



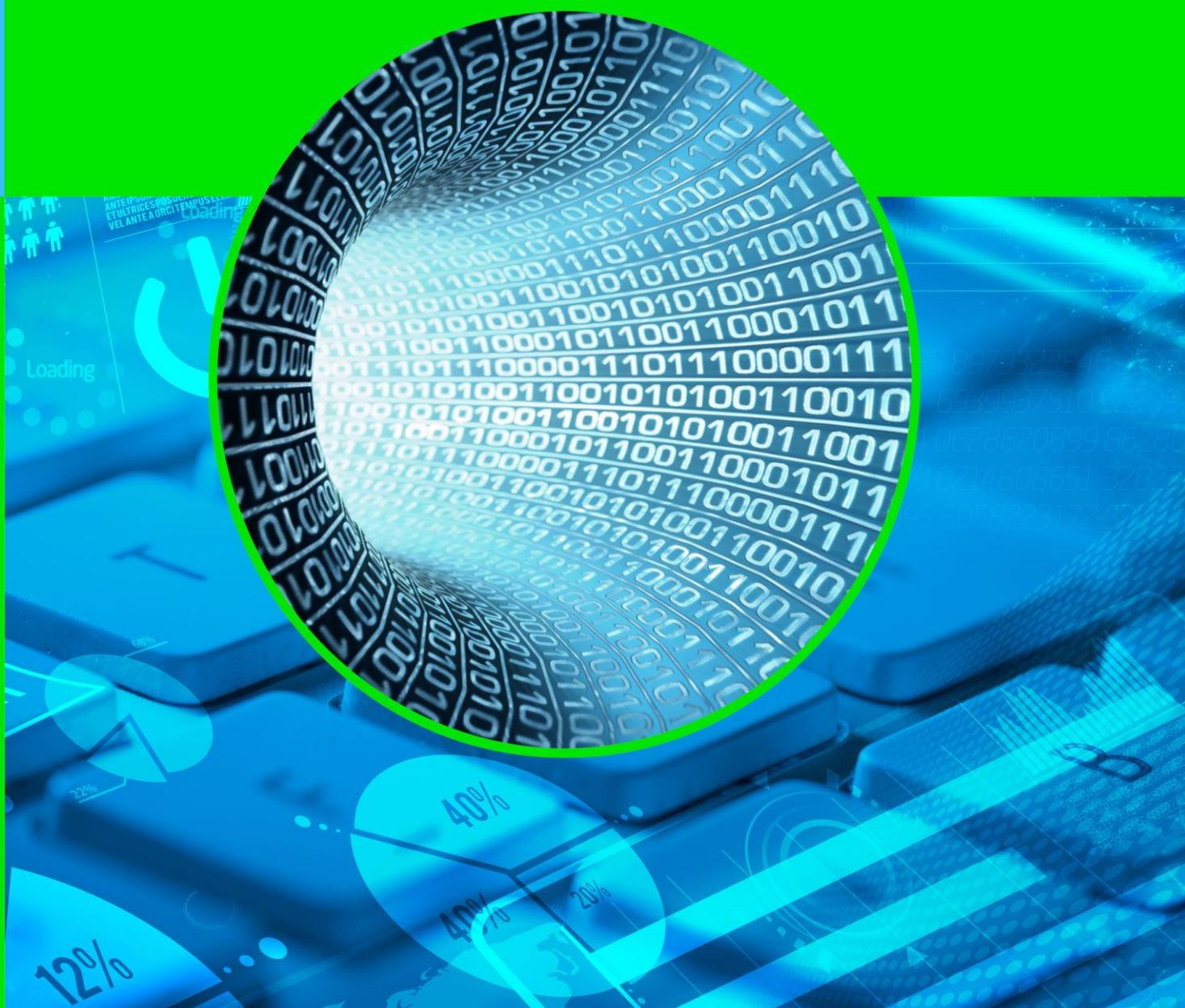
СЕЧЕНОВСКИЙ  
УНИВЕРСИТЕТ

С.В. Грейбо  
Т.Е. Новосёлова  
Н.Н. Пронькин  
И.Ф. Семёнычева

16+

# ИНФОРМАТИКА

УЧЕБНОЕ ПОСОБИЕ



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
ПЕРВЫЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ МЕДИЦИНСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ И.М. СЕЧЕНОВА  
МИНИСТЕРСТВА ЗДРАВООХРАНЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
(СЕЧЕНОВСКИЙ УНИВЕРСИТЕТ)

ГРЕЙБО Сергей Владимирович  
НОВОСЁЛОВА Тамара Евгеньевна  
ПРОНЬКИН Николай Николаевич  
СЕМЁНЫЧЕВА Ирина Флюровна

# **ИНФОРМАТИКА**

**Учебное пособие**

Москва  
2019

УДК 004.42(075.8)  
ББК 32.973.026-018я73  
И74

### Рецензент:

Герасимов А.Н., д.ф.-м..н., проф. Первый Московский государственный медицинский университет имени И.М. Сеченова (Сеченовский Университет)

### Авторы:

Грейбо Сергей Владимирович  
Новосёлова Тамара Евгеньевна  
Пронькин Николай Николаевич  
Семёнычева Ирина Флюровна

Информатика [Электронный ресурс]: учебное пособие – Эл. изд. - Электрон. текстовые дан. (1 файл pdf: 132 с.). - Грейбо С.В., Новосёлова Т.Е., Пронькин Н.Н., Семёнычева И.Ф. 2019. – Режим доступа: <http://scipro.ru/conf/Informatics.pdf>. Сист. требования: Adobe Reader; экран 10'.

ISBN 978-5-907072-88-6

Учебное пособие по изучению дисциплины «Информатика» разработано в соответствии с требованиями государственного образовательного стандарта и включает методические указания по изучению дисциплины, в том числе по проведению практических занятий, список рекомендованной литературы, задания для практических занятий, контрольные вопросы.

Материал излагается простым, доступным языком.

Учебное пособие рассмотрено и одобрено на заседании кафедры «Медицинская информатика и статистика» 21 января 2019 г., протокол № 5.

При верстке электронной книги использованы материалы с ресурсов: Designed by Freepik

ISBN 978-5-907072-88-6



© Грейбо С.В., Новосёлова Т.Е., Пронькин Н.Н., Семёнычева И.Ф. 2019  
© Первый Московский государственный медицинский университет имени И.М. Сеченова (Сеченовский Университет), 2019  
© Оформление: издательство НОО Профессиональная наука, 2019

# Содержание

Введение .....	5
Тема 1. Введение в информатику .....	8
Тема 2. Текстовый редактор MS WORD .....	11
Тема 3. Электронные таблицы MS EXCEL.....	12
<i>Практическое занятие по теме «Введение в информатику» .....</i>	<i>13</i>
<i>Практическое занятие по теме «Текстовый редактор MS Word»</i> .....	<i>14</i>
<i>Практическое занятие по теме «Электронные таблицы MS Excel».....</i>	<i>20</i>
<i>Практическое занятие по теме «БАЗЫ ДАННЫХ MS ACCESS»</i>	<i>38</i>
<i>Программирование НА VISUAL BASIC for APPLICATIONS (VBA)</i>	<i>78</i>
Контрольные вопросы .....	117
Словарь терминов и определений .....	123
Библиографический список .....	128

## Введение

Дисциплина «Информатика» входит в федеральный компонент блока обще- профессиональных дисциплин для подготовки бакалавров по направлениям: 09.03.02 «Информационные системы и технологии», 22.03.01 «Материаловедение и технологии материалов», 28.03.03 «Наноматериалы», 45.03.04 «Интеллектуальные системы в гуманитарной сфере».

Целью дисциплины является изучение процессов информационного обмена в современном обществе, принципов работы вычислительных машин, процессов создания и поддержания электронного офиса.

Задачи дисциплины – приобретение практических навыков по работе на персональном компьютере и по освоению современных пакетов программ; владение техникой использования современного программного обеспечения для задач обработки биомедицинской информации.

### ОСНОВЫ ИНФОРМАТИКИ

#### *Тема 1. Введение в информатику*

Понятие информации, виды и способы ее представления, измерения и получения. Кодирование и обработка информации на компьютере. История развития и роль информатики как науки. Современные информационные компьютерные системы, перспективы и темпы развития.

#### *Тема 2. Аппаратная часть компьютерных систем*

Принципы построения ЭВМ. Архитектура Фон-Неймана. Арифметико-логические основы и структура ЭВМ. Двоичная арифметика и алгебра логики. Структура и принцип функционирования ПК. Понятие архитектуры и конфигурации ПК.

#### **Тема 3. Основные и периферийные устройства ПК.**

Основные устройства ПК, их краткие технические характеристики. Назначение и характеристики дополнительных устройств ПК. Демонстрация работы устройств (отдельно и в сети). Назначение и состав клавиатуры ПК. Порядок подготовки, эксплуатации и завершения работы на ПК.

## ОСНОВЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ

### *Тема 4. Основы программного обеспечения компьютера.*

Программное обеспечение ПК. Операционные системы MS DOS, MAC DOS, UNIX, OS/2, WINDOWS. Системные оболочки. Назначение и возможности. Понятие о иерархической файловой структуре. Работа с каталогами и дисковыми устройствами. Работа с файлами.

### *Тема 5. Операционная система Windows.*

Назначение и возможности Windows. Порядок загрузки Windows. Объекты операционной системы и их свойства. Программа ПРОВОДНИК. Стандартные приложения. *Практическая работа в WINDOWS.* «Создание профиля пользователя». Работа с дисками, папками и файлами. Стандартные приложения. Простейшие графические и текстовые редакторы. Создание простейших текстовых и графических файлов. Контрольная работа.

## ПРИКЛАДНЫЕ ПРОГРАММНЫЕ ПРОДУКТЫ ДЛЯ WINDOWS

### *Тема 6. Текстовый редактор Winword.*

Word как составная часть MS Office. Назначение и возможности редактора Winword. Элементы экрана. Режимы просмотра документов. Масштаб. Создание, открытие и сохранение документов. Ввод и редактирование текста. Спецсимволы. Орфография. Переносы. Форматирование документов. Стили. Колонтитулы. Печать текста.

Работа с командами и инструментами Word. Создание колонок текста. Работа с таблицами. Оформление и заливка. Списковые структуры. Формы. Методы защиты документов и текстовой информации при совместном использовании в локальных и глобальных компьютерных сетях. Вставка и редактирование объектов. Работа с кадрами и графикой. Подготовка учебного текстового документа на оценку.

### *Тема 7. Электронные таблицы Excel*

Основные понятия и элементы ЭТ. Окно Excel. Рабочие листы. Сетка. Масштаб. Создание, открытие и сохранение документа. Ввод данных. Орфография. Типы и форматы данных. Автозаполнение. Автовод. Ввод рядов данных. Копирование информации в соседние ячейки. Выделение ячеек и диапазонов. Присвоение имен и их использование. Автовычисления. Формулы в ЭТ. Ввод и редактирование формул. Операции и аргументы. Автосуммирование. Просмотр формул. Примечания к ячейкам. Зависимости. Форматирование ЭТ: шрифты, выравнивание данных, добавление (удаление) строк и столбцов и изменение их размеров, оформление и заливка. Применение стилей. Ссылки в ЭТ. Абсолютные, относительные и смешанные ссылки. Ссылки на диапазоны, другие рабочие листы и книги. Устранение цикличности. Ввод ссылок. Использование функций. Средства ввода функций: вручную, мастер функций. Примеры функций. Диаграммы. Управление данными: создание и редактирование списка, поиск, сортировка и фильтрация данных, добавление

и удаление итогов. Элементы алгоритмизации и программирования с помощью аппарата встроенных функций и макросов. Формы и методы защиты листов и книг при совместном использовании в локальных и глобальных компьютерных сетях.

*Тема 8. СУБД Access.*

Понятия и типы баз данных. Основные элементы реляционной БД. Системы управления базами данных. СУБД Access. Режимы работы. Окно Access. Таблица БД. Создание схемы БД. Формы и методы защиты информации в БД. Полномочия пользователей БД при работе в локальных и глобальных компьютерных сетях.

Проектирование таблицы. Ввод данных. Редактирование и оформление таблиц. Встраивание и связывание объектов. Отношения между таблицами. Создание и редактирование форм. Управление формами. Проектирование запросов. Типы запросов. Отчеты в БД. Создание отчета.

**ПРОГРАММИРОВАНИЕ НА VISUAL BASIC FOR APPLICATIONS**

*Тема № 9. Программирование в MICROSOFT EXCEL.*

## Тема 1. Введение в информатику

В своих предположениях Нейман основывался на том, что информацию, представленную в числовом виде: 0 и 1, гораздо удобней хранить и обрабатывать, а также такая форма представления информации практически исключает ее потерю. Еще до создания действующего компьютера специалисты в области передачи информации были озадачены преобразованием непрерывных сигналов в цифровую форму. Подобное преобразование стало возможно после опубликования известной в теории информации теоремы Клода Шеннона и разработанного позднее принципа импульсно-кодовой модуляции. Согласно этому принципу: любой непрерывно изменяющийся во времени сигнал может быть представлен последовательностью чисел и, следовательно, передан в цифровой форме. Подобное преобразование называется "кодированием". Под системой счисления понимают совокупность приемов записи и наименования чисел.

На практике применяются четыре системы счисления, которые характеризуются, так называемыми, базовыми числами. Хотя можно использовать любое число в качестве основания системы счисления.

Таблица 1

Базовые числа систем счисления

Основание	Система счисления	Базовые числа
2	Двоичная	0,1
8	Восьмиричная	0,1,2,3,4,5,6,7
10	Десятичная	0,1,2,3,4,5,6,7,8,9
16	Шестнадцатиричная	0,1,2,3,4,5,6,7,8,9,A,B,D,E,F

В общем случае любое десятичное число  $A$  записывается в соответствии с формулой преобразования:

$$A = a_n a_{n-1} a_{n-2} \dots a_1 a_0, a_{-1} a_{-2} \dots = \\ = a_n N^n + a_{n-1} N^{n-1} + \dots + a_2 N^2 + a_1 N^1 + a_0 N^0 + a_{-1} N^{-1} + a_{-2} N^{-2} + \dots ,$$

где  $N$  - основание системы,  $a_i$  - коэффициент в разложении числа по степеням (значение позиции любого числа зависит от расположения запятой в записи числа, поэтому системы счисления называются позиционными).

Представление числа в десятичной системе счисления основано на том, что любое число можно разложить по степеням числа 10.

*Пример.*

$$666,666_{10} = 6 \cdot 10^2 + 6 \cdot 10^1 + 6 \cdot 10^0 + 6 \cdot 10^{-1} + 6 \cdot 10^{-2}$$

В случае кодирования в двоичной системе счисления коэффициенты в разложении представляются одной из двух возможных цифр 0 или 1. Двоичная система счисления - это позиционная, весомо значимая система счисления с основанием

2, где позиция цифры в числе показывает количество единиц соответствующего разряда.

*Пример.*

$$25,75_{10} = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

$$2^{-1} = 0,5$$

$$2^{-2} = 0,25$$

Рассмотрим перевод десятичного числа в двоичную систему счисления.

Информация, обрабатываемая компьютером, запоминается в виде групп двоичных символов 0 и 1. Символы 0 и 1 называют элементами кода или двоичной единицей – бит (от словосочетания - binary digital, аббревиатура - bit), их сочетания образуют кодовые комбинации или кодовые слова. Единицей количества информации принят один двоичный разряд, то есть один бит. Группа из восьми бит называют байтом или машинным словом. Одно машинное слово является базовой логической единицей информации в компьютере: это одна группа обрабатываемых бит, единое выражение или одна команда микропроцессора.

Алгоритм нахождения коэффициентов в разложении целой части состоит в последовательном делении исходного числа на 2 и фиксации остатков деления. Коэффициенты начинаются с младшего разряда и кончат старшим. Поэтому при записи результата в двоичной системе счисления полученные коэффициенты нужно записывать в обратном порядке.

*Пример.*

$$25 \underset{\text{L}}{\underset{2}{\div}}$$

$$24 \quad 12 \mid 2$$

$$1 \quad 12 \quad 6 \mid 2 \quad \text{значение младшего разряда}$$

$$0 \quad 6 \quad 3 \mid 2$$

$$0 \quad 2 \quad 1 \mid 2$$

$$1 \quad 1 \quad 0$$

$$1$$

$$25_{10} = 11001_2$$

Для реализации арифметических действий используются правила для их сложения и умножения, соответственно:

$$0+0=0$$

$$1+0=1$$

$$0*0=0$$

$$1*0=0$$

$$0+1=1$$

$$1+1=10_2$$

$$0*1=0$$

$$1*1=1$$

В восьмеричной системе счисления основанием является 8 и число разлагается по степеням числа 8.

$$A = a_n a_{n-1} a_{n-2} \dots a_1 a_0, a_{-1} a_{-2} \dots =$$

$$= a_n 8^n + a_{n-1} 8^{n-1} + \dots + a_2 8^2 + a_1 8^1 + a_0 8^0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} + \dots$$

Где  $a_i$  принимает любое значение из базисных чисел.

*Пример.*

$$428,25_{10}=654,2_8=6*8^2+5*8^1+4*8^0+2*8^{-1}=6*64+40+4+0,25$$

При переводе чисел из десятичной в восьмеричную систему счисления для перевода целой части воспользуемся делением, а десятичной умножением.

*Пример.*

$$190,6875_{10}=276,54_8$$

$$\begin{array}{r} 190|8 \\ 16 \ 23|8 \\ 30 \ 16 \ 2|8 \\ 6 \ 7 \ 2 \ 0 \end{array}$$

В шестнадцатеричной системе счисления числа с 10 по 15 заменяются буквами латинского алфавита:

$$\begin{array}{ll} 10 - a & 13 - d \\ 11 - b & 14 - e \\ 12 - c & 15 - f \end{array}$$

Число разлагается по степеням числа 16 с указанными коэффициентами.

*Пример.*

$$1505,25_{10}=5E1,4_{16}=5*16^2+14*16^1+1*16^0+4*16^{-1}=5*256+14*16+1+0,25$$
$$4*16^{-1}=4/0,06=0,25$$

При переводе чисел из десятичной в шестнадцатеричную систему счисления для перевода целой части воспользуемся делением, а десятичной умножением.

*Пример.*

$$190_{10}=EB_{16}$$

$$\begin{array}{r} 190|16 \\ 176 \ |11 \\ 14 \end{array}$$

Так как компьютеры обрабатывают информацию, представленную в числовой форме, то при вводе документов, текстов программ происходит процесс автоматической перекодировки символов определенными числами.

## Тема 2. Текстовый редактор MS WORD

Практические занятия начинаются с изучения настроек текстового редактора MS Word, так как удобно созданный интерфейс рабочего пространства положительно влияет на продолжительную работу. Прежде всего необходимо обратить внимание на разновидности, состав и настройку панелей инструментов текстового редактора это необходимо, чтобы пользователь мог самостоятельно удалить или установить «кнопки», реализующие нужные или лишние режимы работы с документом. Затем следует произвести настройку собственных параметров MS Word, которые устанавливаются из пункта меню **Сервис-Параметры**, так как его стандартные значения («по умолчанию») не всегда соответствуют требующимся.

При форматировании готового или при создании нового текстового документа необходимо, чтобы пользователь придерживался определенного алгоритма: на первом этапе устанавливал параметры полей страниц документа, затем производил установку общего шрифтового оформления, далее задавал значения для отступов и выравнивания абзацев документа. На заключительном этапе работы с документом производятся аналогичные действия, но они касаются фрагментов текста и абзацев, имеющих уникальные параметры форматирования.

Элемент типа таблица является одним из наиболее часто используемых при создании документов, причем таблицы могут применяться не только для вывода структурированной информации, но и для производства элементарных вычислений, сложного, оригинального форматирования содержания документа (документ типа бланк, имеющий левую и правую шапки). Обучающиеся должны усвоить, что при грамотном использовании элемента таблица при оформлении документов текстовый редактор MS Word приобретает новые, расширенные возможности.

Использование различных служебных вставок, которыми располагает текстовый редактор, позволяет пользователю значительно сократить трудоемкость по оформлению документа. К ним относятся: использование сносок, нумерации страниц, колонтитулов, режима автозамены, поиска, вставки символов и служебной информации. Особенно следует обратить внимание обучающихся на возможность сборки автоматического оглавления.

Вставка в документ различных видов структурных изображений и диаграмм может производиться с помощью стандартного пакета «Организационная диаграмма» или элементов псевдографики, расположенных на инструментальной панели «Рисование». Практика изучения обучающимися данного режима «вставки» заключается в усвоении некоторых нюансов и особенностей текстового редактора MS Word при создании и внедрении псевдографических изображений.

## Тема 3. Электронные таблицы MS EXCEL

При выполнении заданий необходимо обратить внимание на то, что основное назначение электронных таблиц MS Excel – это производство арифметических вычислений. Соответственно, проблемная ориентация приложения вносит новые элементы управления и интерфейса, изучение которых поможет организации эффективной работы в электронных таблицах.

Особое внимание обучающиеся должны обратить на алгоритм ввода формул для производства вычислений и правильность использования принципов адресации при указании исходного числового материала. Так как при неверном подходе возможности предоставляемые MS Excel по пересчету результатов, копированию формул, подбору параметра и так далее, перестают быть эффективными, возникает ситуация получения неверного результата.

Таблицы составляют основу MS Excel, их грамотное создание и оформление обеспечивает не только удобное восприятие обрабатываемой информации, но и позволяет избежать неудобств в процессе построения диаграмм и графиков.

Использование «Мастера функций» позволяет пользователю избежать усложнения алгоритма вычислений и дополнительного ввода формул при производстве вычислений, так как содержание библиотеки функций MS Excel позволяет использовать уже алгоритмически сформированные стандартные вычисления. Имеющиеся функции включают основные операции из областей математических, логических, статистических, вероятностных вычислений.

## Практическое занятие по теме «Введение в информатику»

**Задание 1. Выполните перевод десятичных чисел:**

$101_{10}$  – в двоичную систему счисления;

$565_{10}$  – в восьмеричную систему счисления;

$918_{10}$  – в шестнадцатеричную систему счисления.

**Задание 2. Выполните перевод в десятичную систему счисления следующих чисел:**

$101001_2$ ,  $174_8$ ,  $3A7_{16}$ .

**Задание 3. Выполните перевод в двоичную систему счисления следующих чисел:**

$111_8$ ,  $345_8$ ,  $195_{16}$ ,  $4C6_{16}$ .

**Задание 4. Выполните перевод в восьмеричную и шестнадцатеричную систему счисления следующих чисел:**

$100101_2$ ,  $1111001_2$ ,  $110010101_2$ .

**Задание 5. Укажите предшествующие и последующие числа:**

\_\_\_  $10111_2$  \_\_\_; \_\_\_  $246_8$  \_\_\_; \_\_\_  $4AC_{16}$  \_\_\_;

\_\_\_  $10101_2$  \_\_\_; \_\_\_  $777_8$  \_\_\_; \_\_\_,  $FBA_{16}$  \_\_\_.

**Задание 6. Произведите попарное сравнение чисел:**

$10101_2 \vee 63_8$ ;  $A_{16} \vee 267_8$ ;  $11101101_2 \vee 255_{16}$ .

**Задание 7. Выполните арифметические действия:**

$10101_2 + 00100_2 =$ ;

$10100_2 + 00110_2 =$ ;

$10100_2 - 01110_2 =$ ;

$38_{16} + 4_{16} =$ ;

$AA_{16} + 1_{16} =$ ;

$78_{16} + C_{16} =$ .

## Практическое занятие по теме «Текстовый редактор MS Word»

Задание 1. Используя приложения Мой компьютер или Проводник, программу Калькулятор, текстовые редакторы Блокнот и MS WordPad, графический редактор MS Paint выполните следующие действия:

1. Создайте в своем рабочем каталоге трехуровневую структуру папок<sup>1</sup>, как показано на рисунке 1.

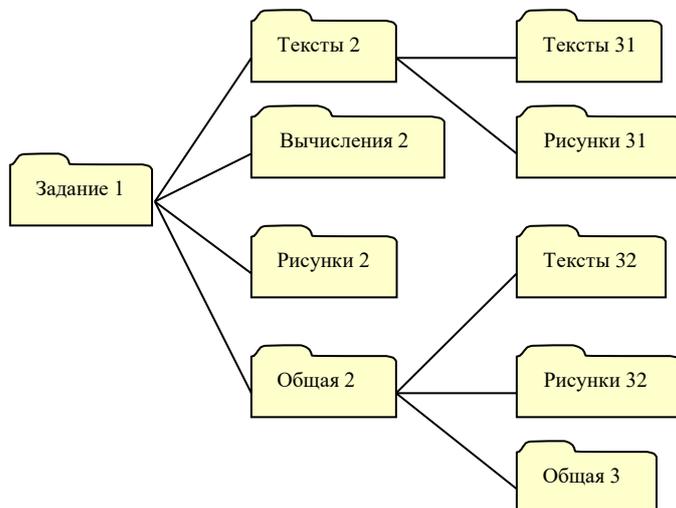


Рисунок 1.

2. Создайте ярлыки на папки третьего уровня и переместите их в папку **Задание 1**.

3. С компьютера преподавателя скопируйте в папку **Общая 2** два файла: **ПРИМЕР\_1.doc** и **ПРИМЕР\_2.doc**.

4. Запустите с помощью главного меню *Пуск-Все программы-Стандартные* приложения **Калькулятор** и **Блокнот**, установите размеры их окон так, чтобы они уместились на **Рабочем столе**.

5. Активизируйте окно программы **Калькулятор**. С помощью пункта меню *Вид* произведите переключение между обычным и инженерным режимами работы **Калькулятора**. Вычислите с его помощью площадь круга радиуса  $R = 7,5$  см ( $S = \pi \cdot R^2$ ). Скопируйте полученный результат в буфер обмена.

6. Активизируйте программу **Блокнот** и введите фразу:

«Площадь круга радиуса 7,5 см = кв. см»,

установите курсор после знака равенства и вставьте из буфера обмена вычисленное ранее число. Затем на следующей строке из пункта меню *Вставка* вставьте текущую дату.

7. Сохраните документ под именем **Запись.txt** и поместите его в папку **Вычисления 2**.

<sup>1</sup> В дальнейшем созданная структура будет использоваться для сохранения выполненных работ.

8. Закройте окна программ **Калькулятор** и **Блокнот**.

9. Запустите с помощью главного меню *Пуск-Все программы-Стандартные* графический редактор **MS Paint**. Установите рабочее поле размером 150x200 точек и нарисуйте автомобиль. Сохраните документ под именем **cars.bmp** в папке **Рисунки 2**. Сверните окно графического редактора **MS Paint** на панель задач.

10. Запустите с помощью главного меню *Пуск-Все программы-Стандартные* текстовый редактор **MS WordPad**. Наберите в нем следующий текст:

«Уважаемый господин Бендер!

Этим письмом я обращаюсь к Вам с просьбой включить мою кандидатуру в список участников конкурса «Самый честный сын лейтенанта Шмидта», объявленного Вами в газете «Золотая гиря» 1 апреля 1922 года, с целью участия в планирующемся автопробеге».

11. Восстановите на экране графический редактор **MS Paint** и используя режим фрагментарного выделения скопируйте рисунок автомобиля в буфер обмена и вставьте его после набранного Вами текста. Затем под вставленным рисунком добавьте лозунг:

«Ударим автопробегом по бездорожью и разгильдяйству!»

Сохраните этот документ в папке **Тексты 2** под именем **Письмо.doc**.

12. Закройте окна программ **MS WordPad** и **MS Paint**.

**Задание 2. Приведите содержимое документа-задания в соответствие с представленным документом-образцом.**

1. Откройте файл **ПРИМЕР\_2.doc**, находящийся в папке **Общая 2**.

2. Требования по оформлению текста:

Поля: верхнее 1,5 см, нижнее 1,5 см, левое 2,25 см, правое 1,5 см.

Отступы: слева, справа, перед, после - 0 см., красная строка – отступ.

Междустрочный интервал – одинарный, выравнивание – по ширине, заголовок – по центру.

Шрифт – Arial, размер шрифта – 12.

3. Удалите ненужные фрагменты текста.

4. Проверьте орфографию.

5. Установите переносы.

6. Используя пункт меню *Правка-Заменить*, замените словосочетание «база данных» на аббревиатуру «БД».

7. Сохраните документ в папке **Тексты 2**.

## Содержание фрагмента документа - задания (файл ПРИМЕР\_2.doc).

### **Раздел 1. Реляционная база данных – основные понятия.**

*Часто, говоря о базе данных, имеют в виду просто некоторое автоматизированное хранилище данных. Такое представление не вполне корректно. Почему  
этого так, будет показано ниже.*

*Действительно, в узком смысле слова, база данных – это некоторый набор данных, необходимых для работы (актуальные данные). Однако **данные - это абстракция; никто никогда не видел "просто данные"; они не возникают и не существуют сами по себе.** Данные суть отражение объектов реального мира. Пусть, например, требуется хранить сведения о деталях, поступивших на склад. Как объект реального мира — деталь - будет отображена в базе данных? Для того, чтобы ответить на этот вопрос, необходимо знать, какие признаки или стороны детали будут актуальны, необходимы для работы.*

*Среди них могут быть название детали, ее вес, размеры, цвет, дата изготовления, материал, из которого она сделана и т.д. В традиционной терминологии объекты реального мира, сведения о которых хранятся в базе данных, называются сущностями - *entities* (пусть это слово не путает читателя - это общепринятый термин), а их актуальные признаки - атрибутами (*attributes*).*

*Каждый признак конкретного объекта есть значение атрибута. Так, деталь "двигатель" имеет значение атрибута "вес", равное "50", что отражает тот факт, что данный двигатель весит 50 килограммов.*

*Было бы ошибкой считать, что в базе данных отражаются только физические объекты. Она способна вобрать в себя сведения об абстракциях, процессах, явлениях - то есть обо всем, с чем сталкивается человек в своей деятельности. Так, например, в базе данных можно хранить информацию о заказах на поставку деталей на склад (хотя он суть не физический объект, а процесс). Атрибутами сущности "заказ" будут название поставляемой детали, количество ...*

**Задание 3. Скопируйте с компьютера преподавателя в папку Общая 2 файл КАК ЗАРЕГИСТРИРОВАТЬ ФИРМУ.doc. Выполните следующие требования по оформлению:**

1. Все абзацы текста должны повторяться по одному разу:

Поля страницы: левое – 1 см, правое – 1,5 см, верхнее – 1,5 см, нижнее – 1 см.

Шрифтовое оформление текста: шрифт – Times New Roman, размер шрифта – 12, начертание – обычное.

Выравнивание: все отступы – 0 см, текст выравнивать - по ширине, межстрочный интервал – полуторный, красная строка – на 2 см.

2. Абзац, содержащий текст «Любая организация, компания, фирма считается созданной только с момента ее государственной регистрации», оформите в виде сноски на текущей странице к заголовку документа.

3. Каждый абзац должен иметь свое начертание: вид шрифта и цвет символов (выбрать самостоятельно).
4. Установите интервал между символами: в первом абзаце – **Обычный**, втором – **Разреженный**, третьем – **Уплотненный**.
5. Заголовок документа оформите стилем «**Заголовок 1**», а заголовок каждого абзаца стилем «**Заголовок 2**».
6. В конце текста соберите оглавление.
7. Вставьте в документ номера страниц – внизу по центру, включая нумерацию первой страницы.
8. Создайте колонтитул. Верхний колонтитул выровняйте по правому краю и введите текст – «Пособие для начинающих деловых людей».
9. Сохраните документ в папке **Тексты 2** под именем **Заявление.doc**.

## Содержание фрагмента документа-задания КАК ЗАРЕГИСТРИРОВАТЬ ФИРМУ.doc

### КАК ЗАРЕГИСТРИРОВАТЬ ФИРМУ

*Любая организация, компания, фирма считается созданной только с момента ее государственной регистрации.*

*Любая организация, компания, фирма считается созданной только с момента ее государственной регистрации.*

*Зная о том, сколько труда в нашем государстве стоит получить любую бумажку, начинающий бизнесмен обычно с ужасом думает о предстоящем хождении по инстанциям. Чтобы облегчить этот этап открытия "своего дела", попробуем разложить процедуру регистрации по полочкам.*

### ДОКУМЕНТЫ

*Для начала будущим компаньонам следует собраться и продумать порядок совместных действий по регистрации своей будущей фирмы. Можно распределить обязанности: одному поручить подготовку документов, другому - открытие счетов в банке, третьему ...*

### **Задание 4. Создайте документ по предложенному заданию-образцу:**

1. Создайте документ типа бланк по предложенному образцу.
2. Документ создается на базе таблицы содержащей три строки и три столбца. Объединяются ячейки в таблице: первой, второй строки второго столбца и первого, второго, третьего столбца третьей строки. После ввода и соответствующего форматирования информации линии обрамления таблицы обесцвечиваются: пункт меню *Формат-Границы и заливка* закладка *Граница* - Вид «Нет».
3. Левая шапка таблицы выравнивается по левому краю, правая – по центру.
4. Рисунки в центральный столбец добавьте из библиотеки рисунков *Clip Gallery*, пункт меню *Вставка-Рисунок-Картинки*.

## Содержание документа задания-образца

Главное управление  
внедрения ЭВМ и каль-  
куляторов  
Адрес: г. Москва.



Тел: 222-333-222

Начальник централизованного  
сектора  
Иванов И.И.

«\_\_\_»\_\_\_\_\_ 2008 г.

### Рисунок 2. Бланк на приобретение средств вычислительной техники

#### Задание 5. Создайте документ по предложенному заданию-образцу:

1. Создайте документ по предложенному заданию-образцу, выполнив следующие требования по оформлению текста:
2. Заголовок таблицы создайте с помощью шаблонов **WordArt**.
3. Рисунки в столбец «Изображение продукции» выберите из библиотеки рисунков **Clip Gallery**, пункт меню *Вставка-Рисунок-Картинки*.
4. Данные в ячейках таблицы выровняйте по вертикали – по центру, по горизонтали - по центру.
5. Рассчитайте общую стоимость товаров на складе, используя вставку формул в таблицы: пункт меню *Таблица-Формула*.
6. Подпись выполните в **MS Paint**, формат рисунка – «перед текстом».
7. Вставьте текущую дату – пункт меню *Вставка-Дата и время*.
8. Отформатируйте документ так, чтобы он уместился на одной странице.

#### Содержание документа задания-образца

*Фирма «Чарт» имеет честь предложить Вам  
разнообразную и высококачественную продукцию.  
Цены и ассортимент указаны в таблице.*

Таблица 2

## ЦЕНЫ И АССОРТИМЕНТ ПРОДУКЦИИ

№	Изображение продукции	Вид продукции	Страна-производитель	Цена (усл. ед.)	На складе (шт.)
1		мебель	Италия	500	4
2		лампочки	Германия	10	2
3		компьютеры	США	1400	1
4		телефонные аппараты	Япония	400	1
6		учебники	Россия	5	17
<i>Общая стоимость товара на складе (усл. ед.)</i>				<i>(формула)</i>	

**Примечание:**

1. Возможно нанесение Вашей символики по договорной цене на товарах позиции 2, 5 и 6.

- Для позиций 2 и 6 используемая краска может быть черная, красная и белая.
- Для позиции 5 используемая краска может быть синяя или белая.

2. Для товаров позиций 1,3 и 4 инструкция дана на двух языках, один из которых русский.

О Вашем решении просим сообщить по тел./факс: 000-00-00.

С уважением менеджер по продажам 

Ипатова Д.А.

вторник, 15 октября 2019 г.

**Задание 6. Создайте документ по предложенному заданию-образцу.**

1. Выведите на экран панель инструментов «Рисование», пункт меню *Вид-Панели инструментов-Рисование*.

2. Заголовок выровняйте по центру.

3. Используя элементы псевдографики панели *Рисование*, создайте предложенную структурную схему.

4. Объедините элементы структурной схемы в единое изображение.

5. Сохраните документ под именем **Структура ЕЭС.bmp** в папке **Рисунки 2**.

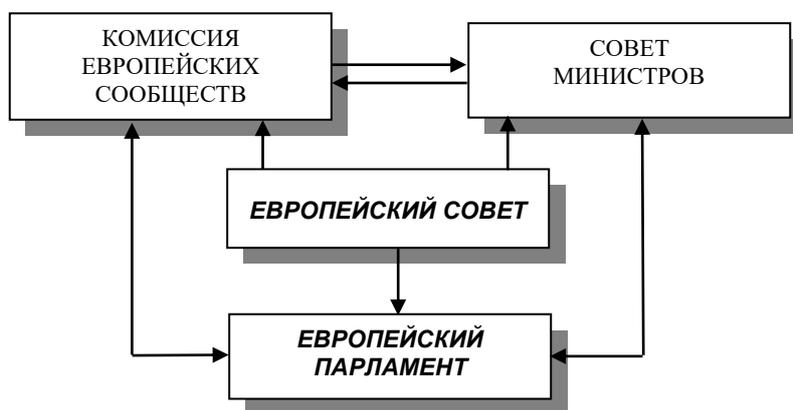


Рисунок 3. Содержание документа задания-образца СТРУКТУРА ЕЭС

## Практическое занятие по теме «Электронные таблицы MS Excel»

### Настройка интерфейса

1. Загрузить на выполнение электронную таблицу **Excel**.
2. Оставить на экране 2 панели инструментов – **стандартную панель инструментов** и **панель инструментов форматирования** (Выполнить пункт меню **Вид/Панели инструментов**, в появившемся списке оставить флажки только на названиях указанных панелей).
3. Вывести **строку формул** (**Вид/Строка формул**).
4. Установить обычный вид таблицы (**Вид/Обычный**).
5. Сохранить файл (который называется **рабочей книгой Excel**) в своей папке под своим именем, например, **Иванов Задание1.xls**. Сохранять изменения в файле после выполнения каждого задания.
6. Определить расположение основных элементов интерфейса:
  - Панелей инструментов
  - Строки формул;
  - Панелей с номерами строк и столбцов;
  - Ярлыков с названиями листов.
7. Расположить панели инструментов в две строки.

### Ввод и форматирование текстовой информации

1. Заменить имя рабочего листа **Лист1** на **Типы данных** (*Щелкнуть правой кнопкой мыши по ярлыку листа, затем из контекстного меню выбрать пункт **Переименовать***). На этом листе будет размещена таблица, в столбцах которой будут приведены примеры основных типов данных, поддерживаемых электронной таблицей.
2. В ячейку **A1** ввести слово **Текст** (*заголовок столбца **A**, поясняющий содержимое столбца*).

Указания по вводу данных:

- ✓ **Имя** (или **адрес**) **ячейки** состоит из номера **столбца** (буквы английского алфавита) и номера **строки** (цифры). Имя выделенной ячейки выводится в левой части строки формул.
- ✓ Для ввода следует поставить выделение на нужную ячейку и ввести с клавиатуры требуемые данные. Ввод можно начинать при отсутствии текстового курсора в ячейке.
- ✓ После набора первого символа в ячейке появляется текстовый курсор, который говорит о том, что таблица перешла в **режим ввода**. В этом ре-

жиме возможности управления электронной таблицей изменяются (часть кнопок на панелях инструментов и пунктов меню становятся недоступными), поэтому после завершения ввода данных необходимо как можно быстрее выйти из этого режима. Все изменения, которые делаются в ячейках таблицы, вступают в силу только после выхода из **режима ввода**.

- ✓ Выход из **режима ввода** выполняется нажатием на клавишу **Enter**.
- ✓ Используйте строку формул для контроля введенных данных, если это трудно сделать по содержимому ячейки.
- ✓ Для исправления ошибок нужно перейти в **режим ввода**, для чего установить выделение на требующую исправления ячейку. При этом в строке формул появится информация в том виде, в котором она была введена в ячейку с клавиатуры. Затем следует щелкнуть мышью по строке формул (появится текстовый курсор) и внести исправления. Вместо щелчка можно нажать клавишу **F2**, тогда текстовый курсор появится прямо в ячейке.
- ✓ Для удаления введенной информации настроить выделение на нужную ячейку и нажать на клавишу **Delete**. Прежняя информация полностью удаляется также, если в ячейку начинается ввод новых данных.

3. В ячейку **A2** ввести текст **Кафедра информатики**, не обращая внимание на выход текста за пределы ячейки.
4. Проверить содержимое ячейки **B2**, на которую накладывается введенный текст (поставить выделение на эту ячейку и посмотреть содержимое строки формул).
5. В ячейку **A3** ввести текст, расположенный в 3 строки (все строки в одной ячейке):

**Медицинская  
информатика  
и статистика**

- ✓ Продолжение ввода текста со следующей строки в той же ячейке происходит после нажатия комбинации клавиш **Alt+Enter**. Завершается ввод, как обычно, нажатием на клавишу **Enter**.
  - ✓ Текст может появиться в изломанном виде, так как в этом режиме не допускается выход текста за пределы ширины столбца. Тогда после завершения ввода следует отрегулировать ширину столбца, в котором находится ячейка. Делается это буксировкой правой границы столбца на панели с обозначениями имен столбцов.
  - ✓ В случае необходимости, используя аналогичный прием, настроить высоту строки.
6. В ячейку **A4** ввести слово **Текст**, расположенное вертикально. Для этого:
    - ✓ Ввести слово как обычно, с горизонтальной ориентацией.
    - ✓ Завершить режим ввода и вернуть выделение на эту ячейку.
    - ✓ Выполнить пункт меню **Формат/Ячейки**, в появившемся диалоговом окне на вкладке **Выравнивание** установить требуемую ориентацию текста.

