procedure»

5 В окне кода между двумя появившимися строчками напишите программный код для данной процедуры. В отличие от функций простые пользовательские процедуры не имеют параметров.

```
Public Sub Pythagor()  
    a = InputBox(«Введите длину катета a»)  
    b = InputBox(«Введите длину катета b»)  
    c = (a ^ 2 + b ^2) ^ (1 / 2)  
    MsgBox («Длина гипотенузы равна» & c)  
End Sub
```

Оператор «&» объединяет две строки.

6 Выполнение процедур происходит аналогично выполнению записанных макросов. Поэтому создайте командную кнопку на рабочем листе для запуска процедуры (рисунок 6).

Также запустить диалоговые окна можно из редактора VBA, выполнив команду Меню *Run / Run Sub / UserForm*. Появится диалоговое окно выбора макроса – выберите название вашей процедуры и нажмите кнопку *Run*. Также в редакторе можно нажать F5 на клавиатуре для быстрого запуска процедуры.

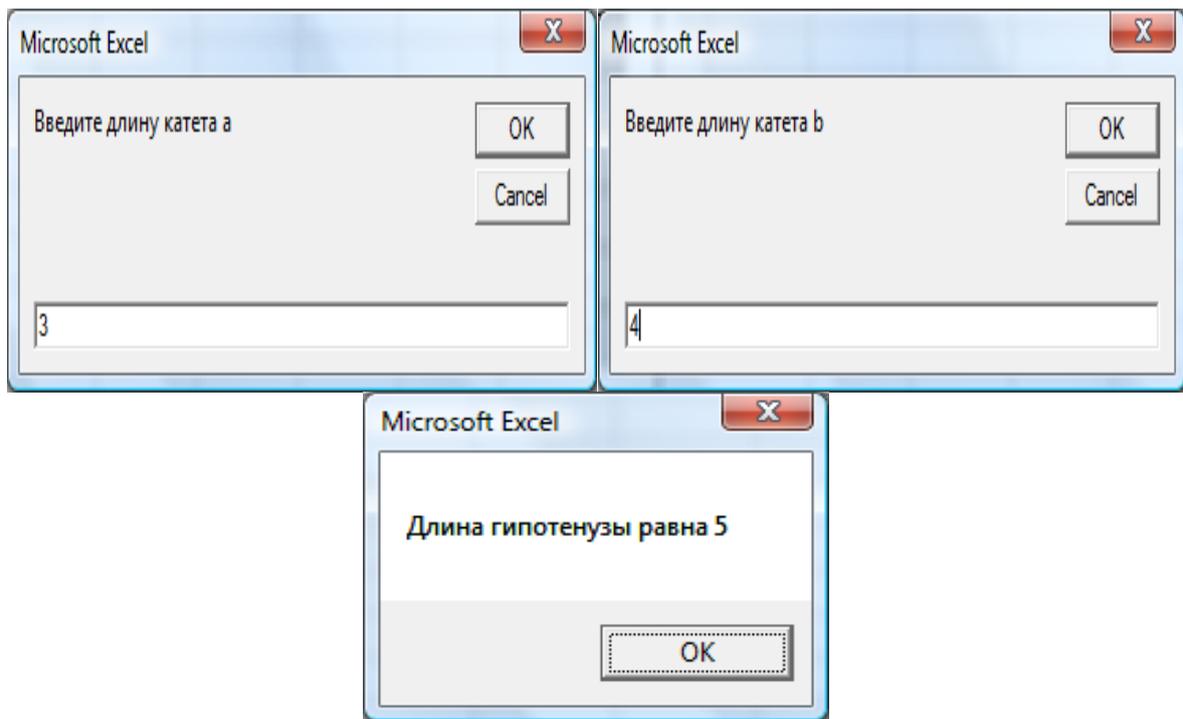


Рисунок 6 – Диалоговые окна при выполнении процедуры

### Практическое задание 1.3

*Цель – научиться создавать пользовательские формы.*

**Примечание:** VBA позволяет создавать и использовать экранные формы, разработанные пользователем. Такие формы представляют собой объекты класса *UserForm*. Для создания новой формы пользователя необходимо в MS Excel перейти в редактор VB и выполнить команду *Insert*→*UserForm*. В результате откроется окно конструктора форм (рисунок 7). При создании формы автоматически отображается панель элементов управления «*Toolbox*», содержащая кнопки, с помощью которых элементы управления можно разместить на создаваемой форме (аналогично тому, как командные кнопки размещались на рабочем листе). Если данная панель инструментов не отображается, выполните команду *View*→*ToolBox*.

После размещения элемента управления на форме с помощью окна свойств (*Properties*) обычно задаются свойства выделенного объекта. В случае, если окно неактивно, вызвать его можно в меню *View*→*Properties Window*, или нажав на клавиатуре F4.

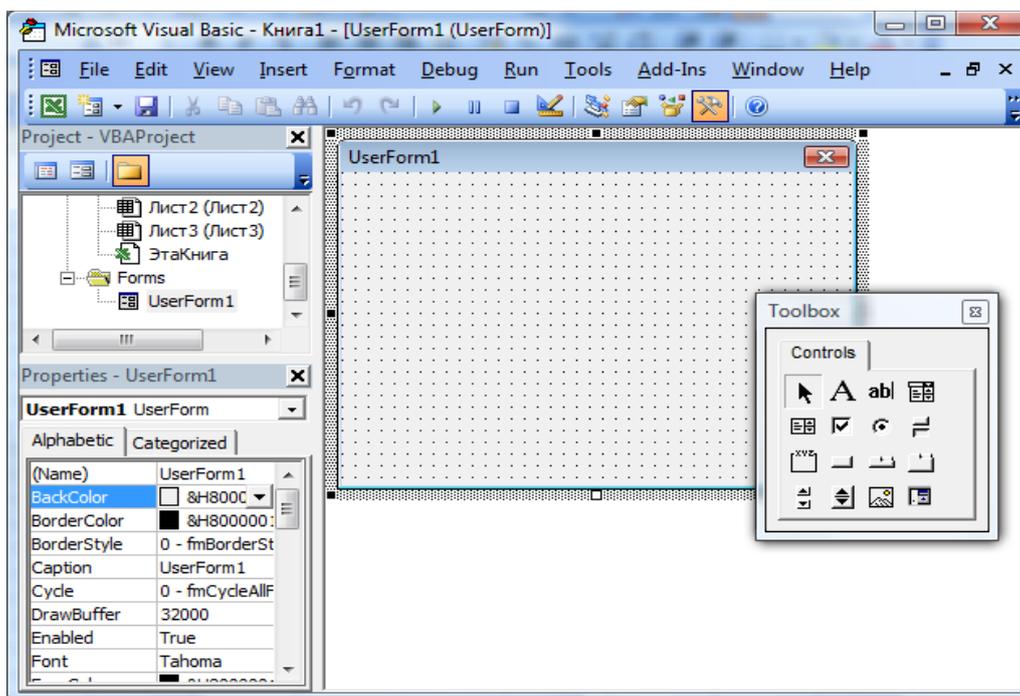


Рисунок 7 – Окно конструктора форм

Создайте пользовательскую форму для вычисления длины гипотенузы треугольника по заданным катетам. При вычислениях использовать созданную ранее функцию «*Пифагор*».

Для этого:

1 Откройте рабочую книгу, содержащую функцию «*Пифагор*». Обратите внимание, что если вы решили создавать форму в новой книге, то функцию Пифагора нужно будет прописать заново, иначе расчет станет невозможным.

2 Перейдите в редактор VB и выполните команду *Insert*→*UserForm* для создания нового макета пользовательской формы.

3 Выделите форму, щелкнув по ней левой кнопкой мыши, в окне «*Properties*» найдите свойство «*Caption*» (данное свойство хранит заголовок формы, текст на кнопке и т.п., т.е. текст, связанный с объектом) и установите для него значение «*Теорема Пифагора*».

4 Добавьте на форму объект «*CommandButton*» (кнопка), три объекта «*TextBox*» (текстовое поле) и три объекта «*Label*» (метка, текст на форме).

5 Для удобства обращения к объектам измените их имена. Для этого необходимо изменить свойство «*Name*». Объекту *TextBox1* задайте имя «*a*», *TexxtBox2* – «*b*», *TextBox3* – «*c*».

6 Для объекта *Label1* установите значение свойства *Caption* – «*a=*», для *Label2* – «*b=*», для *Label3* – «*c=*», для *CommandButton1* – «*Вычислить*» (рисунок 8).

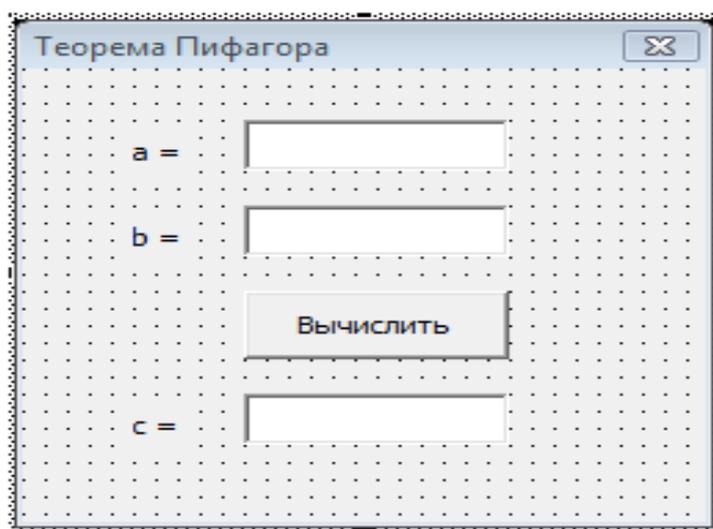


Рисунок 8 – Конструктор формы «Теорема Пифагора»

7 Дважды щелкните по кнопке *CommandButton1*. В результате будет открыт редактор кода и автоматически создастся процедура обработки нажатия кнопки.

8 В окне кода между двумя появившимися строчками напишите программный код для данной кнопки:

```
Private Sub CommandButton1_Click()
    c.Text = Пифагор(a.Text, b.Text)
End Sub
```

Свойство «*Text*» хранит текст, введенный в текстовые поля.

9 Перейдите на «*Лист1*» и создайте кнопку для открытия формы. Перейдите в режим конструктора, выбрав в Меню *Разработчик / Элементы управления / Режим конструктора*. Выделите текст данной кнопки и переименуйте ее, введя название «Открыть форму».

10 Перейдем в редактор *VB*, выделив кнопку и выбрав в Меню *Разработчик / Элементы управления / Просмотр кода*. В появившемся модуле «*Лист1*» введите программный код:

```
Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub
```

В результате при нажатии на кнопку будет отображаться форма «*Теорема Пифагора*» (для этого нужно выйти из режима конструктора). Форму также можно отобразить из редактора *VB*, когда открыт конструктор форм, командой *Run*→*Run Sub / UserForm* или нажав на клавиатуре *F5*.

#### Практическое задание 1.4

Цель – решение задач с условием с использованием функции *ЕСЛИ*.

Для организации процесса вычислений в зависимости от какого-либо Условия служит условный оператор *If/Then/Else*.

Если в зависимости от некоторого *Условия*, необходимо выполнить только одно действие, то используется однострочная форма оператора *If/Then/Else – If Условие Then Оператор1 [Else Оператор2]*.

Если *Условие* выполняется, то выполняется *Оператор1*, в противном случае выполняется *Оператор2*.

Если в зависимости от некоторого *Условия* необходимо выполнить только несколько действий, то используется многострочная форма оператора *If/Then/Else*:

```
If Условие Then
    БлокОператоров1
[Else
    БлокОператоров2]
End If
```

Если *Условие* выполняется, то выполняется *БлокОператоров*, в противном случае, выполняется *БлокОператоров2*.

Блоки операторов могут содержать сколько угодно операторов.

Может возникнуть ситуация, когда при невыполнении *Условия* требуется проверить еще одно *Условие*. В этом случае используется следующая многострочная форма:

```
If Условие1 Then
    БлокОператоров1
[ElseIf Условие2 Then
    БлокОператоров2]
.....
[Else
    БлокОператоров]
End If
```

Если *Условие1* выполняется, то выполняется *БлокОператоров1*, в противном случае проверяется *Условие2*. Если оно выполняется, то выполняется *БлокОператоров2* и т.д. Если ни одно из *Условий* не выполняется, то выполняется *БлокОператоров*. В данной конструкции может содержаться сколько угодно блоков *ElseIf*.

*Условие* может быть как простым (например,  $a > 5$ ), так и составным (например,  $a > 5$  и  $b > 2$ ). Для объединения простых *Условий* используются логические операторы **And** (**И** – два или более *Условий* выполняются одновременно), **Or** (**ИЛИ** – выполняется хотя бы одно из *Условий*), **Not** (**НЕ** – отрицание *Условия*).

Решим следующую задачу. На банковский вклад начисляются проценты в сумме 20% годовых, если сумма вклада превышает 200 000 р., 15% годовых, если сумма от 100 000 до 200 000 р., 10% годовых – на суммы до 50 000 р., 12% годовых – на остальные суммы. Рассчитать сумму полученных вкладчиком процентов по истечении срока.

Для решения поставленной задачи запустим редактор VBA и создадим новую процедуру (*Insert – Module; Insert – Procedure; new Function*). Назовем ее «Процент». Предполагается, что пользователь должен ввести значение суммы (назовем сумму переменной S), а программа в зависимости от этой суммы рассчитать сумму вклада с учетом того, какой процент на данную сумму должен быть начислен.

```
Public Function Процент (S)
    If S > 200000 Then
        Процент = S + S * 0.2
    ElseIf S <= 200000 And S > 100000 Then
        Процент = S + S * 0.15
    ElseIf S < 50000 Then
        Процент = S + S * 0.1
    Else
        Процент = S + S * 0.12
    End If
End Function
```

Решение задачи на листе Excel будет выглядеть следующим образом (рисунок 9).

	A	B	C
1	Сумма вклада, р.	Сумма с процентами, р.	
2	25000	=Процент(A2)	
3	50000	56000	
4	75000	84000	
5	160000	184000	
6	250000	300000	
7			
8			

Рисунок 9 – Решение задачи с использованием функции Процент

Аналогичным образом пропишите в VBA и оформите на листе Excel следующую задачу: Рассчитать стоимость заказа в типографии, если действуют следующие расценки: печать до 100 экземпляров – 10 р. за лист; от 100 до 1000 экземпляров – 7 р. за лист; свыше 1000 – 5 р. за лист. Пропишите формулу и рассчитайте на рабочем листе Excel стоимость печати для 50, 150, 500, 900, 1000, 1200 и 1500 экземпляров.

## 2 ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ В MS ACCESS

*Microsoft Access* – это система управления базами данных (СУБД). *Access* предназначена для хранения и получения данных, представления их в удобном виде и автоматизации часто выполняемых операций. Используя *Access*, можно не только разрабатывать удобные формы ввода данных, но и обрабатывать данные, а также составлять всевозможные сложные отчеты.

В качестве примера будет рассмотрена база данных компьютерной школы, содержащая личные данные студентов и сведения об успеваемости.

### Создание информационной модели данных

Процесс создания базы данных начинается с *создания информационной модели данных*.

Первым шагом в создании информационной модели является выявление сущностей.

**Сущность** – некоторый обособленный объект или событие, информацию о котором необходимо сохранять в базе данных, имеющий определенный набор свойств – атрибутов. Сущности могут быть как физические (реально существующие объекты: например, СТУДЕНТ), так и абстрактные (например, ЭКЗАМЕН).

В нашей базе данных имеется две сущности: *Студент* и *Учебная группа*. Сущность *Студент* обладает такими атрибутами, как *Фамилия*, *Имя*, *Отчество*, *Год рождения*, *Адрес*, *Телефон*, *Номер группы*, *Результаты экзаменов по предметам (Word, Excel, Access)*. Сущность *Учебная группа* обладает атрибутами *Номер группы*, *Преподаватель*. Связи между сущностями отображают в виде диаграммы «сущности-связи» (рисунок 10). Сущности изображают в виде прямоугольников, атрибуты – в овалах, тип связи – в ромбах. Также отображают тип связи: *один-ко-многим (1:M)*, *один-к-одному (1:1)*, *многие-ко-многим (M:M)*.

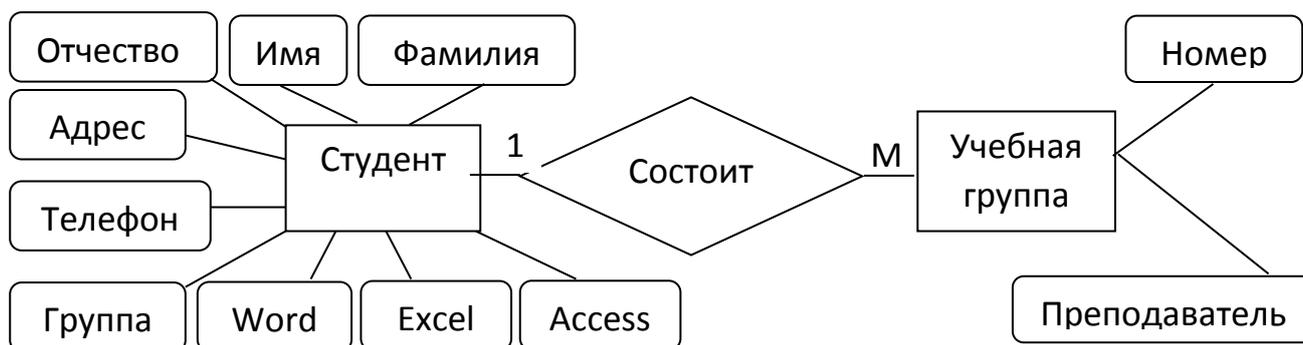


Рисунок 10 – Диаграмма «сущности-связи»

Вторым шагом является создание таблиц по схеме данных. На этом шаге необходимо определить имена и тип полей таблиц. Однако для удобства таблицу *Список* лучше разбить на две таблицы: *Список* и *Личные данные* (таблицы 2 и 3). И последним шагом является физическая разработка базы данных.

## Практическое задание 2.1

Цель – научиться создавать базы данных.

### Создание новой базы данных

Запустите программу Access. Выполните команду *Пустая база данных рабочего стола* и в появившемся диалоговом окне введите имя вашей базы данных (База данных компьютерной школы), выберите папку, в которую данная база будет сохранена и нажмите кнопку *Создать*.

### Создание таблиц

В таблицах баз данных столбцы называются *полями*, строки – *записями*. Создадим 3 таблицы: Группы (таблица 1), Список (таблица 2) и Личные данные (таблица 3).

Таблица 1 – Группы

Имя поля	Тип данных	Описание
Учебная группа	Числовой	номер учебной группы
Преподаватель	Короткий текст	фамилия преподавателя группы

Таблица 2 – Список

Имя поля	Тип данных	Описание
КодСтудента	Счетчик	индивидуальный номер студента
Фамилия	Короткий текст	фамилия студента
Имя	Короткий текст	имя студента
Отчество	Короткий текст	отчество студента
Год рождения	Короткий текст	год рождения студента
Учебная группа	Короткий текст	номер группы студента

Таблица 3 – Личные данные

Имя поля	Тип данных	Описание
КодСтудента	Счетчик	индивидуальный номер студента
Адрес	Короткий текст	адрес студента
Телефон	Короткий текст	телефон студента
Word	Числовой	оценка по Word
Excel	Числовой	оценка по Excel
Access	Числовой	оценка по Access

Для создания таблицы выполните команду *Создание / Конструктор таблиц*. Появится окно *Конструктора таблиц*. В столбце *Имя поля* задается имя столбца таблицы. Значение в столбце *Тип данных* определяет значения, которые можно хранить в этом поле. По умолчанию задается тип данных *Короткий текст*. Любой другой выбирается с помощью *ниспадающего меню*. Столбец *Описание* является необязательным, значение в нем задает описание данных, хранящихся в этом поле

Заполните таблицу *Конструктора* как показано на рисунке 11.

Таблица1		
Имя поля	Тип данных	
Учебная группа	Числовой	номер учебной группы
Преподаватель	Короткий текст	фамилия преподавателя группы

Рисунок 11 – Таблица Конструктора

Установите курсор в ячейку, в которой записано *Учебная группа*, и щелкнув правой кнопки мыши выполните команду *Ключевое поле*. Также задать ключевое поле можно из Меню *Конструктор / Сервис / Ключевое поле*. Ключевое поле должно однозначно идентифицировать запись таблицы. В данном случае ключевым полем является *Учебная группа*, т.к. не может быть двух групп с одинаковыми номерами.

Выполните команду *Сохранить* и в поле *Имя таблицы* введите «Группы». Закройте текущую таблицу. (Если закрыть таблицу, не выполнив предварительно сохранение таблицы, то при закрытии таблицы программа предложит сохранить таблицу и ввести ее имя).

Создайте таблицу *Список* с информацией о студентах в режиме *Конструктора* (рисунок 12).

Таблица		
Имя поля	Тип данных	
КодСтудента	Счетчик	индивидуальный номер студента
Фамилия	Короткий текст	фамилия студента
Имя	Короткий текст	имя студента
Отчество	Короткий текст	отчество студента
Год рождения	Числовой	год рождения студента
Учебная группа	Числовой	номер группы студента

Рисунок 12 – Таблица *Список* в режиме *Конструктора*

Значения поля *Учебная группа* надо вводить не вручную, а выбирать из списка, содержащегося в таблице *Группы*. Для этого установите курсор в ячейку *Учебная группа* и в окне *Свойства поля* на вкладке *Подстановка* в поле *Тип элемента управления* выберите *Поле со списком*, *Тип источника строк* – *Таблица или запрос*, *Источник строк* – *Группы* (рисунок 13).

Имя поля	Тип данных	Описание (необязательно)
КодСтудента	Счетчик	индивидуальный номер студента
Фамилия	Короткий текст	фамилия студента
Имя	Короткий текст	имя студента
Отчество	Короткий текст	отчество студента
Год рождения	Числовой	год рождения студента
Учебная группа	Числовой	номер группы студента

Свойства поля

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	Группы
Присоединенный столб	1
Число столбцов	1
Заголовки столбцов	Нет
Ширина столбцов	
Число строк списка	16
Ширина списка	Авто
Ограничиться списком	Нет
Разрешить несколько значений	Нет
Разрешить изменение списка	Да
Форма изменения элементов	
Только значения источника	Нет

Имя поля может содержать не более 64 знаков (включая пробелы). Для получения справки по именам полей нажмите клавишу F1.