## ЧИСЛОВЫЕ ДАННЫЕ

1. В ячейку **B1** ввести слово **Числа**.
2. В ячейку **B2** ввести число **25**. Обратить внимание на особенности ввода числовой информации:
  - ✓ **Excel** автоматически распознает формат вводимой информации. Текстовые данные выравниваются по левому краю ячейки, а числовые – по правому. Выравнивание в ячейках легко изменить, но делать этого пока не следует, так как при вводе числовой информации часто возникают ошибки, а по выравниванию легко визуально определить, распознана ли ваша информация как числовая.
  - ✓ Иногда после ввода числовых данных в ячейке появляется запись в формате даты. В этом случае следует принудительно изменить формат ячейки (**Формат/Ячейки**, затем на вкладке **Число** установить числовой формат) и повторить ввод.
3. В ячейку **B3** ввести число **5,125**.
  - ✓ По выравниванию в ячейке убедиться, что данные восприняты как числовые. В противном случае исправить ошибку, которая, скорее всего, связана с символом десятичного разделителя.
  - ✓ Освоить технику увеличения и уменьшения числа десятичных разрядов в представлении числа в ячейке (кнопки с рядами нулей на **панели инструментов форматирования**). Сравнить данные в строке формул и в ячейке таблицы при изменении разрядности.
4. В ячейку **B4** ввести число **6,00**. Добиться, чтобы при выходе из **режима ввода** сохранялись нули после десятичного разделителя.
5. В ячейку **B5** ввести число в процентном формате **25%**. Для этого:
  - ✓ Сначала ввести это число в обычном формате **0,25**.
  - ✓ Завершить **режим ввода**.
  - ✓ Вернуть выделение на эту ячейку.
  - ✓ На панели инструментов форматирования щелкнуть по кнопке **процентного формата** (на кнопке изображен символ процента).
  - ✓ Отмена **процентного формата** выполняется с помощью пункта меню **Формат/Ячейки**, затем на вкладке **Число** установить **общий** формат.

## ССЫЛКИ

1. Объединить ячейки **C1, D1 и E1**. Для этого:
  - ✓ Выделить **блок** из указанных трех ячеек. Выделение начинать при форме курсора в виде креста белого цвета, который следует отбуксировать по выделяемому диапазону.

- ✓ На панели инструментов форматирования щелкнуть по кнопке **Объединение ячеек** (*на кнопке всплывает подсказка **Объединить и поместить в центре***).
- 2. В объединенные ячейки ввести слово **Ссылка**. В столбцах, объединенных общим заголовком, будут приведены примеры использования ссылок и приемы работы с ними.
- 3. В ячейке **C2** ввести ссылку на ячейку **A2**:  
**=A2**

для этого:

- ✓ Первым символом ввести с клавиатуры знак равенства (признак формулы, функции или ссылки).
  - ✓ Щелкнуть мышью по ячейке **A2**. При этом **адрес ячейки** автоматически вставится после знака равенства, по месту текстового курсора. Этот прием полезно использовать во всех случаях, когда в формулу нужно подставлять **ссылку** на ячейку.
  - ✓ **Адрес ячейки** можно вводить и с клавиатуры, но этот способ не застрахован от ошибок ввода.
4. Убедиться в том, что содержимое ячейки **C2** повторяет содержимое ячейки **A2**.
  5. Изучить поведение ссылки при смене содержимого исходной ячейки. Изменить содержимое ячейки **A2**, ввести в нее другой текст, например, **Лечебный факультет**. Посмотреть, как и в какой момент изменилось содержимое ячейки **C2**.
  6. Изучить поведение **ссылки** при перемещении содержимого исходной ячейки. Для этого перенести содержимое ячейки **A2** в ячейку **B6**. Перемещение можно выполнить с помощью стандартных операций (*вырезать – вставить*), но наиболее эффективный метод перемещения - **буксировка**. Для ее выполнения необходимо выполнить следующие действия:
    - ✓ поставить выделение на буксируемую ячейку;
    - ✓ подвести курсор к любой стороне буксируемой ячейки (кроме правого нижнего угла) и добиться изменения формы курсора на стрелку белого цвета;
    - ✓ В этом положении отбуксировать содержимое ячейки в другое место.
  7. После переноса поставить выделение на ячейку **C2** и проверить, как изменилась введенная ранее в эту ячейку ссылка. Запомнить правила изменения адресов ссылок при перемещении ячейки, на которую установлена ссылка.
  8. Отменить перенос и восстановить исходное состояние таблицы (щелчком по **кнопке отмены операции** на стандартной панели инструментов, изображение на кнопке – изогнутая стрелка влево).
  9. Изучить поведение таблицы при копировании ячеек, содержащих ссылки. Скопировать содержимое ячейки **C2** в ячейку **C3**, а затем в ячейку **D2**. Копирование выполнить буксировкой, так же как перемещение (см. пункт 5), но с нажатой клавишей **Ctrl**, которую следует отпустить только после завершения буксировки.

10. Сравнить адреса ссылок в исходной ячейке **C2** и в конечных ячейках **D2** и **C3**. Определить, что происходит с адресами ссылок при копировании ячеек, содержащих ссылки. Установить правила изменения адресов ссылок при копировании.
11. Освоить метод копирования в последовательные ячейки таблицы с использованием метода **автозаполнения**. Содержимое ячейки **C2** скопировать в ячейки столбца до **C4** включительно. Для этого:
- ✓ Установить выделение на ячейку **C2**.
  - ✓ Подвести курсор мыши к правому нижнему углу выделенной ячейки. Добиться, чтобы курсор изменил свою форму на небольшой черный крест.
  - ✓ Отбуксировать ячейку **C2** до ячейки **C4** включительно не нажимая клавиши на клавиатуре. При этом все промежуточные ячейки будут заполнены копиями буксирной ячейки.

По той же схеме скопировать содержимое ячейки **D2** в ячейки столбца до **D5** включительно.

12. Внимательно изучить механизм изменения адресов ссылок в копируемых ячейках, в каких случаях и на какую величину изменяются адреса строк и столбцов. Сравнить адреса ссылок в исходной и скопированных ячейках. Механизм копирования выражений со ссылками придает электронной таблице уникальные свойства, позволяет повысить эффективность ввода формул для вычислений массивов данных.
13. Освоить технику заполнения ячеек последовательными числами. Для этого:
- ✓ В ячейку **E2** ввести число **1**;
  - ✓ В ячейку **E3** ввести формулу

$$=E2+1$$

Эта формула задает значение в ячейке **E3** на **1** большее содержимого ячейки **E2**.

- ✓ Скопировать формулу из ячейки **E3** в ячейки столбца **E** до **E7** включительно, используя технику **автозаполнения**. Объяснить механизм появления возрастающей последовательности чисел с шагом **1**;
- ✓ Изменить содержимое ячейки **E2** на **10**, затем на **-10**. Объяснить результат.

## ФОРМУЛЫ

1. В ячейку **F1** ввести текст **Формулы**.
2. В ячейку **F2** ввести с клавиатуры формулу (запись формул и функций начинается со знака равенства):

$$=2*3$$

После завершения ввода вернуть выделение на эту ячейку и сравнить содержимое ячейки и строки формул.

3. В ячейку **F3** ввести формулу со скобками

$$=(2+2)*(3+3)$$

4. В ячейку **F4** ввести формулу со ссылкой

$$=B2*B3$$

Для уменьшения вероятности ошибок адреса ячеек вводить щелчком мыши. Изменить содержимое ячеек **B2** и **B3** и проконтролировать изменение содержимого ячейки **F4**.

5. В ячейку **F5** ввести формулу возведения в степень

$$=2^16$$

Для ввода знака возведения в степень следует переключить клавиатуру на английский язык и нажать комбинацию клавиш **Shift + 6**

1. Функции

2. В ячейку **G1** ввести текст **Функции**.

3. В ячейку **G2** с помощью **мастера функций** ввести функцию вывода текущих даты и времени, которая после завершения работы мастера будет выглядеть так:

$$=ТДАТА()$$

Для этого:

- ✓ Поставить выделение на ячейку **G2**.
- ✓ Вызвать мастер функций (кнопка с изображением  $f_x$  на стандартной панели инструментов или пункт меню **Вставка/Функция**). Мастер функций работает по шагам. На каждом шаге работы мастера открывается диалоговое окно для последовательного выполнения действий, приводящих к конечному результату.
- ✓ На первом шаге открывается диалоговое окно, в котором можно выбрать категорию и конкретную функцию из данной категории. Выбрать категорию функций **Дата и время**, в этой категории найти приведенную выше функцию.
- ✓ Данная функция не требует аргументов, поэтому на втором шаге еще раз **ОК** без выполнения каких-либо действий и вставка функции завершена.
- ✓ Если результат вставки представляет собой ряд звездочек, то следует увеличить ширину столбца.

Имя функции можно вводить непосредственно с клавиатуры, но этот способ чреват ошибками ввода. Кроме того, запомнить точное написание более сотни поддерживаемых таблицей функций и назначение их аргументов не представляется возможным, подсказки мастера функций значительно облегчают задачу.

4. В ячейку **G3** ввести функцию получения случайного числа

$$=СЛЧИС()$$

Функция находится в категории **Математические**. Результат выполнения функции – случайное число в диапазоне от 0 до 1. Обратите внимание, что при каждом изменении любых ячеек электронной таблицы генерируются новые случайные числа.

5. В ячейку **G4** ввести функцию с аргументом

**=КОРЕНЬ(36)**

*В случае задания функции, требующей аргументов, на втором шаге мастера функций открывается диалоговое окно с одним или несколькими полями, в которое надо ввести аргументы.*

6. В ячейку **G5** ввести функцию с аргументом в виде ссылки

**=EXP(-B2)**

В этом случае в поле аргумента следует ввести адрес ячейки. Разумнее всего это сделать щелчком мыши по нужной ячейке. Если ячейка закрыта диалоговым окном, его можно отодвинуть буксировкой или свернуть в узкую полоску щелчком мыши по кнопке в правом конце поля аргумента.

Изменить содержимое ячейки **B2** и проследить за изменением ячейки **G5**.

7. Скопировать содержимое ячейки **G5** в ячейку **G6**. Проверить содержимое исходной и скопированной ячеек, объяснить результат.

8. В ячейку **B6** (ячейка под столбцом числовых данных) ввести функцию суммирования чисел. Для этого:

- ✓ Поставить выделение на ячейку **B6**
- ✓ Щелкнуть по кнопке вызова **функции суммирования** на стандартной панели инструментов (кнопка с изображением  $\Sigma$ ).
- ✓ По появившемуся плавающему прямоугольнику, который показывает определенный по умолчанию диапазон суммирования, проконтролировать диапазон, который должен охватывать числовые данные столбца **B**.
- ✓ Нажать на клавишу **Enter**.

9. Получить такую же сумму в ячейке **G7**. Для этого после вызова функции буксировкой мыши изменить выбранный по умолчанию диапазон суммирования.

## НАСТРОЙКА ВИДА ЭЛЕКТРОННОЙ ТАБЛИЦЫ

1. Выделить заголовки столбцов жирным шрифтом с выравниванием по центру ячеек. Для этого:

- ✓ Выделить блок ячеек с заголовками столбцов.
- ✓ На панели инструментов форматирования нажать кнопку с изображением буквы **Ж**.
- ✓ Не отменяя выделения на той же панели нажать кнопку выравнивания по центру.

2. Обвести жирными границами заполненные столбцы, строку заголовков, а также подобрать различный цветной фон для столбцов с различными типами данных:

- ✓ Выделить блок ячеек, который необходимо заключить в рамку;
- ✓ Выполнить пункт меню **Формат/Ячейки**,
- ✓ В появившемся диалоговом окне найти вкладку, на которой расположены инструменты управления видом границы, и освоить технику работы с ними;

- ✓ В этом же диалоговом окне найти вкладку, на которой расположены инструменты управления цветом фона, и освоить технику работы с ними;
- ✓ Все операции этого пункта можно выполнить с помощью кнопок на панели инструментов форматирования, освоите эти операции самостоятельно.

## ПОСТРОЕНИЕ ДИАГРАММ

1. Выделить ячейки столбца **Е**, содержащие числовые данные.
2. На стандартной панели инструментов щелкнуть по кнопке вызова **мастера диаграмм** (кнопка с изображением цветных столбиков).
3. На первом шаге мастера диаграмм на вкладке **Стандартные** выбрать тип диаграммы **Гистограмма** и обычный вид гистограммы.
4. На втором шаге перейти на вкладку **Ряд** и убедиться, что в списке рядов стоит надпись **Ряд 1**, а в поле **Значения** – ссылка на диапазон числовых данных выбранных ячеек. На графическом образце диаграммы должен быть ряд столбиков с последовательно увеличивающейся высотой.
5. На следующем шаге ввести заголовок диаграммы **Последовательно возрастающие числа**. На вкладке **Легенда** убрать флажок **Добавить легенду**.
6. На следующем шаге задать вывод диаграммы на текущем листе и завершить построение диаграммы.
7. Изменить число в ячейке **Е2** и наблюдать за изменениями на диаграмме.
8. Сохранить файл (**рабочую книгу**) электронной таблицы в своей папке.
9. Обмен данными между электронной таблицей и редактором текста
10. Загрузить на выполнение редактор текста **Word**.
11. В новом документе установить альбомную ориентацию страницы (**Файл/Параметры страницы**)
12. Ввести заголовок документа: Типы данных электронной таблицы **Excel**.
13. Перейти на созданный лист электронной таблицы и выделить прямоугольный блок ячеек, содержащий все введенные данные.
14. Через буфер обмена перенести блок в документ **Word**, затем туда же перенести диаграмму, снабдив ее комментариями (указать, содержимое каких ячеек отражает диаграмма).
15. Сохранить файл в своей папке.

## ПРАКТИЧЕСКИЕ ПРИЕМЫ ИСПОЛЬЗОВАНИЯ ССЫЛОК

1. Переключиться на **Лист2** и присвоить ему имя **Парабола**.
2. Оформить ячейки таблицы, содержащие текстовую информацию и параметры функциональной зависимости по следующему образцу:

Таблица 3

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
1	Аргумент	Функция		Уравнение параболы	
2				$y=a*(x-b)^2 + c$	
3				Параметры параболы	
4				1	a – крутизна и направление ветвей
5				10	b - положение вершины
6				10	c – высота вершины над осью X
7				1	Шаг аргумента

Пояснения к образцу:

- ✓ Столбец **A** служит для размещения значений аргументов параболы.
- ✓ Столбец **B** служит для размещения вычисленных значений параболы
- ✓ Уравнение параболы представлено в ячейке **D2** в таком виде, чтобы ее параметры имели очевидный смысл (см. комментарий в столбце **E**).
- ✓ Начальные значения параметров, приведенные в столбце **D (1, 10, 10, 1)**, показывают, что парабола будет направлена ветвями вверх ( $a>0$ ), положение вершины будет в точке **10** по оси абсцисс ( $b=10$ ), вся парабола будет расположена выше точки **10** по оси ординат ( $c=10$ ), шаг аргумента в столбце **A** равен **1** (ячейка **D7**).

3. Заполнить столбец аргументов последовательными числами от **1** до **20**, используя технику копирования формул. В качестве шага аргумента использовать содержимое ячейки **D6**. Проверить правильность заполнения:

- ✓ Изменить число в ячейке **A2**. После замены весь столбец чисел должен перестроиться;
- ✓ Изменить число в ячейке **D6**. При этом должен измениться шаг последовательности чисел.

4. Заполнить столбец **B** значениями параболы. Для этого:

- ✓ В ячейке **B2** ввести формулу для вычисления значений параболы в обозначениях, распознаваемых электронной таблицей:

$$=D\$4*(A2-D\$5)^2+D\$6$$

Сравнить эту формулу с приведенным в образце таблицы уравнением параболы. Определить, как представлены в формуле параметры (коэффициенты **a**, **b**, **c**) параболы и аргумент (**x**).

- ✓ Особое внимание обратите на знак доллара, который может присутствовать в адресах ячеек. Он служит для закрепления ссылок и может стоять перед цифрой, обозначающей номер строки, или перед буквой, обозначающей номер столбца, или перед тем и другим. Компонент адреса (номер строки и/или столбца), перед которым стоит доллар, не меняется при копировании.

- ✓ Скопировать формулу на ячейки столбца **B** до **B21** включительно. Проверить правильность вычисления значений параболы. Минимум должен быть при значении аргумента **10**.
- ✓ Поменять параметры параболы и шаг аргумента в ячейках **D4-D7** и проследить за изменениями вычисленных значений. Проверить соответствие параметров приведенному выше назначению.

## ПОСТРОЕНИЕ ГРАФИКА ПАРАБОЛЫ

1. Выделить ячейки столбца **B** с вычисленными значениями параболы вместе с названием столбца (ячейка **B1**).
2. На стандартной панели инструментов щелкнуть по кнопке вызова мастера диаграмм.
3. На первом шаге мастера диаграмм на вкладке **Стандартные** выбрать тип диаграммы **График** и тип графика – **График с маркерами**.
4. На втором шаге перейти на вкладку **Ряд** и убедиться, что в списке рядов стоит надпись **Парабола**, в поле **Имя** – ссылка на заголовок столбца, в поле **Значения** – ссылка на диапазон числовых данных столбца. Если этого нет, то воспользоваться техникой, описанной в следующем пункте, для того чтобы исправить положение (или начать построение сначала).
5. Настроить подписи по оси **X**. Для этого:
  - ✓ Щелкнуть мышкой по полю **Подписи по оси X** и убедиться, что там появился текстовый курсор.
  - ✓ Отодвинуть панель мастера диаграмм так, чтобы она не заслоняла столбец аргументов.
  - ✓ Отбуксировать мышку по числовым ячейкам столбца аргументов (ячейки с **A2** по **A21**). После завершения буксировки убедиться в том, что поле отражает нужный диапазон.
  - ✓ Перейти к следующему шагу (кнопка **Далее**). Ввести заголовок диаграммы **График параболы**. Подписи по осям **X** и **Y** оставить пустыми. На вкладке **Легенда** убрать флажок **Добавить легенду**.
  - ✓ На следующем шаге задать размещение диаграммы на имеющемся листе и щелкнуть по кнопке **Готово**. Создание диаграммы завершено.

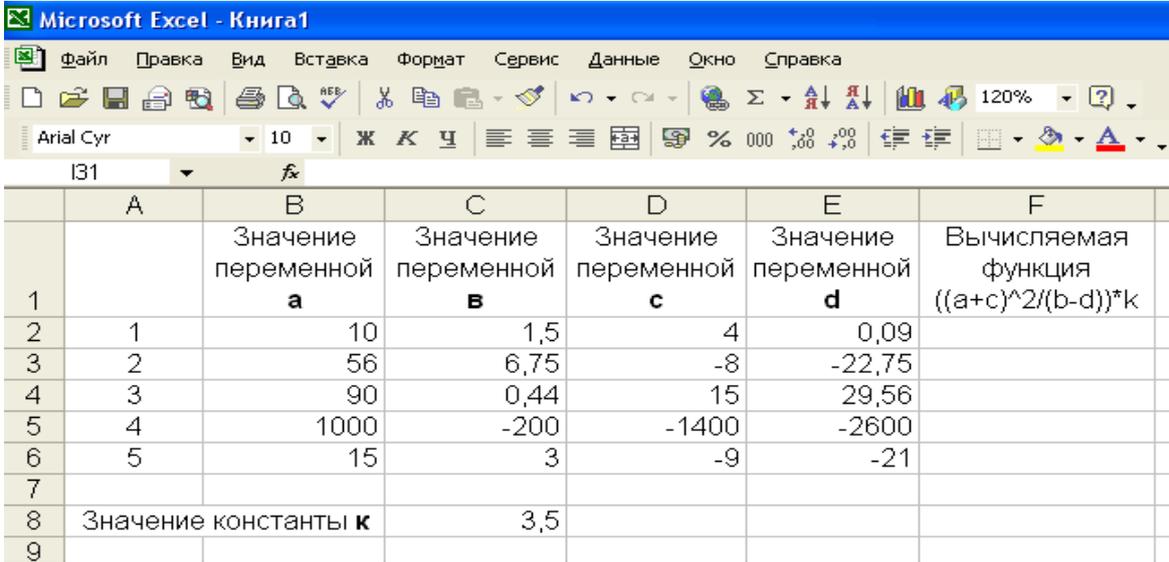
### ***Задание 1. Создайте документ по предложенному образцу:***

1. Заполните ячейки Книги MS Excel текстовой информацией и числовым материалом в соответствии с представленным образцом.
2. Параметры форматирования ячеек: пункт меню **Формат-Ячейки** закладка **Выравнивание** пункт **Перенос по словам**.
3. Введите в ячейку F2 формулу для вычисления значений некоторой функции:

$$f = \frac{(a + c)^2}{(b - d)} k.$$

4. Скопируйте формулу во все строки.
5. Сохраните документ под именем **Расчет.xls** в папке **Вычисления 2**.

### Содержание документа задания-образца



	A	B	C	D	E	F
1		Значение переменной <b>a</b>	Значение переменной <b>b</b>	Значение переменной <b>c</b>	Значение переменной <b>d</b>	Вычисляемая функция $((a+c)^2/(b-d))*k$
2	1	10	1,5	4	0,09	
3	2	56	6,75	-8	-22,75	
4	3	90	0,44	15	29,56	
5	4	1000	-200	-1400	-2600	
6	5	15	3	-9	-21	
7						
8	Значение константы <b>k</b>		3,5			
9						

Рисунок 4.

### Задание 2. Создайте таблицу по предложенному образцу.

1. В графы, которые не выделены цветом, введите исходные данные.
2. Составьте и введите формулы в графы, которые закрашены серым цветом, учитывая следующее:
  - графа «Итого» равна сумме реализации книг издательством за четыре квартала;
  - строка «Всего» равна сумме реализации книг издательствами за каждый квартал;
  - графа «% от выручки» определяет долю выручки, полученной от продажи каждым издательством относительно «Всего».
3. Постройте диаграммы:
  - реализация книг по кварталам;
  - динамика реализации книг любых трех издательств на одном графике;
  - объемную круговую диаграмму по графе «% от выручки»
4. Сохраните документ под именем **Книги.xls** в папке **Вычисления 2**.

**Содержание документа задания-образца**  
**Таблица реализации книг издательствами в 2008 г.**

<i>Наименование издательств</i>	<i>1-й квартал</i>	<i>2-й квартал</i>	<i>3-й квартал</i>	<i>4-й квартал</i>	<i>Итого</i>	<i>% от выручки</i>
«ЭКОМ»	2,3	2,7	2,2	1,9		
«СОФТ»	3,1	2,6	2,9	2,2		
«Финансы и статистика»	3,4	3	2,9	3,5		
«Мир»	2,8	3,1	2,4	3,2		
«АВФ»	1,1	1,9	2	2,1		
<i>Всего</i>						

**Задание 3. Создайте таблицу по предложенному образцу.**

1. Под ячейкой «Процент доплаты 1,5» подразумевается 150%, или коэффициент 1,5.

2. В графы, которые не выделены цветом, введите исходные данные.

3. Составьте и введите формулы в графы, которые закрашены серым цветом, учитывая следующее:

- норматив рабочего времени 160 часов и процент доплаты 1,5 (т.е. 150 %). Эти значения внесите в отдельные ячейки. Обращайтесь к этим ячейкам, используя абсолютную адресацию;
- графа «В т.ч. сверхурочно» вычисляется по формуле с условием: если «Фактически отработанное время Всего» больше «Норматива рабочего времени», тогда («Всего» – «Норматив рабочего времени»), иначе 0;
- графа «Основная зарплата» вычисляется по формуле с условием: если «В т.ч. сверхурочно» больше 0, тогда «Основная зарплата» равна «Норматив рабочего времени» \* «Тарифную ставку», иначе «Фактически отработанное время Всего» \* «Тарифную ставку»;
- графа Начислено «Доплата» вычисляется по формуле с условием: если «В т.ч. сверхурочно» больше 0, тогда «Тарифная ставка» \* «В т.ч. сверхурочно» \* «% доплат», иначе 0;
- графа «Начислено «Всего» вычисляется по формуле: «Основная зарплата» + «Доплата».

4. Рассчитайте среднюю, максимальную, минимальную оплату за месяц с помощью функции СРЗНАЧ, МАКС, МИН соответственно.

5. Постройте диаграммы:

- круговую по графе «Начислено Всего».
- сравнительную диаграмму по графам «Основная зарплата» и «Доплата».

14. Сохраните документ под именем **Зарплата.xls** в папке **Вычисления 2**.

### Содержание документа задания-образца

#### Ведомость начисления заработной платы за фактически отработанное время рабочих-повременщиков цеха № 1 за май 2018 года

№ п/п	ФИО работающего	Табельный номер	Тарифная ставка	Фактически отработанное время		Начислено			
				Всего	В т.ч. сверхурочно	Основная зарплата	Доплата	Всего	
1	Сурков И.П.	132	50,5	160	0	8080	0	8080	
2	Никанорова В.М.	117	45,9	180	20	7344	1377	8721	
3	Этагорова Г.В.	212	60,1	100	0	6010	0	6010	
4	Звальной И.И.	317	66	160	0	10560	0	10560	
5	Кузнецова Р.Д.	222	64,7	162	2	10352	194,1	10546,1	
6	Панфилова Т.И.	127	63,4	100	0	6340	0	6340	
7	Белова Л.М.	32	62,1	200	40	9936	3726	13662	
8	Храбрый И.С.	56	60,8	210	50	9728	4560	14288	
Итого								78207,1	
Норматив рабочего времени				160	Средняя оплата за месяц				9775,89
Процент доплаты				1,5	Максимальная оплата за месяц				14288
					Минимальная оплата за месяц				6010

Рисунок 5.

***Задание 4. Скопируйте с компьютера преподавателя в папку Вычисления 2 файл Кадры.xls и выполните указанные действия***

	A	B	C	D	E	F	G	H	I	J
1	ФИО	Дата рождения	Пол	Возраст	Дата приема на работу	Стаж	Оклад	Семейное положение	Кол-во детей	
2	Антонова И.П.	04.12.1951	ж	56	09.05.1988	20	4300	замужем	1	
3	Артемова Г.И.	06.08.1979	ж	28	05.09.1997	10	4500	незамужем		
4	Афоничева Л.К.	09.01.1952	ж	56	05.10.1985	22	4900	замужем	1	
5	Афоничева М.Е.	08.06.1978	ж	29	22.09.1998	9	3000	замужем		
6	Ахундова О.Г.	01.05.1967	ж	41	23.05.1986	22	4400	незамужем		

Рисунок 6. Фрагмент документа задания, файл Кадры.xls, лист Кадры

	A	B	C	D	E	F
1	<b>Задание КАДРЫ</b>					
2						
3	1.	На листе <i>Кадры</i> размещен список сотрудников.				
4		Определите: количество мужчин,				
5		количество женщин,				
6		средний возраст мужчин,				
7		средний возраст женщин,				
8		средний стаж у мужчин,				
9		средний стаж у женщин.				
10	2.	Добавьте в книгу лист с именем <i>Итоги1</i> и поместите на нем таблицу:				
11						
12				Количество	Средний возраст	Средний стаж
13			Мужчины			
14			Женщины			
15						
16		Заполните эту таблицу.				
17	3.	На листе <i>Итоги1</i> постройте две диаграммы:				
18		круговую диаграмму, отражающую состав организации,				
19		и гистограмму, отражающую распределение возраста и стажа.				
20		Таблица и диаграммы должны поместиться на одной странице формата А4.				
21	4.	Задайте колонтитулы.				
22	5.	Определите количество сотрудников в каждой из трех возрастных групп:				
23		не старше 30 лет,				
24		старше 30 и не старше 45,				

Рисунок 7. Фрагмент документа задания, файл *Кадры.xls*, лист *Задание*

**Задание 5. Моделирование спада температуры тела пациента под действием жаропонижающих препаратов**

1. Создать в своей папке новую рабочую книгу под своим именем, например, **Иванов Задание2.xls**.
2. Присвоить листу **1** имя **Температура**.
3. Оформить заголовки столбцов и ячейки с параметрами модели по приведенному образцу.

Таблица 5

	A	B	C	D	E
1	Дни	Температура		Зависимость температуры (t) от времени (x) в линейной модели	
2				t = -ax+b если t >= 36,6 ; значения t < 36,6 заменяются на t = 36,6	
3				Параметры модели	
4				0,5	a – эффективность лекарственного препарата
5				40	b - начальная температура

В столбце **A** будут приведены дни приема лекарственного препарата, в столбце **B** – значения температуры в соответствующие дни.

4. Заполнить столбец **A** числами от **0** до **20**, используя технику копирования формул.
5. В ячейку **B2** ввести формулу для вычисления температуры по линейной модели. С учетом того, что температура тела не должна опускаться ниже нормальной, формула должна описывать линейный участок спада до температуры **36,6** а затем поддерживать постоянный уровень. В электронной таблице для этого удобно использовать функцию из категории логических **ЕСЛИ(...)**, которая имеет 3 аргумента.

✓ Первый аргумент представляет собой логическое условие, при выполнении или невыполнении которого дальнейшие вычисления происходят по различным формулам. В данном случае таким условием является превышение модельной температурой значения **36,6**:

$$-ax+b > 36,6$$

✓ Второй аргумент – формула, по которой рассчитывается температура при выполнении логического условия, в данном случае

$$t = -ax+b$$

✓ Третий аргумент – формула, которая выполняется при невыполнении логического условия, в данном случае

$$t = 36,6$$

Задать функцию и ее аргументы, обращая внимание на закрепление ссылок на ячейки, содержащие параметры. (*Формулы на место аргументов функции ЕСЛИ(...) вставляются без знака равенства*). Скопировать формулу на весь столбец, проверить правильность полученных значений.

6. Построить диаграмму типа графика, на котором должны быть представлены значения модели. В качестве **подписей по оси X** использовать данные столбца **A**.
7. Изменить параметры модели и проследить за изменениями вычисленных значений по числовым данным столбца **B** и по графику.

**Задание 6. Моделирование процесса нормализации систолического давления под действием лекарственных препаратов**

1. Переключиться на **Лист 2** и присвоить ему имя **Гипертония**.
2. Оформить заголовки столбцов и параметры модели по приведенному образцу.

Таблица 6

	A	B	C	D	E	F
1	Дни	Точная модель	Разброс	Модель с разбросом	Уравнение изменения давления	
2					$D = (D_0 - D_n) \text{EXP}(-kt/M) + D_n + R_N$	
3					Параметры модели	
4					200	$D_0$ – Начальное давление
5					120	$D_n$ – Давление в норме
6					15	$k$ - Эффективность препарата
7					20	$N$ - диапазон разброса случайных чисел
8					100	$M$ – Масштабный коэффициент

В столбцах таблицы будут приведены следующие данные:

- ✓ **A** – дни приема лекарственного препарата (длительность курса лечения – 30 дней);
  - ✓ **B** – значения систолического давления, рассчитанные по точной математической модели, которая показывает его плавное снижение от начальной величины до нормы в течение курса лечения;
  - ✓ **C** – значения статистического разброса, учитывающего тот факт, что на давление оказывают влияние не только лекарственные препараты, но и другие причины, строгий учет которых практически невозможен. К ним относятся, например, стрессовые ситуации, повышенная метеочувствительность, вредные привычки и т.п.
  - ✓ **D** – модельные значения давления с учетом разброса, которые представляют собой сумму значений столбцов **B** и **C**.
3. Заполнить столбец **A** числами в пределах от 0 до 29, при заполнении использовать копирование формул.
  4. В столбце **B** получить значения давления для экспоненциальной модели, показывающей плавное снижение давления до нормального значения без учета случайного разброса. Эта модель описывается формулой

$$D = (D_0 - D_n) \text{EXP}(-kt/M) + D_n$$

где

**D** – текущее значение давления, которым должны быть заполнены ячейки столбца **B**;

**t** – время, прошедшее с начала лечения (приведено в столбце **A**).

*Параметры модели:*

**D<sub>0</sub>** – начальное значение давления пациента до лечения (ячейка **E4**);

**D<sub>n</sub>** – давление в норме (ячейка **E5**);

**k** – эффективность лекарственного препарата (ячейка **E6**);

**M** – масштабный коэффициент, позволяющий работать в удобном масштабе.

Для этого в ячейку **B2** ввести формулу, вычисляющую значения давления, записанную по правилам, принятым в **Excel** и скопировать ее на рабочий диапазон столбца. В таблице параметрам модели присвоены некоторые начальные значения, задающие настройку модели. Впоследствии они будут изменяться, давая тем самым возможность моделирования различных реальных ситуаций.

5. В столбце **C** получить значения случайного разброса в значениях давления. Функция получения случайного числа в **Excel** выглядит следующим образом:

**=СЛЧИС()**

Однако следует иметь в виду, что эта функция возвращает случайное число в диапазоне от 0 до 1, а для построения модели следует получить случайное число в диапазоне от –10 до 10 (для того чтобы максимальный случайный разброс был равен 20 – значению параметра, приведенного в ячейке **E7**), а среднее значение случайного разброса равно нулю. Формула, которая дает случайное число в этом диапазоне, выглядит следующим образом:

$$R_N = N(R_1 - 0,5)$$

где

**R<sub>N</sub>** – случайный разброс в заданном диапазоне;

**N** – значение диапазона (содержимое ячейки **E5**);

**R<sub>1</sub>** – случайный разброс в диапазоне 0 – 1.

В ячейку **C2** ввести вышеприведенную формулу, выраженную по правилам, принятым в **Excel**, и скопировать ее на рабочий диапазон столбца.

6. В столбце **D** получить окончательный результат – модель, описывающую экспоненциальный спад с учетом случайного разброса. Для этого в ячейку **D2** ввести формулу, по которой подсчитывается сумма чисел из ячеек **B2** и **C2** и скопировать ее на весь рабочий диапазон.
7. Построить диаграмму типа графика, рядами данных на диаграмме должны быть числовые значения столбцов **B** и **D**, а подписями по оси **X** – числовые данные столбца **A**.
8. Исследовать поведение модели в зависимости от эффективности лекарственного препарата. Подобрать такое значение эффективности, чтобы давление приблизилось к норме к концу курса лечения.
9. Исследовать поведение модели при различных значениях максимального случайного разброса.
10. С помощью разработанной модели посмотреть динамику изменения давления пациента с **5** по **35** день.

**Задание 7. Моделирование процесса выведения из организма пациента лекарственного препарата**

1. *Постановка задачи.* При внутривенном введении препарат сразу поступает в кровь, его концентрация принимает максимальное значение. В процессе циркуляции по кровеносной системе препарат постепенно выводится из организма по закону, близкому к экспоненциальному. Для характеристики скорости выведения используется величина называемая **периодом полувыведения**, численно равная времени, в течение которого концентрация препарата уменьшается в два раза.
2. С учетом сказанного зависимость концентрации препарата от времени описывается следующей формулой:

$$K = K_0 \cdot \text{EXP}(-t \cdot \ln(2) / T_{1/2})$$

где:

- ✓  $K$  – концентрация препарата;
- ✓  $K_0$  – начальная концентрация;
- ✓  $t$  – время, прошедшее с момента внутривенного введения препарата;
- ✓  $T_{1/2}$  – период полувыведения;
- ✓  $\ln(2)$  – натуральный логарифм 2, численное значение которого равно **0,69**.

*Параметрами модели являются начальная концентрация и период полувыведения.*

3. Используя в качестве образца предыдущие задания, разместить на новом листе, которому присвоить имя **Препарат**, числовую модель процесса. Взять суточный интервал времени, значения концентрации вычислить для каждого часа, прошедшего с момента инъекции. Задать начальное значение периода полувыведения равным 5 часам и произвольную начальную концентрацию.
4. Построить график зависимости концентрации препарата от времени.
5. Исследовать поведение модели в зависимости от значения параметров. Проследить, как ведет себя модель для малых и больших периодов полувыведения.

## Практическое занятие по теме «БАЗЫ ДАННЫХ MS ACCESS»

*БАЗА ДАННЫХ – это набор сведений, относящихся к определенной теме или задаче, такой как отслеживание заказов клиентов или хранение коллекции звукозаписей.*

Работа в базе данных происходит в 3 этапа:

- создание структуры таблиц, т.е. дается название полям (в Access – полем называется столбец, а строка – записью) и определяется тип данных, которые будут вводиться в поля;
- ввод данных в таблицы либо в формы, которые создаются на основе уже созданных таблиц;
- работа с данными (выбор необходимой информации при помощи запросов, создание отчетов, диаграмм и т.п.).

### СОЗДАНИЕ БАЗЫ ДАННЫХ ACCESS

1. Загрузить на выполнение базу данных **Access**.
2. Создать в своей папке новую базу данных и присвоить ей имя **Клиника.mdb**.
3. Познакомиться со структурой **основного окна** базы данных. Отметить следующие элементы:
  - Ярлыки объектов (Таблицы, Запросы, Формы, Отчеты, Макросы и Модули);
  - Кнопки управления объектами (Открыть, Конструктор, Создать).
4. Назначение объектов:
  - **Таблицы** – объект для хранения данных. База данных может содержать несколько таблиц, таблицы могут быть связаны между собой для организации эффективного хранения информации.
  - **Запросы** – объект, с помощью которого осуществляется выборка из таблиц необходимых данных.
  - **Формы** – позволяют реализовывать 2 функции:
    - Служат для представления данных в удобном для просмотра и анализа виде;
    - Являются основой для программного управления базой данных. Позволяют встраивать **элементы управления** (текстовые поля, кнопки, флажки и т.п.) и использовать программную поддержку на языке **Visual Basic**.

- **Отчеты** – объект, позволяющий готовить отобранные из базы данные для вывода на принтер с широким набором возможностей, включая построение диаграмм и графиков.
  - **Макросы** – объект, позволяющий управлять базой данных с помощью последовательности команд. В простых случаях заменяет программирование.
  - **Модули** – программы на языке **Visual Basic for Applications (VBA)**, управляющие работой базы данных.
5. Кнопки управления объектами:
- **Открыть** – открывает объект для работы;
  - **Конструктор** – открывает объект в режиме конструктора, что дает возможность вносить изменения в его структуру;
  - **Создать** – открывает конструктор для создания нового объекта.

## СОЗДАНИЕ ТАБЛИЦ

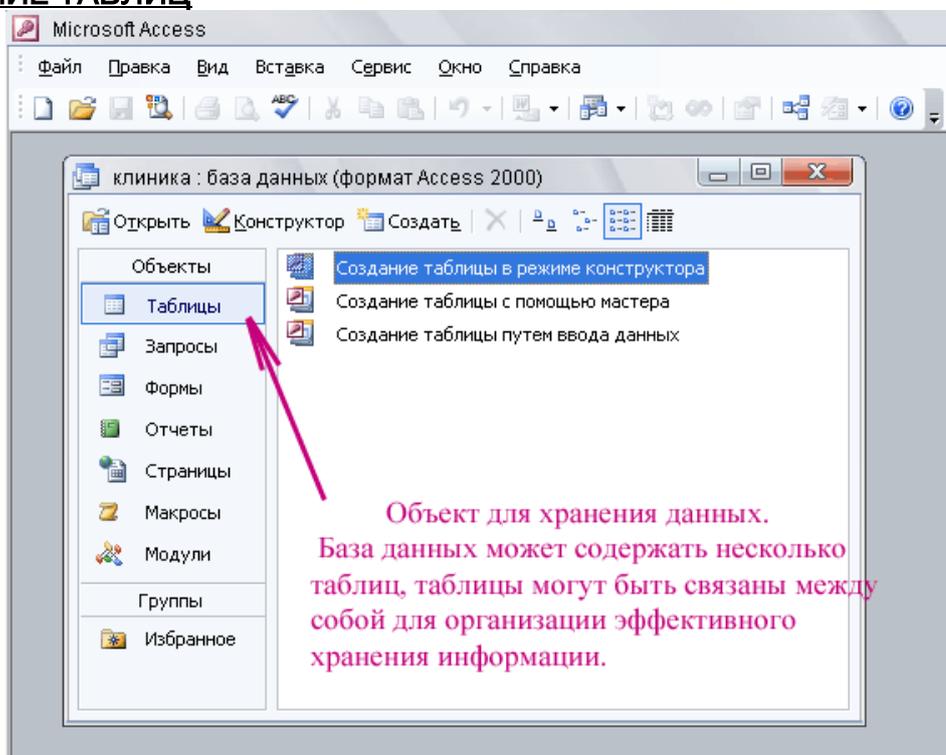


Рисунок 8.