Рисунок 13 – Значения поля *Учебная группа*, выбранные из списка, содержащегося в таблице *Группы*

Задайте ключевое поле *КодСтудента*, сохраните и закройте таблицу. Значение поля *КодСтудента* будет меняться автоматически при добавлении записей в таблицу, т.к. проставлен тип данных *Счетчик*.

Создайте таблицу *Личные данные* с ключевым полем *КодСтудента* (рисунок 14).

Имя поля	Тип данных	
КодСтудента	Счетчик	индивидуальный номер студента
Адрес	Короткий текст	адрес студента
Телефон	Короткий текст	телефон студента
Word	Числовой	оценка по Word
Excel	Числовой	оценка по Excel
Access	Числовой	оценка по Access

Рисунок 14 – Таблица *Личные данные* с ключевым полем *КодСтудента*

Обратите внимание, если вы создаете таблицу и не задаете ключевое поле, то при сохранении таблицы программа предложит вам задать его, при согласии автоматически будет создано дополнительное поле *Код* с типом данных *Счетчик*, выбранное ключевым. В случае, если необходимо внести изменения в таблицу, нужно открыть ее в режиме конструктора. Для этого щелкните на название нужной вам таблице, нажмите правую кнопку мыши и в ниспадающем меню выберите пункт *Конструктор*.

## Практическое задание 2.2

### Создание схемы данных

После того как созданы все необходимые таблицы, необходимо установить соответствие между записями таблиц. В поле *Учебная группа* таблицы *Список* должно подставляться значение из поля *Учебная группа* таблицы *Группы* (для этого в свойствах поля в конструкторе таблицы было выбрано поле со списком из таблицы «Группы»). Поле *КодСтудента* таблицы *Личные данные* будет соответствовать полю *КодСтудента* таблицы *Список*. Именно по значению этого поля одной записи таблицы *Список* будет ставиться в соответствие запись таблицы *Личные данные*.

Для этого необходимо установить связи между таблицами.

Выполните команду *Схема данных* (Меню *Работа с базами данных / Отношения / Схема данных*). В диалоговом окне *Добавление таблицы* выберите таблицы *Группы*, *Список* и *Личные данные* (выбрав таблицу, для добавления нажмите на кнопку *Добавить* или просто добавьте таблицу двойным щелчком мыши по ее названию). Закройте диалоговое окно *Схема данных*. В окне *Схема данных* появится условный вид этих таблиц (рисунок 15).

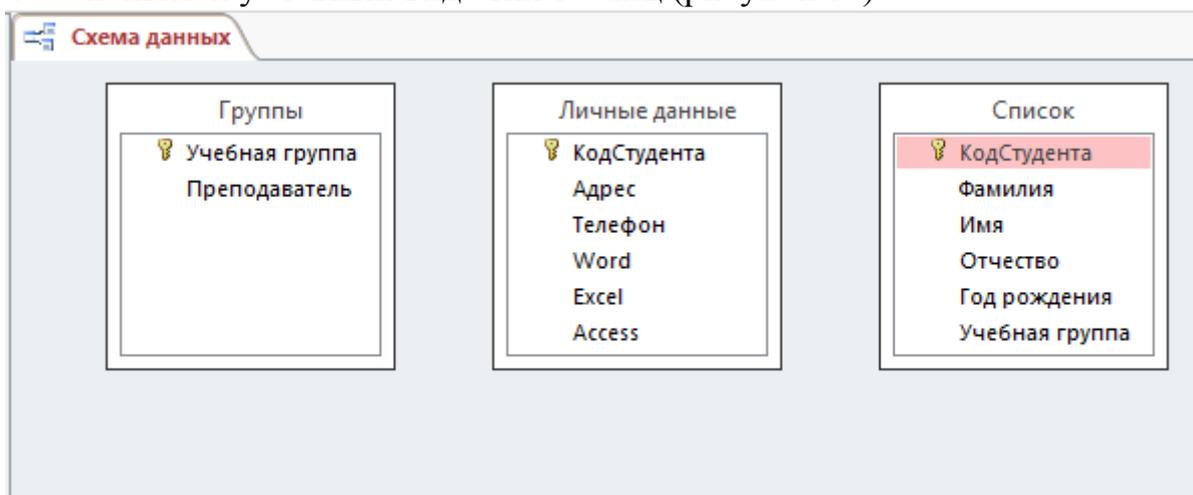


Рисунок 15 – Условный вид таблиц *Группы*, *Список* и *Личные данные*

Установите курсор мыши на имя поля *Учебная группа* в таблице *Группы* и, не отпуская левую кнопку мыши, перетащите ее на поле *Учебная группа* в таблице *Список*. Отпустите кнопку мыши. Появится окно *Изменение связей* (рисунок 16). Установите в этом окне флажки *Обеспечение целостности данных* (это невозможно будет сделать, если типы связываемых полей различны, т.е. в обеих таблицах типом данных поля *Учебная группа* должно быть *Числовое значение*), *Каскадное обновление связанных полей* (это приведет к тому, что при изменении номера группы в таблице *Группы* автоматически изменятся соответствующие номера групп в таблице *Список*), *Каскадное удаление связанных полей* (это приведет к тому, что при удалении записи с номером группы в таблице *Группы*, будут удалены все записи из таблицы *Список*, в которой стояли соответствующие номера групп). *Тип отношения* определится автоматически – *Один-ко-Многим* (это означает, что одной записи таблицы *Группы* может соответствовать несколько записей таблицы *Список*).

Нажмите кнопку *Создать*.

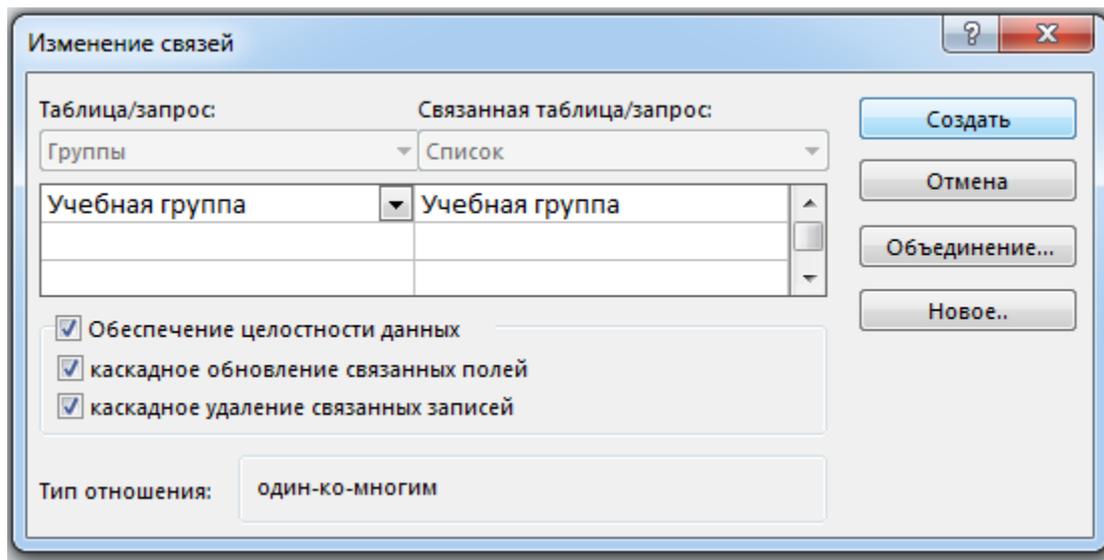


Рисунок 16 – Диалоговое окно *Изменение связей*

Аналогичным образом установите связь между полями *КодСтудента* таблицы *Личные данные* и *КодСтудента* таблицы *Список*. Установите флажок *Обеспечение целостности данных*. Должна установиться связь *Один-к-Одному*. В результате должна получиться схема данных как на рисунке 17.

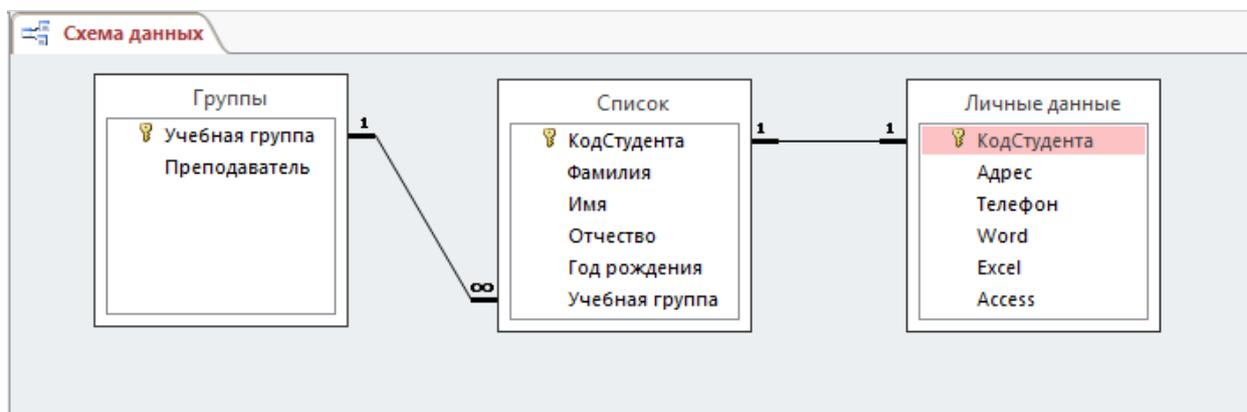


Рисунок 17 – Условная итоговая схема данных

Закройте схему данных, сохранив изменения.

### Практическое задание 2.3

#### Заполнение таблиц

В окне объектов Access выберите таблицу *Группы* и дважды щелкните по ней левой кнопкой мыши. Откроется пустая таблица *Группы*. Заполните таблицу значениями (рисунок 18).

	Учебная группа	Преподаватель	Щелкните для добавления
+	101	Верзаков С.А.	
+	102	Белоусов А.И.	
+	103	Макарова В.А.	
+	104	Новикова П.О.	
+	105	Качесова И.Н.	
*	0		

Рисунок 18 – Заполненная таблица *Группы* в окне объектов Access

Закройте таблицу, сохранив изменения.

### Практическое задание 2.4

#### Создание форм для заполнения таблиц

Каждую из созданных таблиц можно заполнять отдельно, но удобнее создать единую форму для заполнения таблиц.

Выполните команду *Мастер форм*. Меню *Создание / Мастер форм*. В списке *Таблицы и запросы* выбирается таблица, поля которой необходимо добавить на форму. Нажатие кнопки > добавляет выбранное поле из списка *Доступные поля* в список *Выбранные поля*. Нажатие кнопки >> добавляет все поля из списка *Доступные поля* в список *Выбранные поля*.

Перенесите в список *Выбранные поля* все поля из таблицы *Список* и поля *Адрес, Телефон, Word, Excel, Access* из таблицы *Личные данные*. Нажмите кнопку *Далее*. Выберите внешний вид формы *В один столбец*. Нажмите кнопку *Готово*. В результате будет создана форма *Личные данные*.

В окне объектов Access выберите форму *Личные данные* (рисунок 19) и дважды щелкните по ней левой кнопкой мыши. Откроется форма для заполнения таблиц *Список* и *Личные данные* (рисунок 20).

Значения кнопок работы с записями:

- Переход к первой записи;
- Переход к предыдущей записи;
- Переход к следующей записи;
- Переход к последней записи;
- Добавление новой записи.