1. Переключиться на вкладку **Таблица**.
2. Двойной щелчок по пиктограмме **Создание таблицы в режиме конструктора**, находящейся на рабочей области Вашей базы данных. При этом откроется **конструктор таблиц**, который служит для создания табличной структуры. Он имеет вид таблицы, состоящей из 3 столбцов:

- **Имя поля** – название столбца таблицы (в названиях полей **нельзя** использовать следующие символы: !, ., **двойные кавычки**, **нельзя в начале названия ставить пробел**);
- **Тип данных**, которые должны заноситься в данное поле;
- **Комментарий** – необязательное поле, в котором может располагаться поясняющий текст к данному столбцу таблицы. При работе с таблицей комментарий выводится в строке состояния.

### 3. Типы данных, поддерживаемых Access:

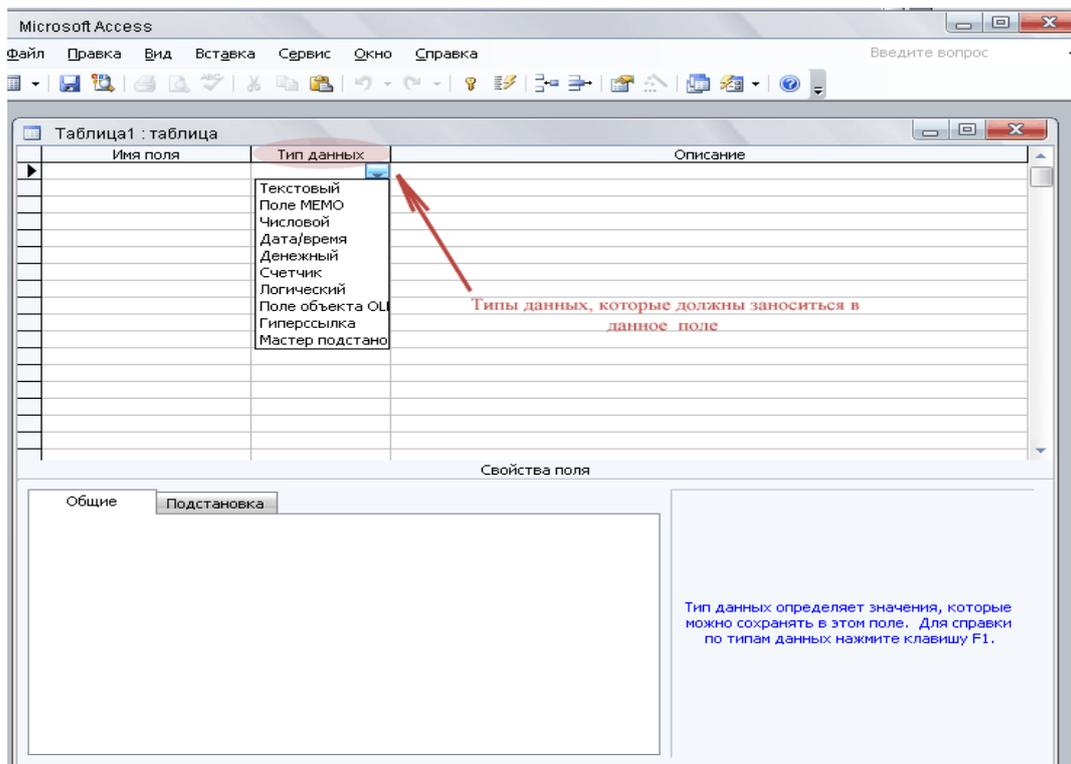


Рисунок 9.

- **Текст** – текстовые данные размером не более 255 символов.
- **Поле MEMO** – текстовые данные большого объема размером до 65535 символов.
- **Числовой** – числовые данные различных форматов, основные из них:
  - **Байт** – целое положительное число в диапазоне от 0 до 255;
  - **Целое** – целое число в диапазоне от -32768 до + 32767;
  - **Длинное целое** – целое число в диапазоне от -2 147 483 648 до +2 147 483 647;
  - **Одинарное с плавающей точкой** – число в пределах от -3.402823E38 до -1.401298E-45 для отрицательных чисел и от 1.401298E-45 до 3.402823E38 для положительных;



4. Создать структуру таблицы со следующими полями (в названиях полей **нельзя** использовать следующие символы: !, ., **двойные кавычки**, **нельзя** в начале названия ставить пробел):

- № пациента
- № карты
- Ф И О
- Место жительства
- Дата рождения
- Пол
- Возраст
- Скидка

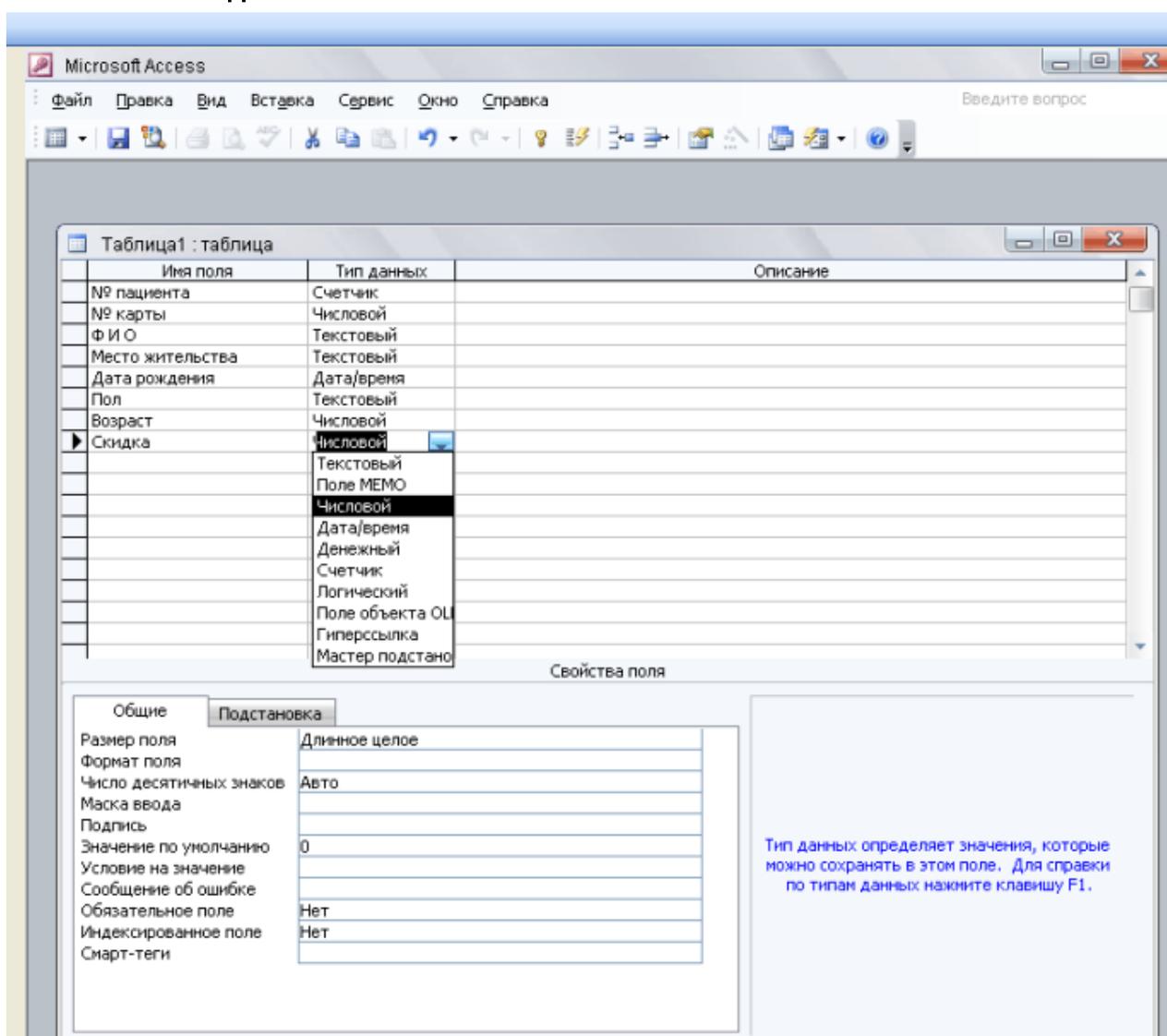


Рисунок 11.

- ✘ Поле **№ пациента** присвоить ему тип данных – **счетчик** и сделать ключевым (щелкнуть по кнопке с изображением ключа на панели инструментов либо щелкнуть на это поле правой кнопкой мыши и в контекстном меню выбрать

соответствующую опцию). (*С помощью ключевых полей создаются межтабличные связи*).

- ✘ Дополнительные настройки числового и других форматов производятся в разделе **Свойства поля** в нижней части конструктора таблиц. Типы данных полей задать по смыслу содержащейся в них информации. Здесь и в последующих таблицах для полей типа **Дата/Время** установить краткий формат даты. Для этого в разделе **Свойства поля** щелкнуть по строке **Формат поля** и выбрать из выпадающего списка нужный формат.

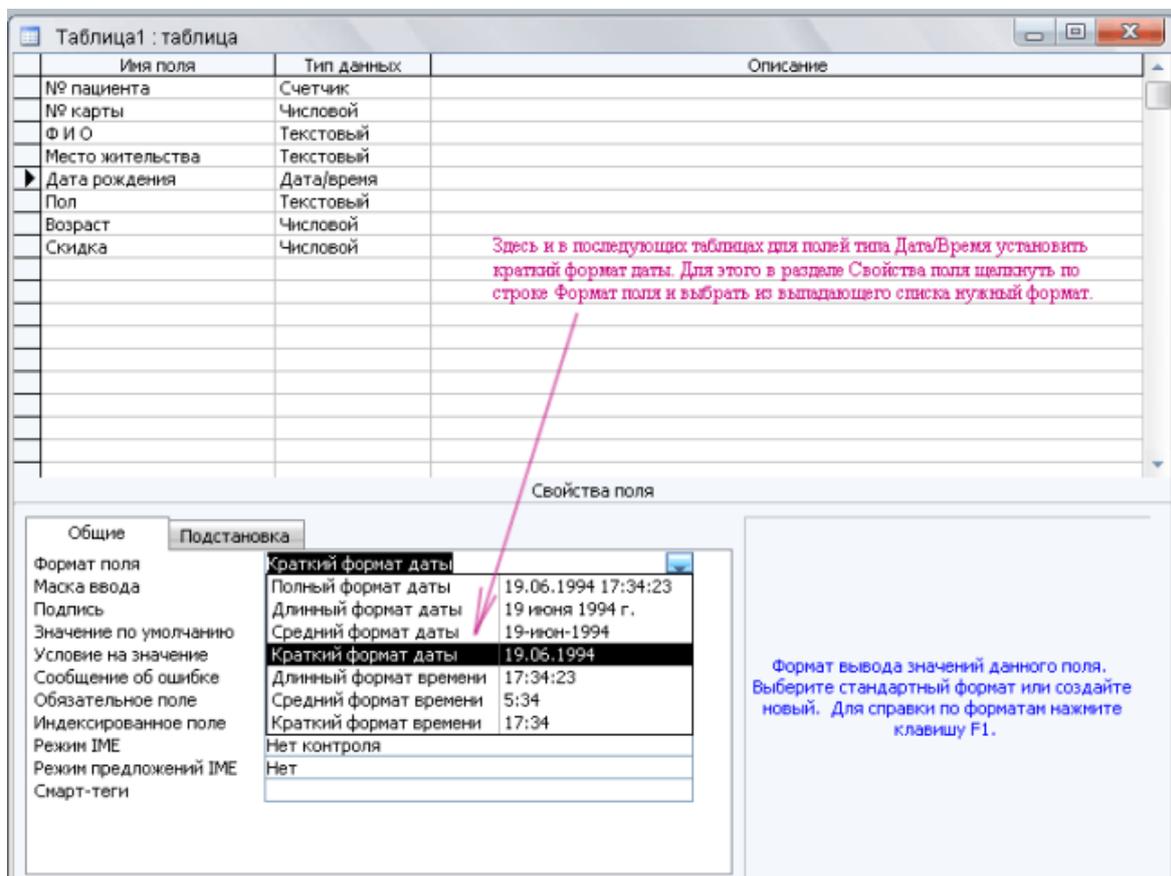


Рисунок 12.

- ✘ Тип данных поля **№ карты** сделать числовым и в разделе **Свойства поля** в нижней части конструктора таблицы установить размер **Целое**, а также установить **Индексирование (да – совпадения не допускаются)**. Совпадения не допускаются, т. к. номер карты каждого пациента индивидуален, исключителен и не может повторяться в списке пациентов. **Индексирование – вторичный ключ**. В основном требуется индексировать поля, в которых часто осуществляется поиск, поля сортировки или поля, **объединенные**. Объединенные поля устанавливают связь между полем одной таблицы или запроса и полем другой таблицы или запроса, имеющим тот же тип данных и содержащие одни и те же данные на пациентов. Объединение определяет взаимосвязь поля одной таблицы с полями из других таблиц.

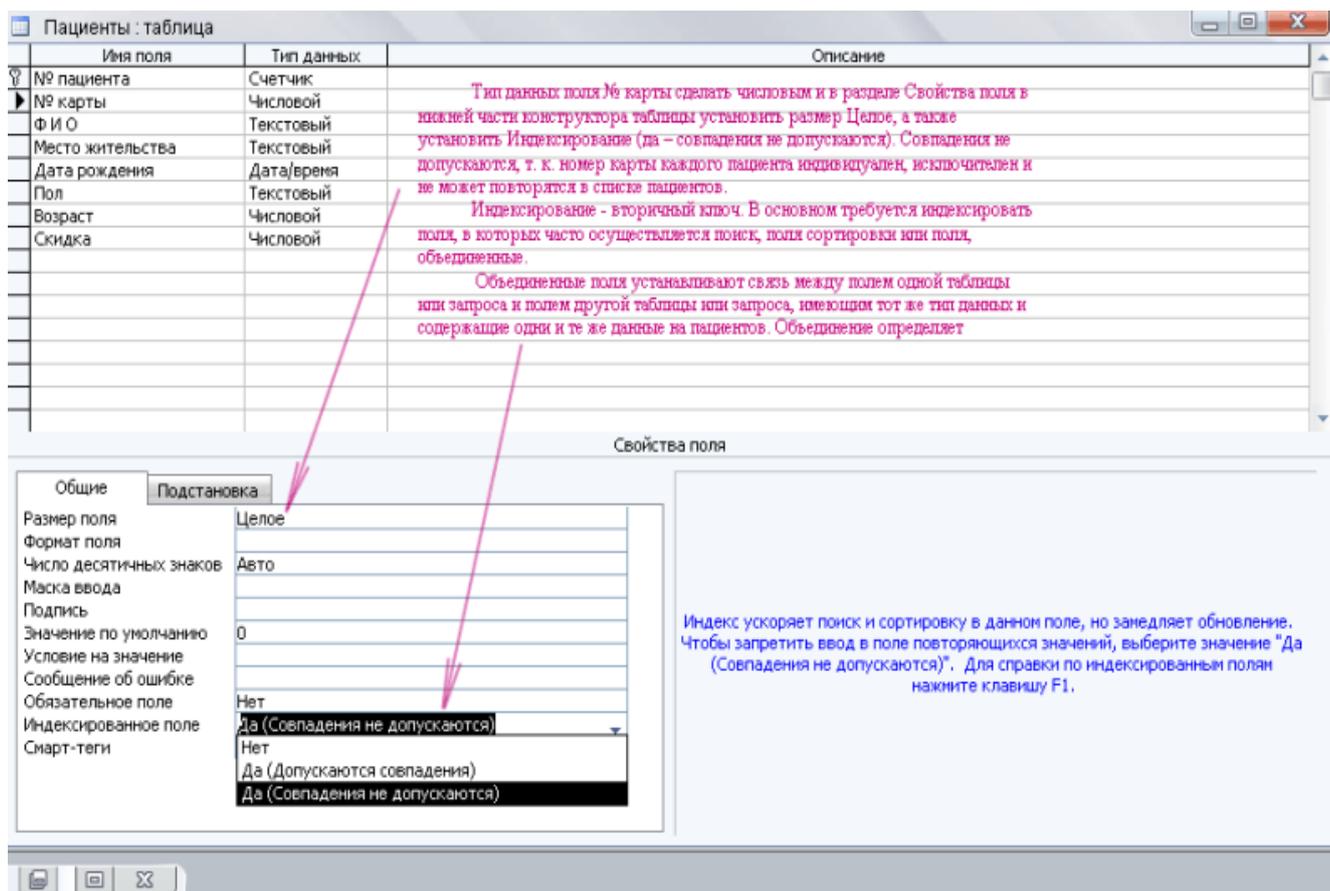


Рисунок 13.

- ✘ Тип данных поля **Место жительства** сделать текстовым и в разделе **Свойства поля** в нижней части конструктора таблицы ввести **в строке значение по умолчанию** название какого-либо города. (Введенное название города будет автоматически добавляться в ячейку поля при вводе новой записи, но при желании это название можно заменить на другое непосредственно в таблице).

5. Присвоить таблице имя **Пациенты**. Для этого закрыть конструктор таблицы (**Файл/Заккрыть**), после чего появится диалоговое окно с запросом на имя таблицы.

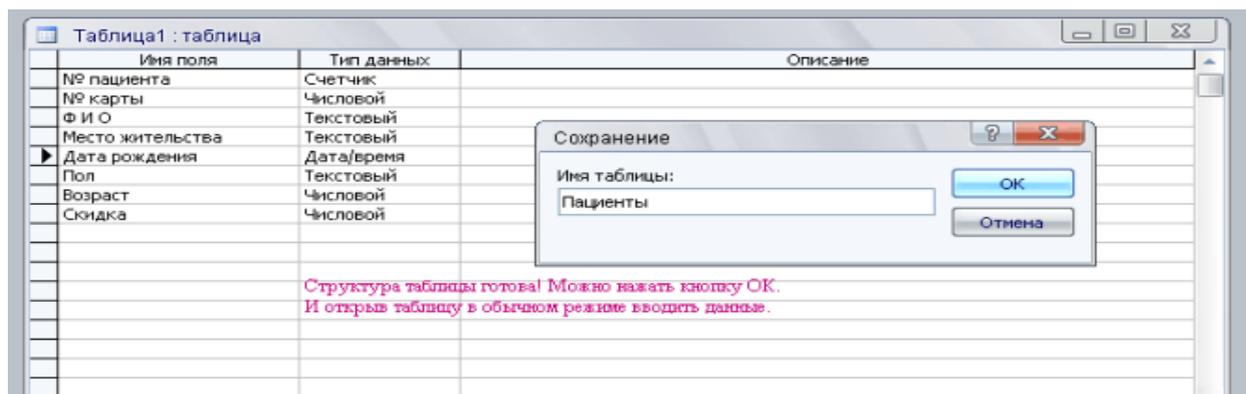


Рисунок 14.

6. Самостоятельно создать еще одну таблицу с именем **Госпитализация** со следующими полями:

- № госпитализации
- № карты
- Дата поступления
- Врач
- Диагноз
- Лекарство
- Дата выписки
- Оплата

- № госпитализации сделать ключевым полем, тип данных – счетчик
- Обе таблицы будут связаны друг с другом с помощью одноименных полей № карты. Поэтому тип данных поля № карты таблицы Госпитализация сделать числовым и в разделе **Свойства поля** в нижней части конструктора таблицы установить **целый размер**, а также **индексирование (да – совпадения допускаются)**

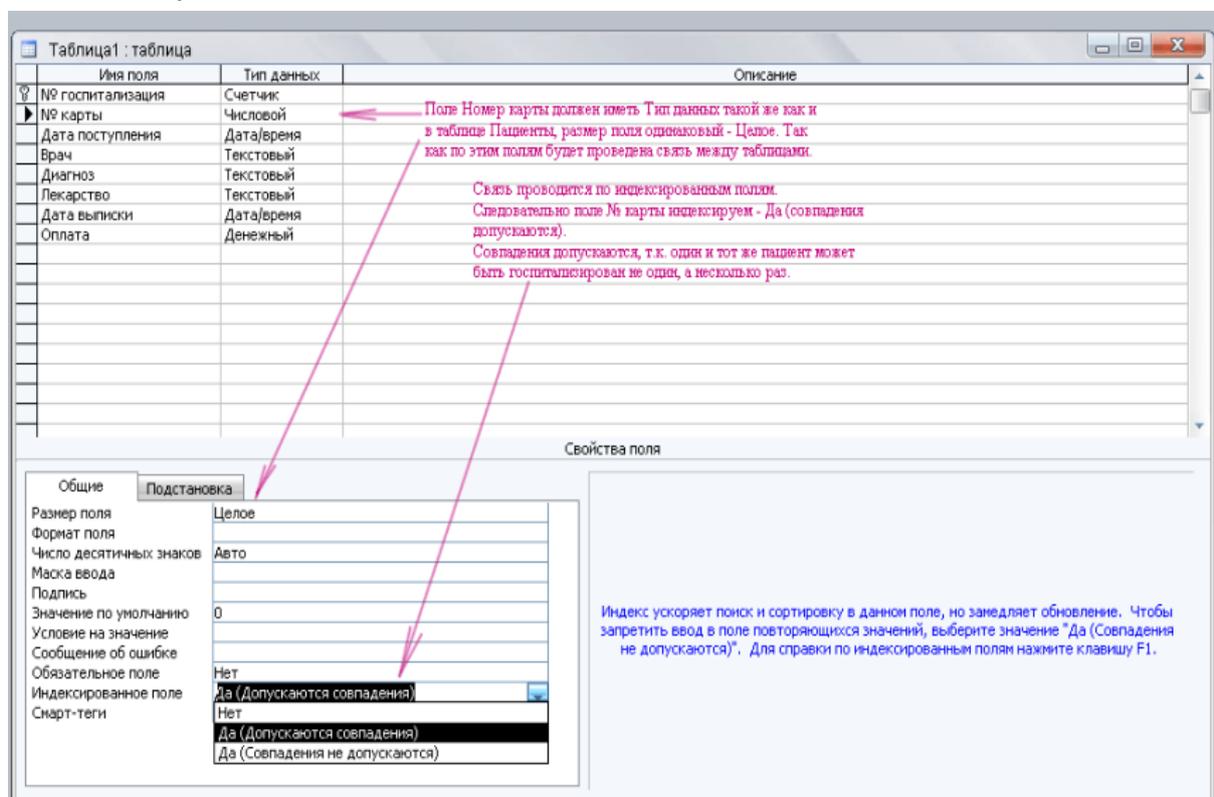


Рисунок 15.

- Для полей типа **Дата/Время** установить краткий формат даты.
- Тип данных поля **Лекарство** сделать текстовым и в разделе **Свойства поля** в нижней части конструктора таблицы установить **индексирование (да – сов-**

падения допускаются.), т. к. разным пациентам может быть установлено лечение одним и тем же лекарственным препаратом.

- Тип данных поля **Оплата** сделать денежным.

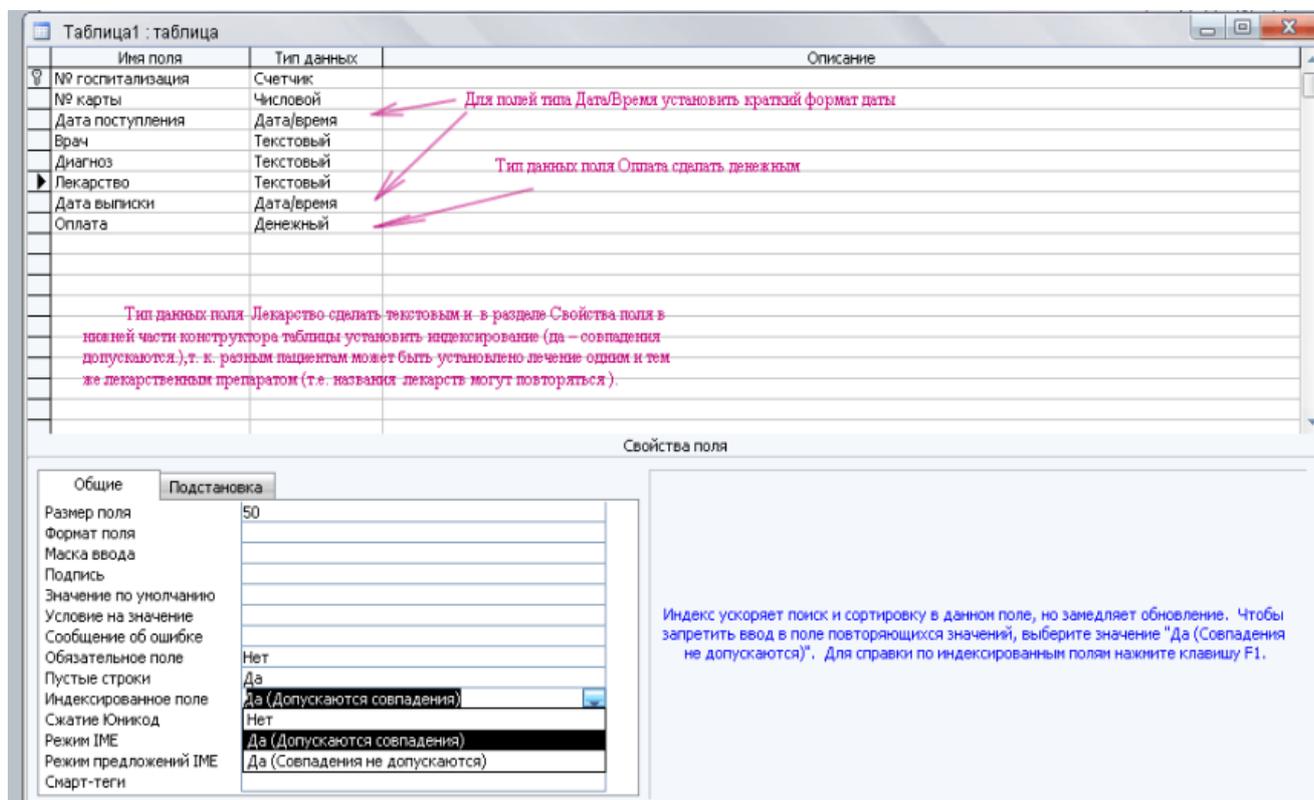


Рисунок 16.

7. Самостоятельно создать еще третью таблицу с именем **Препараты** со следующими полями:

- **№ препарата**
- **Лекарство** (не забыть про индексирование, т.к. по этому полю будет проводиться связь. И т.к. названия лекарств не повторяются, то и **совпадения не допускаются**).
- **Стоимость**

Внимательно и аккуратно определить тип данных и свойства для полей таблицы.

## СОЗДАНИЕ МЕЖТАБЛИЧНЫХ СВЯЗЕЙ

*Связь между таблицами необходима для того, чтобы иметь доступ ко всей информации, введенной в разные таблицы. В большинстве случаев связывают поле одной таблицы с соответствующим ему полем (часто имеющим одно и то же имя) второй таблицы. (Например, поле **№ карты** в таблице **Пациенты** и **№ карты** в таблице **Госпитализация**).*

1. Закрывать все таблицы. Если по неосторожности оказалась закрытой база данных, то открыть пункт меню **Файл** и в нижнем разделе меню в списке недавно открывавшихся файлов выбрать пункт с именем вашей базы.
2. Вывести на экран окно **Схемы данных** (**Сервис/Схема данных** или кнопка на панели инструментов, на которой изображены 3 соединенные между собой прямоугольника). При этом на экране появится окно **схемы данных**, на которое наложено окно добавления таблицы.
3. Добавить три таблицы и закрыть окно добавления. (Таблицы добавляются либо двойным щелчком по названию таблицы, либо после выделения названия щелчком по кнопке «**добавить**»).
4. Выделить поле **№ карты** из таблицы **Пациенты** и буксировкой наложить его на поле **№ карты** в таблице **Госпитализация**. В результате должно появиться диалоговое окно настройки межтабличной связи.

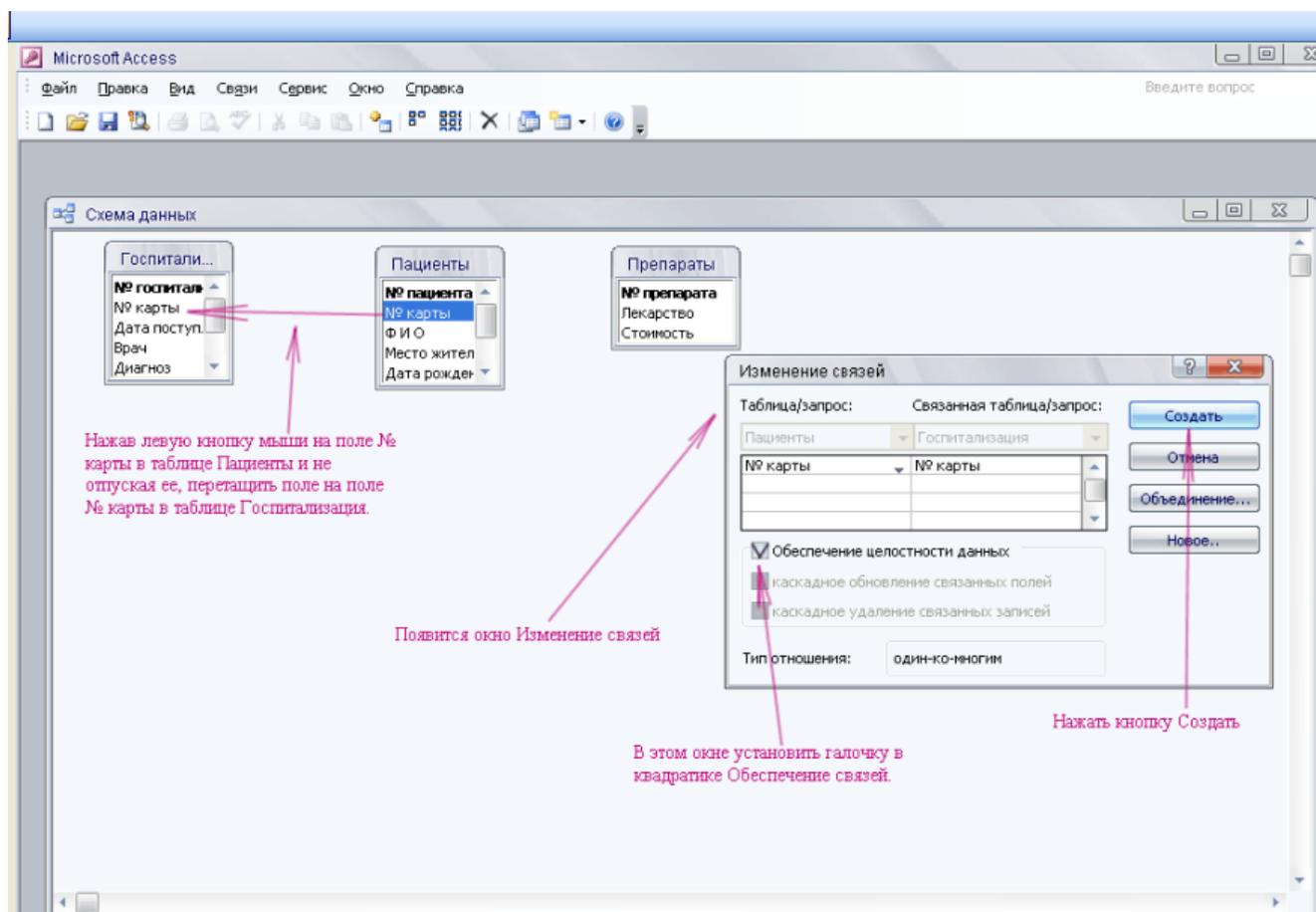


Рисунок 17.

5. Поставить флажок **Сохранение целостности данных**, убедиться в том, что создается связь типа **Один-ко-многим**. Щелкнуть по кнопке **ОК**.
6. После закрытия окна настройки на схеме данных должна появиться связь в виде линии, соединяющей задействованные поля двух таблиц. На стороне один должна стоять 1, на стороне многие должен стоять знак бесконечности.

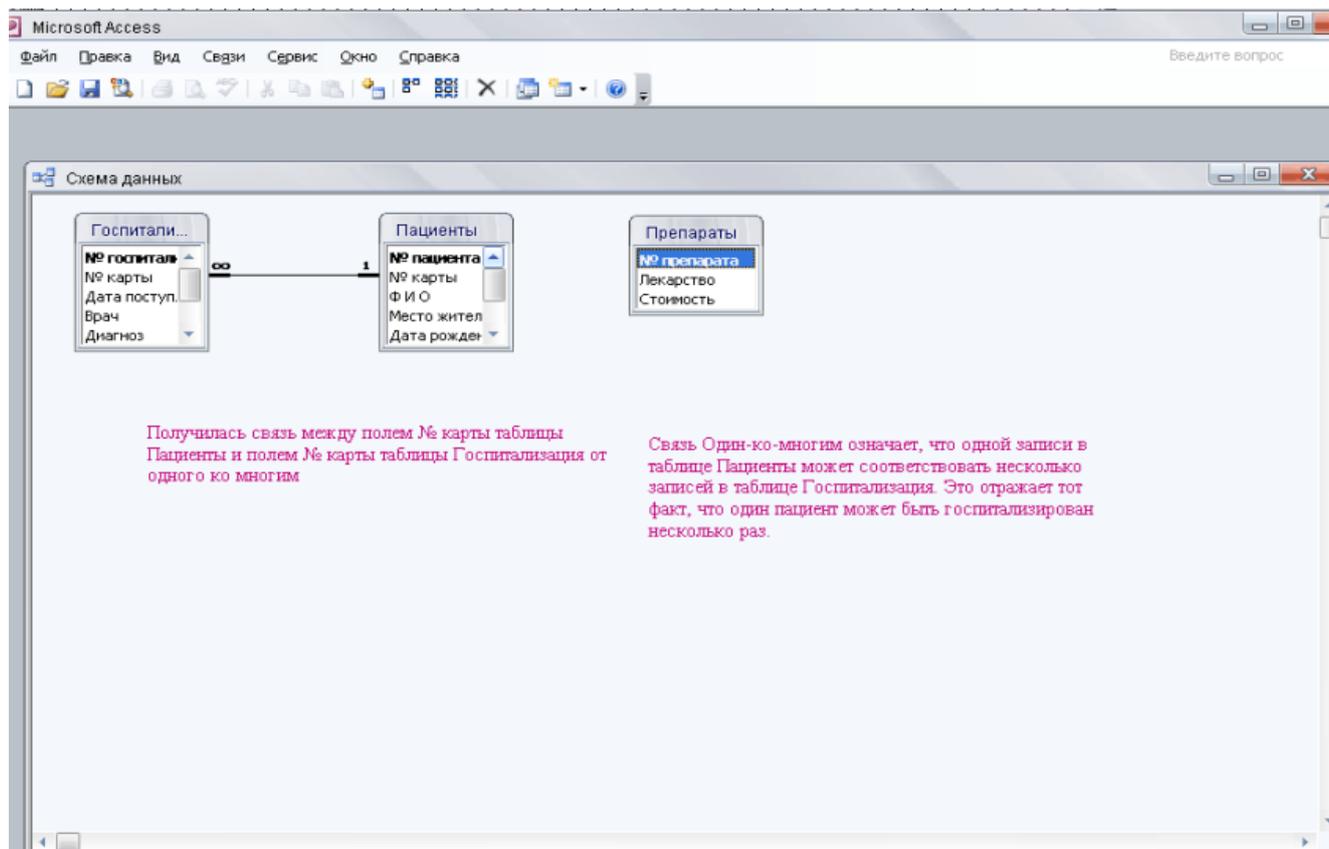


Рисунок 18.

7. Аналогично создать связь между таблицами Госпитализация и Препараты по полю Лекарство.

Связь **Один-ко-многим** означает, что одной записи в таблице Пациенты может соответствовать несколько записей в таблице Госпитализация. Это отражает тот факт, что один пациент может быть госпитализирован несколько раз.

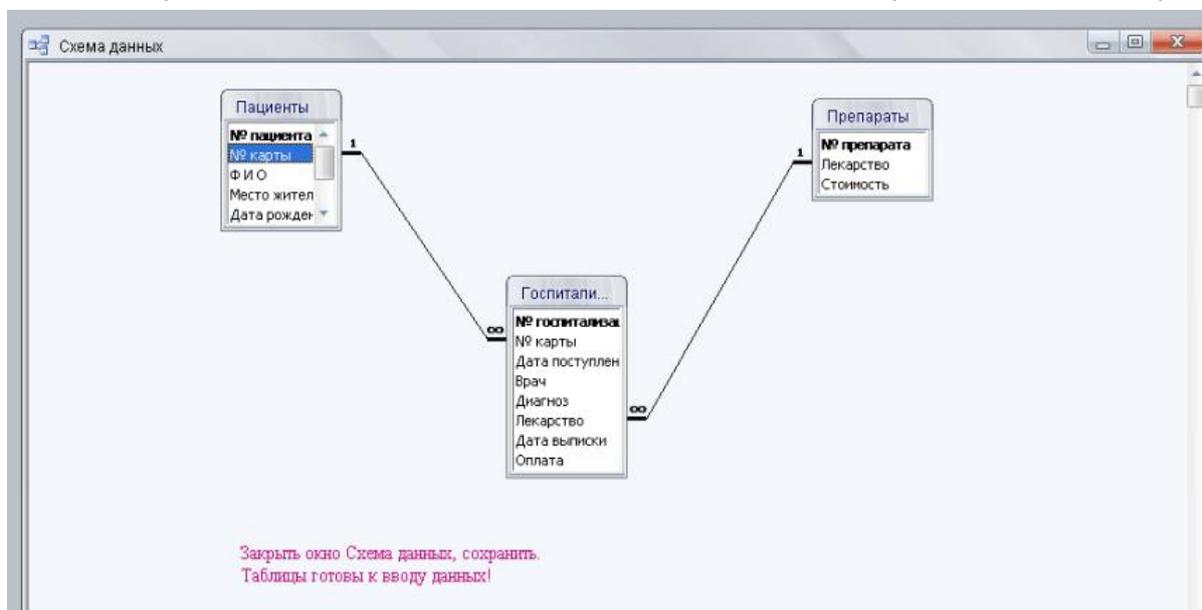
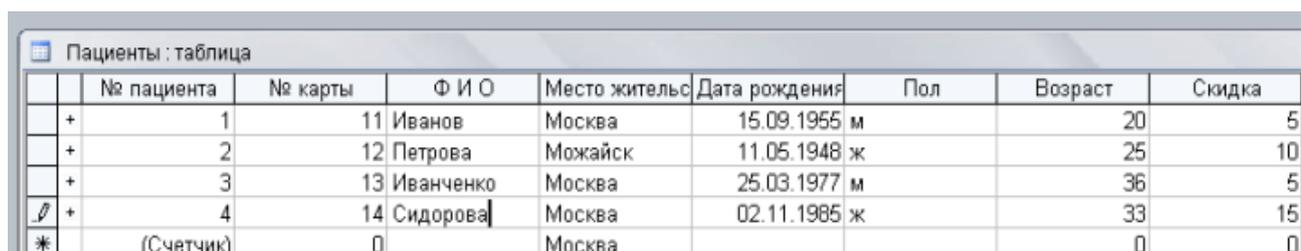


Рисунок 19.

## ЗАПОЛНЕНИЕ ТАБЛИЦЫ ПАЦИЕНТЫ

Открыть таблицу **Пациенты** и заполнить 7 записей (строк таблицы) произвольными, но разумными данными, соблюдая следующие условия:

- Числа в поле **№ карты** не должны повторяться и для экономии времени вводим только двузначные числа.
- Фамилии двух пациентов должны начинаться на одну и ту же букву, быть максимально короткими и без инициалов (можно использовать имена).
- В качестве места жительства указать 2 различных города (т.е. города в некоторых записях должны повторяться).
- Заполнить поле дата рождения, учитывая, что 4 пациента (два мужчины и две женщины) должны иметь пенсионный возраст. (год рождения для пенсионера мужчины д. б. <1953 г., а для женщины – <1956 г.)
- Поле **пол** заполнить буквами **м** и **ж** (обратить внимание на то, что для Access **м**, **м.**, **М**, **М.**, **муж** и т.д. будет считаться разным полом, а также необходимо разное количество полов, например, 3 женщины и 4 мужчины).
- Поле **возраст** заполнить числами в пределах от 20 до 40 лет, не просчитывая от даты рождения, т.к. впоследствии при помощи запроса на обновление это поле заполнится реальным возрастом.
- Поле **скидка** заполнить числами в пределах от 5 до 20 (не надо устанавливать процентный формат, т.к. при расчете оплаты скидка поделится на 100).



	№ пациента	№ карты	Ф И О	Место жительства	Дата рождения	Пол	Возраст	Скидка
+	1	11	Иванов	Москва	15.09.1955	м	20	5
+	2	12	Петрова	Можайск	11.05.1948	ж	25	10
+	3	13	Иванченко	Москва	25.03.1977	м	36	5
+	4	14	Сидорова	Москва	02.11.1985	ж	33	15
*	(Счетчик)	0		Москва			0	0

Рисунок 20.

Выделение столбцов и строк, изменение ширины столбцов и высоты строк, а также форматирование шрифта производится по аналогии работе в текстовом редакторе WORD и табличном редакторе EXCEL.

Удалить столбцы можно только через удаление в структуре таблицы, а вот строки и данные можно удалять в самой таблице.

## ЗАПОЛНЕНИЕ ТАБЛИЦ ГОСПИТАЛИЗАЦИЯ И ПРЕПАРАТЫ

1. Открыть таблицу **Препараты** и заполнить всего 3 записи в таблице. Названия лекарств **не должны** повторяться.
2. Открыть таблицу **Госпитализация** и заполнить 11 записей в таблице, соблюдая следующие условия:
  - Данные в таблице должны отражать тот факт, что один из пациентов, зафиксированных в таблице **Пациенты**, был госпитализирован 1 раз, второй и третий пациенты по 2 раза, а четвертый и пятый пациенты по 3 раза. Т.е. номера карт должны быть взяты из таблицы **Пациенты** и номера карт у четверых пациентов должны еще и повторяться. Остальные пациенты не были госпитализированы и их данные не нашли отражения в таблице **Госпитализация**.
  - При заполнении полей **Дата поступления** и **Дата выписки** обратить внимание на то, чтобы пациенты, госпитализированные по 2 - 3 раза не находились бы в больнице в один этот же промежуток времени, и этот промежуток не должен быть больше 2-х недель. Желательно всех пациентов госпитализировать в один этот же месяц и год. (Например, с 01.05.2010 по 31.05.2010 г.)
  - В таблице должны встречаться 3 различных диагноза, 3 различных названия лекарства и 2 различные фамилии врача. (т.е. диагнозы, лекарства и фамилии врачей будут повторяться и совсем необязательно, чтобы один врач лечил одного и того же пациента и ставил ему и в первом и во втором случае один и тот же диагноз).
  - Поле **оплата** не заполнять.



№	№ карты	Дата поступле	Врач	Диагноз	Лекарство	Дата выписки	Оплата
1	11	15.05.2010	Пипеткин	ангина	Анальгин	28.05.2010	0,00р.
2	12	02.05.2010	Таблеткин	орз	Аспирин	11.05.2010	0,00р.
3	12	18.05.2010	Пипеткин	ангина	Анальгин	29.05.2010	0,00р.
4	14	05.05.2010	Таблеткин	ангина	Анальгин	16.05.2010	0,00р.
5	14	17.05.2010	Пипеткин	орз	Аспирин	30.05.2010	0,00р.
6	15	01.05.2010	Таблеткин	орз	Аспирин	12.05.2010	0,00р.
7	15	13.05.2010	Таблеткин	ангина	Анальгин	18.05.2010	0,00р.
8	15	19.05.2010	Пипеткин	скарлатина	Панадол	30.05.2010	0,00р.
* (Счетчик)	0						0,00р.

Рисунок 21.

## СОЗДАНИЕ ФОРМ

*С помощью форм можно представлять данные, содержащиеся в таблицах и запросах, в удобном для просмотра и анализа виде, а также как продолжить, так и начать ввод данных. Кроме того, в формы можно встраивать элементы управления (кнопки, флажки, текстовые поля и др.), работой которых управляют создаваемые пользователем программы или макросы.*

1. Перейти на вкладку **Формы** и щелкнуть по кнопке **Создать**.
2. Во вновь открывшемся окне **Новая форма** выбрать в качестве источника данных таблицу **Пациенты**, а в качестве инструмента создания – **мастер форм**. Щелкнуть по кнопке **ОК**. После этого начинает работать **Мастер построения форм**.
3. На первом шаге мастера следует перенести все поля из списка **Доступные поля** в список **Выбранные поля**, нажав кнопку с двойной галочкой и щелкнуть по кнопке **Далее**.
4. На втором шаге выбрать вид формы **в один столбец** (этот вид дает возможность самостоятельно выбрать дизайн формы и представить каждую запись в виде карточки, все остальные виды форм похожи на табличную).
5. На третьем шаге выбрать стиль формы.
6. На четвертом шаге задать имя формы (**Пациенты**) и щелкнуть по кнопке **Готово**.
7. В открывшейся форме каждая запись (данные на пациента) представлена в виде карточки. На нижней строке окна, строке состояния, в ячейке –счетчике записей отражен номер записи (номер пациента) на которой открыта форма – активная запись, можно вносить изменения. На этой же строке есть кнопки с пиктограммой в виде треугольника, позволяющие перелистывать записи. Кнопка с треугольником, острым углом направленным налево – переход к предыдущей записи, острым углом направленным налево и упирающимся в черту – переход к первой записи. Кнопки, острый угол которых направлен направо – это переход к следующей записи и к последней записи. И последняя кнопка **Новая запись** с треугольником и звездочкой – переход к новой записи. После кнопки **Новая запись** отражена цифра, указывающая, сколько всего записей введено в вашу форму. Ввести 3 новые записи (не забыть для пациентов задать различный пол, возраст от 20 до 40 лет и т.п.)
8. Проверить как отобразились записи в таблице.

## НАСТРОЙКА СВОЙСТВ ФОРМЫ

1. Открыть созданную форму в режиме конструктора (Вид/Конструктор).
2. Структура формы состоит из 3 областей: область заголовка формы, область данных и примечание формы. Подвести курсор на границу между **заголовком формы** и **областью данных** (курсор изменит начертание на горизонтальную линию с вертикальными стрелками), нажать правую кнопку мыши и, не отпуская потянуть вниз приблизительно на 2-3 см, затем отпустить кнопку мыши.

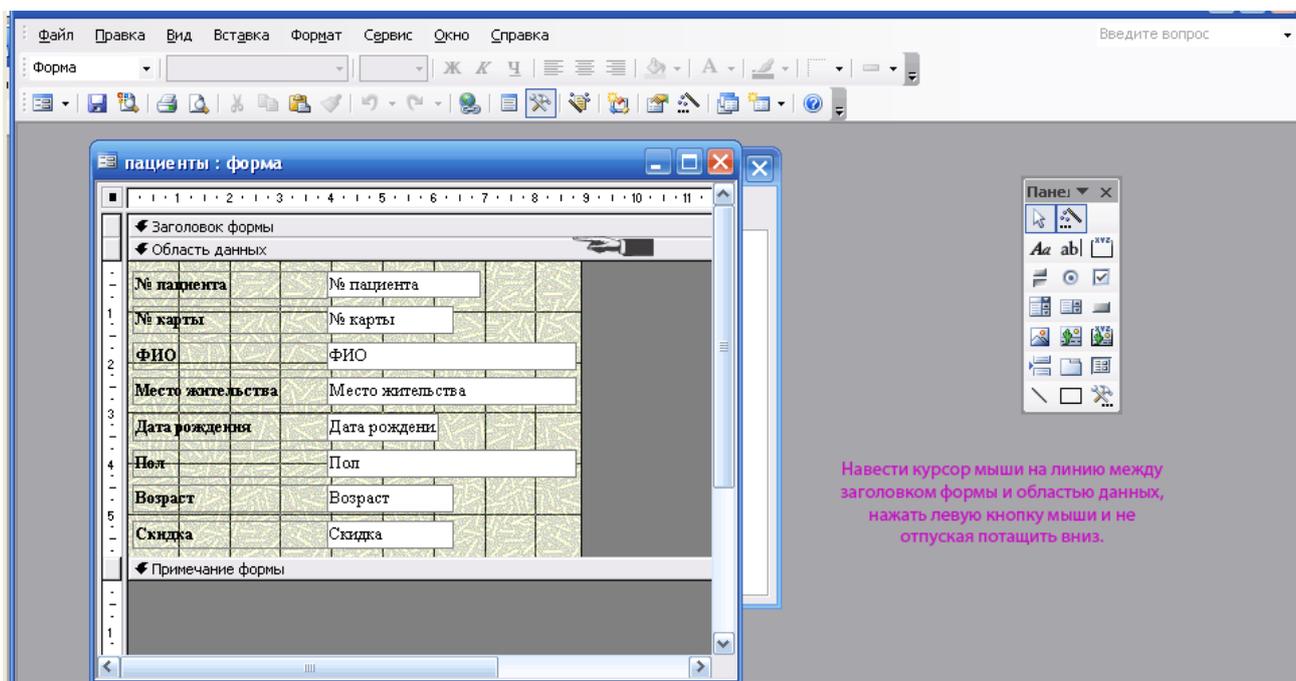


Рисунок 22.

3. Вызвать панель элементов. Для этого на панели инструментов надо утопить кнопку с пиктограммой молотка и гаечного ключа.

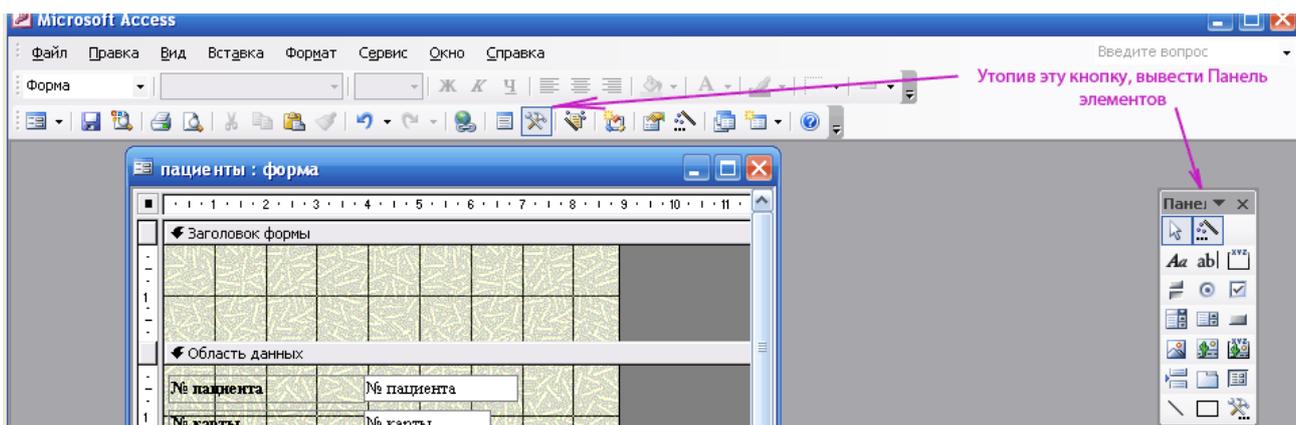


Рисунок 23.

На панели элементов найти элемент с пиктограммой **Aa** (наз. **надпись**), затем в увеличенной области заголовка формы растянуть небольшой прямоугольник и напечатать в нем название формы **Пациенты**. Закрепить название, т.е. выйти из текстового режима нажатием клавиши **Enter** на клавиатуре или при помощи щелчка мыши в любом месте формы.

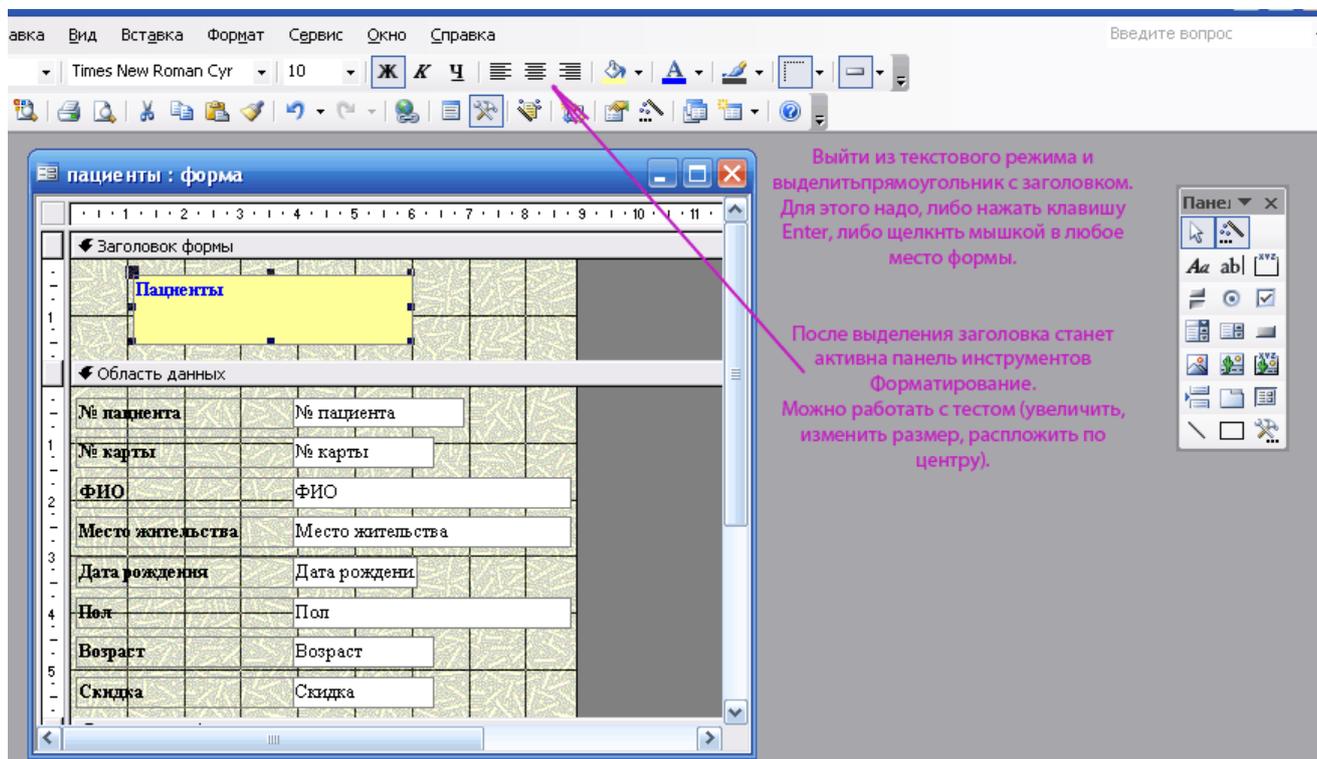


Рисунок 24.

4. Выделить надпись **Пациенты**, щелкнув по ней мышью. На панели инструментов станут активными кнопки с пиктограммами в виде кружки (заливка фона), ручки (изменение цвета шрифта), и кисточки (цвет границы), а также кнопки толщины линии и оформления. При помощи этих кнопок отформатировать надпись (изменить цвет шрифта, фона и т.д.).
5. Выйти из режима конструктора, сохранив изменения, открыть форму в обычном режиме и убедиться в правильности настройки элементов формы.

## ЗАПРОСЫ

*Запросы позволяют получать из базы данных только необходимые данные, давая возможность успешно решать задачи и получать ответы на вопросы. С помощью запросов можно отбирать, объединять, многократно использовать и анализировать данные, собранные в нескольких связанных между собой таблицах. При создании запроса используется язык SQL (структурированный язык за-*

*просов), который используется также для обновления и управления реляционными базами данных.*

## **СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ**

1. Перейти на вкладку **Запросы**.
2. Щелкнуть по кнопке **Создать** и выбрать **конструктор** для создания запроса. При этом на экране появятся два окна: окно конструктора запросов, на которое наложено **окно добавления таблицы**. С помощью последнего окна выполнить операцию добавления таблицы **Пациенты** (предварительно выделив ее в списке таблиц), после этого закрыть окно добавления. Изображение таблицы появится в верхней части конструктора в виде прямоугольного окна со списком полей таблицы. Если случайно появились лишние окна, их нужно убрать щелчком правой кнопки по таблице с последующим выбором пункта **Удалить** из контекстного меню.
3. В нижней части окна конструктора запросов располагается механизм формирования запроса. Он представляет собой таблицу, в столбцах которой следует расположить требуемые поля одной или нескольких таблиц, а в строках – следующие параметры:
  - **Поле** – имя поля таблицы.
  - **Имя таблицы** – имя таблицы, которой принадлежит поле.
  - **Сортировка** – порядок сортировки записей в запросе – алфавитный или обратный, необязательное поле.
  - **Вывод на экран** – флажок, управляющий выводом поля на экран при выполнении запроса. Поле с выключенным выводом на экран допускает сортировку и установку условий отбора, но после выполнения запроса в окне запроса это поле не будет отражено.
  - **Условие отбора** – вводится условие отбора для вывода тех записей, которые удовлетворяют этим условиям.
4. **Создать запрос на выборку пациентов, фамилии которых начинаются на определенную букву (например, на С), а также выбрать их дату рождения и место жительства.** Для этого заполнить первые три ячейки строки **поле** именами полей таблицы: **ФИО**, **Дата рождения** и **Место жительства**. Это можно сделать следующими способами:
  - Отбуксировать имя поля из таблицы в ячейку строки **Поле**.
  - Щелкнуть по ячейке в строке **Поле**, открыть выпадающий список при помощи кнопки со стрелкой, указывающей вниз и выбрать нужное поле.
  - Выполнить двойной щелчок по нужному полю в таблице, находящейся в верхней части конструктора. При этом имя поля появится в ближайшей свободной ячейке.

- **Не следует вводить имя поля с клавиатуры во избежание ошибок!!!**
- В строке **Условие отбора** в первой ячейке для поля **ФИО** зададим условие отбора **P\***

Таблица 7

поле	ФИО	Дата рождения	Место жительства	
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора	P*			

*В строке Условие отбора можно использовать некоторые символы, которые заменяют букву, группу букв, цифру:*

\* – заменяет любое количество любых букв.

? – заменяет одну какую-либо букву.

# – заменяет одну какую-либо цифру.

5. Выполнить запрос. Для этого можно не закрывая его щелкнуть по кнопке на панели инструментов с изображением восклицательного знака или выполнить пункт меню **Запрос/Запуск**.
6. После выполнения в окне запроса будут выведены все пациенты, фамилии которых начинаются на букву **P**, а также их даты рождения и место жительства.
7. Закрывать запрос (**Файл/Закреть**) и присвоить ему имя **Фамилия**.
8. **Создать запрос на выборку ФИО пациентов мужского пола в возрасте до 30 лет.** Для этого заполнить первые три ячейки строки **поле** именами полей таблицы: **ФИО**, **Пол** и **Возраст**. Далее ввести условия отбора для полей **Пол** и **Возраст**. Схема запроса:

Таблица 8

поле	ФИО	Пол	Возраст	
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора		<b>м</b>	<b>&lt;30</b>	

9. Выполнить запрос, посмотреть есть ли **пациенты, удовлетворяющие этим 2 условиям**. (если нет, то изменить в условии отбора для пола **м** на **ж**). Сохранить под именем **Мужчины до30лет**.
10. **Создать следующий запрос**, выбрав те же поля **ФИО**, **Пол** и **Возраст**, но добавив еще одно условие отбора **или**:

Таблица 9

поле	ФИО	Пол	Возраст	
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора			<b>&lt;30</b>	
или		<b>м</b>		

т.е. условие отбора для поля **Пол** перенесли на строку ниже **Или** (логическое **И**, которое дает возможность выполнения и того и другого условия).

11. После выполнения запроса обратить внимание на различие данных, полученных после выполнения этих двух запросов (в данном запросе **выводятся мужчины любого возраста и пациенты обоих полов, но моложе 30 лет, не выводятся только женщины старше 30 лет**). Сохранить под именем **И** пол и возраст.
12. Создать еще один запрос на выборку пациентов из 2-х связанных таблиц, которые были госпитализированы с определенным диагнозом (*например, ОРЗ*) и лечившихся у определенного врача (*например, у Таблеткина*). В окне добавления таблицы добавить таблицы Пациенты и Госпитализация.

Таблица 10

поле	ФИО	Врач	Диагноз	
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора		Таблеткин	орз	

13. Сохранить под именем **Врач и Диагноз**.
14. Создать запрос **Отбор по дате**, который отбирает пациентов моложе 30 лет, принимавших определенное лекарство. (Добавляем 2 таблицы).

Таблица 11

поле	ФИО	Дата рождения	Лекарство	
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора		> 1.01.80	аспирин	

15. Сохранить под именем **Отбор по дате моложе 30 лет**.

## СОЗДАНИЕ ЗАПРОСОВ С ПАРАМЕТРАМИ

*Запрос с параметрами дает возможность не изменять условие отбора для одного поля в конструкторе, создавая несколько запросов, а изменять это условие при запуске запроса. Т.е. при запуске на выполнение одного и того же запроса, задавая разные параметры в окне **Введите значение параметра**, будут выбираться разные данные, соответствующие тому параметру, который будет введен.*

1. Используя те же приемы, создать новый запрос по таблицам **Пациенты и Госпитализация**. Но в условии отбора ввести следующий текст, заключенный в квадратные скобки: **[Введите фамилию врача]**.

Таблица 12

поле	ФИО	Возраст	Врач	Диагноз
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора			<b>[Введите фамилию врача]</b>	

2. При его выполнении вначале возникает диалоговое окно **Введите значение параметра**, в ячейку которого следует ввести одну из фамилий врача, и будут выведены данные на пациентов, которых он лечил. Преимущество в том, что

не надо на каждого врача создавать запрос и в условии отбора вводить его фамилию. Убедиться в правильности выполнения запроса.

3. Сохранить запрос под именем **Параметр – Врач**.
4. Создать еще один запрос с параметром, в качестве параметра используя диагноз.

Таблица 13

поле	ФИО	Возраст	Врач	Диагноз
Имя Таблицы				
Сортировка				
Вывод на экран	✓	✓	✓	
Условие отбора				[Введите фамилию диагноз]

5. Сохранить запрос под именем **Параметр - Диагноз**.

## СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ С ИСПОЛЬЗОВАНИЕМ ГРУППОВЫХ ОПЕРАЦИЙ

*Групповым называется такой запрос, в котором выполняется группировка данных по содержимому одного из полей таблицы, а для остальных включенных в запрос полей могут вычисляться значения, общие для каждой группы. Основные из них:*

- **Sum** – сумма значений в группе
- **Avg** – среднее значение
- **Min** – минимальное значение
- **Max** – максимальное значение
- **Count** – число значений
- **StDev** – стандартное отклонение
- **Var** – дисперсия

1. Создать новый запрос, добавив в него одну таблицу – Пациенты. Выбрать поля Пол и еще раз Пол (или можно выбрать поле, где тип данных – счетчик. Это поле - № пациента).
2. Установить групповой тип запроса. Для этого выполнить пункт меню **Вид/Групповые операции** (или щелкнуть по кнопке с изображением знака суммирования на панели инструментов). При этом в конструкторе запросов должна появиться строка **Групповая операция**, а в обоих полях в этой строке должно появиться название операции **Группировка**. Это название должно сохраниться в поле Пол, определяя тем самым, что именно по этому полю будет произведена группировка данных для вычисления задаваемых параметров, вместо группировки для второго поля Пол (или для поля № пациента) выбирать значение **Count**.

Таблица 14

поле	№ пациента	Пол	
Имя Таблицы			
Групповые операции	<b>Count</b>	<b>Группировка</b>	
Сортировка			
Вывод на экран	✓	✓	
Условие отбора			
или			

3. Выполнить запрос и получить данные по количеству мужчин и женщин.
4. Сохранить под названием **Количество мужчин и женщин**.
5. **Создать новый запрос**, добавив в него две таблицы – **Госпитализация и Препараты**. При помощи этого запроса подсчитать на какую сумму выписал лекарств каждый врач. Схема этого запроса выглядит так:

Таблица 15

поле	Врач	Стоимость		
Имя Таблицы				
Групповые операции	<b>Группировка</b>	<b>Sum</b>		
Сортировка				
Вывод на экран	✓	✓		
Условие отбора				
или				

6. Сохранить под названием **Сумма лекарств**.
7. Создать групповой запрос, который вычисляет средний возраст для мужчин и для женщин.

## СОЗДАНИЕ ЗАПРОСОВ НА ОБНОВЛЕНИЕ

Запрос на обновление это такой тип запроса, с помощью которого можно изменить содержимое полей таблиц.

1. Создать новый запрос по таблицам **Пациенты, Госпитализация и Препараты**. Включить в него то поле, которое будет обновляться – поле **Оплата**.
2. Установить требуемый тип запроса. Для этого выполнить пункт меню **Запрос/Обновление** либо на панели инструментов нажать кнопку **Тип запросов** (на ней пиктограмма в виде двух таблиц).
3. В появившейся в конструкторе строке **Обновление** для поля **Оплата** щелкнуть правой кнопкой мыши, вызвав контекстное меню и выбрать опцию **Построить**. В верхнюю часть **Построителя выражений**, где моргает курсор, необходимо ввести выражение, которое будет просчитывать оплата за вычетом скидки. Для этого в левой нижней ячейке двойным щелчком открыть папку **Таблицы**, выбрать таблицу **Препараты**, раскрыть ее и двойным щелчком перенести **Стоимость** в верхнюю ячейку, нажать знак «-», который находится под этой ячейкой в числовом блоке (**при вводе математических знаков**

и скобок пользоваться только этим числовым блоком), еще раз перенести **Стоимость**, выбрать знак умножения, затем из таблицы **Пациенты** перенести двойным щелчком поле **Скидка**, выбрать знак деления и ввести с клавиатуры 100. (**Стоимость - Стоимость \* Скидка/100**). Закрывать построитель, и выполнить запрос. При выполнении запроса **Access** предупредит, что будут обновлены данные в таблице.

4. Убедиться в правильности обновления можно только раскрыв таблицу **Пациенты** и посмотрев поле **Оплата**. Сохранить запрос под именем **Оплата**.

## СОЗДАНИЕ ЗАПРОСОВ С ПОМОЩЬЮ ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ

1. Создать новый запрос по таблице **Пациенты**, включив в него поля **Фамилия** и **Пол**.
2. Определить новое поле, в котором будет вычисляться возраст пациента. Для этого в свободную ячейку (третью по счету) строки **Поле** щелкнуть правой клавишей мыши, вызвав контекстное меню и выбрать в нем опцию **Построитель**. Откроется окно **Построителя выражений**. В верхнюю часть **Построителя выражений**, где моргает курсор, необходимо ввести выражение:

**Возраст: Year(Now()) - Year([Дата рождения])**

Элементы этого выражения обозначают:

- **Возраст** – название нового поля. Отделяется от остальной части выражения двоеточием;
- **Now()** – встроенная функция, которая возвращает текущую дату, аргументов не имеет, но пустые круглые скобки обязательны;
- **Year(...)** – встроенная функция, в качестве аргумента требует данные в формате даты, возвращает содержащийся в дате год.
- **Year(Now())** – вложенные функции: функция **Now()** возвращает текущую дату, которая становится аргументом функции **Year(...)**, последняя возвращает выделенный из даты текущий год. Например, если запрос выполняется 25 сентября 2006 года, то результатом выполнения вложенных функций будет число 2006.
- **Year([Дата рождения])** – аргументом функции **Year(...)** является значение поля таблицы **[Дата рождения]** (название поля задается в квадратных скобках).
- Для этого в левой нижней ячейке двойным щелчком открыть папку **Функции**, и найти папку **Встроенные функции**, раскрыть. В средней ячейке показаны все типы встроенных функций, выбрать тип **Дата и время** или **Все**, затем в правой ячейке найти функцию **Year** и двойным

щелчком по ней перенести ее в верхнюю ячейку. Затем нажать на знак «-» и еще раз двойной щелчок по функции **Year**.

В верхней ячейке получено выражение:

**Year(«number») - Year(«number»)**, щелчком мыши в скобках первого **Year(«number»)** выделить «number» и вместо его выбрать функцию **Now()** из третьей ячейки, вместо второго «number» выбрать из таблицы **Пациенты** поле **Дата рождения**. Получено выражение: **Year(Now()) - Year([Дата рождения])**. Нажать кнопку **ОК**, выполнить запрос. В запросе поле, в котором просчитан возраст имеет название **Выражение1**, чтобы это исправить, надо вернуться в конструктор запроса и в ячейке, где прописано выражение, аккуратно вместо **Выражение1** напечатать **Возраст**.

3. Выполнить запрос и убедиться в правильности вычислений для каждой записи.
4. Сохранить запрос под именем **Возраст пациентов**.
5. Создать новый запрос по таблицам **Пациенты** и **Госпитализация**, включив в него поля: **ФИО**, **№ карты**, **Врач** и **Диагноз**, для выбора пациентов, которые пролежали в стационаре менее 8 дней. Далее в свободную ячейку (пятую по счету) строки **Поле** щелкнуть правой клавишей мыши, вызвав контекстное меню и выбрать в нем опцию **Построитель**. Откроется окно **Построителя выражений**. В верхнюю часть **Построителя выражений**, где моргает курсор, необходимо по аналогии с предыдущим запросом ввести выражение: **Day([Дата выписки])-Day([Дата поступления])**.  
*Можно упростить: не использовать встроенную функцию **Day(...)**, а сразу ввести **[Дата выписки]- [Дата поступления]***
6. Далее в конструкторе запроса в ячейке (пятой по счету) строки **Условие отбора** записать условие **<10**.
7. Выполнить запрос и убедиться в правильности вычислений. Сохранить запрос под именем **Число дней**.

## ПОСТРОЕНИЕ ДИАГРАММ

*Построить диаграмму типа гистограммы, по запросу **Количество мужчин и женщин**, в которой данными являются количество пациентов, а по оси **X** отложен пол пациентов.*

1. Перейти на вкладку **Формы**.
2. Щелкнуть по кнопке **Создать**.
3. В качестве источника данных выбрать из списка созданный ранее запрос **Количество мужчин и женщин**. Установить тип создаваемой формы **Диаграмма**

- и щелкнуть по кнопке **ОК**. После этого начинает работать мастер построения диаграмм.
4. На первом шаге мастера следует перенести названия обоих полей из списка **Доступные поля** в список **Поля Диаграммы**. По кнопке **Далее** перейти к следующему шагу.
  5. На втором шаге выбрать простую гистограмму в качестве типа диаграммы.
  6. На третьем шаге проверить установленные по умолчанию параметры диаграммы. По оси **X** должен быть отложен пол, по оси **Y** – количество. Если это не так, то отбуксировать на нужные места названия полей, приведенные в правой части окна мастера.
  7. На этом же шаге откорректировать (**только, если это является необходимостью**) вид вычислений, которые выполняются при построении диаграммы. Для этого нужно выполнить двойной щелчок по названию поля данных (обычно это **Сумма ...**) и выбрать тот вид вычислений, который необходим.
  8. На четвертом шаге следует задать имя диаграммы, можно оставить приведенное по умолчанию. После щелчка по кнопке **Готово** на экране появится диаграмма. Однако вид диаграммы, как правило, требует коррекции.
  9. Перейти в конструктор формы, выполнив пункт меню **Вид/Конструктор**. На экране появится окно конструктора формы (серого цвета с черной сеткой) с наложенным на него окном диаграммы (белый фон с гистограммой и фрагментами надписей). Надписи типа **1, 2, 3** и столбцы **A, B, C, D** представляют общий условный вид таблицы данных для изображения диаграммы в **Microsoft Graph**, который можно отредактировать (*убрать лишние столбцы – ряды данных, добавить необходимые, также поступить со строками, или внести изменения в числовые и текстовые данные*) по своему усмотрению. В данном случае со всем этим справится **Мастер диаграмм**. В данном случае надо оставить две строки, которые будут соответствовать полу – **м** и **ж**, и один столбец, который будет соответствовать количеству пациентов.
  10. Для изменения цвета столбцов щелкнуть мышью по столбцу один раз – выделение обоих столбцов, еще щелчок – выделение только одного столбца, двойной щелчок на выделенный столбец – вызов в **Microsoft Graph** окна для редактирования цвета. После изменения цвета, хотя бы у одного столбца, сразу в области диаграммы раскрывается легенда, где каждый цвет указывает на определенный пол.
  11. Для того чтобы изменить размеры рисунка необходимо перейти в конструктор диаграмм. Выполните двойной щелчок по диаграмме, при этом на экране возникнет конструктор диаграмм (**Microsoft Graph**), в рабочей области которого находятся 2 окна – таблица данных и диаграмма. Не трогая таблицу, следует растянуть окно диаграммы на доступную часть экрана.
  12. Выйти из конструктора диаграмм (**Файл/Выход и возврат в Формат**). Затем выйти из конструктора форм (**Вид/Режим формы**). Убедиться, что все изме-

нения правильны, в противном случае вернуться в конструктор формы, затем в конструктор диаграмм и повторить настройку.

13. Закрывать, сохранив форматирование.

## **ВСТРАИВАНИЕ В ФОРМУ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ ИЛИ СОЗДАНИЕ ПРОЦЕДУРЫ ОБРАБОТКИ СОБЫТИЯ**

*Программа на языке Microsoft Visual Basic пишется в виде блоков, которые называют процедурами. Процедура содержит набор инструкций Visual Basic, с помощью которых выполняются действия или рассчитывается значение. Процедурой обработки события называют процедуру, выполняющуюся в ответ на событие, инициируемое пользователем или программой или генерируемое системой. Событием называют определенное действие, возникающее при работе с конкретным объектом. Microsoft Access реагирует на целый ряд событий: нажатие кнопки мыши, изменение данных, открытие или закрытие форм и т.д. С помощью процедуры обработки события пользователь имеет возможность определить собственные отклики на события, происходящие в форме, отчете или элементе управления. Существуют процедуры двух типов: **Sub** и **Function**.*

*Процедура-подпрограмма **Sub** выполняет действие или набор действий, но не возвращает значение. Пользователь имеет возможность создавать процедуры **Sub** самостоятельно или использовать процедуры обработки событий, определенные в Microsoft Access.*

*Процедура-функция **Function** (часто такие процедуры называют просто функциями) возвращает значение, например, полученное в результате расчетов. Visual Basic включает ряд встроенных функций; например, функция **Now** возвращает текущее значение даты и времени. В дополнение к встроенным функциям, пользователь имеет возможность самостоятельно создавать функции, которые называют специальными или пользовательскими функциями*

**Рассмотрим пример встроенной процедуры обработки событий (например, при помощи мастера кнопок):**

1. Открыть форму Пациенты в режиме конструктора.
2. Вывести на экран панель элементов. Установить кнопку мастера встраивания в нажатое положение (вторая кнопка на панели с рисунком в виде палочки, из которой сыплется песок).

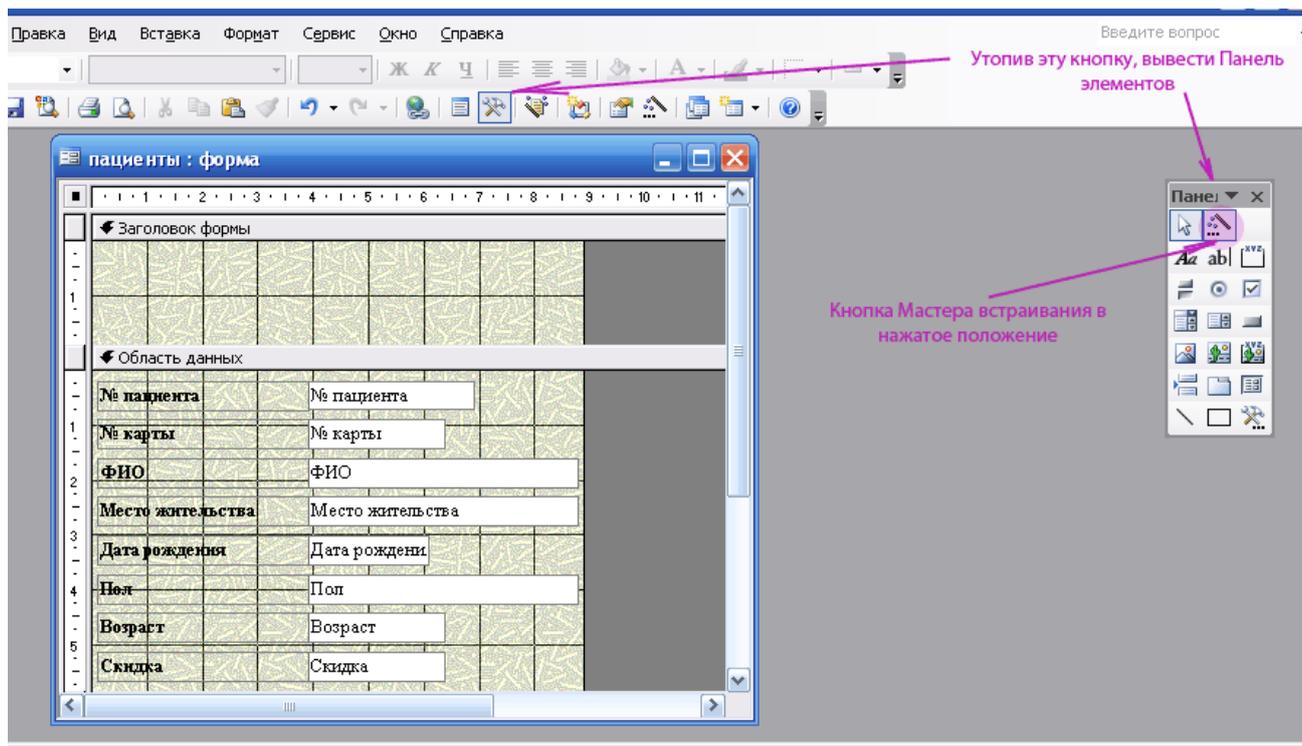


Рисунок 25.

- Щелкнуть по элементу под названием **Кнопка**. Затем в увеличенной области заголовка формы рядом с надписью растянуть небольшой квадратик. При этом должен вступить в действие мастер встраивания кнопки, который связывает с нажатием кнопки определенное действие.

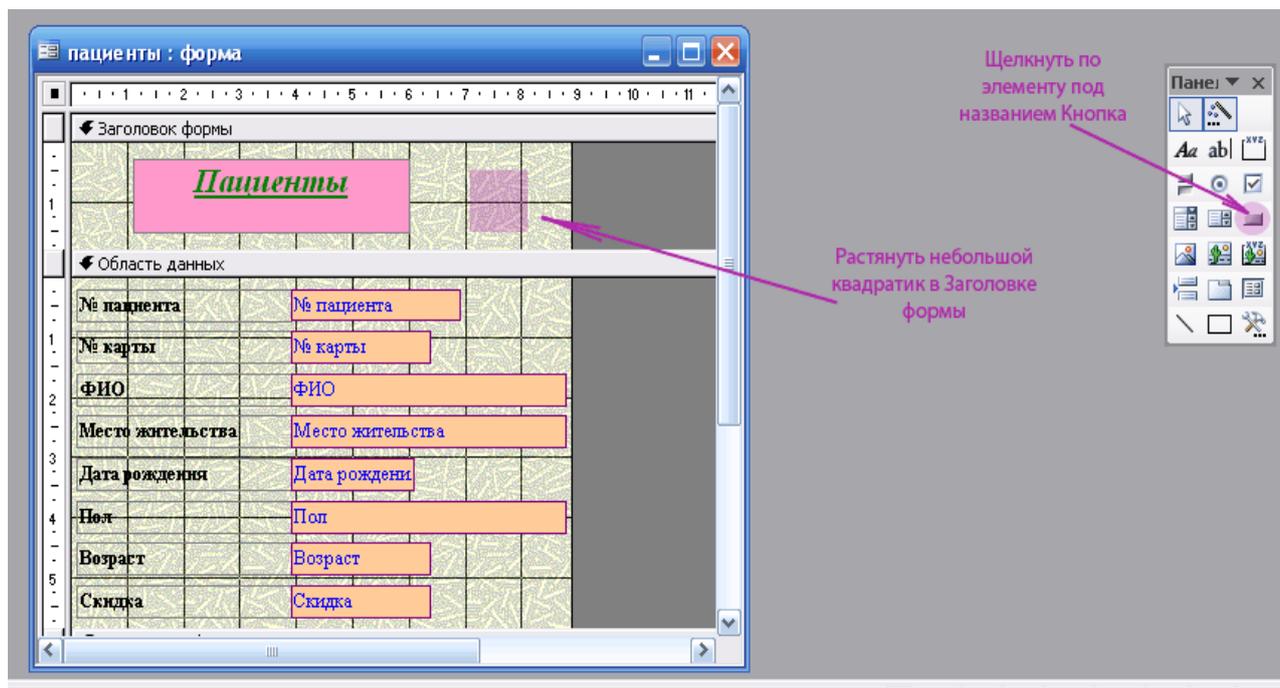


Рисунок 26.

Пользуясь подсказками мастера связать с нажатием кнопки закрытие формы.