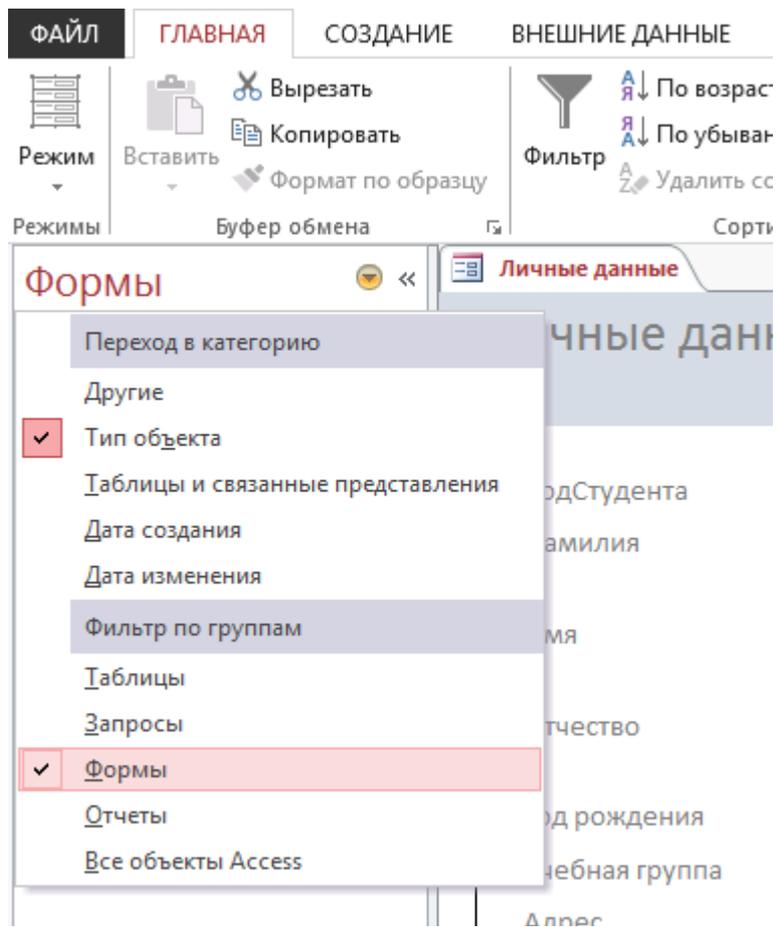


Рисунок 19 – Выбор формы

The image shows the 'Личные данные' (Personal Data) form. The form has a title bar with 'Личные данные' and a tab icon. The main area contains the following fields:

- КодСтудента: Text box containing '(No)'
- Фамилия: Text box
- Имя: Text box
- Отчество: Text box
- Год рождения: Text box
- Учебная группа: Dropdown menu
- Адрес: Text box
- Телефон: Text box
- Word: Text box
- Excel: Text box
- Access: Text box

At the bottom of the form, there is a status bar with the text 'Запись: 1 из 1', 'Нет фильтра', and 'Поиск'.

Рисунок 20 – Форма для заполнения таблиц *Список* и *Личные данные*

Заполните с помощью созданной формы таблицы *Список* (рисунок 21) и *Личные данные* (рисунок 22).

КодСтудент	Фамилия	Имя	Отчество	Год рождения	Учебная группа
1	Иванова	Анна	Ивановна	1994	101
2	Баранова	Ирина	Алексеевна	1993	102
3	Корнилова	Ирина	Алексеевна	1994	103
4	Воробьев	Алексей	Петрович	1993	101
5	Воробьев	Алексей	Иванович	1994	104
6	Воробьев	Олег	Григорьевич	1995	105
7	Скоркин	Александр	Евгеньевич	1992	101
8	Володина	Анна	Алексеевна	1994	102
9	Новоселов	Алексей	Антонович	1993	103
10	Александрова	Елена	Алексеевна	1994	101
*	(№)			0	

Рисунок 21 – Таблица *Список*

КодСтудент	Адрес	Телефон	Word	Excel	Access
1	Центральная, 51-17-22	51-17-22	5	5	5
2	Солнечная, 8-	51-18-22	4	4	4
3	Сиреневый, 7-	51-19-22	3	4	5
4	Центральная, 51-20-22	51-20-22	5	5	4
5	Сиреневый, 7-	51-21-22	5	4	4
6	Солнечная, 2-	51-22-22	4	4	4
7	Школьная, 5-8	51-23-22	5	5	5
8	Центральная, 51-24-22	51-24-22	5	5	5
9	Сиреневый, 7-	51-25-22	3	4	4
10	Солнечная, 6-	52-26-22	5	5	5
*	(№)		0	0	0

Рисунок 22 – Таблица *Личные данные*

Откройте таблицы *Список* и *Личные данные* и убедитесь, что в них появились записи.

## Практическое задание 2.5

### Создание фильтров

Фильтры используются в таблицах для отбора записей по определенным критериям.

Выражения в *Фильтре* могут состоять из точных значений, которые Access использует для сравнения в том виде, в котором они вводятся. Числа вводятся без ограничителей, например 22. Текст должен быть заключен в кавычки, например «Александров». Даты ограничиваются символами #, например, #10/01/99#. Элементы выражения могут быть связаны операторами:

- арифметическими:  $*$ ,  $+$ ,  $-$ ,  $/$ ,  $^$ ;
- сравнениями:  $<$ ,  $<=$ ,  $>=$ ,  $=$ ,  $<>$ ;
- логическими: **And (И)**, **Not (Нет)**, **Or (Или)**;
- **Like** – для использования логики замены в выражениях;
- **In** – для определения, содержится ли элемент данных в списке значений;

- **Between...And** – для выбора значений из определенного интервала.

Откройте таблицу *Список*. Отфильтруем все записи студентов с годом рождения 1994. Для этого установите курсор в поле *Год рождения* со значением 1994. Щелкните правой кнопкой мыши и выберите фильтр *Равно 1994*. Вы автоматически попадете в таблицу, в которой будут только выбранные надписи. Отменить фильтр можно, выбрав в Меню *Главная / Сортировка и фильтр / Фильтр* или, щелкнув в ячейке со значением фильтра (1994) и снова нажав правую кнопку мыши, выберите пункт *Снять фильтр с Год рождения*.

Для ввода своего фильтра выберите в Меню *Главная / Сортировка и фильтр / Дополнительно / Фильтр по форме*. Чтобы получить записи студентов, фамилии которых начинаются, например, на букву «В», в соответствующем поле наберите *Like «B\*»* (В – в данном случае, русская буква). Символ «\*» означает, что после буквы В могут идти любые символы. После ввода выражения нажмите Меню *Главная / Сортировка и фильтр / Дополнительно / Применить фильтр*.

## Практическое задание 2.6

### Создание запросов

Выполните команду *Конструктор запросов*. Меню *Создание / Запросы / Конструктор запросов*. Добавьте нужные таблицы (*Личные данные* и *Список*) – для этого дважды щелкните на нужную таблицу или, выбрав таблицу, нажмите на кнопку *Добавить*. Выберите поля *Фамилия*, *Имя* и *Отчество* из таблицы *Список*, *Телефон* – из таблицы *Личные данные* (рисунок 23). Для этого достаточно сделать двойное нажатие мышкой по имени поля. Второй вариант – перетащить мышкой название поля в клетки запроса.

Сохраните запрос под именем *Номера телефонов*. В результате выполнения (открытия) запроса отобразится таблица с именами, фамилиями, отчествами и номерами телефонов.

Чтобы вернуться в режим *Конструктора запросов*, выполните команду *Перейти в режим конструктора*. Щелкните на нужном запросе правой кнопкой мыши и выберите *Конструктор*.

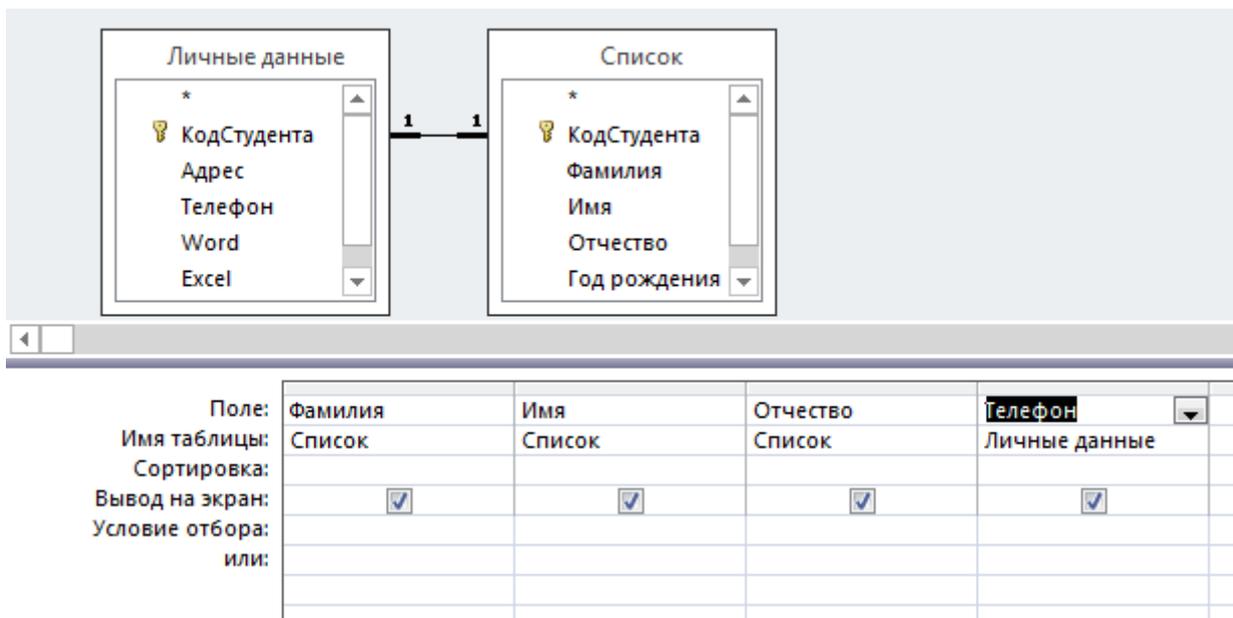


Рисунок 23 – Выбор полей *Фамилия, Имя и Отчество* из таблицы *Список*, *Телефон* – из таблицы *Личные данные*

Предположим, что вам нужно составить ведомость для выплаты стипендии всем студентам, которые учатся без троек. Для этого нужно выбрать записи, в которых оценки по предметам 4 или 5. Для этого откройте *Конструктор запросов* и добавьте нужные таблицы (*Личные данные* и *Список*). Выберите поля *Фамилия, Имя и Отчество* из таблицы *Список* и *Word, Excel* и *Access* – из таблицы *Личные данные*. В строке *Условие отбора* под полями *Word, Excel* и *Access* поставьте 4 Or 5 (рисунок 24).