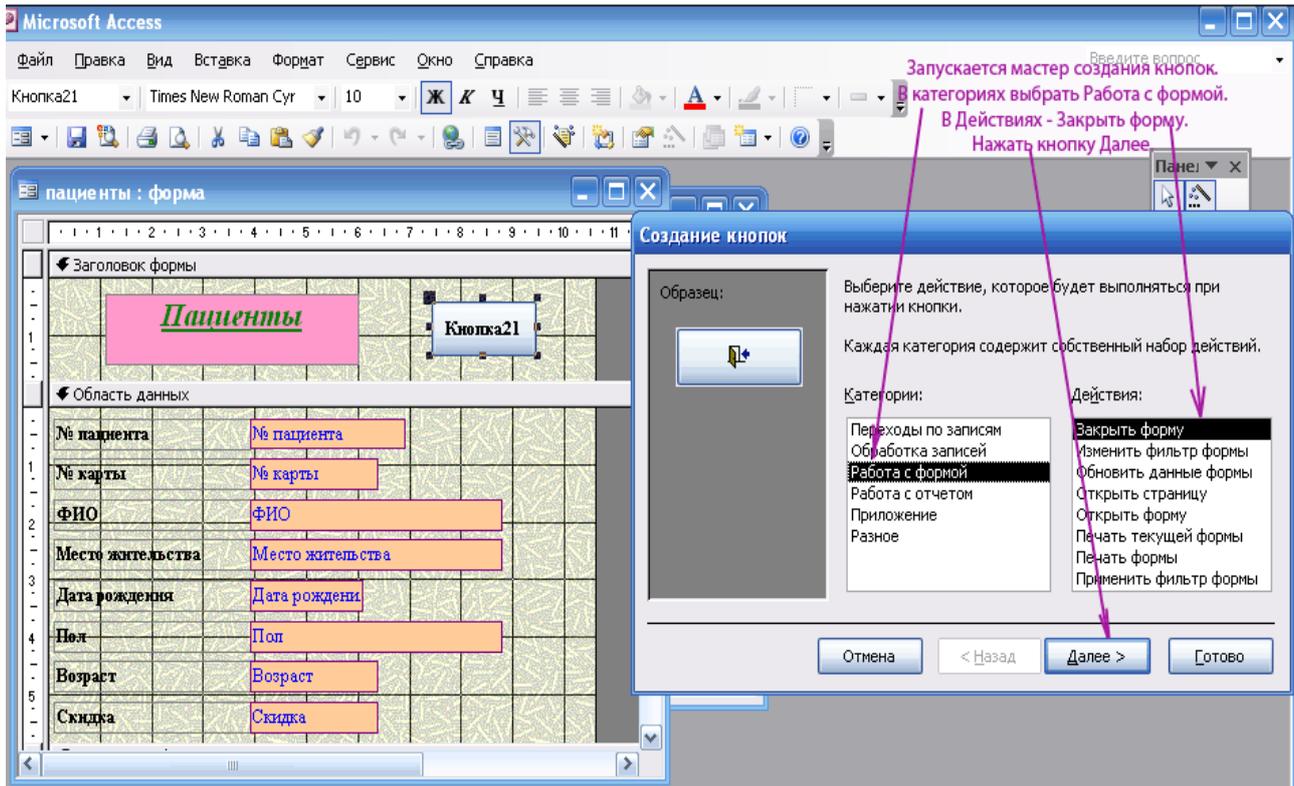


Рисунок 27.

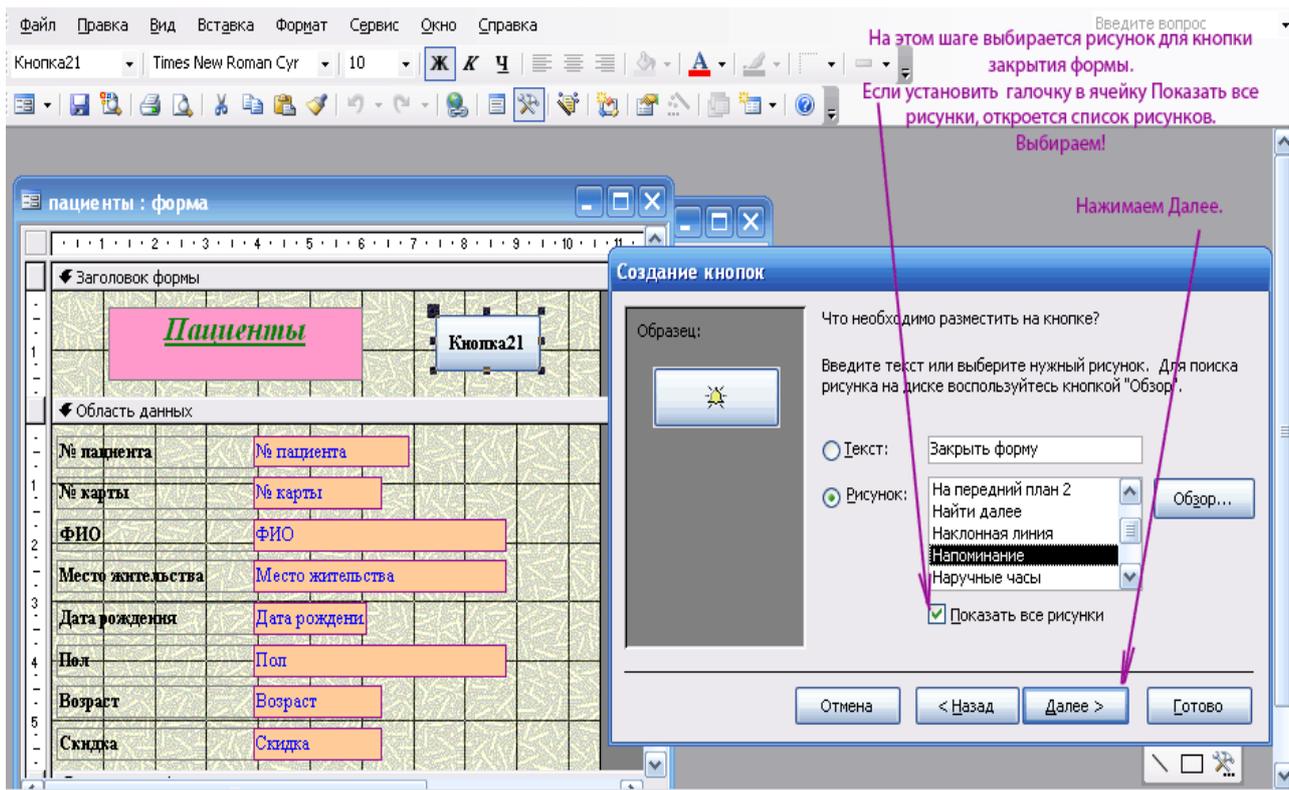


Рисунок 28.

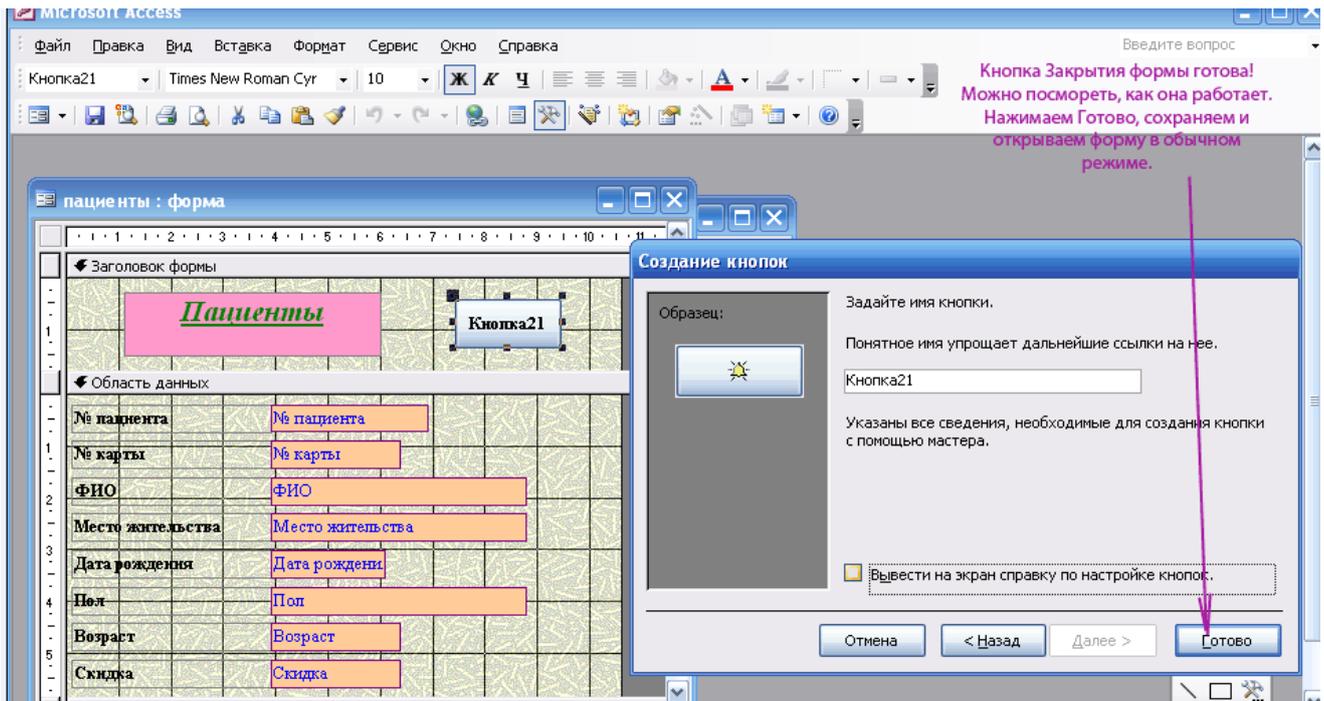


Рисунок 29.

#### 4. Закрыть конструктор. Сохранить. Проверить.

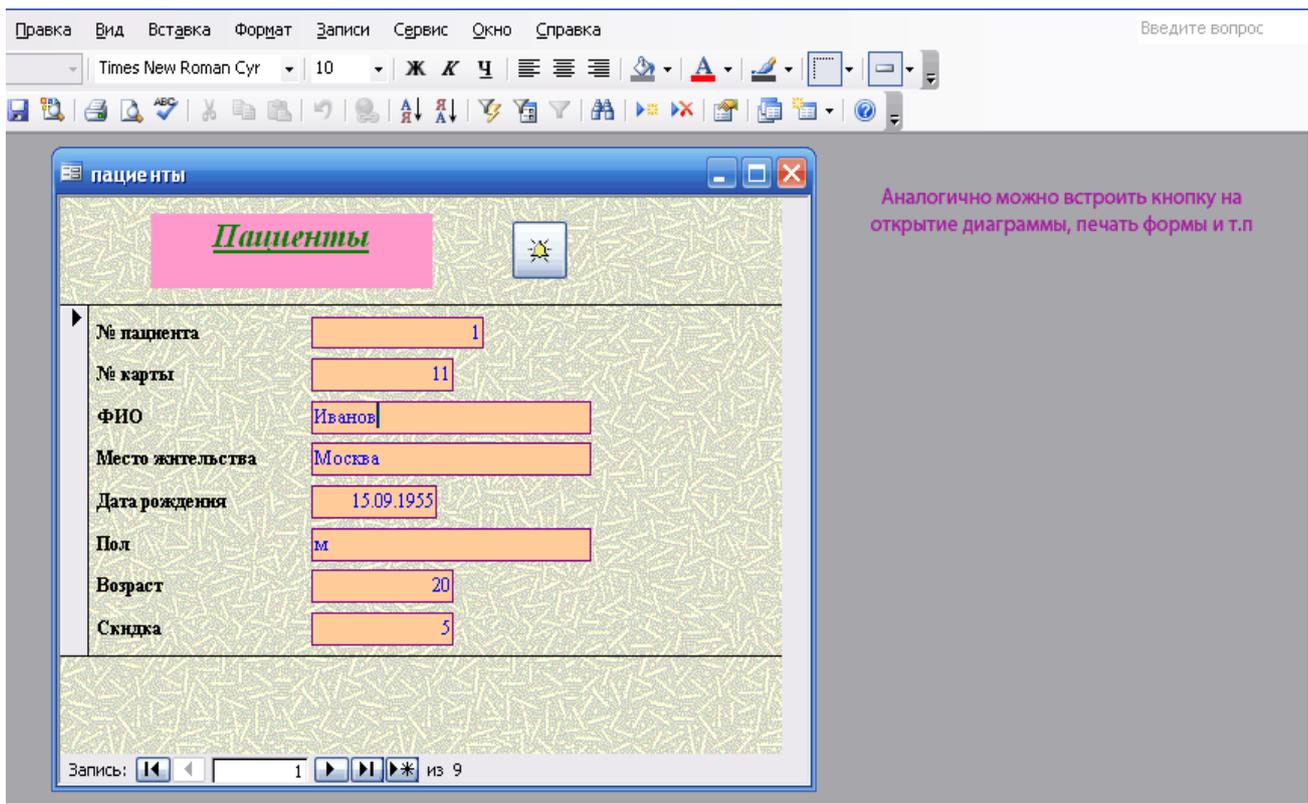


Рисунок 30.

*Аналогично можно встроить кнопку на открытие диаграммы, печать формы и т.п.*

Теперь рассмотрим возможность создания процедуры Sub самостоятельно:

5. Открыть форму Пациенты в режиме конструктора.
6. Вызвать панель элементов. Для этого на панели инструментов надо утопить кнопку с пиктограммой молотка и гаечного ключа. На панели элементов найти элемент с пиктограммой **Aa** (наз. **Надпись**), затем растянуть небольшой прямоугольник рядом с полем «**Возраст**». В этом прямоугольнике напечатать «+» и выйти из текстового режима (щелчок мыши в любом месте формы или Enter).

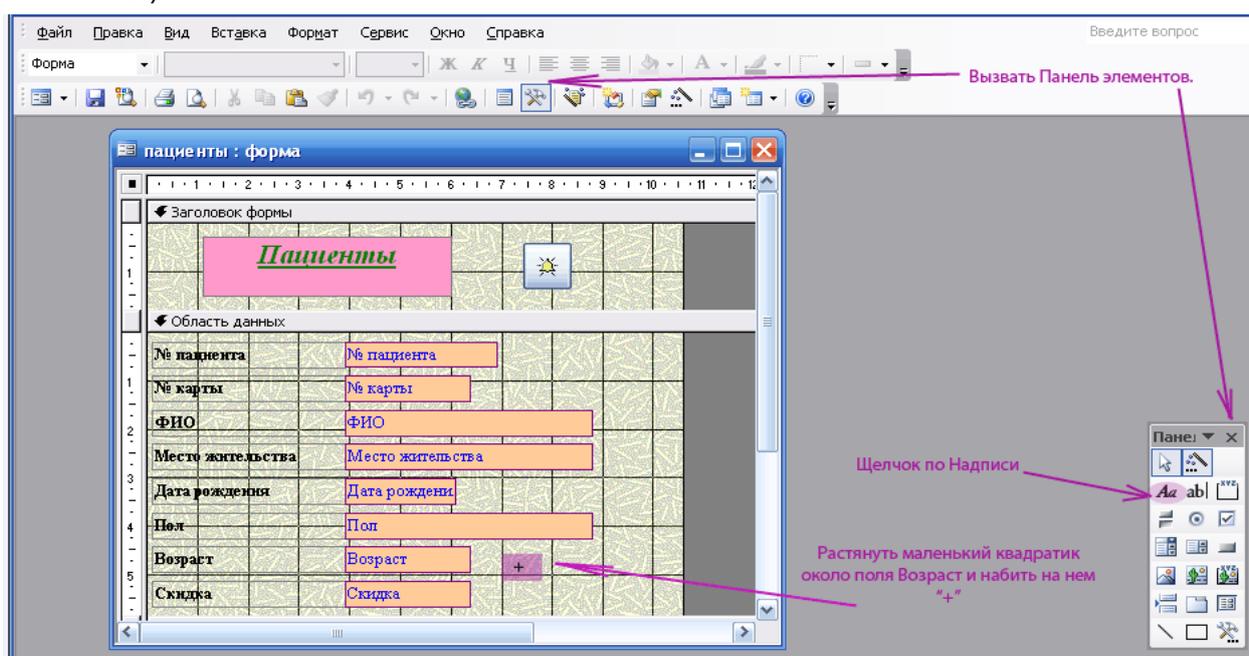


Рисунок 31.

7. Для написания процедуры обработка события щелкнуть правой клавишей мыши на надпись «+» (одновременно выделяя ее и вызывая контекстное меню) и в контекстном меню выбрать опцию обработка событий. В открывшемся окне Построитель выбрать Программы. Нажать ОК.

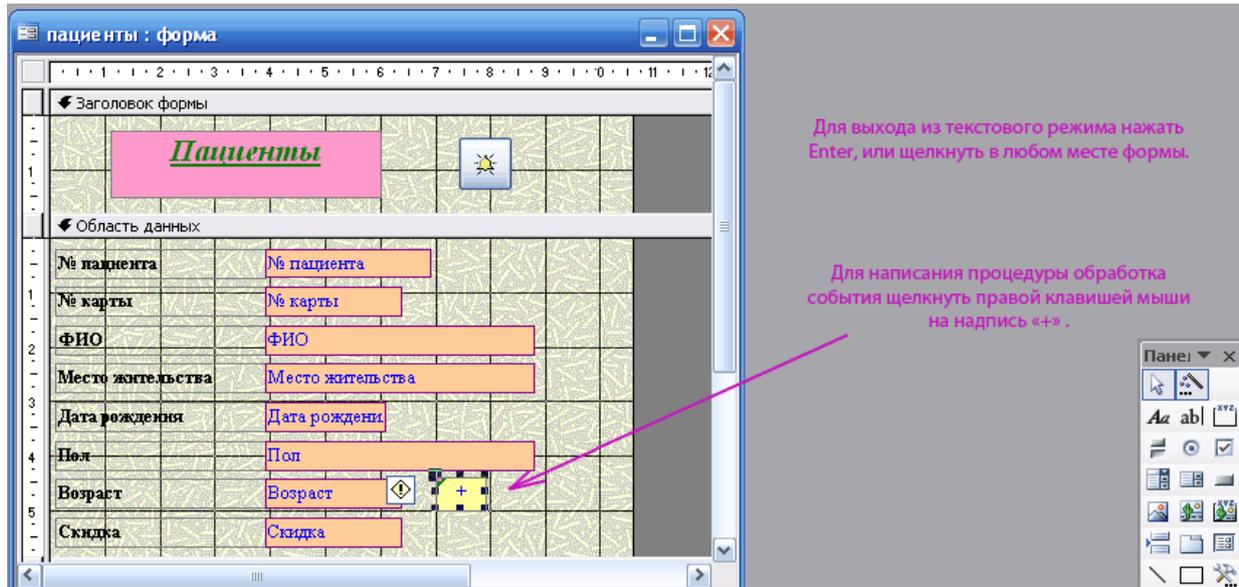


Рисунок 32.

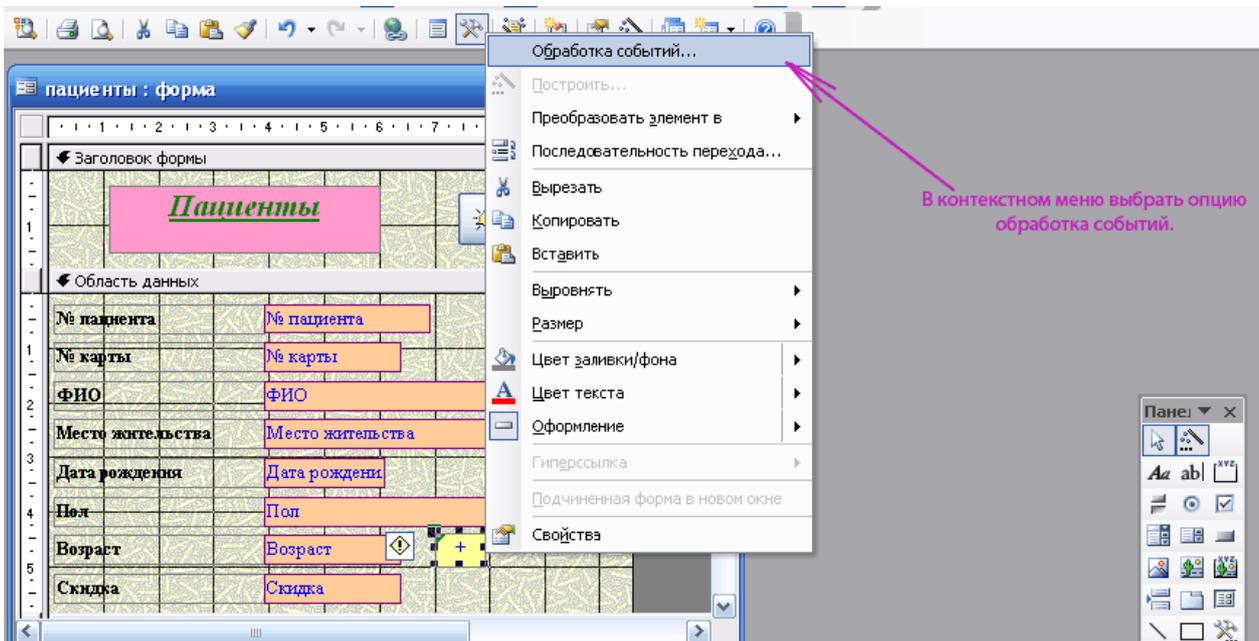


Рисунок 33.

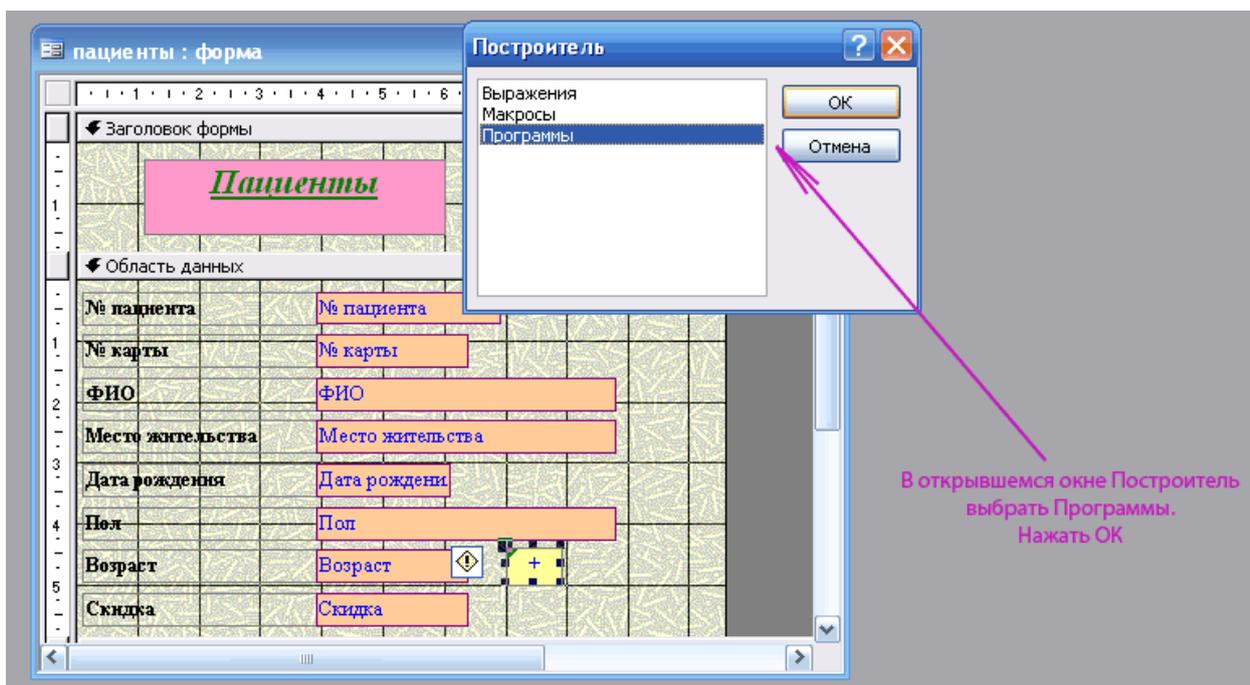


Рисунок 34.

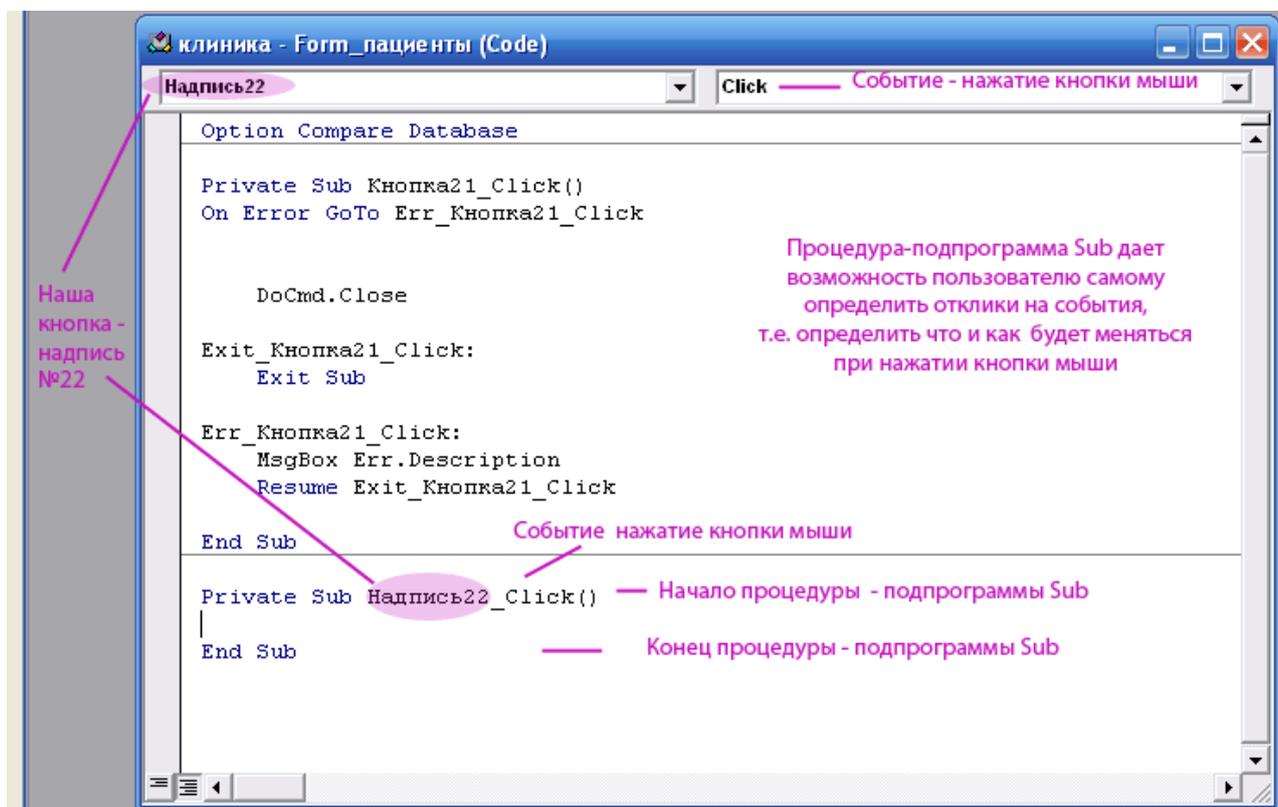


Рисунок 35.

8. В процедуре - подпрограмме Sub для кнопки №...для события **Нажатие кнопки** (в этом месте моргает курсор), напечатать присвоение полю Возраст увеличение на 10: **[Возраст] = [Возраст]+10**

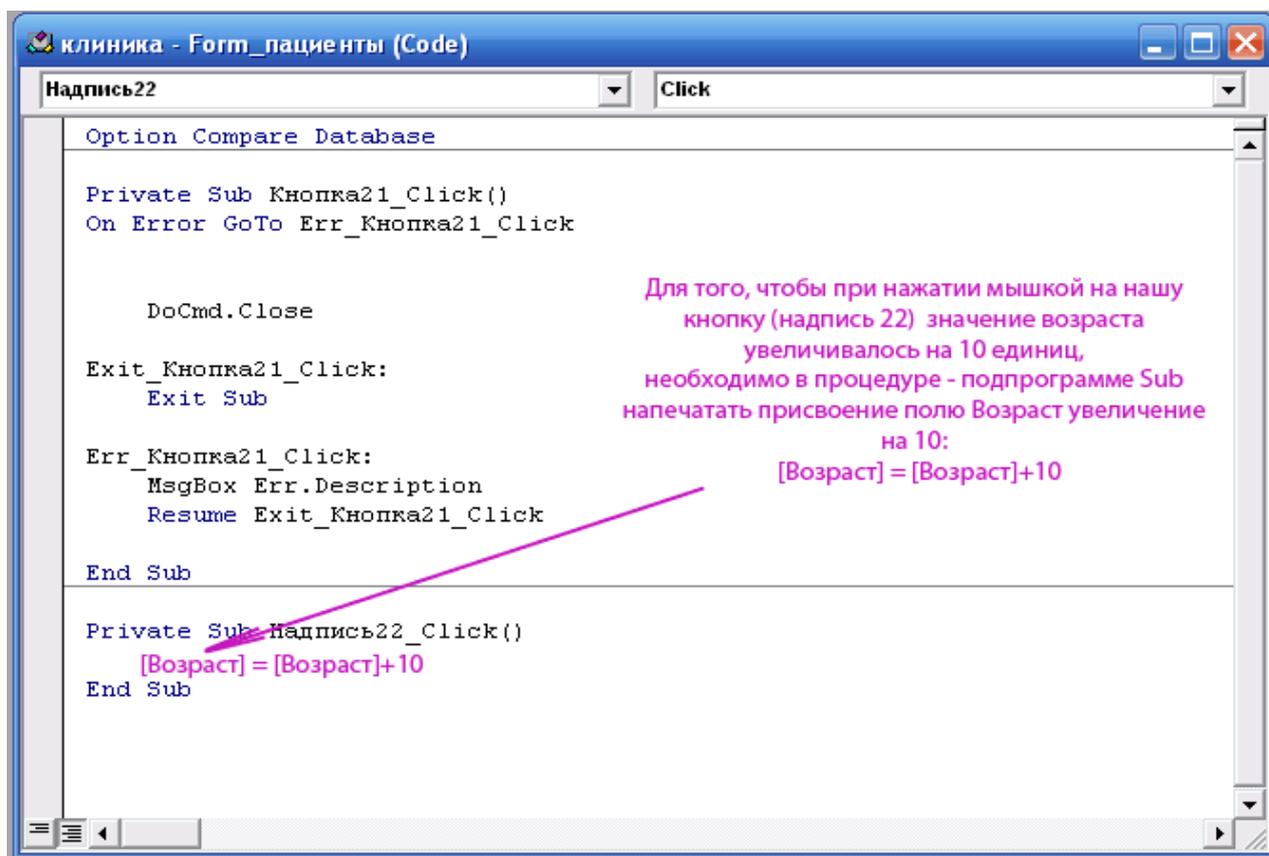


Рисунок 36.

9. Используя условный оператор If ... Then (если ... тогда), определить ограничение и возврат на меньшее значение для возраста. На следующей строке напечатать `If [Возраст] >100 Then [Возраст]=5`

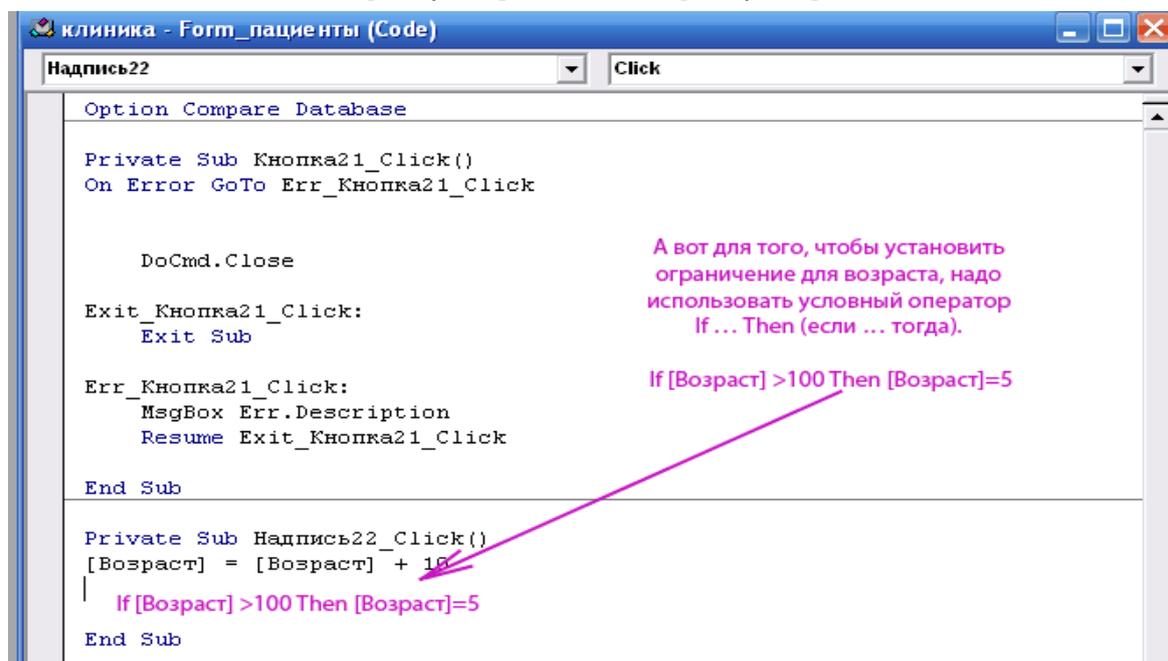


Рисунок 37.

10. Для проверки правильности написания процедуры необходимо на панели инструментов нажать кнопку Компилировать, в виде стопки листов на которую сверху указывает стрелка.

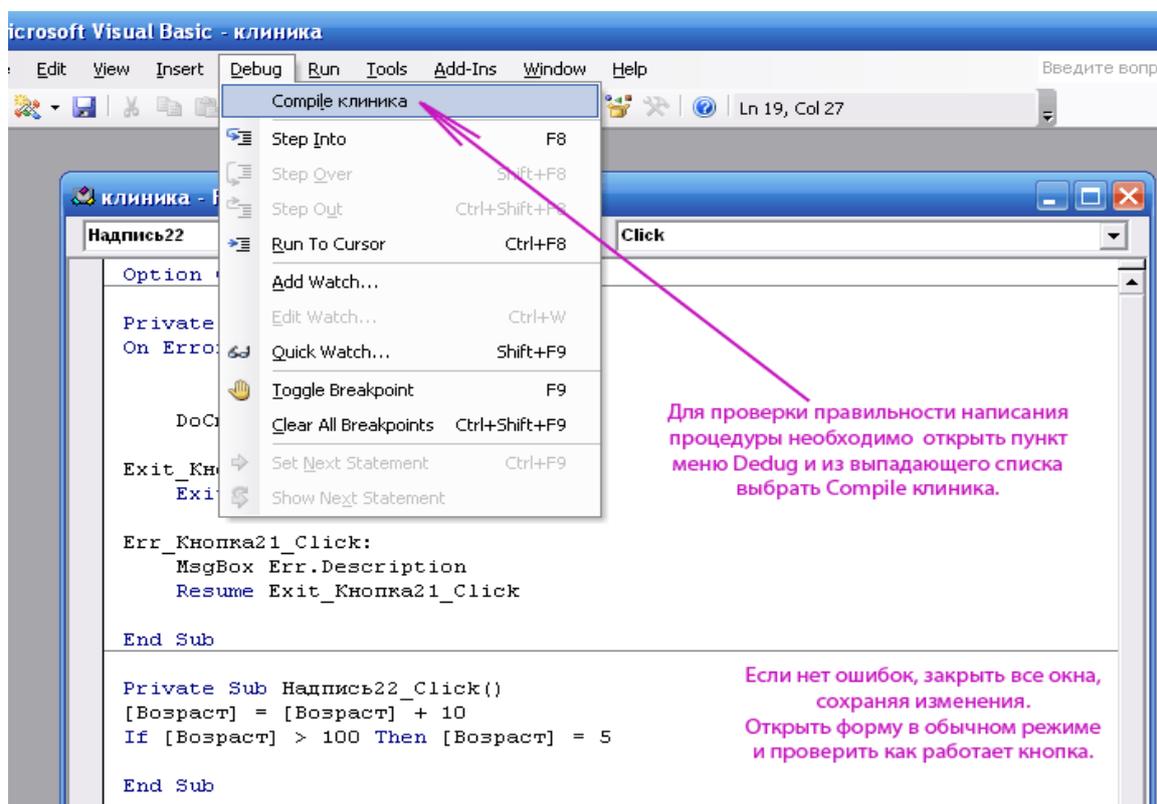


Рисунок 38.

11. Затем все закрыть, сохранить и проверить работу кнопки, открыв форму.

### Создание процедуры обработки события

1. Перейти на вкладку **Формы** и создать новую форму на основе таблицы **Пациенты**, назвав ее Пациент2.
2. Раскрыть форму в режиме **Конструктор**. Щелкнуть правой кнопкой мыши в серой области формы, вызвать контекстное меню и выбрать в нем либо **Обработку событий**, либо **Свойства**, вызвав этим окно **Свойств формы** (→ закладка **События**), где необходимо выбрать свойство события в ответ на которое должна выполняться **процедура обработки события**. Выбрать **Текущая запись** и вызвать **Построитель**, нажав кнопку с многоточием справа от ячейки.

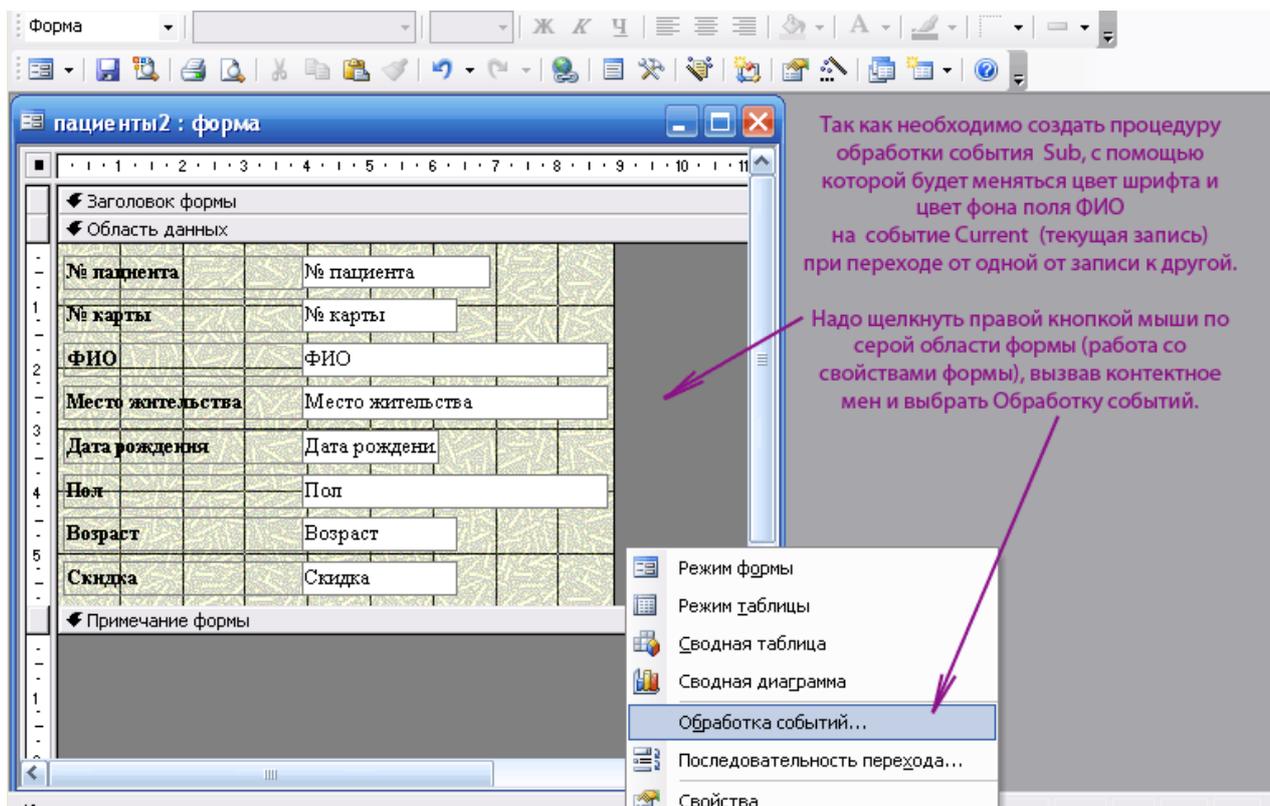


Рисунок 39.

Если в контекстном меню выбрать опцию **Обработка событий**, то сразу же открывается окно **Процедуры обработки события**, но в правой ячейке надо изменить свойство события на **Current - Текущая запись**.

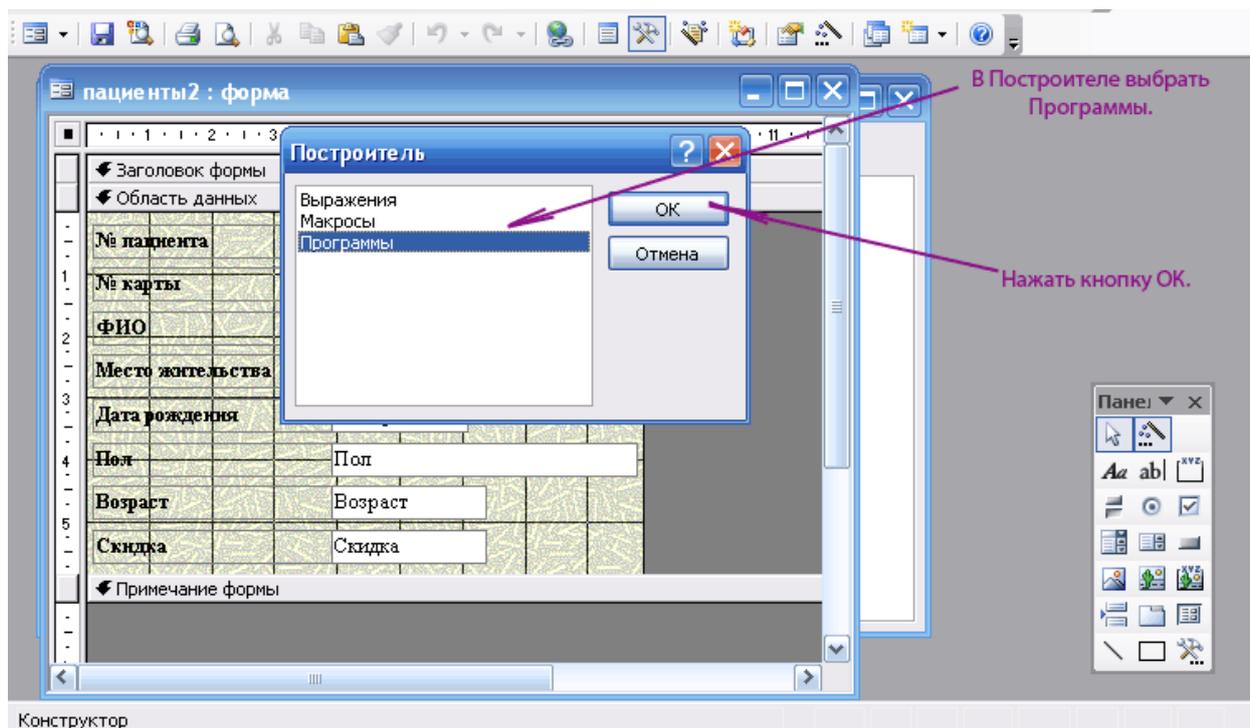


Рисунок 40.

В Построителе выбрать Программы и выйти в окно Процедуры обработки события (подпрограммы Sub)

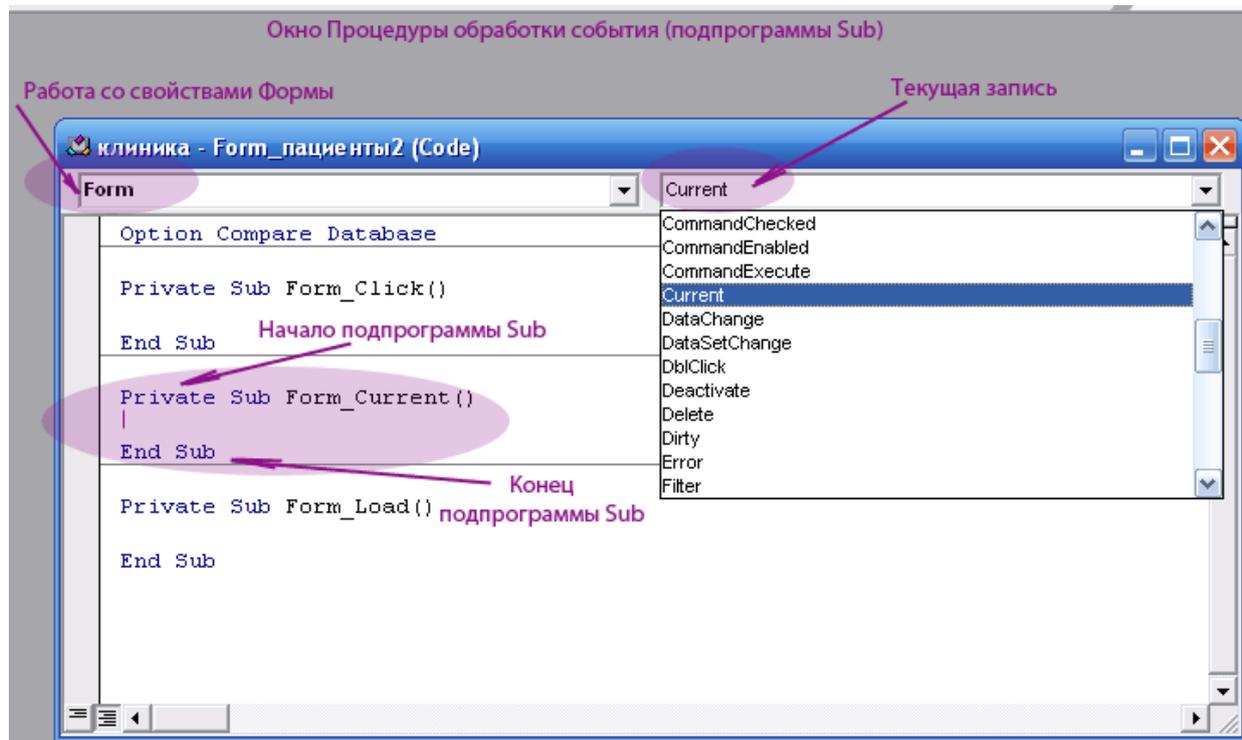


Рисунок 41.

В верхней части окна есть две ячейки: в левой показано в каком объекте происходит событие (в нашем случае – это **форма**), в правой показано выбранное нами свойство события – **Текущая запись - Current** (переход из одной записи в другую). Курсор моргает в том месте **подпрограммы Sub**, где надо написать инструкцию.

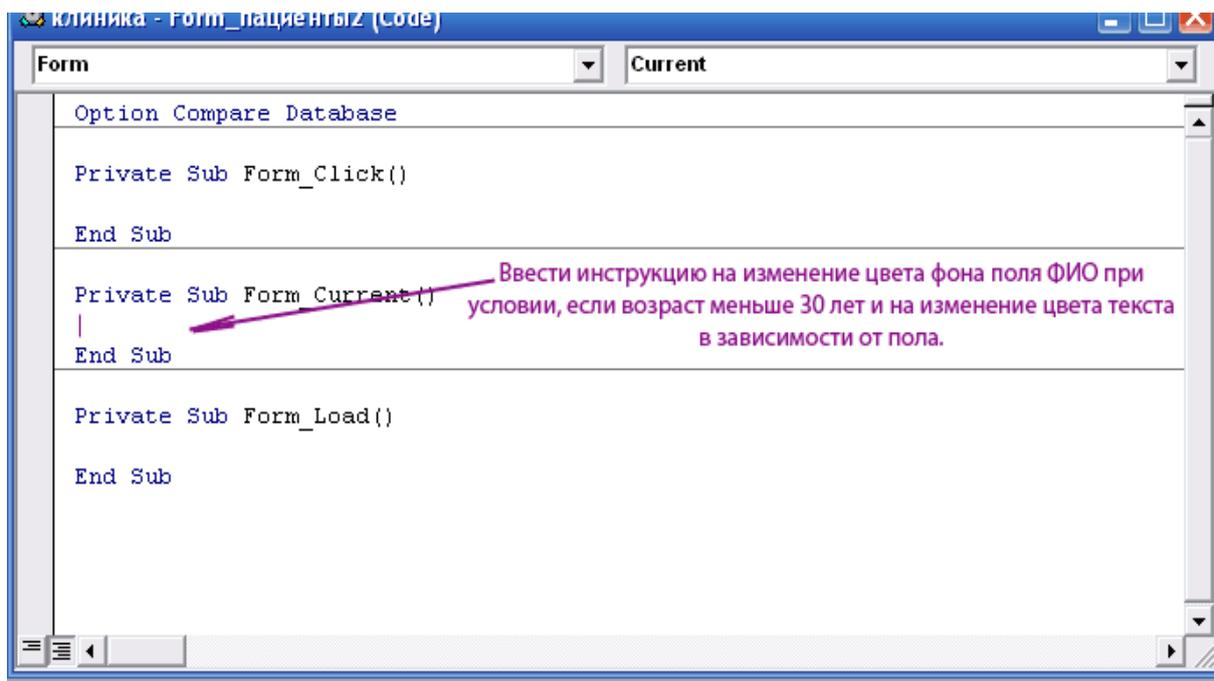


Рисунок 42.

Вводим инструкцию на изменение цвета фона поля **ФИО** при условии, если возраст меньше 30 лет и изменение цвета текста в зависимости от пола.

Цвета фона описывает оператор **BackColor**, а цвет текста – **ForeColor**, значения функции **RGB (red, green, blue)** определяют отображаемый цвет и находятся в интервале от 0 до 255. В следующей таблице приведены некоторые стандартные цвета и их красная, зеленая и синяя составляющие:

Таблица 16

<i>Цвет</i>	<i>Красная составляющая</i>	<i>Зеленая составляющая</i>	<i>Синяя составляющая</i>
Черный	0	0	0
Синий	0	0	255
Зеленый	0	255	0
Бирюзовый	0	255	255
Красный	255	0	0
Малиновый	255	0	255
Желтый	255	255	0
Белый	255	255	255

При вводе инструкции надо соблюдать некоторые правила написания:

- ✓ Если имя поля введено русскими буквами, то это имя заключается в квадратные скобки, если английскими, то без скобок.

- ✓ Если значение поля введено буквами, то эти буквы заключаются в кавычки, если цифрами, то без кавычек.

*Пример: pol=1, Pol=«М», [пол]=1, [пол]=«М».*

Ввести инструкцию:

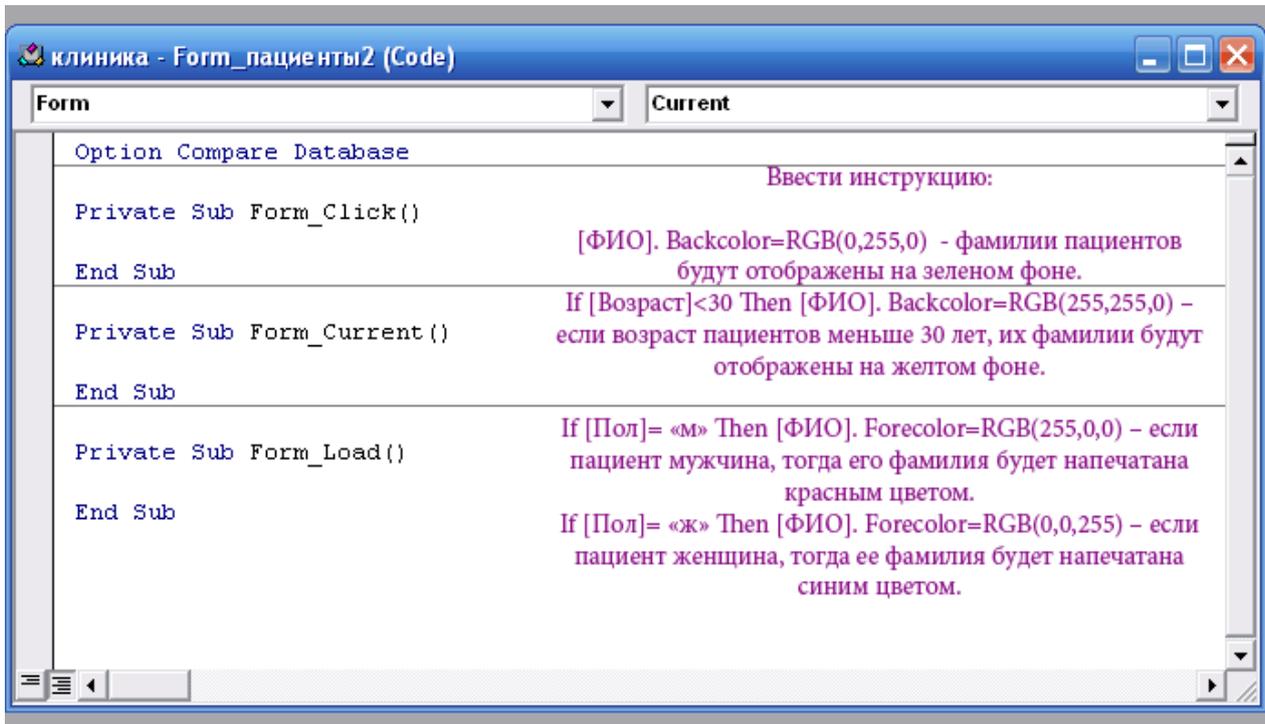


Рисунок 43.

*[ФИО]. Backcolor=RGB(0,255,0) – фамилии пациентов будут отображены на зеленом фоне.*

*If [Возраст]<30 Then [ФИО]. Backcolor=RGB(255,255,0) – если возраст пациентов меньше 30 лет, их фамилии будут отображены на желтом фоне.*

*If [Пол]= «М» Then [ФИО]. Forecolor=RGB(255,0,0) – если пациент мужчина, тогда его фамилия будет напечатана красным цветом.*

*If [Пол]= «ж» Then [ФИО]. Forecolor=RGB(0,0,255) – если пациент женщина, тогда ее фамилия будет напечатана синим цветом.*

3. Для проверки написания нажать на панели инструментов кнопку **Компилировать**. Если проверка прошла нормально, закрыть все окна, сохранив изменения в форме.

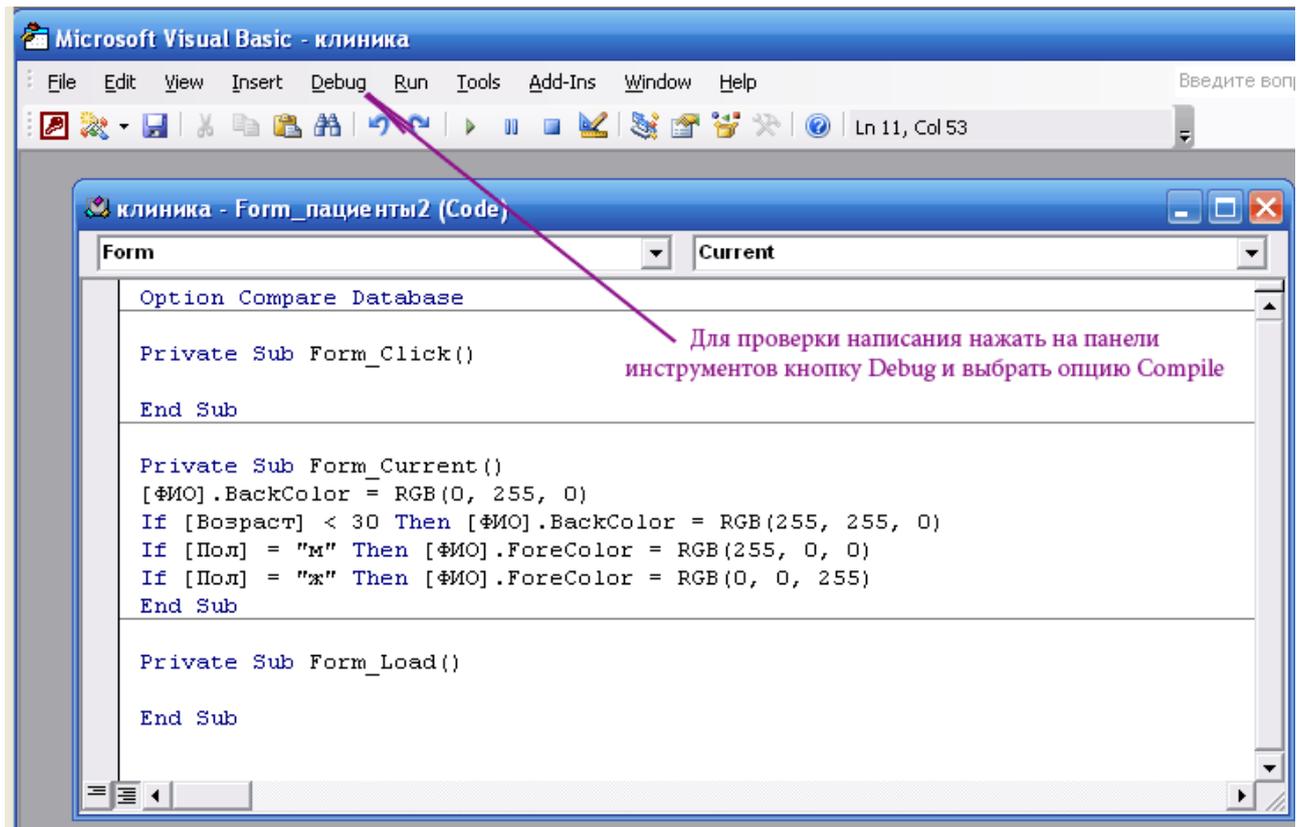


Рисунок 44.

4. Открыть форму и проверить изменение цвета.

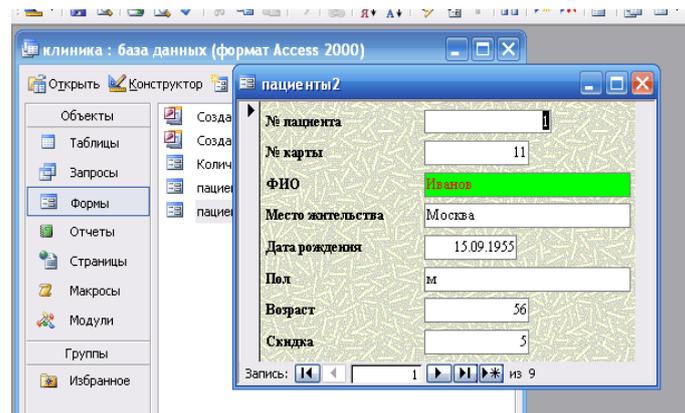


Рисунок 45.

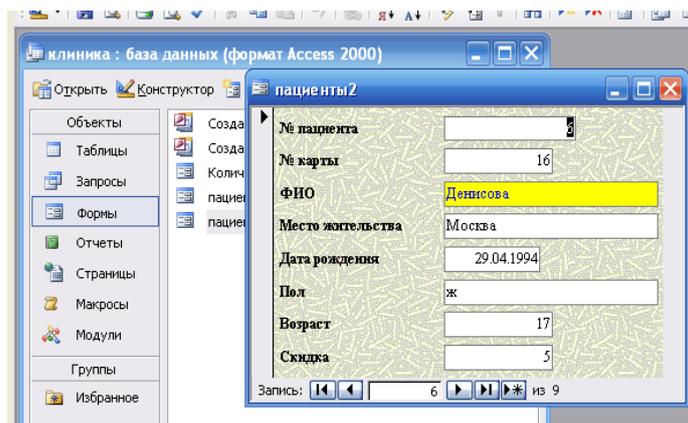


Рисунок 46.

## КОНТРОЛЬНОЕ ЗАДАНИЕ ПО ЗАПРОСАМ

1. Создать групповой запрос, который вычисляет средний возраст для мужчин и для женщин.
2. Создать запрос с параметром, взяв в качестве параметра данные поля **Диагноз**, а также в запросе должны быть выбраны поля: **ФИО, дата рождения, врач**.
3. Создать запрос **Вычисление месяца**, в котором вычисляется номер месяца поступления и выписки пациентов. (Использовать встроенную функцию **Month(...)**, которая работает аналогично функции **Year(...)**, но в качестве результата возвращает номер месяца).
4. Создать запрос на **обновление** данных в поле **Возраст** (учесть, что числовой тип данных в этом поле должен иметь размер поля – целое и число десятичных знаков – 0).

## КОНТРОЛЬНОЕ ЗАДАНИЕ ПО ФОРМАМ

1. Построить диаграмму по запросу **Сумма лекарств**, отформатировать цвет столбцов и название диаграммы.
2. Создать форму вида **в один столбец** по запросу **Сумма лекарств** и встроить в нее кнопку, при нажатии на которую будет открываться диаграмма, построенная по этому запросу.

## КОНТРОЛЬНОЕ ЗАДАНИЕ

1. Создать базу данных **Библиотека.mdb**, состоящую из двух таблиц: **Авторы** и **Учебники**. Основной таблицей является таблица **Авторы**, каждому автору может принадлежать авторство нескольких учебников, приведенных в таблице **Учебники**. Структура таблиц должна предусматривать возможность создания межтабличной связи типа **один-ко-многим**.

Таблица **Авторы** содержит следующие поля:

- Код автора
- Фамилия И О
- Дата рождения
- Телефон
- Пол
- Место работы

Таблица **Учебники**:

- Код учебника
- № автора
- Название
- Число страниц
- Год издания

Определить ключевые поля, обратить внимание на настройку типов и форматов данных ключевых полей и поля связи.

2. Создать межтабличную связь типа **один-ко-многим** с сохранением целостности данных (руководствуясь условием: один автор – несколько учебников).
3. Заполнить обе таблицы данными на 6 авторов. Одному автору принадлежит авторство одного учебника, другому двух, третьему трех.
4. Создать **форму** по таблице **Авторы**. Вставить **Заголовок формы** и отформатировать текст в **области заголовка** и в **области данных**.
5. Создать отчеты по таблицы **Авторы** и **Учебники**. **Отформатировать отчеты**.

# Программирование на VISUAL BASIC for APPLICATIONS (VBA)

## *ВВЕДЕНИЕ*

Язык **Visual Basic for Applications** является одной из версий популярного языка разработки приложений **Visual Basic**.

### *В чём же отличия двух языков?*

- Язык Visual Basic разрабатывался для создания законченных самостоятельных приложений, VBA используется для автоматизации существующих приложений.
- Visual Basic имеет собственную среду программирования, VBA использует среду приложения.
- Для выполнения разработанной программы, созданной в Visual Basic, пользователю не обязательно иметь доступ к самой среде программирования, т.к. разработанные приложения можно скомпилировать в .EXE файлы. VBA – приложения могут выполняться только в том приложении, в котором они были созданы.

### *Общими для VBA и VB являются.*

- Язык программирования (во всех аспектах – типы данных, правила объявления и использования переменных, процедур и функций).
- Во многом похожие средства разработки и отладки программ.
- Аналогичные модели объектно-ориентированного и событийно-управляемого программирования.
- Общие библиотеки объектов, механизмы подключения и использования библиотек.

Знакомство с одним из языков программирования создаёт основу для успешного изучения другого языка, то есть если Вы знакомы с Visual Basic, то легко будете программировать на VBA, верно и обратное утверждение.

## МАКРОСЫ

### Запись макросов

**Макрос** – это записанная последовательность команд и действий пользователя, сохранённая под уникальным именем, которую может выполнить Microsoft Office Excel.

**Код макроса** может служить основой для дальнейших разработок и является учебным материалом, т.к. по коду макроса можно научиться записывать последовательность действий в VBA.

**Имя макроса** может содержать до 255 символов и должно начинаться с буквы. В имя могут входить буквы, цифры и знаки подчёркивания. Пробелы в именах макросов не допускаются.

Рассмотрим процедуру записи простого макроса, который изменяет шрифт и цвет в выделенном диапазоне ячеек.

1. Откройте новую рабочую книгу.
2. В ячейку A1 введите вашу фамилию, в ячейку B1 – имя, в ячейку C1 – номер группы.
3. Перейдите в ячейку A1.
4. Выполните команду Сервис/Макрос/Начать запись. Появится диалоговое окно Запись макроса, показанное на Рисунке 47.

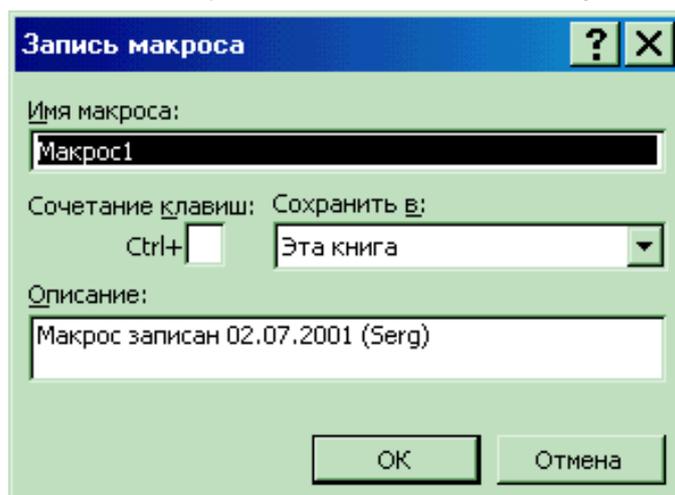


Рисунок 47. Диалоговое окно Запись макроса

5. Введите название макроса МойПервыйМакрос, и нажмите ОК. На экране появится панель инструментов Остановить запись.
6. Выполните команду Формат/Ячейки. В диалоговом окне Формат ячеек перейдите на вкладку Шрифт.
7. Установите размер шрифта 18 пунктов, цвет – красный. Щёлкните на кнопке ОК.

- Щёлкните на кнопке Остановить запись одноимённой панели инструментов.

### Выполнение макросов

При выполнении макроса Excel повторяет те же действия, которые вы выполняли в процессе его создания. Для выполнения макроса нужно проделать следующую последовательность действий:

- Выделите ячейку В1.
- Выполните команду Сервис/Макрос/Макросы. Появится диалоговое окно Макрос.

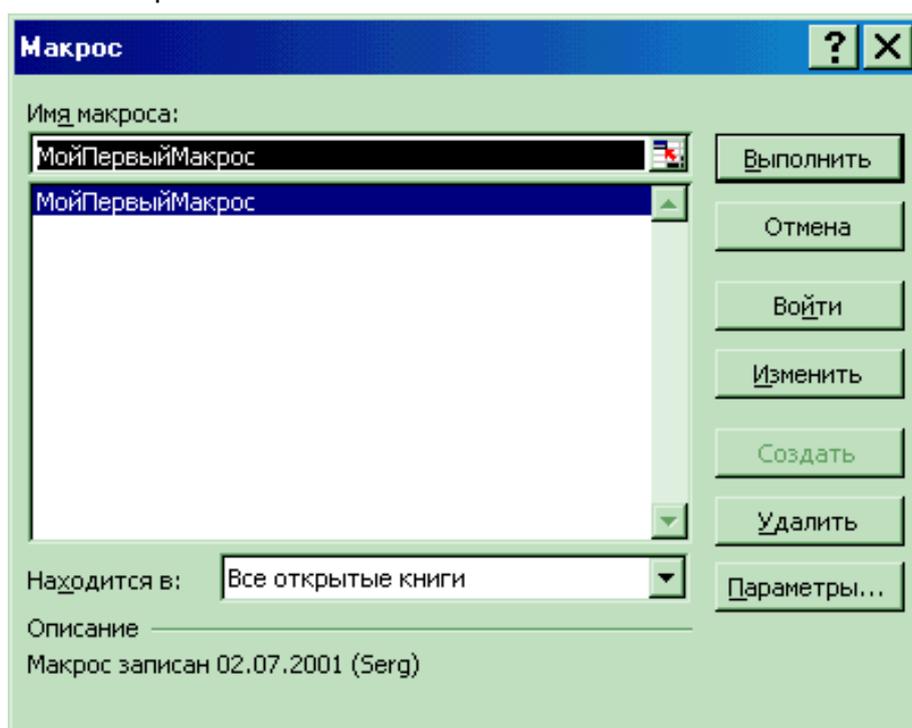


Рисунок 48. Диалоговое окно Макрос

- Выделите макрос МойПервыйМакрос и щёлкните на кнопке Выполнить. Шрифт в ячейке В1 станет красным и размером 18 пунктов.
- Проделайте ту же последовательность действий для ячейки С1.

### Просмотр кода макроса

По ходу записи макроса Excel запоминает действия пользователя и одновременно преобразует их в код VBA. Чтобы увидеть полученный код, выполните следующие действия:

- Выберите команду Сервис/Макрос/Макросы. Отобразится диалоговое окно Макрос.

2. Выделите макрос МойПервыйМакрос и щёлкните на кнопке Изменить. Откроется окно редактора Microsoft Visual Basic, показанное на Рисунк 49.

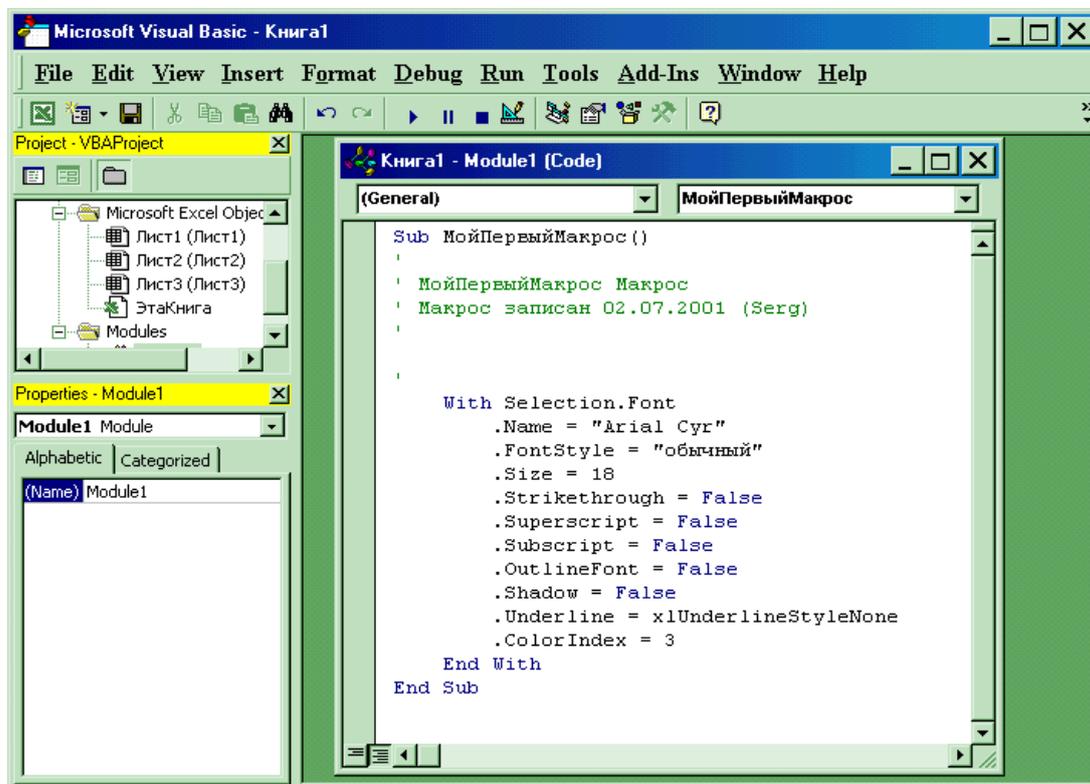


Рисунок 49. Окно редактора Microsoft Visual Basic

Первая строка макроса Sub МойПервыйМакрос () – представляет собой начало процедуры выполнения макроса и его имя. Следующие строчки, вначале которых стоит одиночная кавычка, являются комментариями, поясняют отдельные строки макроса.

### Редактирование кода макроса

Использовать возможность редактирования макроса пользователь может в том случае, когда возникает необходимость изменить какие-либо параметры макроса или возникли ошибки при записи макроса и пользователь хочет изменить код макроса

Редактировать код можно в редакторе Visual Basic. Попробуем изменить код макроса:

1. Выделите строку, которая начинается с .Name.
2. Удалите эту строку, нажав клавишу Delete.
3. Продолжайте удалять строки, пока не останется следующий фрагмент программы.

```
Sub МойПервыйМакрос()  
,  
' МойПервыйМакрос Макрос  
' Макрос записан 02.07.2001 (Serg)  
,  
  
With Selection.Font  
    .Size = 18  
    .ColorIndex = 3  
End With  
End Sub
```

4. Закройте окно редактора Visual Basic и вернитесь в свою рабочую книгу.
5. В ячейке E1 введите слово Тест.
6. Не снимая выделения с ячейки E1, выполните макрос МойПервыйМакрос, Вы видите, что макрос работает точно также, как и до удаления строк кода.
7. Выполните команду Сервис/Макрос/Макросы.
8. В диалоговом окне Макрос выделите макрос МойПервыйМакрос и щёлкните на кнопке Изменить.
9. Измените размер шрифта с 18 пунктов на 24. Код макроса должен выглядеть следующим образом:

```
Sub МойПервыйМакрос()  
,  
' МойПервыйМакрос Макрос  
' Макрос записан 02.07.2001 (Serg)  
,  
  
With Selection.Font  
    .Size = 24  
    .ColorIndex = 3  
End With  
End Sub
```

10. Закройте окно редактора Visual Basic и вернитесь в рабочую книгу.
11. Выделите ячейку A1 и выполните макрос МойПервыйМакрос. Вы заметите, что шрифт в ячейке A1 увеличился.

**Что нельзя сделать с помощью макросов?**

1. Ввести какую-либо информацию во время выполнения макроса.
2. Отобразить диалоговые окна Excel, например, Сохранить документ.

### Удаление макросов

1. Выполните команду Сервис/Макрос/Макросы. Появится диалоговое окно Макрос.
2. Выделите макрос, подлежащий удалению, и щёлкните на кнопке Удалить.
3. Появится окно, требующее подтверждения на удаление макроса.

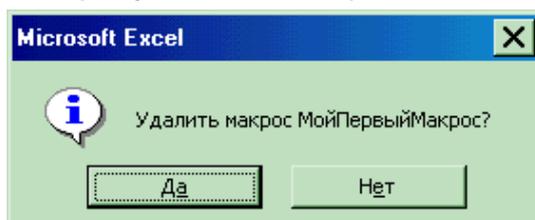


Рисунок 50. Окно Excel запрашивающее подтверждение удаления макроса

4. Щёлкните на кнопке Да, и макрос будет удалён.

### Назначение макросов командным кнопкам

Пользователь, который занимается разработкой приложений, должен быть заинтересован в автоматизации задач, и находить простые пути реализации намеченной цели. Самый доступный способ – использование командных кнопок непосредственно на листе рабочей книги Excel. Вставить командную кнопку можно с помощью панели инструментов Формы. Затем можно назначить ей любой макрос. После выполнения этих действий щелчок на командной кнопке будет вызывать на выполнение назначенный макрос.

Давайте приступим к созданию макроса, создадим командную кнопку и с её помощью выполним макрос. Для этого выполним следующие действия:

1. Выполним команду Вид/Панели инструментов/Формы. Появится панель, как на Рисунок 51.

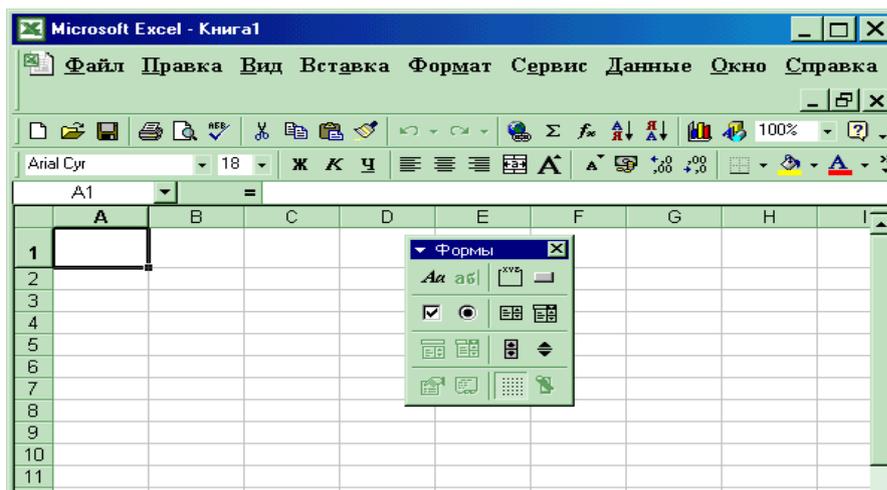


Рисунок 51. Панель инструментов Формы

2. Щёлкните на инструменте Кнопка панели инструментов Формы. Указатель мыши примет форму небольшого чёрного перекрестия.
3. Поместите указатель мыши в то место, где хотите разместить командную кнопку, нажмите и не отпускайте левую кнопку мыши. Перетащите указатель мыши для создания прямоугольника. Придав командной кнопке нужный размер, отпустите кнопку мыши. После появления на рабочем листе командной кнопки одновременно откроется диалоговое окно Назначить макрос объекту.
4. Выберите макрос МойПервыйМакрос и щёлкните на кнопке ОК. Этот макрос будет назначен созданной кнопке.
5. Щёлкните и выделите надпись Кнопка1 на командной кнопке. Нажав клавишу Delete и удалив надпись, введите текст новой надписи, например, РазмерШрифта.

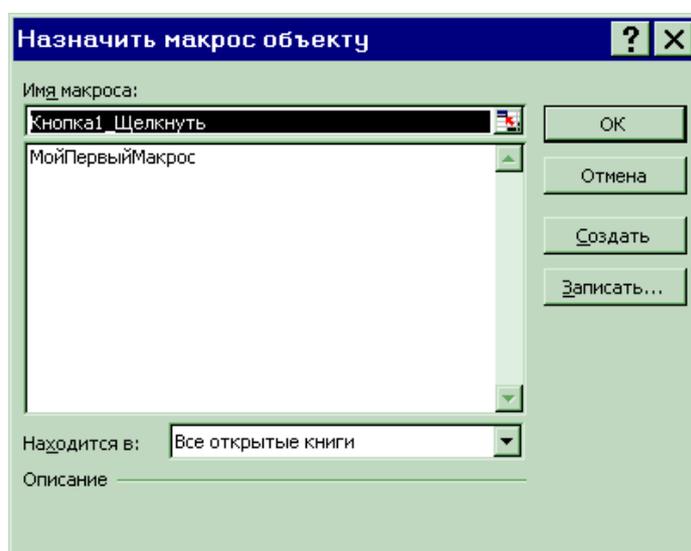


Рисунок 52. Окно Назначить макрос объекту

6. Установив указатель мыши на один из маркеров, которые окружают командную кнопку, Вы можете изменить размеры созданной кнопки.
7. Щёлкните вне кнопки, чтобы снять с неё выделение.
8. В ячейку A5 введите номер своей группы и нажмите Enter.
9. Щёлкните на ячейке A5, а затем на созданной командной кнопке. Выполнится макрос МойПервыйМакрос.

### Назначение макроса графическим изображениям

Макрос, назначенный графическим объектам, будет выполняться более эффективно. Для назначения макроса графическому изображению выполните следующие действия:

1. Выделите ячейку рабочего листа, где будет вставлено изображение.
2. Выполните команду Вставка/Рисунок/Картинки.
3. Выберите изображение для вставки в рабочий лист.

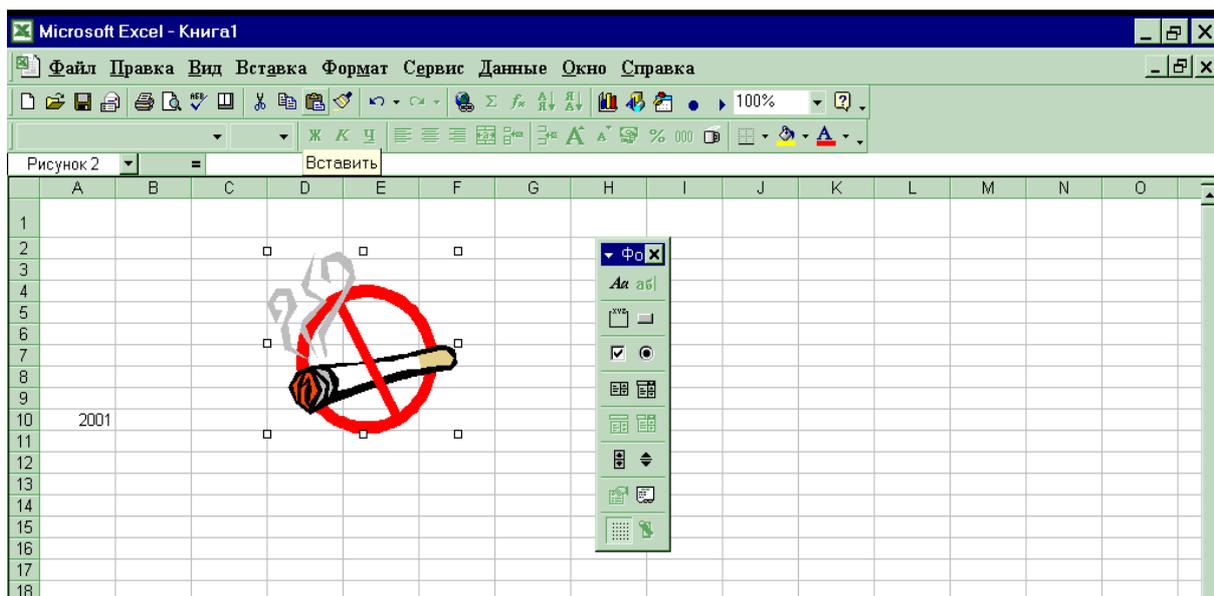


Рисунок 53. Вставка графического объекта

4. После вставки изображения измените его размеры, чтобы они составляли 5х5 см.
5. Щёлкните правой кнопкой мыши на изображении и в открывшемся меню выберите команду Назначить макрос. Отобразится окно, представленное на Рисунке 54.