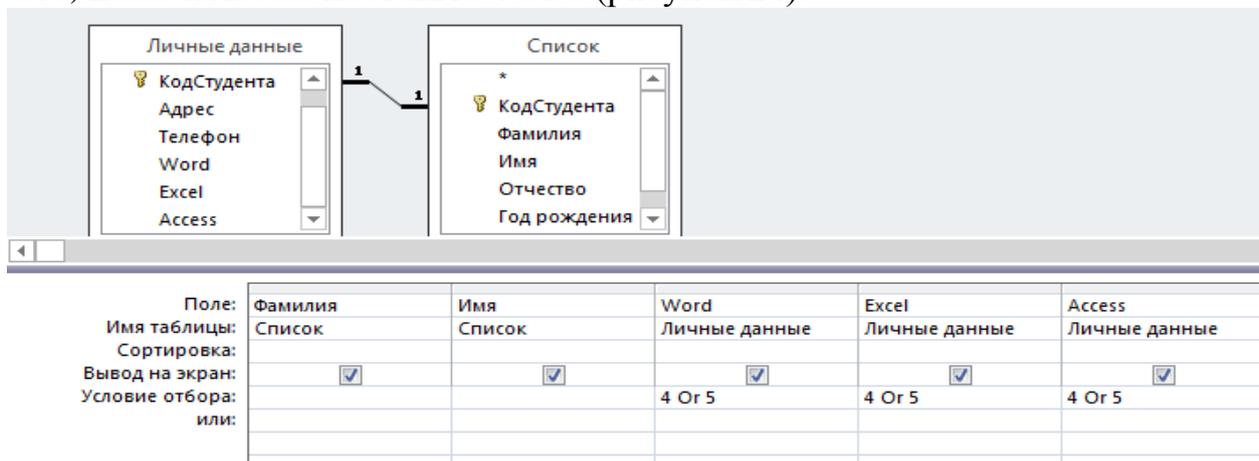


Рисунок 24 – Таблица с записями оценок только 4 или 5

Сохраните запрос под именем «Успеваемость». Выполните запрос. В результате будет отображена таблица, в которой содержатся записи с оценками только 4 или 5.

### Создание вычисляемых полей

Откройте *Конструктор запросов* и добавьте нужные таблицы (*Личные данные* и *Список*). Выберите поля *Фамилия* и *Имя* из таблицы *Список* и поля

Word, Excel и Access – из таблицы *Личные данные*.

Поставьте курсор на клетку правее Access в строке *Поле* и введите в нее *Среднее: ([Word]+[Excel]+[Access])/3*. Сохраните запрос с именем *Средний балл* (рисунок 25).

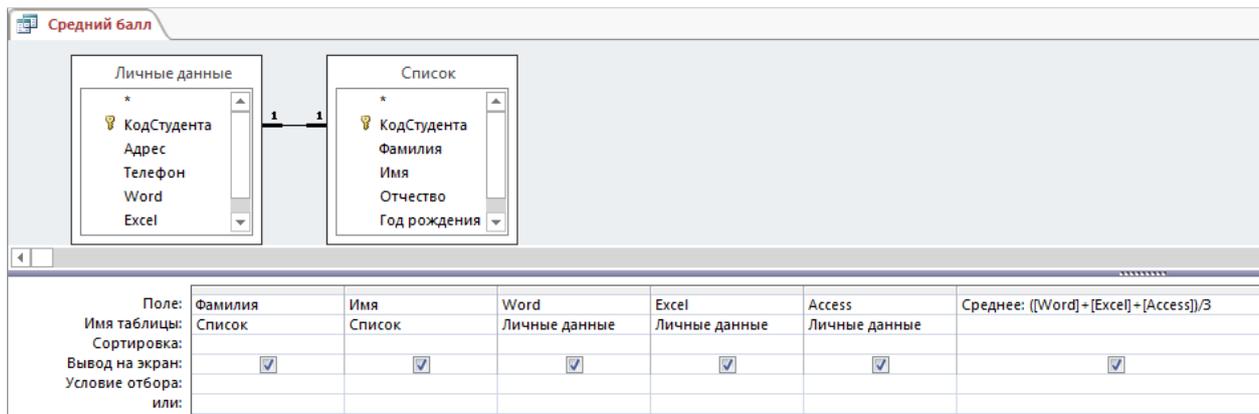


Рисунок 25 – Таблица с именем *Средний балл*

### Создание отчетов

Создайте отчет на основе запроса *Успеваемость*. Выполните команду *Мастер отчетов. Меню / Создание / Отчеты / Мастер отчетов*. В качестве источника отчета выберите *Запрос: Успеваемость*. Добавьте все доступные поля в список *Выбранные поля*. Нажмите кнопку *Готово*. Отобразится отчет на основании запроса *Успеваемость* (рисунок 26).

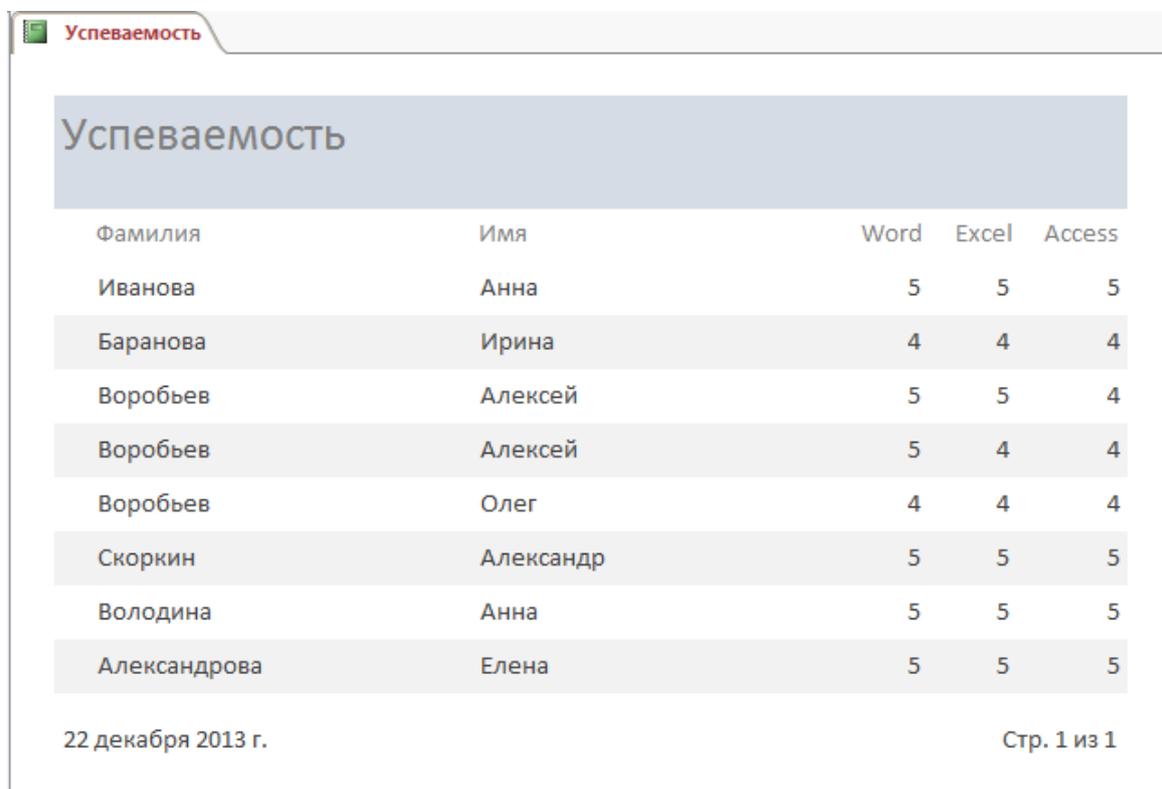


Рисунок 26 – Отчет на основании запроса *Успеваемость*

## Практическое задание 2.7

### Создание кнопочной формы

Выполните команду *Конструктор форм. Меню / Создание / Формы / Конструктор форм*. Появится пустая заготовка формы. Мышью, растягивая за края, можно изменить размеры формы. В *Меню / Конструктор / Элементы управления* выберите объект *Надпись*, добавьте его на форму и напишите *База данных «Компьютерная школа»*. Измените размеры надписи так, чтобы текст располагался в две строки. Щелкните по надписи правой кнопкой мыши и в контекстном меню выберите *Свойства*. Для свойства *Выравнивание текста* в открывшемся окне свойств задайте значение *По центру*. В этом же окне можно задать параметры оформления надписи (рисунок 27).

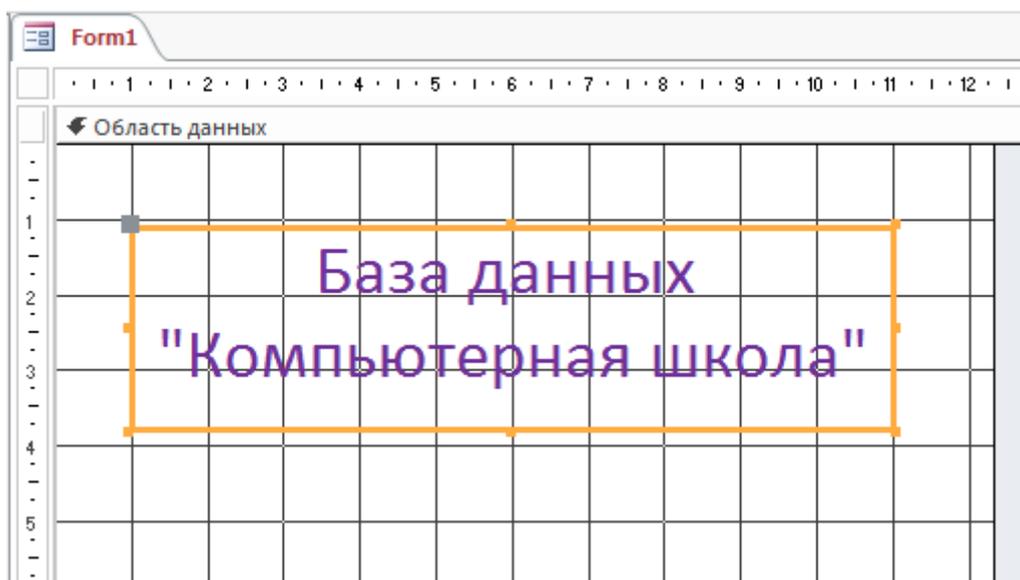


Рисунок 27 – Параметры оформления надписи

Добавьте на форму элемент *Кнопка*. Автоматически отобразится окно *Создание кнопок* (рисунок 28). В списке *Категории* выберите *Работа с формой*, в списке *Действие* – *Открыть форму*, нажмите кнопку *Далее*. В следующем окне выберите форму *Личные данные* и нажмите кнопку *Далее*. В следующем окне установите переключатель *Открыть форму и показать все записи*, нажмите *Далее*. Установите переключатель *Текст* и в соседнее поле впишите *Форма Личные данные*. Нажмите кнопку *Готово*.

Аналогичным образом создайте кнопку для отчета *Успеваемость*, выбрав категорию *Работа с отчетами (Просмотр отчета)*.

Создайте кнопку для закрытия базы данных. Для этого в диалоговом окне *Создание кнопок* выберите категорию *Приложение* и действие *Выйти из приложения*.

Закройте форму и сохраните ее под именем *Заставка*. Законченный вид формы показан на рисунке 29.