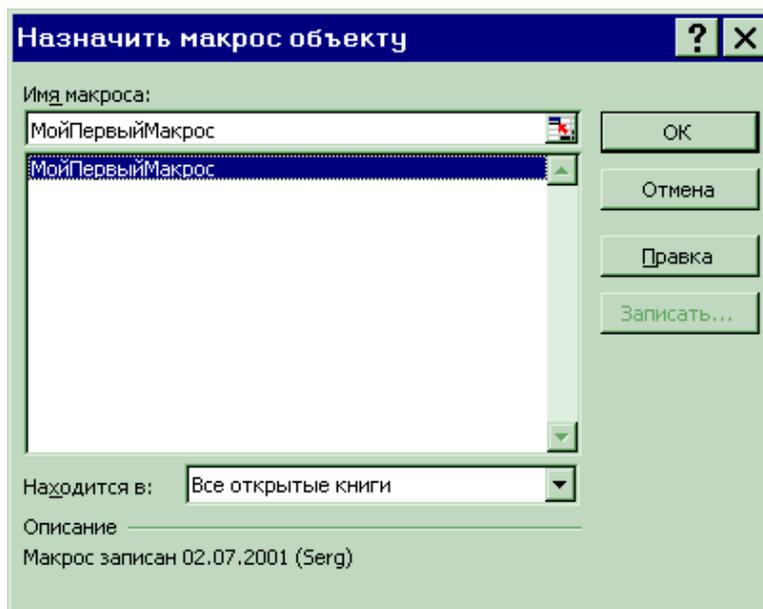


Рисунок 54. Диалоговое окно Назначить макрос объекту

6. Выберите макрос МойПервыйМакрос и щёлкните на кнопке ОК.
7. Щёлкните вне графического изображения, чтобы снять с него выделение
8. Введите число 2001 в ячейку A10.
9. Выделите ячейку A10 и щёлкните на графическом изображении. Выполнится макрос МойПервыйМакрос.

### Назначение макросов кнопкам панелей инструментов

Excel позволяет изменять панели инструментов путём добавления на них новых кнопок. Новой кнопке можно назначить макрос. Чтобы воспользоваться этой возможностью, выполните следующую последовательность действий.

Выполните команду Сервис/Настройка. Появится диалоговое окно Настройка.

1. Перейдите на вкладку Команды (Рисунок 55).
2. В списке Категории выберите Макросы.
3. В списке Команды выберите команду Настраиваемая кнопка. Перетащите команду Настраиваемая кнопка на какую-либо панель инструментов.
4. Щёлкните правой кнопкой мыши на вновь добавленной кнопке. Появится контекстное меню (Рисунок 56).

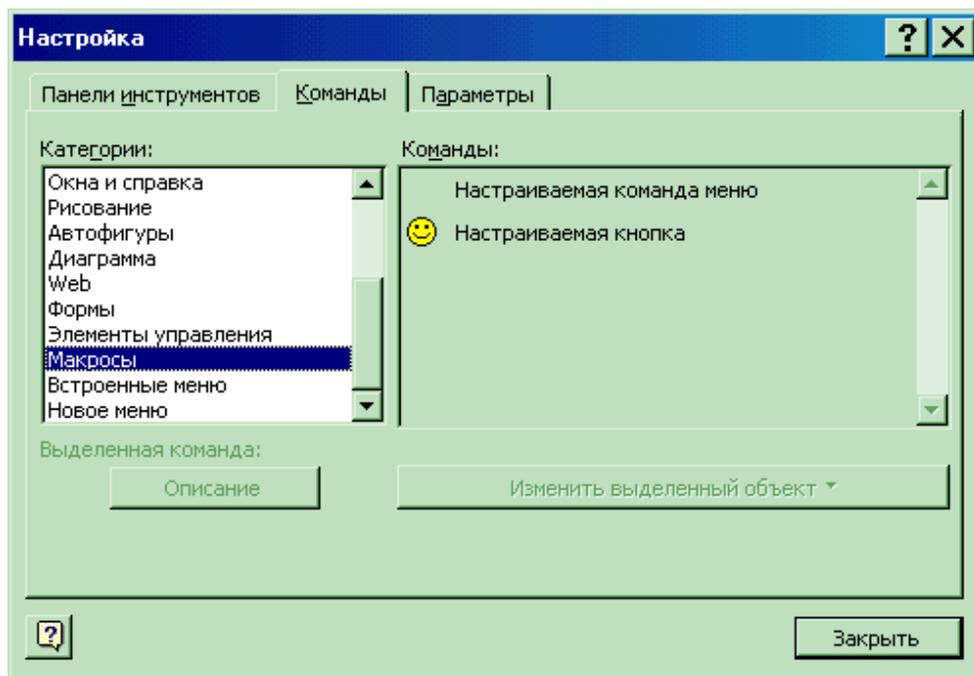


Рисунок 55. Диалоговое окно для добавления кнопок на панели инструментов

5. В контекстном меню кнопки выберите пункт Выбрать значок для кнопки и в открывшемся списке значков выберите любой значок.
6. Снова щёлкните правой кнопкой мыши на новой кнопке и выполните команду Назначить макрос. Откроется диалоговое окно.
7. Выберите макрос МойПервыйМакрос и щёлкните ОК
8. Щёлкните на кнопке Заккрыть диалогового окна Настройка
9. В ячейку B10 введите число 2001.
10. Выделите ячейку B10 и щёлкните на созданной кнопке. Должен выполниться макрос МойПервыйМакрос.

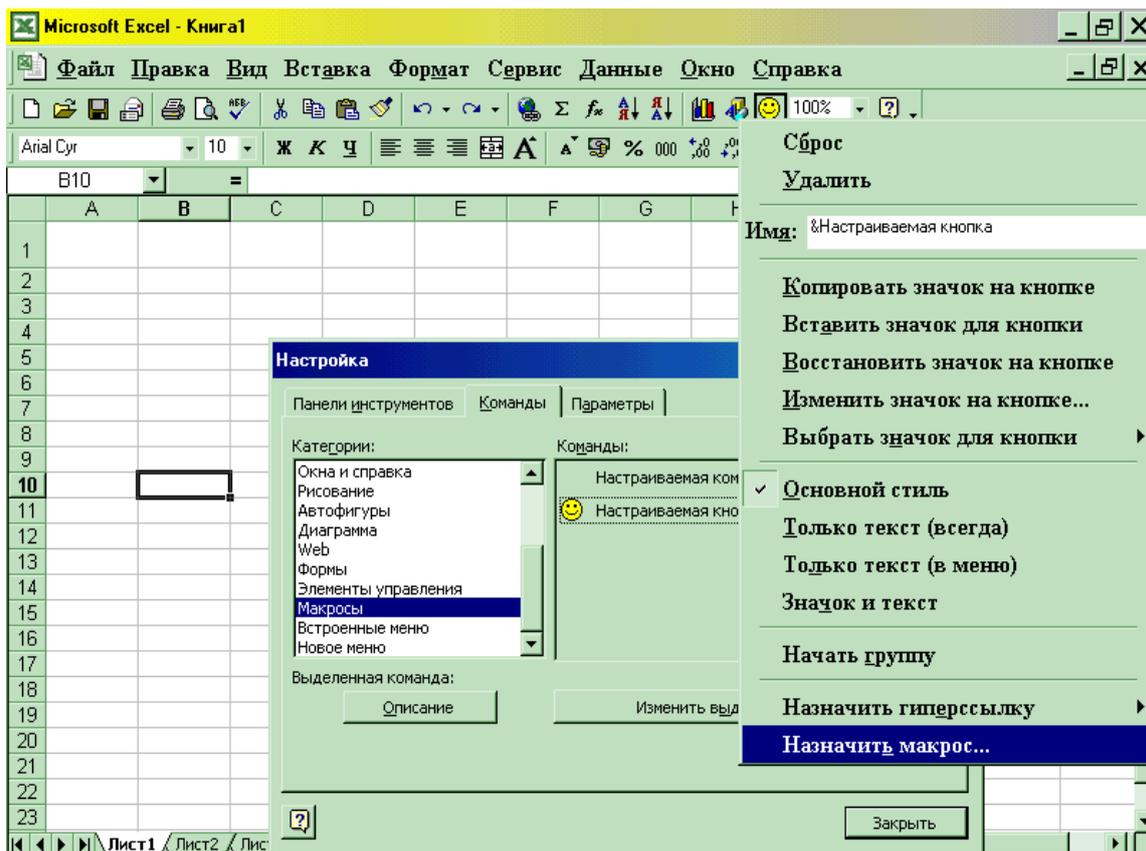


Рисунок 56. Контекстное меню при назначении макроса командной кнопке

## **ВВЕДЕНИЕ В ПРОЦЕСС РАЗРАБОТКИ ПРИЛОЖЕНИЙ**

Если Вы собираетесь заниматься проектированием приложения, то, наверное, представляете не только, как оно будет выглядеть (внешний вид интерфейса), но и степень сложности работы с Вашей программой, учитывая то обстоятельство, что с ней может работать и новичок, и опытный пользователь.

Вы должны представлять, как и откуда будут поступать данные, а также, где и как будут храниться данные, полученные с помощью разработанного приложения, в каком виде Вы собираетесь выводить полученные результаты. Эти вопросы не поставят Вас в тупик, если владеть тем инструментарием, который предоставляет пользователю Excel.

Если Вы хотите, чтобы с Вашим приложением с удовольствием работали и новичок, и профессионал, то Вам следует воспользоваться Экранными формами или как их ещё называют – пользовательскими формами (от английского – UserForm). Созданием на форме объектов управления и установкой значений свойств этих объектов (о чём будет рассказано позже), пользователь создаёт себе

условия для работы с будущим приложением как с обычным диалоговым окном. Прделайте следующие действия по вставке Экранной формы:

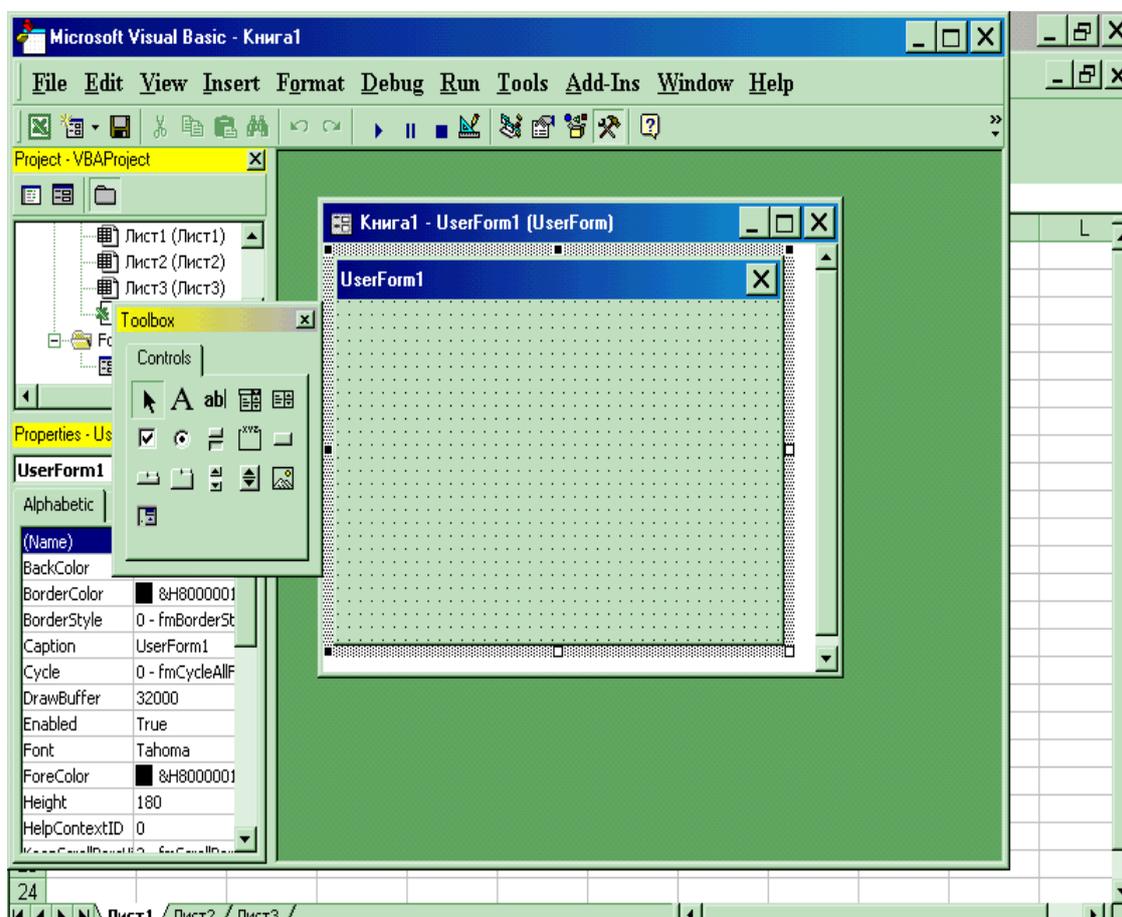


Рисунок 57. Экранная форма для разрабатываемого приложения

### Типы элементов управления

При создании экранной формы автоматически отображается панель инструментов Панель элементов (ToolBox). В Экранную форму можно вставить большое число разнообразных типов элементов управления. Элемент Надпись (Label) – это статическая область текста, обычно содержащая какую – либо поясняющую информацию, например, к элементам интерфейса.

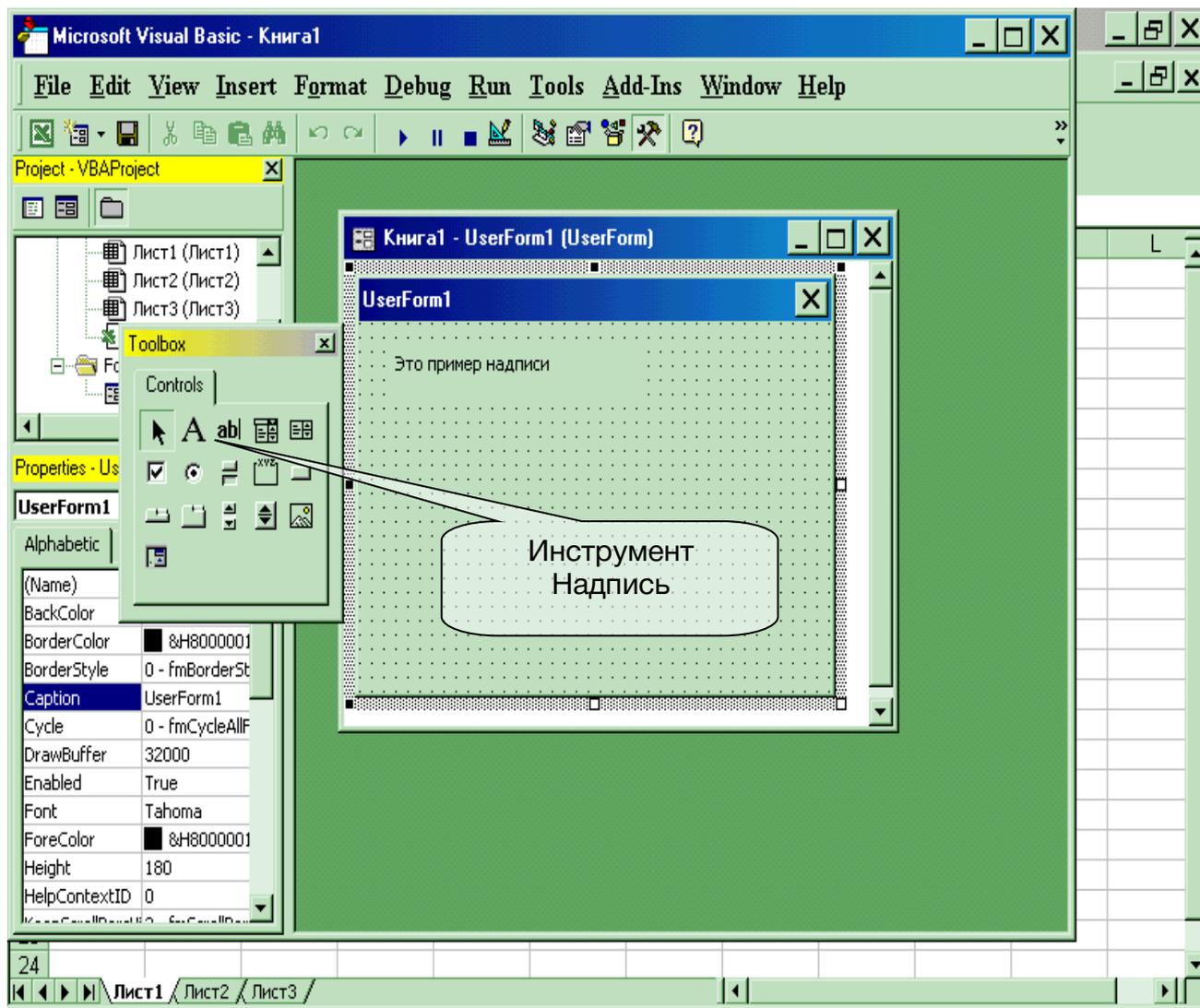


Рисунок 58. Надпись – статическая область текста

Следующий инструмент, присутствующий на Панели элементов называется Текстовое поле (TextBox). В окнах, которые будут созданы с помощью этого инструмента на форме, Вы будете вводить данные и получать результаты (Рисунок 59).

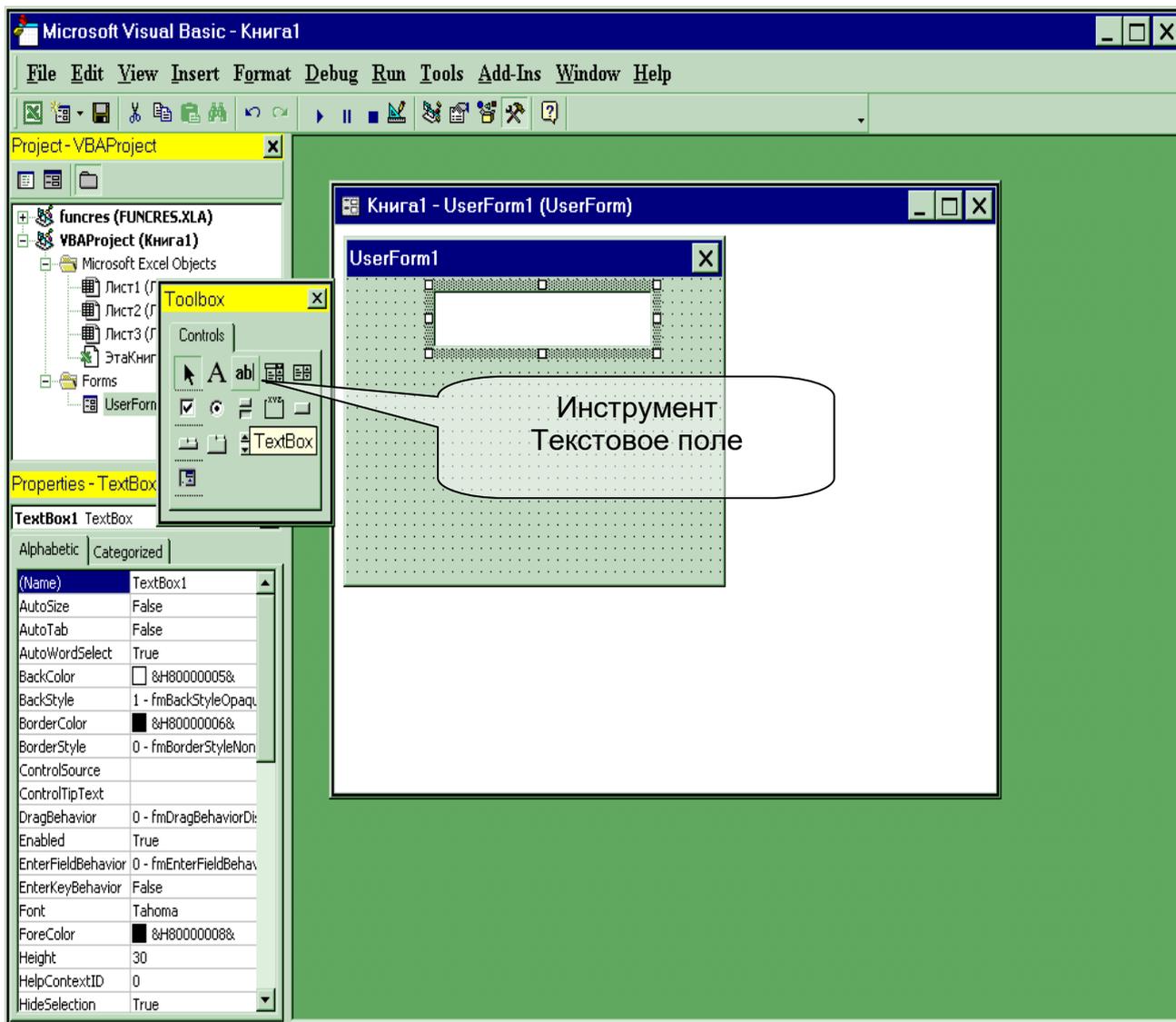


Рисунок 59. Текстовое поле – залог успешного ввода и вывода данных

С помощью инструмента Кнопка (CommandButton) на форме можно создать командную кнопку. Этот элемент управления знаком каждому пользователю, т.к. в Windows на каждом шагу мы встречаемся с кнопками ОК, Отмена, Да, Нет.

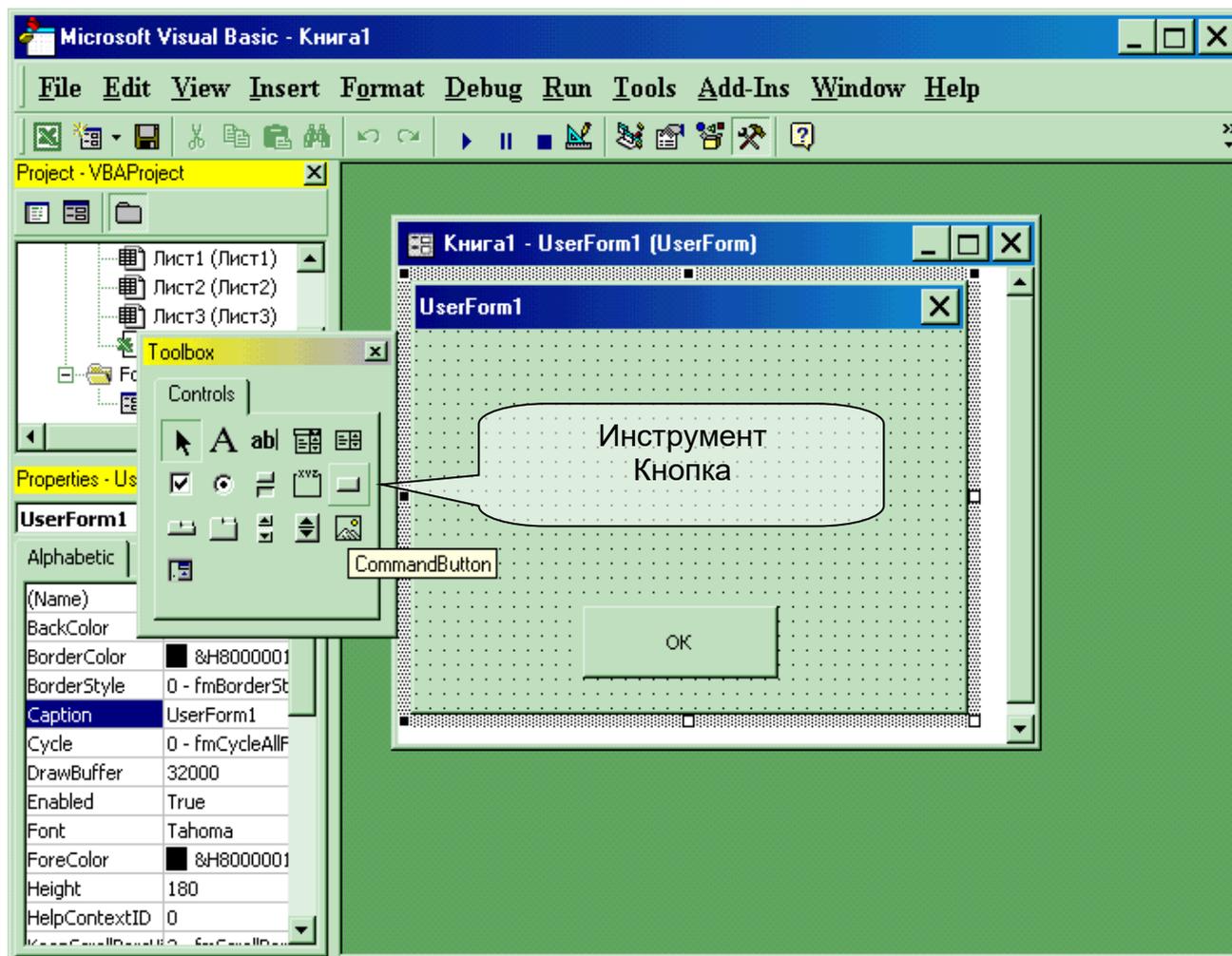


Рисунок 60. Создание командной кнопки

Следующие два элемента управления, флажок (CheckBox) и переключатель (OptionButton) (Рисунки 61, 62), используются в ситуациях, когда пользователю надо сделать выбор из нескольких возможностей. Эти элементы имеют два устойчивых состояния: включено и выключено. Если флажки организованы в группу, то пользователь может установить в этой группе один или несколько флажков, а то и все. В отличие от флажков, в группе переключателей можно выбрать только один.

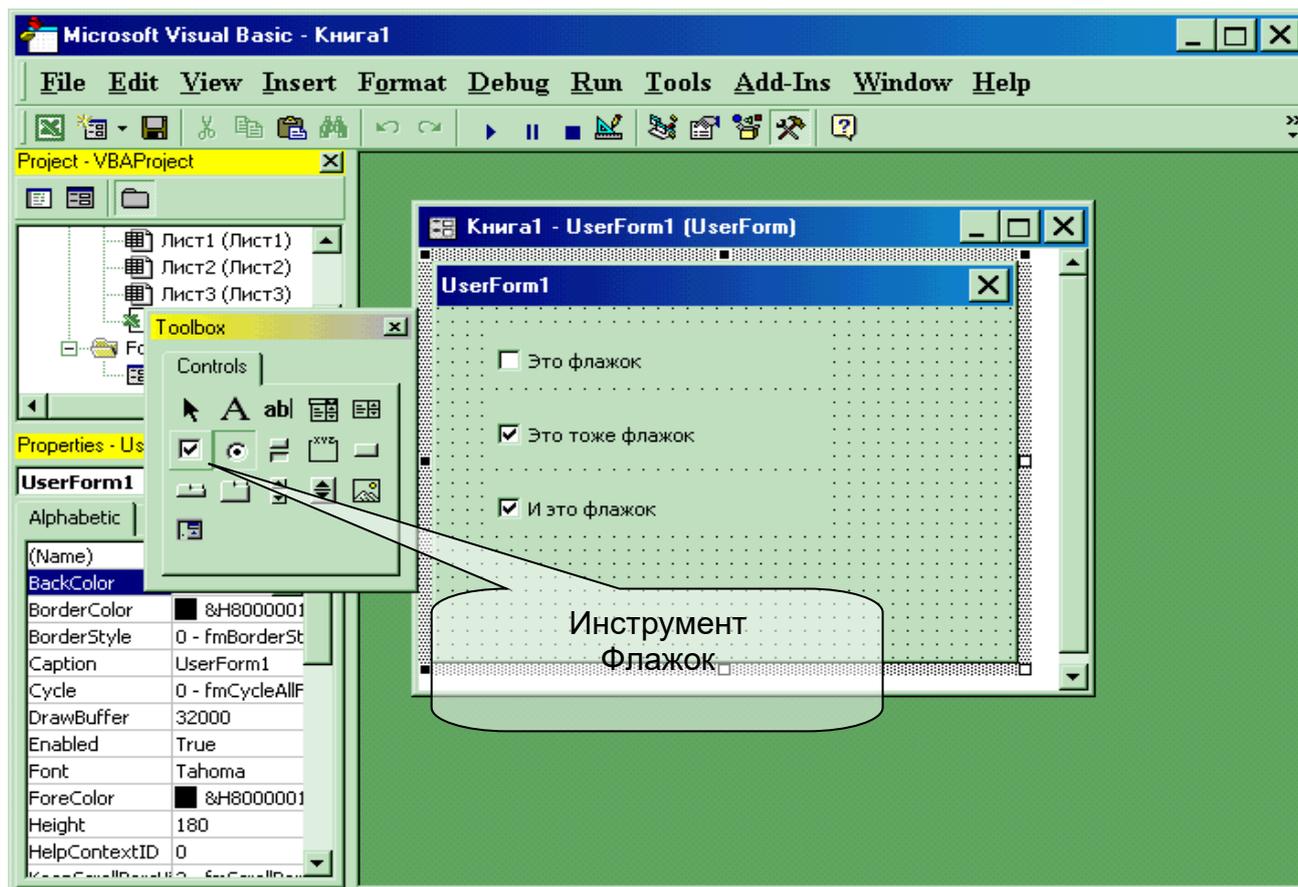


Рисунок 61. Окно формы с созданными флажками

Ещё два элемента управления, которые можно поместить на форму, – полосы прокрутки (ScrollBar) и счётчик (SpinButton). Полосы прокрутки как элементы управления – это не совсем то, чем мы пользуемся для прокрутки рабочего листа, в данном случае полосы прокрутки применяются в качестве механизмов выбора, например, в графических программах с их помощью можно сделать изображение более светлым или более тёмным.

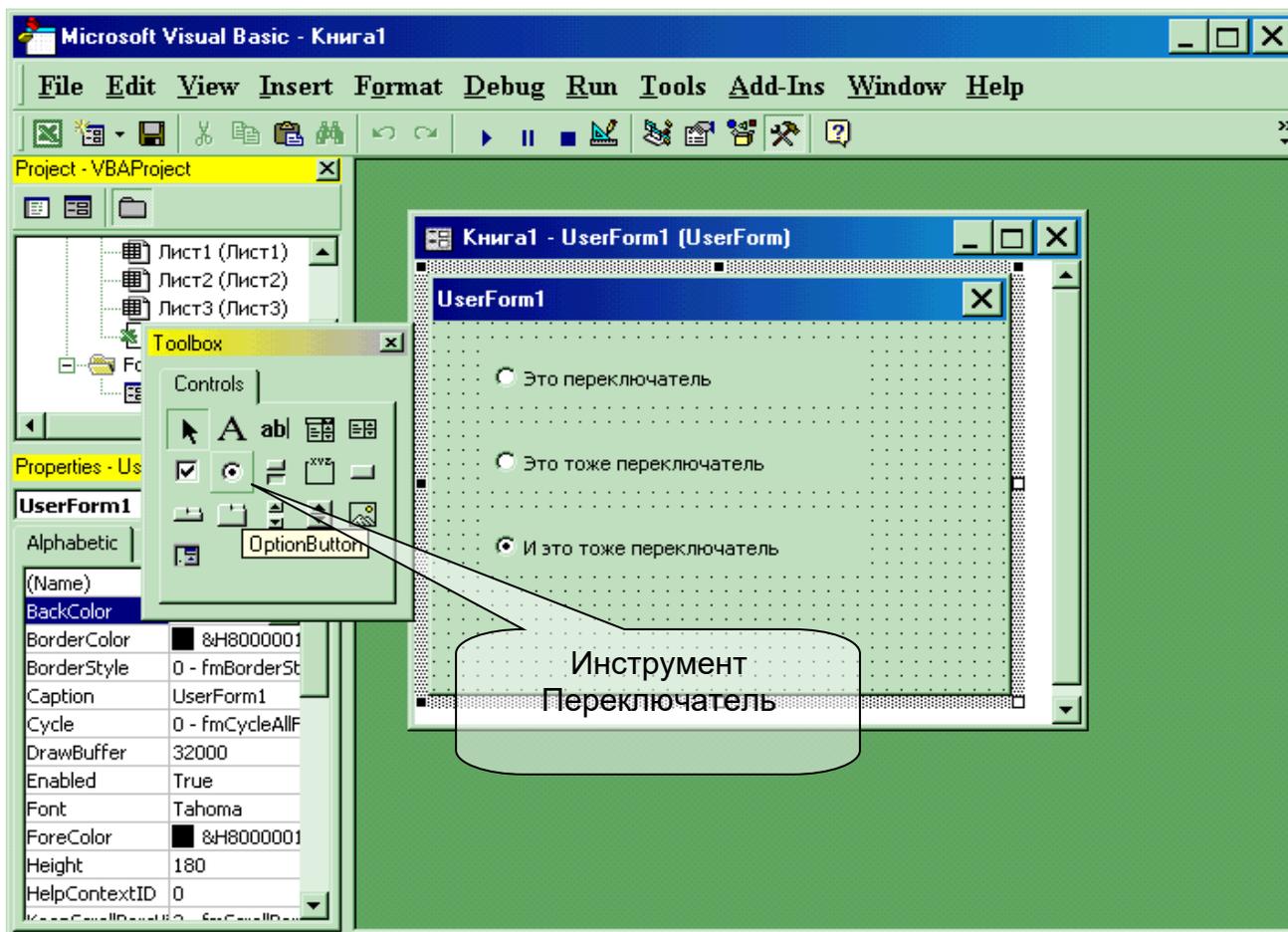


Рисунок 62. Окно формы с созданными переключателями

Элементы управления можно вставлять не только в экранную форму, но и в рабочие листы.

### Вставка элементов управления в рабочий лист

Процесс вставки элемента в рабочий лист почти ничем не отличается от рисования прямоугольника в программе Paint и заключается в следующем:

1. Выполните команду Вид/Панели инструментов/Формы.
2. На панели элементов выберите элемент любой управления.
3. Поместите указатель мыши в то место рабочего, где должен находиться верхний левый угол элемента управления.
4. Нажмите левую кнопку мыши и протащите указатель, рисуя прямоугольник. Отпустите кнопку мыши.

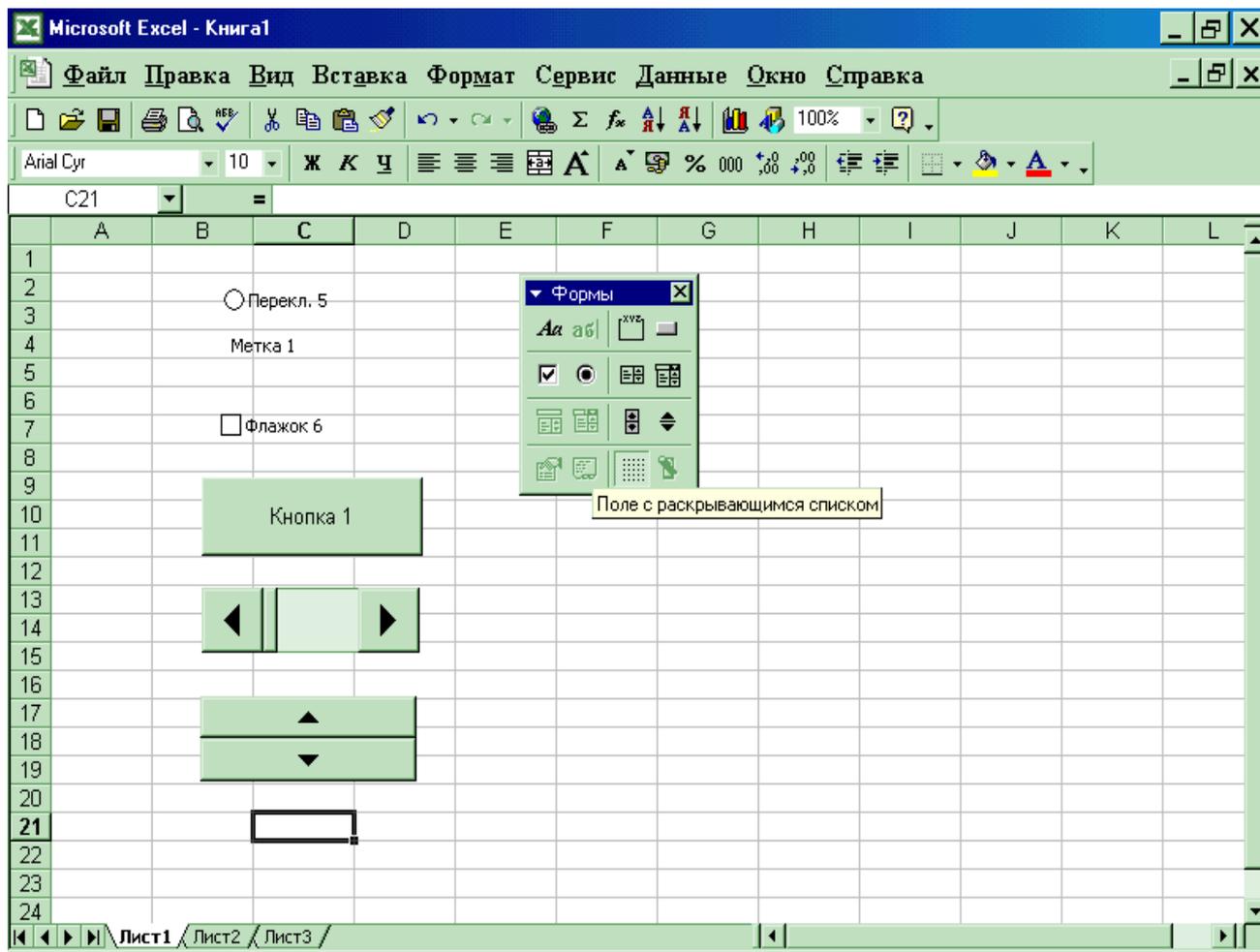


Рисунок 63. Окно Excel с элементами управления

### Как осуществляется связь элемента управления с рабочим листом?

Элементам управления можно назначить макросы, но можно и использовать их для ввода значений в рабочий лист. Выполните следующие действия:

1. Поместите на рабочий лист флажок.
2. Выделите его щелчком мыши.
3. Щёлкните на нём правой кнопкой мыши и в контекстном меню выберите команду Формат объекта. Откроется диалоговое окно Формат элемента управления (Рисунок 64).

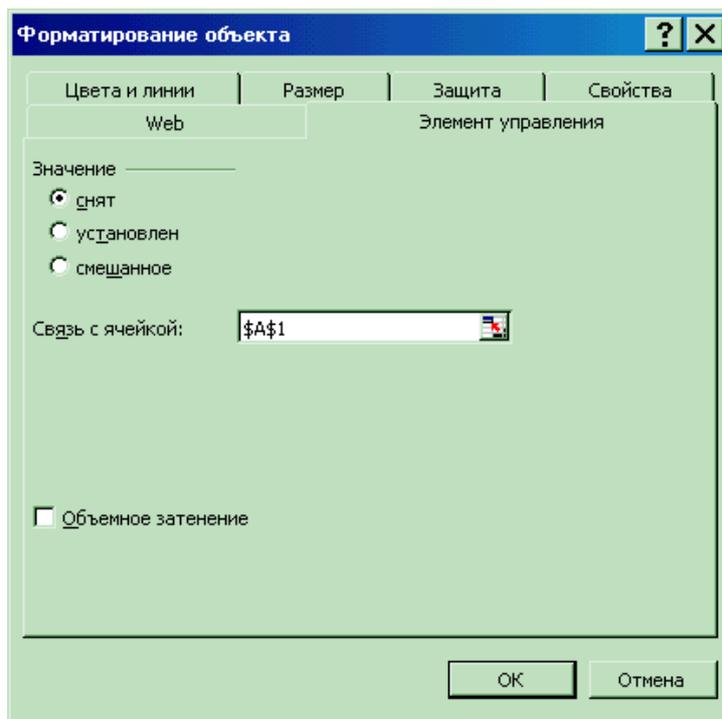


Рисунок 64. Вкладка Элемент управления

4. Перейдите на вкладку Элемент управления.
5. Щёлкнув в поле Связь с ячейкой и установив там курсор, щёлкните в ячейке A1, а затем на ОК.
6. Щёлкните вне области флажка для снятия с него выделения.
7. Установите флажок, щёлкнув на элементе Флажок. В ячейке A1 появится слово ИСТИНА.
8. Сняв флажок, можно увидеть, что в ячейке A1 появилось слово ЛОЖЬ.

### Поработаем с инструментом Полоса прокрутки

1. Создайте на рабочем листе полосу прокрутки.
2. Щёлкните на ней правой кнопкой мыши и в контекстном меню выберите команду Формат объекта. Откроется диалоговое окно Формат элемента управления (Рисунок 65).