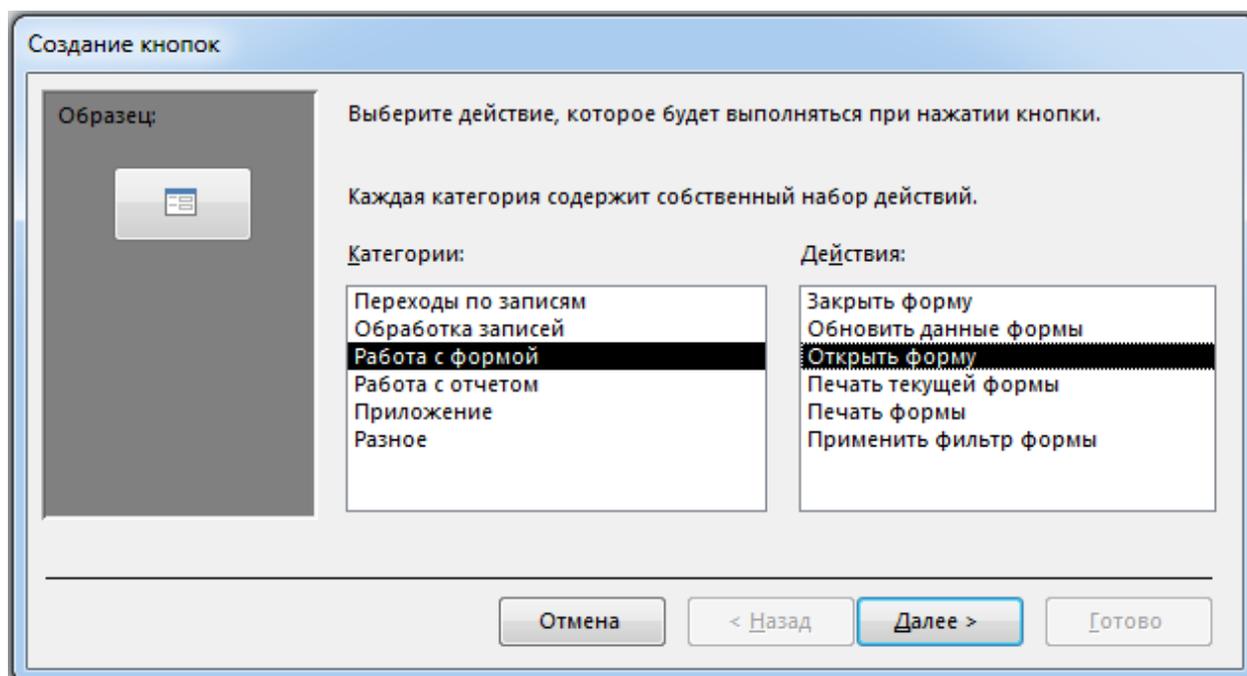


Рисунок 28 – Диалоговое окно *Создание кнопок*

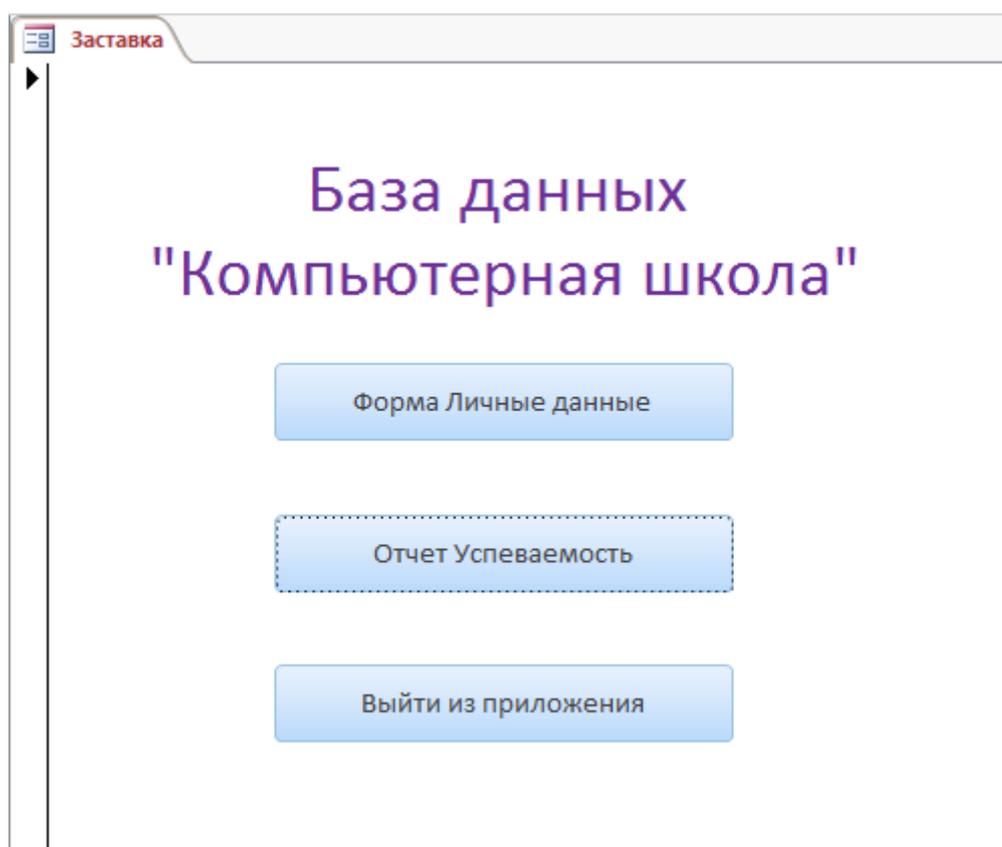


Рисунок 29 – Законченный вид формы под именем *Заставка*

Чтобы кнопочная форма автоматически появлялась при запуске базы данных, выполните команду Меню *Файл / Параметры / Текущая база данных /*

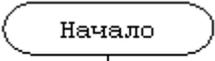
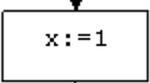
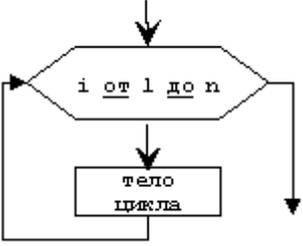
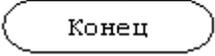
Форма просмотра и выберите *Заставка*, нажмите кнопку *ОК*. Закройте форму, ответив утвердительно на возможный вопрос о ее сохранении. Завершите работу с программой *Access*.

### 3 БЛОК-СХЕМЫ И АЛГОРИТМИЗАЦИЯ

Внутри алгоритмов можно выделить группы шагов, отличающиеся внутренней структурой – алгоритмические конструкции.

Основными алгоритмическими конструкциями являются линейная последовательность шагов, ветвление и цикл.

Таблица 4 – Условные обозначения блок-схем

Условное графическое обозначение	Название	Комментарий
	Стрелка	Означает, к исполнению какого действия следует перейти. С помощью стрелки обозначают последовательность исполнения команд
	Начало	Точка блок-схемы, с которой начинается исполнение алгоритма
	Ввод или Вывод	Блок, означающий, что в этом месте алгоритма необходимо произвести ввод или вывод данных
	Простое действие (в данном случае – присваивание)	Один элементарный шаг алгоритма. Присваивание значения выражения переменной (в данном случае переменной x присвоено значение 1)
	Условие	Исполнение предполагает вычисление условия – логического выражения. Если условие истинно (Истина, True), то необходимо перейти к действию по стрелке помеченной Т, если условие ложно – то по стрелке F (Ложь, False)
	Модификация (цикл со счетчиком)	Повторение исполнения тела цикла для каждого из последовательных значений целочисленного счетчика i от 1 до n.
	Конец	Точка блок-схемы, дойдя до которой прекращается процесс исполнения алгоритма.

### Практическое задание 3.1

#### Линейная последовательность шагов

Группа шагов алгоритма, всегда выполняемых последовательно друг за другом без каких-либо условий, называется линейной последовательностью. Если весь алгоритм представляет собой линейную последовательность шагов, то его называют линейным.

На рисунке 30 изображена блок-схема линейного алгоритма, состоящего из двух шагов.

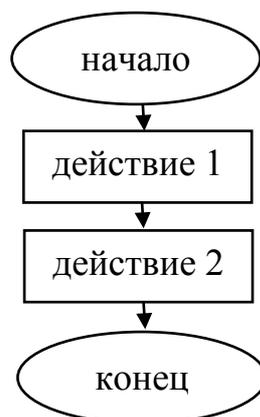


Рисунок 30 – Пример линейного алгоритма

Пример блок-схемы алгоритма, решающего следующую задачу. Даны три вещественных положительных числа  $a$ ,  $b$  и  $c$ . Найти площадь треугольника, стороны которого равны  $a$ ,  $b$  и  $c$ . Для нахождения площади треугольника по трем его известным сторонам воспользуемся формулой Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где  $p$  – полупериметр треугольника, равный

$$p = \frac{a + b + c}{2}.$$

Алгоритм будет выглядеть так, как показано на рисунке 31.

Задача решена. Обратите внимание, что в отличие от математики, где приводимые формулы декларируют некоторые факты, соотношения и порядок указания которых не так важен (или важен только с точки зрения логичности изложения), в алгоритмических схемах важна в первую очередь правильная последовательность действий (формул), которая и определяет порядок выполнения шагов. Например, если в приведенной блок-схеме переставить местами шаги, вычисляющие  $S$  и  $p$ , то алгоритм не будет правильным, так как до вычисления  $S$  необходимо предварительно вычислить  $p$ .

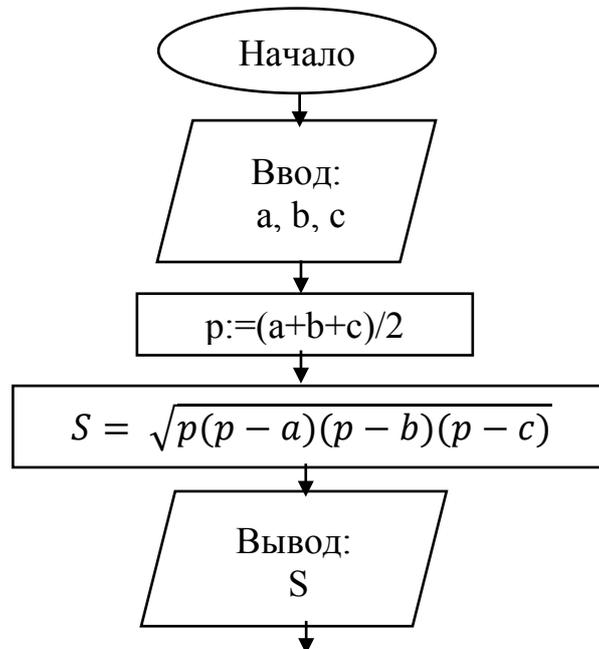


Рисунок 31 – Блок-схема решения задачи линейного алгоритма

### Задача для самостоятельного решения

Аналогичным образом создайте блок-схему для решения следующей задачи. Необходимо вычислить сумму НДС (N) реализованного магазином товара и его общую стоимость (S) по заданному количеству реализованного товара (K) и цене единицы товара (P), при условии, что НДС = 18%. Вывести значения НДС и общей стоимости.

### 2 Ветвление

Ветвление представляет собой алгоритмическую конструкцию, в которой выполнение того или иного шага зависит от истинности условия.

На рисунке 32 приведена блок-схема ветвления.

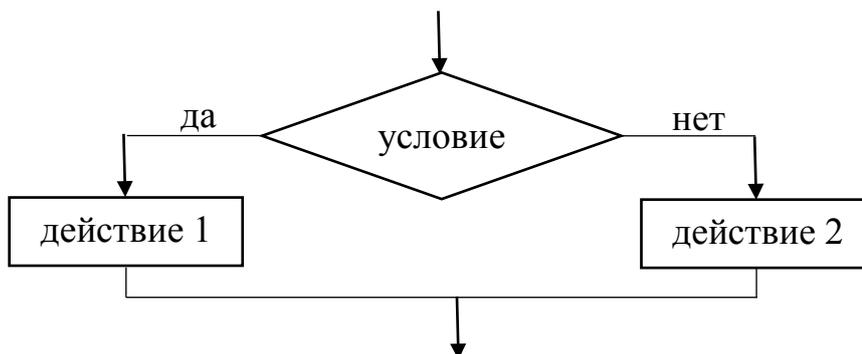


Рисунок 32 – Пример блок-схемы с ветвлением

Если условие истинно, то будет выполнено только действие1, в противном случае будет выполнено только действие2.

В языках программирования высокого уровня ветвление обычно реализуется с помощью оператора (команды) IF. (Например, оператор If в Паскале, оператор If в языке Basic).

Пример решения задачи на алгоритм с ветвлением. Даны значения трех действительных переменных  $a$ ,  $b$  и  $c$ . Найти наибольшее значение из  $a$ ,  $b$  и  $c$ . Для решения подобной задачи вначале нужно найти наибольшее значение из  $a$  и  $b$ , а потом сравнить его и  $c$ , снова найти наибольшее из двух. Это можно представить такой блок-схемой (рисунок 33).

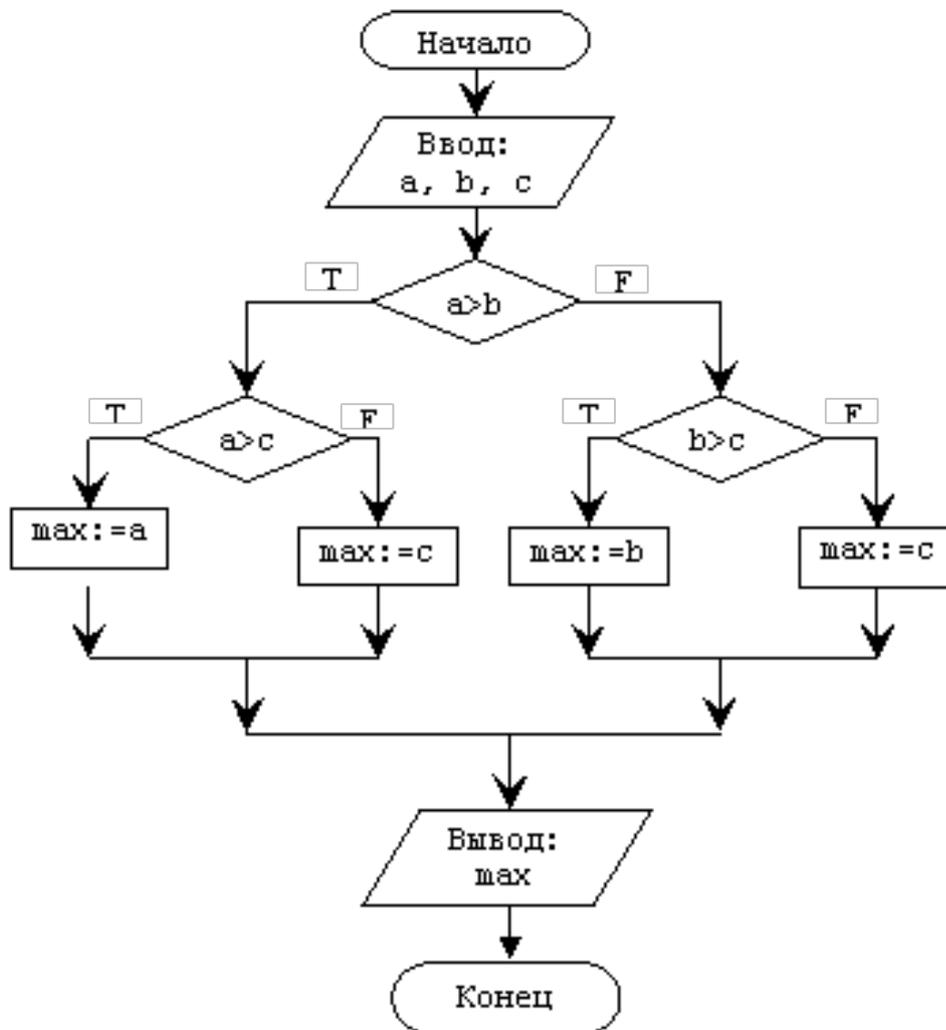


Рисунок 33 – Блок-схема решения задачи алгоритма с ветвлением

Примечание: в данной блок-схеме функцию Да выполняет Т (True – Истина), а функцию Нет – F (False – Ложь).

### Задача для самостоятельного решения

Определить сумму (SP), которую получит вкладчик по прошествии года, если он вкладывает сумму –  $S$ , и проценты начисляются следующим образом: до 10 000 р. – 5%; от 10 000 р. до 50 000 р. – 10%; более 50 000 р. – 15%.

### 3 Цикл

Цикл представляет собой алгоритмическую конструкцию, в которой многократно выполняется одна и та же последовательность шагов, называемая телом цикла. Каждое однократное исполнение тела цикла называется

итерацией. Если тело цикла было выполнено  $N$  раз, говорят, что было произведено  $N$  итераций.

Для того, чтобы определить момент прекращения выполнения тела цикла, используется условие цикла. Если при истинности условия цикл продолжается, то такое условие называется условием продолжения цикла. Иными словами, цикл продолжается, пока условие цикла истинно.

Если при истинности условия цикл завершается, то такое условие называется условием завершения цикла. В этом случае цикл продолжается до тех пор, пока условие цикла не станет истинным.

Различают циклы с проверкой условия перед выполнением очередной итерации (рисунок 34) и циклы с проверкой условия после выполнения очередной итерации (рисунок 35). Первые называются циклами с предусловием, вторые – с постусловием.

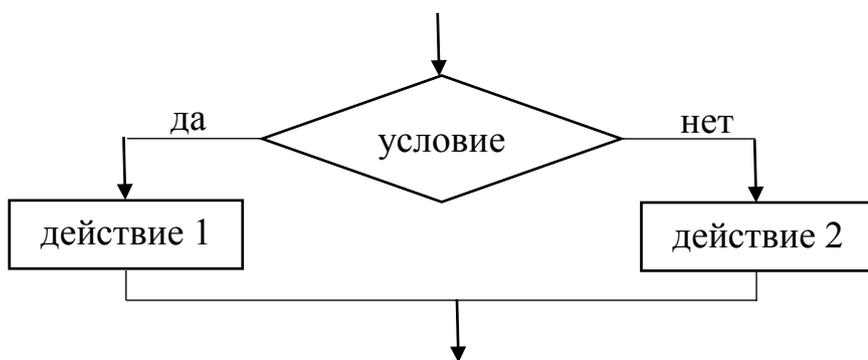


Рисунок 34 – Блок-схема цикла с предусловием продолжения

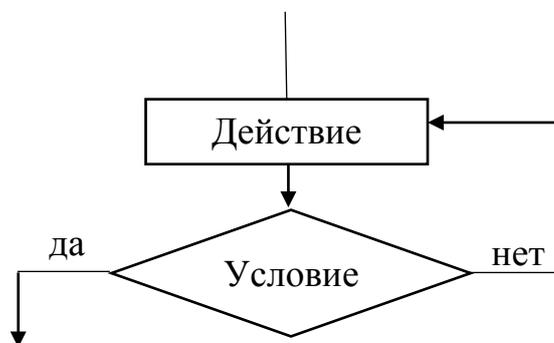


Рисунок 35 – Блок-схема цикла с постусловием завершения

Тело цикла с постусловием всегда выполнится хотя бы один раз.

*Циклический алгоритм типа «Пока»* (рисунок 36) – это такой циклический алгоритм, действия которого будут выполняться до тех пор **пока** выполняется заданное условие. В цикле типа «Пока» – пока выполняется условие происходит действие (или последовательность действий). *Циклический алгоритм типа «Для»* (рисунок 37) – это такой циклический алгоритм, в котором число повторений известно. В цикле типа «Для» – действие (последовательность действий) выполняется для  $i$  от  $i_1$  до  $i_2$ .

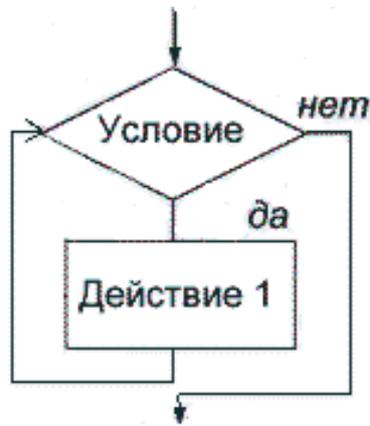


Рисунок 36 – Цикл типа «Пока»



Рисунок 37 – Цикл типа «Для»

В языках программирования высокого уровня существуют различные операторы циклов (например: циклы в Паскале, циклы в Basic), в том числе реализующие циклы с заранее заданным количеством итераций, так называемые циклы со счетчиком.

Цикл со счетчиком состоит из заголовка и тела цикла. В заголовке указывается начальное и конечное значение счетчика. На каждой итерации значение счетчика автоматически увеличивается. Цикл завершается, когда счетчик достигнет конечного значения. Фактически, цикл со счетчиком представляет собой разновидность цикла с предусловием продолжения, заключающемся в том, что значение счетчика находится в заданных границах.

**Пример решения задачи на алгоритм типа «Пока»**

Дано: Клиент положил в банк денежную сумму в 10 000 р. (S)  
 Ежегодный процент = 14%. Через сколько лет сумма в банке превысит 25 000 р.? (Начисляется сложный процент).

Решение задачи приведено на рисунке 38.

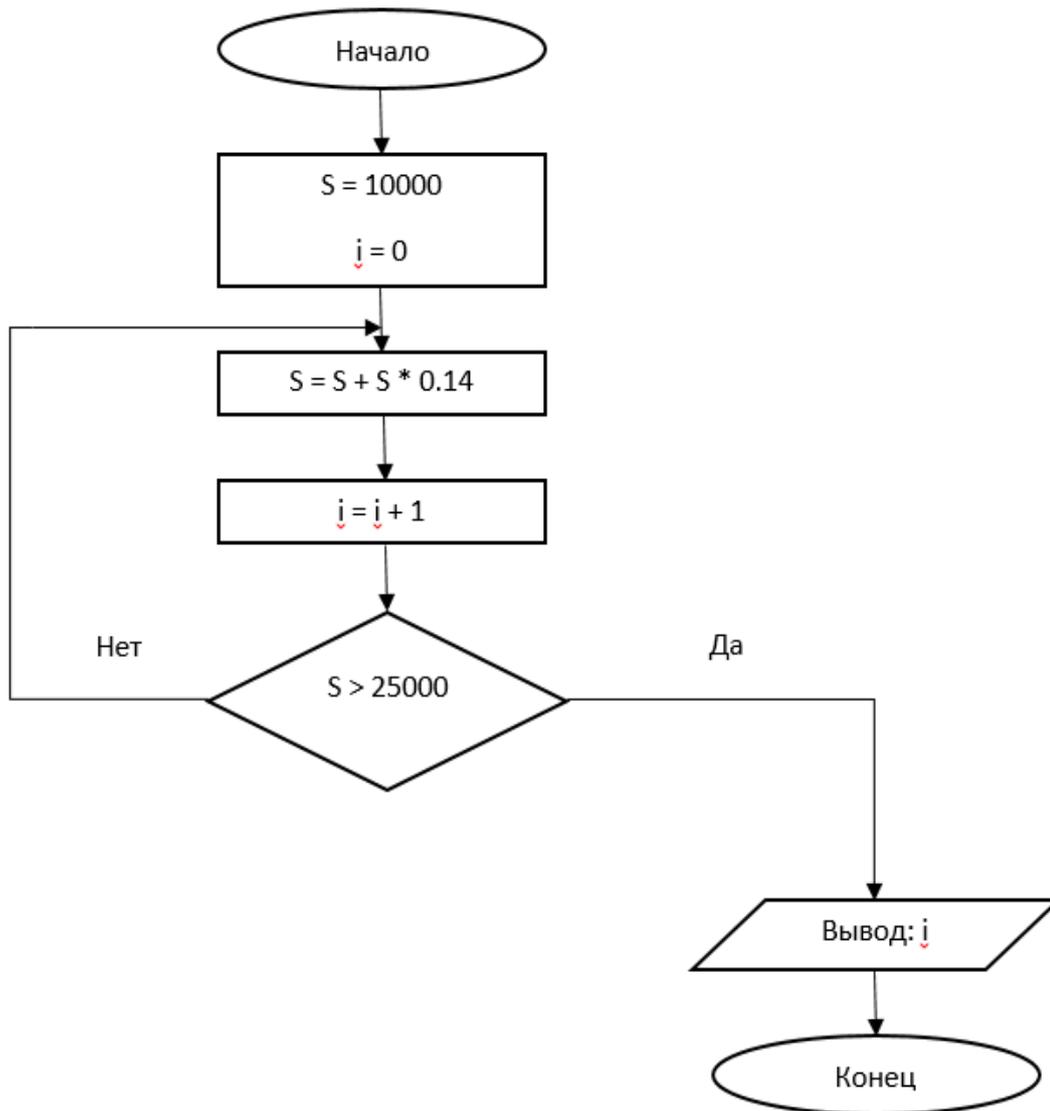


Рисунок 38 – Блок-схема решения задачи алгоритма типа «Пока»

### Задача для самостоятельного решения

Дано: Машина поставлена на платную стоянку – за каждый день клиент оплачивает 50 р. За 30 дней – плата составляет 1000 р., за дни свыше 30 – плата 45 р. за каждый день. Вывести количество дней, за которое плата за стоянку превысит 2000 р.

### Пример решения задачи на алгоритм типа «Для»

Дано: Машина поставлена на стоянку на 10 дней ( $i$  – количество дней (или счетчик) от 1 до 10). Вывести общую сумму стоянки автомобиля на каждый из 10 дней при цене за день равной  $P$ .

Решение задачи приведено на рисунке 39.

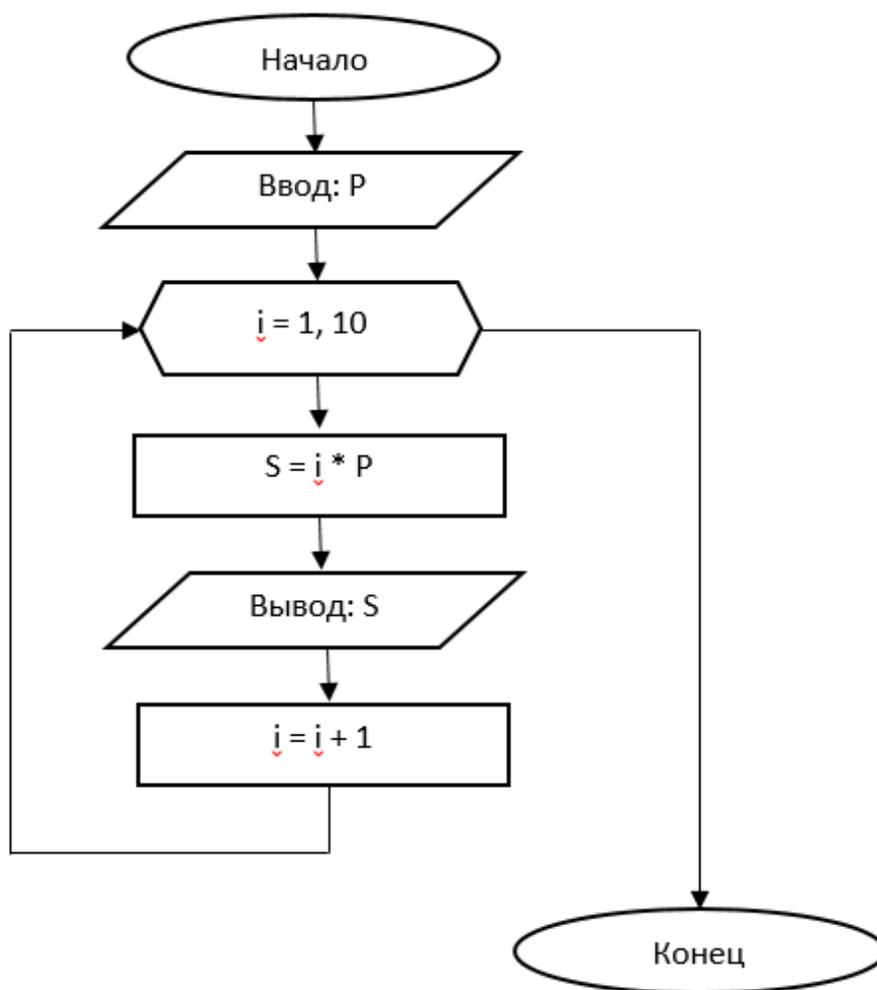


Рисунок 39 – Блок-схема решения задачи алгоритма типа Для

### Задача для самостоятельного решения

Дано: Денежная сумма положена на банковский счет на 5 лет ( $i$  – количество лет (или счетчик) от 1 до 5). Изначальная сумма равна  $S$ . Ежегодный процент начисляется по сложной процентной ставке и равняется 12%. Вывести сумму на счете на каждый год пребывания денежных средств в банке.

### 4 Решение задач по вариантам

#### Вариант 1

1) Торговый агент получает вознаграждение в размере некоторой доли от суммы совершенной сделки: если объем сделки до 12000 р., то в размере 3%; если объем до 19 000 р. – 5%; если выше – 7%. Определить размер вознаграждения торгового агента.

2) Написать алгоритм, который вычислял бы сумму ряда  $F(N)$ . Члены ряда, это все числа кратные 3. Т.е. каждый элемент должен делиться на 3 без остатка.  $F(N)=3 + 6 + 9 + 12 + \dots + N$ .

### Вариант 2

1) Менеджер получает комиссионные в зависимости от объема продаж за месяц: если объем продаж менее 50 000 р., то ставка комиссионных составляет 5%, если от 50 000 до 70 000 р. – 7%, более 70 000 р. – 9,5%. Определить сумму комиссионных менеджера.

2) Написать алгоритм, который бы генерировал случайным образом следующие данные. Дан периметр треугольника  $P$  (целое число). Используя циклическую структуру «Пока» – подобрать размер сторон  $a, b, c$  (целые числа) треугольника таким образом, чтобы  $a+b+c = P$ , а также  $a^2+b^2=c^2$ . Случайное число можно сгенерировать с помощью функции **Random**.

### Вариант 3

1) Рассчитать заработную плату сотруднику с учетом подоходного налога. Подоходный налог высчитывается по следующей схеме: если начисленная зарплата составляет менее 4200 р., то налог не взимается, если от 4200 р. до 100 тыс. р., то он составляет 13% от начисленной суммы, а если начисленная зарплата составляет более 100 тыс. р., то к 13% от разницы начисленной суммы и 100 тыс. р. прибавляется еще 18 200 р.

2) Написать алгоритм, который вычислял бы сумму ряда  $F(a)$ , если каждый элемент ряда вычисляется по формуле:  $2*\sin(0) + 2*\sin(0,3) + \dots + 2*\sin(a)$ , шаг  $0,3$ .

### Вариант 4

1) Сотрудник фирмы получает премию по итогам работы за месяц по следующей схеме: если заключено сделок менее чем на 10 000 р., то премия составляет 5% от общей стоимости сделок, если не меньше чем на 25 000 р., но менее чем на 50 000 р. – 7,5%, если же не менее 50 000 р. – 11,3%. Определить размер премии сотрудника.

2) Используя циклическую структуру «Пока», нарисовать блок-схему алгоритма, который бы генерировал случайным образом следующие данные. Дано число  $S$  (запрашивается у пользователя), сгенерировать  $s1$  и  $s2$ , таким образом, чтобы  $s1+s2=S$ , числа  $s1$  и  $s2$  были положительными и не повторялись.

### Вариант 5

1) Рассчитываясь в одном из магазинов, покупатель получает скидку: если количество покупаемого им товара не менее 100, то скидка составляет 10%, если от 50 до 100, то 5%, в остальных случаях скидка не предоставляется. Определить, какую сумму надо заплатить покупателю, если он покупает товар в количестве  $N$  штук по цене  $S$  р. каждая.

2) Написать алгоритм, который вычислял бы сумму ряда  $F(N)$  при следующей функции  $F(N) = 1+3+5+7+\dots+M$ .  $M=N$ , если  $N$  – нечетное и  $M = N-1$ , если  $N$  – четное.

## СПИСОК ЛИТЕРАТУРЫ

- 1 Гобарева, Я. Л. Технология экономических расчетов средствами MS Excel [Текст] : учебное пособие / Я. Л. Гобарева, О. Ю. Городецкая, А. В. Золотарюк. – М.: КноРус, 2010.
- 2 Золотарюк, А. В. Технология работы с Microsoft Office [Текст] : учебное пособие / А. В. Золотарюк. – М. : Академический проект, 2010. – 411 с.
- 3 Косарев, В. П. Информатика. Практикум для экономистов [Текст] : учебное пособие / В. П. Косарев, Е. А. Мамонтова. – М. : Финансы и статистика; ИНФРА-М, 2011. – 544 с.
- 4 Практикум по экономической информатике : учебное пособие / под ред. Е. Л. Шуримова, Н. А. Тимаковой, Е. А. Мамонтовой. – М. : Перспектива, 2010. – 300 с.
- 5 Практикум по экономической информатике : учебное пособие для студентов вузов в 3 ч. Ч. 1 / под ред. Е. Л. Шуримова, Н. А. Тимаковой, Е. А. Мамонтовой. – М. : Финансы и статистика : Перспектива, 2009.
- 6 Экономическая информатика : учебник / под ред. В. П. Косарева. – 3-е изд. – М. : Финансы и статистика, 2010. – 656 с.
- 7 Экономическая информатика : учебное пособие / под ред. Д. В. Чистова. – М. : КноРус, 2009. – 514 с.

Филимонов Сергей Михайлович  
Студентова Екатерина Александровна

**ЭКОНОМИЧЕСКАЯ ИНФОРМАТИКА  
(ЧАСТЬ 2)**

Методические указания  
к выполнению практических и самостоятельных заданий  
для студентов направлений 080100.62, 080200.62, 100762.62 и  
специальностей 080101.65, 036401.65  
очной формы обучения

Редактор Е.А. Могутова

---

Подписано в печать 30.09.06	"Формат 60x84 1/18	Бумага 65 г/м <sup>2</sup>
Печать цифровая	Усл. печ.л. 2,5	Уч.-изд. л. 2,5
Заказ 437	Тираж 27	Не для продажи

---

РИЦ Курганского государственного университета.  
640000, г. Курган, ул. Советская, 63/4.  
Курганский государственный университет.