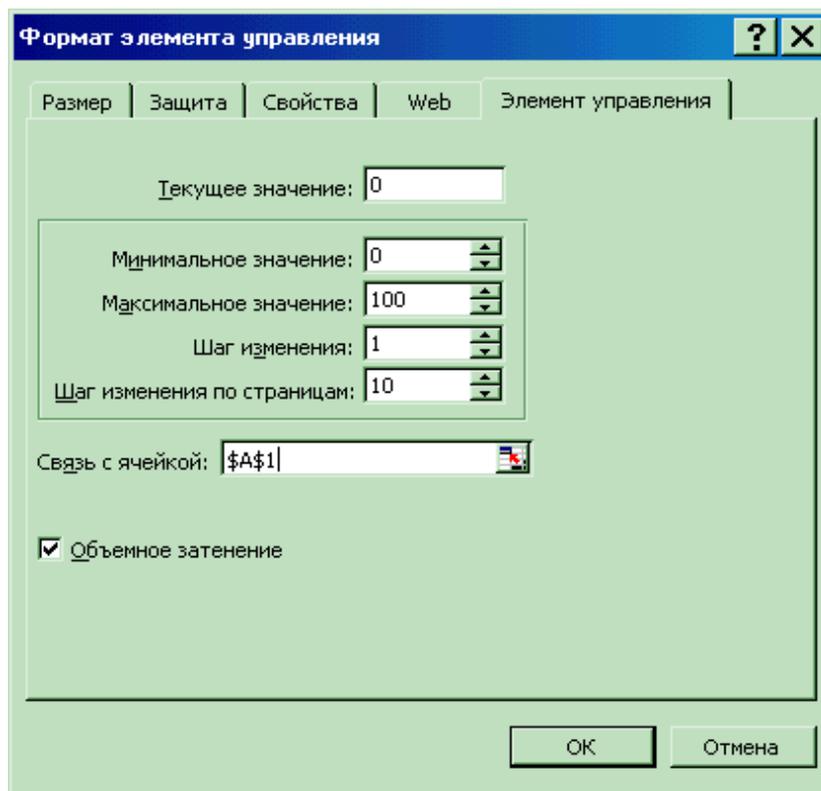


Рисунок 65. Вкладка Элемент управления

3. Перейдите на вкладку Элемент управления.
4. Щёлкнув в поле Связь с ячейкой и установив там курсор, щёлкните в ячейке A1, а затем на ОК.
5. Щёлкните на полосе прокрутки на кнопке со стрелкой, направленной вправо. Значение в ячейке A1 увеличится на единицу. Продолжая щёлкать на этой кнопке, Вы увеличиваете значение в ячейке A3.

## ПРОГРАММИРОВАНИЕ НА VBA

Выше было сказано, что процесс создания экранной формы, называется *разработкой интерфейса*. Под **интерфейсом** понимают совокупность средств, обеспечивающих взаимодействие пользователя и программ вычислительной системы. При разработке интерфейса пользователь должен быть заинтересован в максимально простом и удобном способе ввода и вывода данных. Важна и эстетическая форма подачи программы: это и размер формы, её фон, шрифт надписей, звуковое и графическое сопровождение. Также мы говорили о том, что на экранной форме пользователь создаёт элементы управления: текстовые поля, метки, командные кнопки. Эти элементы являются объектами языка Visual Basic for Applications. **Объект** – одно из основных понятий не только VBA, но и других **объектно-**

**ориентированных языков программирования.** Каждый объект имеет своё имя и является представителем некоторого класса подобных себе объектов. Под классом объектов подразумевается общее описание таких объектов, для которых характерно наличие множества общих свойств и общих действий, которые способны выполнять эти объекты. Существует, например, класс Командная кнопка – общее описание кнопок в окнах приложений. Кнопки могут быть разных размеров и цветов, иметь разные подписи, но имеют множество общих свойств и событий (например, щелчок мыши по кнопке). Началом работы алгоритма, а, следовательно, и программы, является какое-нибудь событие. **Событием** (Event) называется характеристика класса объектов, описывающая внешнее воздействие, на которое реагирует объект этого класса во время работы приложения. Объектно-ориентированное программирование основано на следующей идее: программы управляются событиями. Пояснить это можно так: после проектирования экранной формы устанавливаются события, и только затем программируются действия, связанные с этими событиями.

Код программы, написанной пользователем на VBA, хранится в модулях. **Модуль** – это совокупность объявлений и процедур, хранящихся как единое целое. Рассмотрим термин “процедура”.

Если в программе возникает необходимость частого обращения к некоторой группе операторов, то рационально сгруппировать такую группу в самостоятельный блок, к которому можно обращаться, указывая его имя. Такие программные блоки называются подпрограммами пользователя. Подпрограммы в языках программирования реализуются посредством процедур и функций.

**Процедура** – это независимая часть программы, которую можно вызвать по имени для выполнения определённых действий. Процедура выполняет один или несколько операторов и не возвращает значения. Примером процедуры является код макроса.

Например,

```
Sub МойПервыйМакрос_Click()  
  With Selection.Font  
    .Size = 18  
    .ColorIndex = 3  
  End With  
End Sub
```

Процедура обозначается словом Sub, End Sub говорит о том, что процедура закончена. Процедура выполняется в ответ на какое-либо событие или говорят, что

процедура обрабатывает событие. Имя процедуры состоит из названия объекта МойПервыйМакрос, и события, Click (щелчок), которое отделяется знаком подчёркивания от названия объекта.

**Функция** аналогична процедуре, но отличается тем, что возвращает в точку вызова основной программы единственный результат. Например,  $Y = ABS(X)$ ; – определяет модуль числа  $X$  и возвращает результат в точку вызова  $Y$ .

VBA можно использовать для создания собственных функций. Например, можно определить функцию Процент (число), которая будет определять процент от числа.

```
Public Function Процент (число)
    Процент = число*0.01/100
End Function
```

Функция может иметь аргумент Число. Аргумент Число может быть числом или ссылкой на ячейку рабочего листа. Величина процента является тем значением, которое возвращает функция.

В VBA существуют определённые правила для имён процедур и функций:

- Первые три символа имени должны быть буквами.
- Имя может содержать буквы, цифры и знаки подчёркивания.
- Имя не может содержать пробелы, точки, запятые, восклицательные знаки и символы @, &, \$, #
- Имя не должно иметь более 255 символов.

### Объявление типа переменной

Переменные 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 17 содержится информация о размере данных, т. е. об объёме памяти, выделяемом для хранения данных.

## Типы данных в VBA

Таблица 17

<i>Тип данных</i>	<i>Размер в байтах</i>	<i>Диапазон значений</i>
Byte	1	0 ... 255
Integer	2	-32768 ... +32767
Long	4	-2 147 483 648 ... +2 147 483 647
Boolean	2	True, False
String	10 + 1 байт на символ	0 ... 2 млрд. символов
Variant	16	-1.797693E308 ... -4.9406564E-324 для отрицательных чисел; 4.9406564E-324 ... 1.79769313E308 для положительных чисел

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

Dim ИмяПеременной As ТипДанных

### Общие правила написания имени переменной

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

Можно не указывать тип данных при написании своих процедур и тогда по умолчанию VBA применит для Ваших переменных тип Variant. Но, просмотрев таблицу 1 можно увидеть, что ни один из типов данных не требует 16 байт для сохранения значений переменных. Неплохо бы экономить ресурсы памяти компьютера. Кроме того, не объявляя тип переменных вы рискуете получить результат выполнения Вашей программы, который будет далёк от ожидаемого.

### Описание констант

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

Const ИмяКонстанты As ТипДанных = Значение

Например,  
Const Gruppa As Integer = 25

## ПРИМЕНЕНИЕ УСЛОВНЫХ ОПЕРАТОРОВ

До сих пор мы учились вводить данные и присваивать значения переменным. Теперь надо научиться организовывать различные потоки выполнения приложения в зависимости от ситуации, которая складывается в ходе работы программы. Не раз и не два Вы видели в окнах сообщений кнопки Да и Нет, Ok и Cancel, и знаете, что дальнейший ход программы будет зависеть от того, на какой кнопке щёлкнет пользователь.

### Разветвляющийся алгоритм

Алгоритм называется разветвляющимся, если последовательность выполнения шагов алгоритма изменяется в зависимости от выполнения некоторых условий. Условие – это логическое выражение, которое может принимать одно из двух значений: “ДА” – если условие верно (истинно), и “НЕТ” – если условие неверно (ложно).

### Простой условный оператор

Синтаксис простого условного оператора, если Вы пишете программу структурно, следующий:

```
If <логическое выражение> Then  
P1  
Else  
P2  
End If
```

Возможна и другая запись:

```
If <логическое выражение> Then P1 Else P2
```

где If, Then, Else, End If – зарезервированные слова, а P<sub>1</sub>, P<sub>2</sub> – операторы.

Простой условный оператор работает по следующему алгоритму:

Сначала вычисляется логическое выражение. Если результат есть TRUE (истина), то выполняется оператор  $P_1$ , а оператор  $P_2$  пропускается. Если результат есть FALSE (ложь), то выполняется оператор  $P_2$ , а оператор  $P_1$  пропускается.

### Сокращённый условный оператор

Если необходимо выполнить некоторое действие только при истинности проверяемого условия, то в таком случае применяется сокращённый условный оператор.

Синтаксис сокращённого условного оператора, если Вы пишете программу структурно, следующий:

```
If <логическое выражение> Then
    P1
End If
```

Возможна и другая запись:

```
If <логическое выражение> Then P1
```

где If, Then, Else, End If – зарезервированные слова, а  $P_1$  – оператор.

### Составной условный оператор

Если при некотором условии надо выполнить определённую последовательность операторов, то их объединяют в один составной оператор.

Синтаксис составного условного оператора, если Вы пишете программу структурно, следующий:

```
If <логическое выражение> Then
    P1
    P2
    .
    .
    PN
Else
    M1
    M2
    .
    .
    Mn
End If
```

Возможна и другая запись:

If <логическое выражение> Then P<sub>1</sub> P<sub>2</sub> . . P<sub>N</sub> Else M<sub>1</sub> M<sub>2</sub> . . M<sub>n</sub>

где If, Then, Else, End If – зарезервированные слова, а P<sub>1</sub>, P<sub>2</sub>, P<sub>N</sub>, M<sub>1</sub>, M<sub>2</sub>, M<sub>n</sub> – операторы.

### Многозначные ветвления

Очень часто приходится выбирать путь решения задачи не из двух, а из нескольких возможных. В программировании это можно реализовать, используя несколько условных операторов.

Синтаксис многозначных ветвлений, если Вы пишете программу структурно, следующий:

```
If <логическое выражение1> Then
    P1
Elseif <логическое выражение2> Then
    P2
Elseif <логическое выражение3> Then
    P3
Else
    P4
End If
```

Возможна и другая запись:

```
If <логическое выражение1> Then P1 Elseif <логическое выражение2> Then P2 Elseif
<логическое выражение3> Then P3 Else P4
```

где If, Then, Else, End If – зарезервированные слова, а P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub> – операторы.

Алгоритм работы такой конструкции следующий: если логическое выражение1 истинно, то выполняется оператор или блок операторов, следующих за конструкцией Then, в противном случае этот блок пропускается. Если логическое выражение, следующее за оператором If ложно, то анализируется логическое выражение, следующее за Elseif. Если оно истинно, то выполняется оператор или блок операторов, следующих за Then, а остальные операторы пропускаются. Операторы, следующие за последним Else, выполняются лишь в том случае, если ложны все логические выражения в конструкциях If. Любая встретившаяся часть Else принадле-

жит ближайшей к ней части Then условного оператора. Условные операторы If в такой конструкции называются вложенными.

Рассмотрим пример использования простого условного оператора If.

1. Откройте новую рабочую книгу.
2. Создайте таблицу, показанную на Рисунок 66.

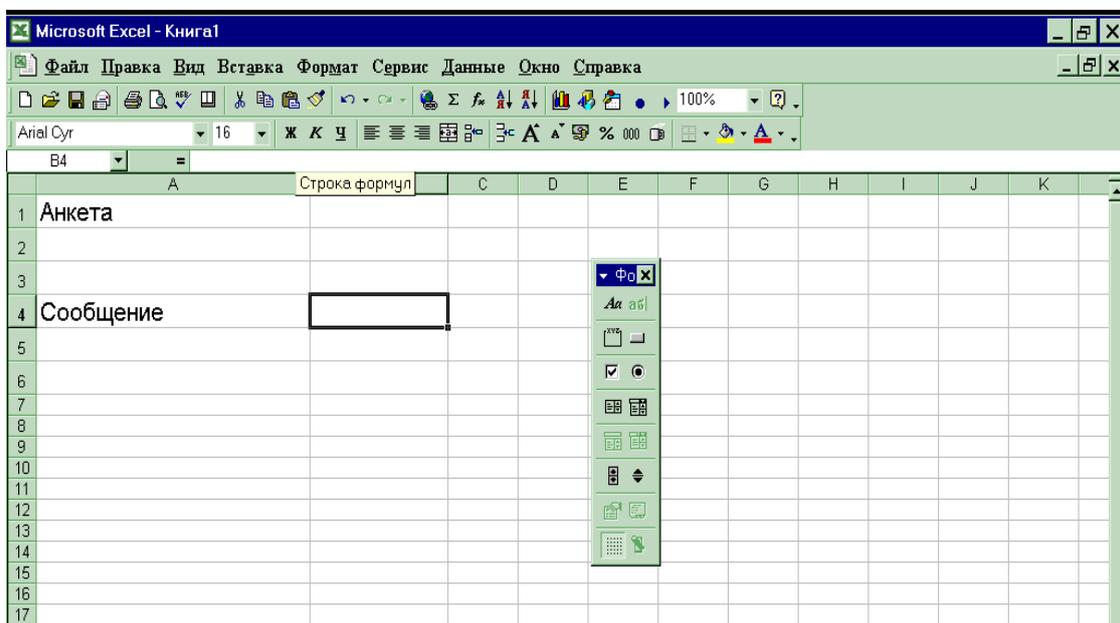


Рисунок 66. Рабочий лист с примером

2. Вставьте модуль в рабочую книгу.
4. Создайте процедуру и назовите её Абитуриент.
5. Введите следующий код процедуры:

```
Dim ball As String
ball = MsgBox("Вы хотите поступить в ПМГМУ?", vbYesNo)
If ball = vbYes Then
    Range("b4").Value = "Надо сдать экзамены"
Else
    Range("b4").Value = "Идите в ПТУ"
End If
```

6. Перейдите в рабочую книгу.
7. Вставьте командную кнопку в рабочий лист. В списке Имя макроса выберите процедуру Абитуриент и щёлкните на кнопке Ок.
8. Дайте кнопке новое имя – Вопрос. Получим лист как на Рисунок 67.

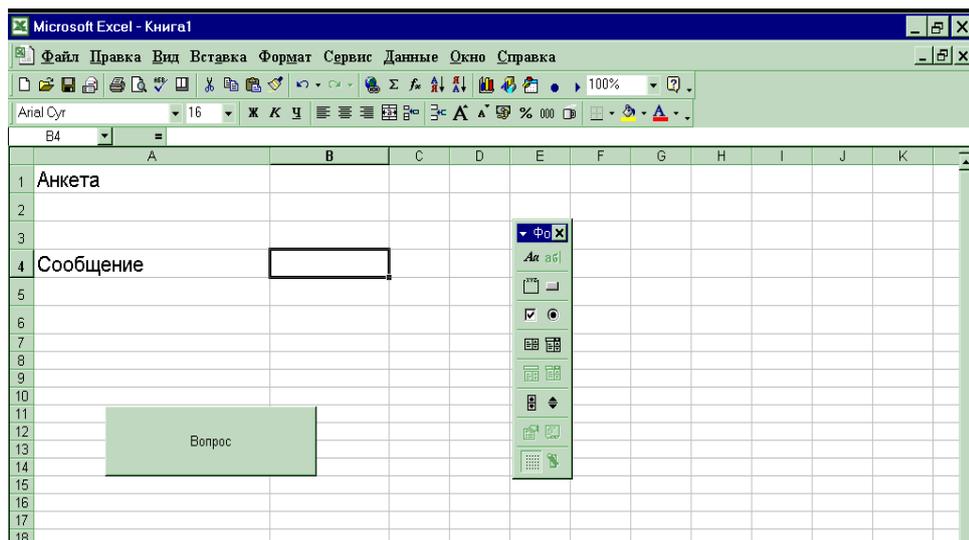


Рисунок 67. Законченный рабочий лист

9. Щёлкните на кнопке Вопрос. Отобразится окно сообщения.

В этом примере значение переменной ball проверяется условным оператором If. После применения оператора Range("b4"). Value = "Надо сдать экзамены" в ячейку b4 рабочего листа заносится, то выражение, которое находится справа от оператора присваивания либо – "Надо сдать экзамены", либо – "Идите в ПТУ".

Другой пример посвящён более сложной конструкции – многозначным ветвлениям. Но прежде несколько таблиц, содержимое которых пригодится в дальнейшем. Для создания логических условий используются операторы сравнения.

Таблица 18

<i>Оператор сравнения</i>	<i>Его назначение</i>
=	Равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
<>	Не равно

Таблица 19

<i>Логический оператор</i>	<i>Описание</i>
And (И)	Если оба условия, соединённые этим оператором имеют значение ИСТИНА, то результат – ИСТИНА, в противном случае – ЛОЖЬ
Or (ИЛИ)	Если хотя бы одно из условий, соединённых этим оператором, имеет значение ИСТИНА, то результат – ИСТИНА, в противном случае – ЛОЖЬ
Not (Не)	Если логическое выражение имеет значение ЛОЖЬ, то результат применения к нему этого оператора – ИСТИНА. Если логическое выражение имеет значение ИСТИНА, то результат применения оператора – ЛОЖЬ.
Хор (исключающее ИЛИ)	Если одно и только одно из условий, соединённых этим оператором, имеет значение ИСТИНА, то результат – ИСТИНА. Если оба условия имеют значения ИСТИНА или ЛОЖЬ, то результат – ЛОЖЬ

### Оператор выбора Select Case

В следующем примере с помощью вложенных операторов If реализуется задача перевода баллов, набранных студентами в пятибалльную систему оценок.

1. В редакторе Visual Basic напишите код процедуры, как на Рисунок 68.

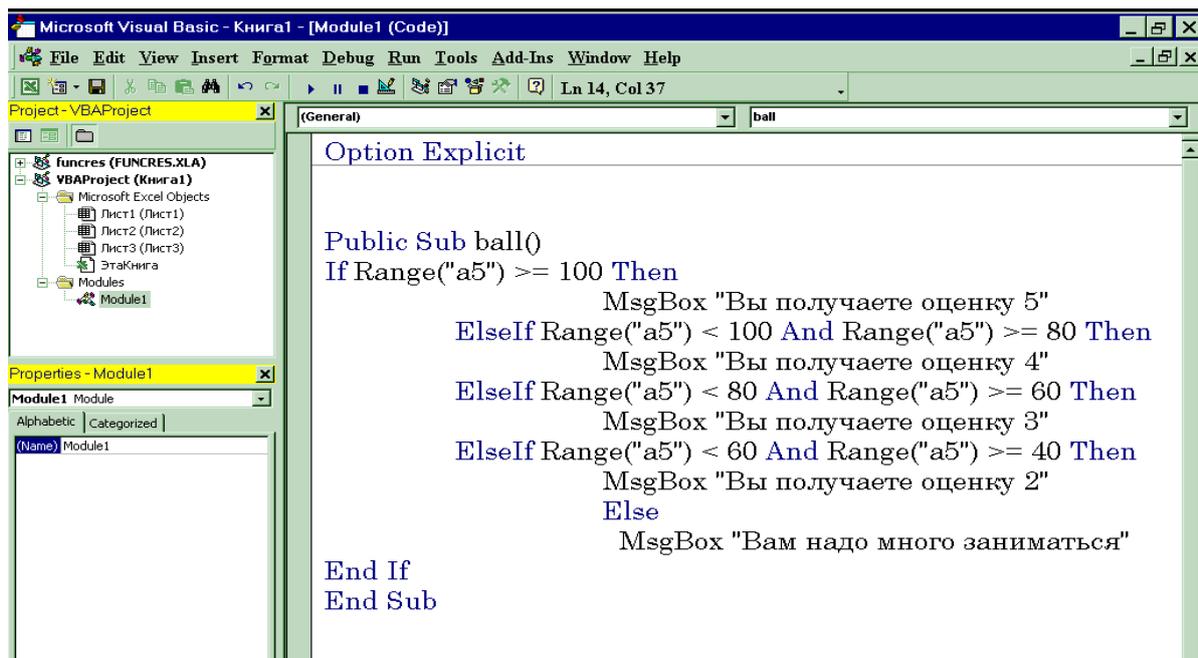


Рисунок 68. Код процедуры **Ball**

2. Запустите её на выполнение и убедитесь в том, что она работает.

На этом примере видно, что за счёт вкладывания одного условного оператора в другой, конструкция усложняется, становится тяжёлой для восприятия. В подобных случаях хорошей альтернативой оператору If служит оператор выбора Se-

lect Case, который позволяет выбрать одно из нескольких возможных продолжений программы.

Синтаксис оператора Select Case следующий:

```
Select Case КлючВыбора
    Case Is выражение
        оператор
    Case диапазон значений
        оператор
    Case диапазон значений
        оператор
    Case диапазон значений
        оператор
    Case Else
        оператор
End Select
```

Например,

```
voznrast = Application.InputBox("Введите возраст")
Select Case voznrast
    Case Is <=7
        MsgBox "Ты дошкольник"
    Case 8 to 16
        MsgBox "Ты учишься в школе"
    Case 17 to 30
        MsgBox "Тебе пора заняться делом"
    Case 31 to 60
        MsgBox "Кто не работает, тот не ест"
    Case Else
        MsgBox "Вы заслужили отдых"
End Select
```

Если значение переменной voznrast меньше или равно 7, отображается сообщение "Ты дошкольник". Если значение переменной voznrast находится в диапазоне от 8 до 16, отображается сообщение "Ты учишься в школе". Если значение переменной voznrast находится в диапазоне от 17 до 30, отображается сообщение "Тебе пора заняться делом". Если значение переменной voznrast находится в диапазоне от 31 до 60, отображается сообщение "Кто не работает, тот не ест". Если значение

возраста не равно ни одному из предложенных диапазонов значений, выводится сообщение "Заслуженный отдых".

1. В редакторе Visual Basic напишите код процедуры, как на Рисунок 69.
2. Запустите её на выполнение и убедитесь в том, что она работает.

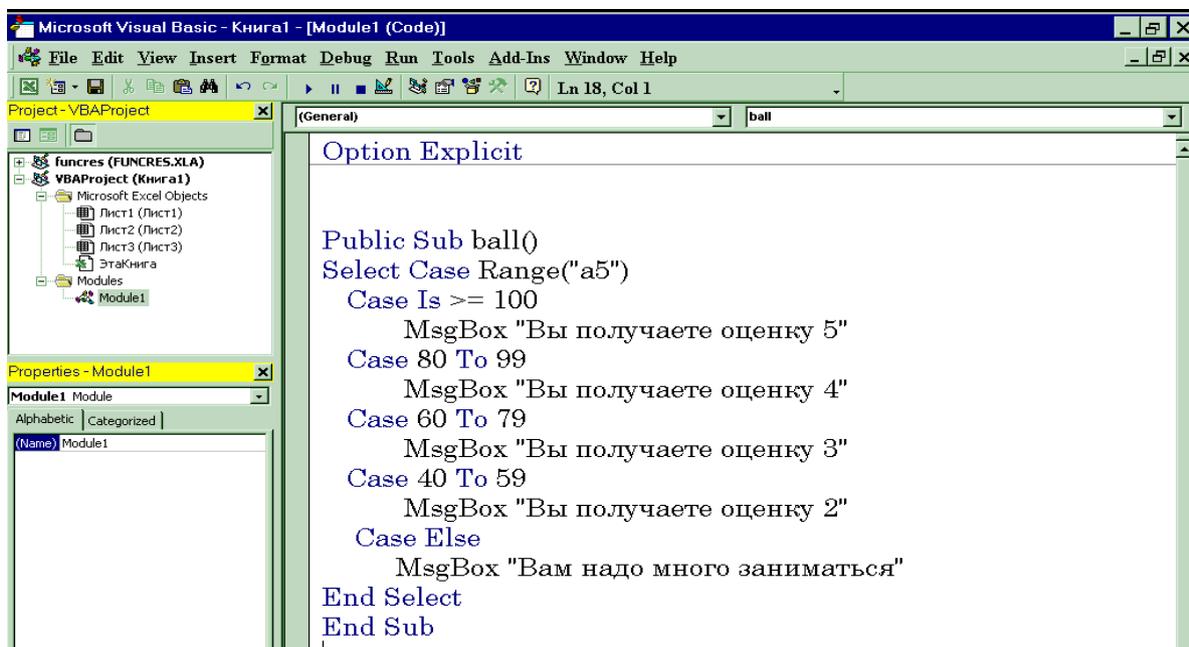


Рисунок 69. Код процедуры **Ball** с оператором **Select Case**

Вы видите, что код этой процедуры более прост для восприятия, и в случае трёх и более возможных разветвлений в программе лучше использовать оператор **Select Case**.

### Циклический алгоритм

Очень часто в программах надо выполнить определённые операторы несколько раз. Не логично записывать эту последовательность действий 20 или 50 раз подряд. В этих случаях организуют циклические вычисления. Алгоритм называется циклическим, если определенная последовательность шагов выполняется несколько раз в зависимости от заданной величины, которая называется параметром цикла.

### Оператор цикла **For...Next**

Синтаксис для цикла **For...Next**, который позволяет выполнить группу операторов или один оператор заданное количество раз, следующий:

For I= N<sub>1</sub> To N<sub>2</sub> Step h

```
      P1
      .
      .
      Pn
    } Тело цикла
Next
```

где

For(для), To(до), Step(шаг), Next(следующий) – служебные слова VBA, а P<sub>1</sub>, P<sub>n</sub> – операторы. Step является необязательным параметром. Если он опущен в программе, то значение параметра I увеличивается на 1. Параметр Step может быть любым действительным числом, как целым, так и дробным, как положительным, так и отрицательным. Работает оператор For следующим образом:

При первом вхождении в цикл, параметр цикла I принимает значение равное величине нижней границы N<sub>1</sub> и выполняется оператор или операторы в теле цикла. Затем значение параметра увеличивается на величину шага и вновь выполняется тело цикла. Подобные действия будут повторяться до тех пор, пока значение параметра цикла не станет больше величины N<sub>2</sub>, после чего осуществляется выход из цикла.

В качестве примера выполните следующее задание:

1. В редакторе Visual Basic создайте процедуру и назовите её Цикл.
2. Введите код процедуры, как на Рисунок 34.
3. Запустите её на выполнение и убедитесь в том, что она работает.

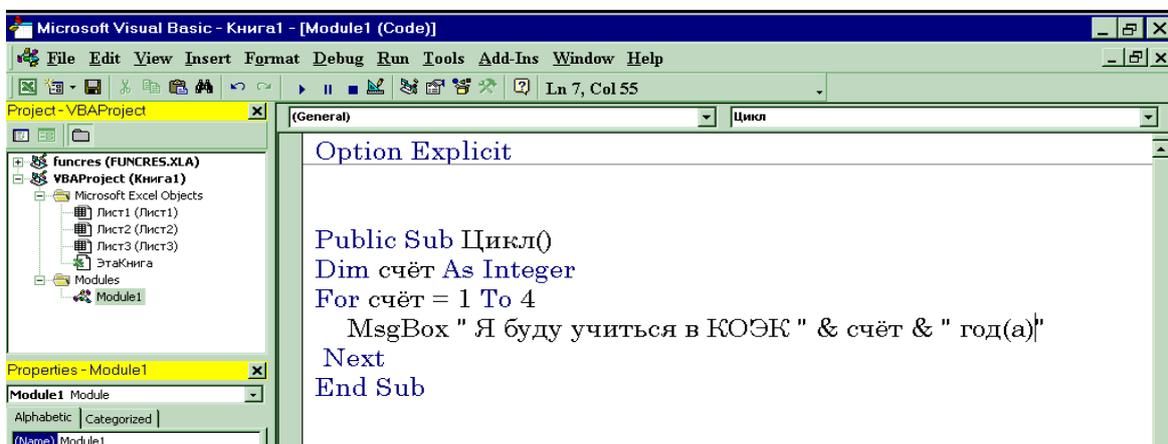
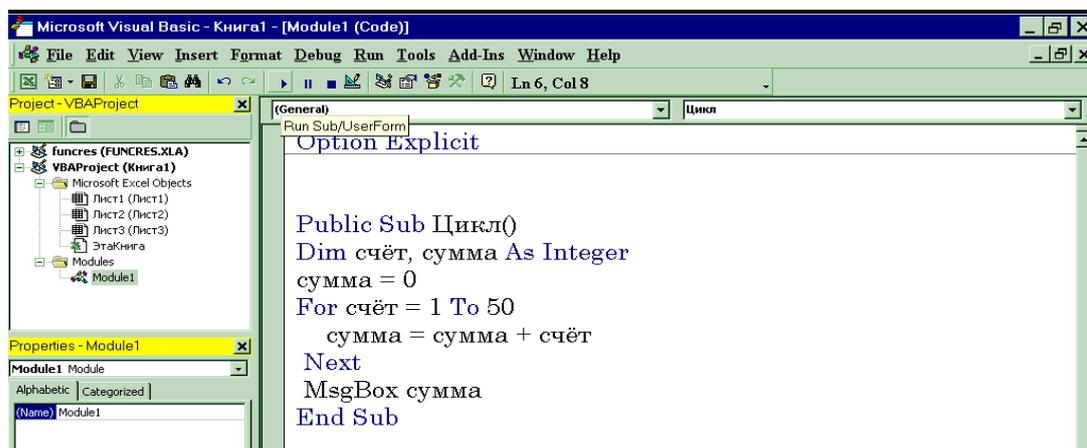


Рисунок 70. Код процедуры Цикл

Переменная *счёт* выступает в этом примере в качестве параметра цикла, и будет менять своё значение от 1 до 4. В цикле 4 раза будет появляться окно сообщения за счёт использования функции MsgBox. С помощью значка конкатенации к сообщению будет добавляться значение переменной *счёт*, а также слово год(а).

Ещё одним примером, демонстрирующим работу оператора For...Next, является решение задачи: Найти сумму целых чисел от 1 до 50.

Код процедуры представлен на Рисунок 71.



```
Option Explicit

Public Sub Цикл()
    Dim счёт, сумма As Integer
    сумма = 0
    For счёт = 1 To 50
        сумма = сумма + счёт
    Next
    MsgBox сумма
End Sub
```

Рисунок 71. Код процедуры *Сумма целых чисел*

В этом примере, описав, переменные *счёт* и *сумма* как целочисленные, обнуляется значение переменной *сумма*. Для чего это делается? Мы берём произвольную ячейку памяти компьютера и должны быть уверены в том, что в ней ничего не содержится. Ведь в цикле будет выполняться оператор  $сумма = сумма + счёт$ , то есть к тому что было в ячейке *сумма* прибавляется значение переменной *счёт*. И мы должны знать, что предварительно туда занесёно значение нуля. В результате запуска процедуры на выполнение Вы получите ответ: 1275.

## Сложный циклический процесс с использованием оператора For...Next

### Вложенные циклы

Если телом цикла является циклическая структура, то такие циклы называются вложенными. Цикл, содержащий в себе другой цикл, называют внешним, а цикл, содержащийся в теле другого цикла, называют внутренним.

Синтаксис операторов сложного цикла:

```
For I= N1 To N2
  For J= M1 To M2
    P1
    .
    .
    Pn
  Next
Next
```

} тело цикла

При первом вхождении в цикл, параметр внешнего цикла *I* принимает значение равное  $N_1$  и управление передаётся во внутренний цикл, в котором параметр цикла *j* принимает значение, равное  $M_1$  и выполняется оператор (операторы), которые записаны во внутреннем цикле. Затем параметр внутреннего цикла *j* увеличивается на 1, и вновь выполняется тело цикла. Операторы  $P_1, P_n$  будут выполняться до тех пор, пока параметр цикла *j* не станет больше величины  $M_2$ . Затем параметр внешнего цикла *I* увеличивается на 1, и вновь начинает свою работу внутренний цикл, в котором параметр цикла *j* будет изменяться от  $M_1$  до  $M_2$ , и при каждом прохождении цикла будут выполняться операторы  $P_1$  и  $P_n$ . Внешний цикл закончит свою работу, когда параметр цикла *I* станет больше величины  $N_2$ .

1. В редакторе Visual Basic введите код процедуры (Рисунок 72).
2. Разобравшись в работе сложного цикла, посмотрите, какие значения принимает переменная *произв*.

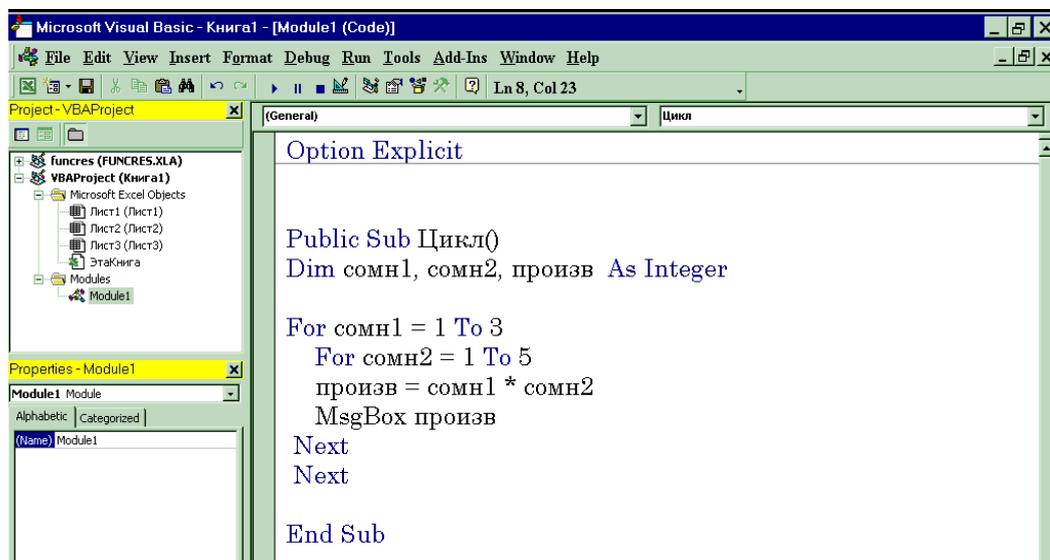


Рисунок 72. Пример работы сложного цикла

## Оператор Exit For

Оператор Exit For позволяет выйти из цикла For...Next до его завершения. Тем самым программа сможет среагировать на определённое событие, не завершая выполнение цикла заданное число раз. Например, в следующем примере

```
Dim i As Integer
Dim name As String
For i = 1 To 10
    name = InputBox("Введите Ваше имя или слово Обед для выхода")
    If name = "Обед" Then
        Exit For
    End If
    MsgBox name
Next
```

Программа запрашивает у пользователя ввод до 10 имён, но если введено слово "Обед", то программа прервёт своё выполнение и приступит к обработке инструкций, следующих за оператором Next. Оператор Exit For обычно используется с оператором If или Select Case.

## Программирование циклов Do

Для выполнения оператора For необходимо задать параметры, которые будут определять, сколько раз должен выполняться оператор(ы) цикла. Альтернативой циклу с For...Next является цикл Do, в котором группа операторов выполняется до тех пор, пока определённое логическое выражение имеет значение True (истина). Такие циклы нужно применять в тех задачах, где мы не можем знать точно, сколько раз будет повторен цикл. Например, Вы хотели бы, чтобы пользователь вводил пароль в вашей программе до тех пор, пока он не совпадёт с Вашим паролем. Существует несколько разновидностей цикла Do, в зависимости от условий его выполнения.

## Цикл Do...While

Синтаксис оператора следующий:

Do While <логическое выражение>

P<sub>1</sub>

P<sub>2</sub>

.

.

P<sub>n</sub>

Loop

где

Do(выполнить), While(пока), Loop(петля) – зарезервированные слова, P<sub>1</sub>, P<sub>2</sub>, P<sub>n</sub> – операторы

Если логическое выражение после служебного слова While имеет значение True, то выполняются операторы P<sub>1</sub>, P<sub>2</sub>, P<sub>n</sub>, после чего проверка логического выражения повторяется. Если логическое выражение имеет значение False, то происходит выход из цикла. Если условие в заголовке цикла не является истинным с самого начала, цикл Do не выполняется ни разу.

Другая возможная запись цикла:

Do While Until <логическое выражение>

P<sub>1</sub>

P<sub>2</sub>

.

.

P<sub>n</sub>

Loop

Смысл связки Do While Until <логическое выражение> – до тех пор, пока логическое выражение истинно выполнять операторы P<sub>1</sub>, P<sub>2</sub>, P<sub>n</sub>.

Например, следующий цикл Do обрабатывает данные до тех пор, пока не будет введено слово “Обед”:

```
Dim i As Integer
Dim name As String
Do While name <> "Обед"
    name = InputBox("Введите Ваше имя или слово Обед для выхода")
If name <> "Обед" Then
    MsgBox name
End If
Loop
```

Вернёмся к задаче, которая была разобрана ранее: Найти сумму целых чисел от 1 до 50. В цикле с оператором For ответом было число 1275. Что изменится в программе, если применить оператор Do While? Прежде всего, надо позаботиться о том, чтобы какая-нибудь переменная менялась в цикле от 1 до 50. Ведь такой величины как параметр цикла нет в конструкции Do While – Loop. В нашем примере это будет переменная K. Задав в качестве условия выхода из цикла  $k <> 50$ , и применяя в цикле, оператор  $sum = sum + k$  мы просуммируем все 50 слагаемых и должны получить в ответе 1275.

1. В редакторе Visual Basic создайте процедуру и назовите её Сумма.
2. Введите код процедуры, как на Рисунок 73.

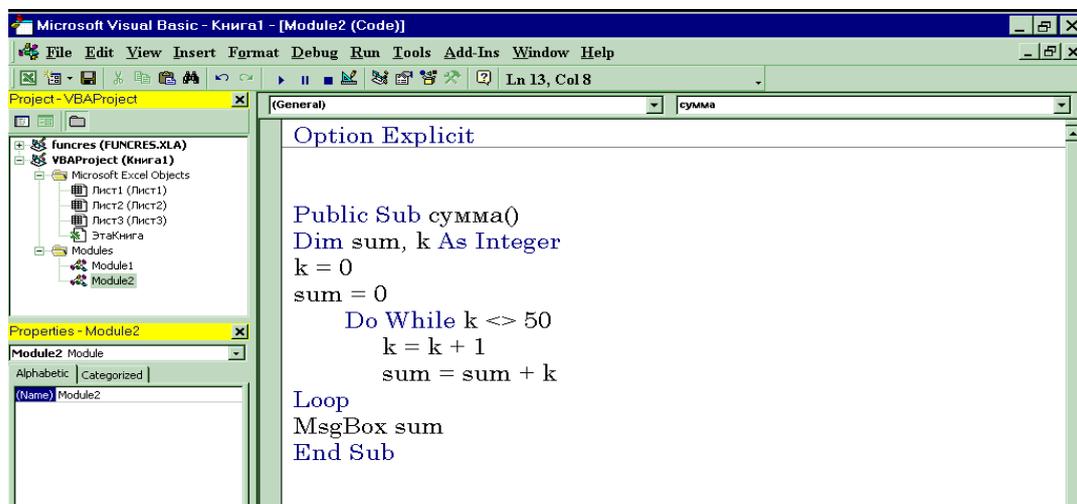


Рисунок 73. Код процедуры Сумма целых чисел

2. Убедитесь, что ответ равен 1275.

### Цикл Do...Until

Повторяющиеся вычисления можно организовать, используя в программах, другой оператор Do...Until.

Синтаксис оператора следующий:

```
Do
  P1
  P2
  .
  .
  Pn;
Loop Until <логическое выражение>
```

Цикл с оператором Do...Loop Until начинается с выполнения операторов внутри цикла, поэтому цикл, организованный с помощью этого оператора, в любом случае выполнится хотя бы один раз. Затем проверяется логическое выражение и если оно истинно, то осуществляется выход из цикла. Если же значение логического выражения ложно, то выполнение операторов P<sub>1</sub>, P<sub>2</sub>, P<sub>N</sub> повторяется, а затем вновь проверяется логическое выражение.

Другая возможная запись цикла:

```
Do
  P1
  P2
  .
  .
  Pn;
Loop While <логическое выражение>
```

Как будет организовано выполнение операторов при решении задачи: Найти сумму целых чисел от 1 до 50 в цикле Do...Loop While? Переменная K всё так же необходима для подсчёта количества повторений цикла, а так как мы знаем, что слагаемых 50, то логично в качестве условия выхода из цикла задать условие k=50. Таким образом, накопление суммы в ячейке SUM будет происходить до тех пор, пока не будут просуммированы все 50 слагаемых.

1. В редакторе Visual Basic создайте процедуру и назовите её Сумма.
2. Введите код процедуры, как на Рисунок 74.
3. Убедитесь, что ответ равен 1275.

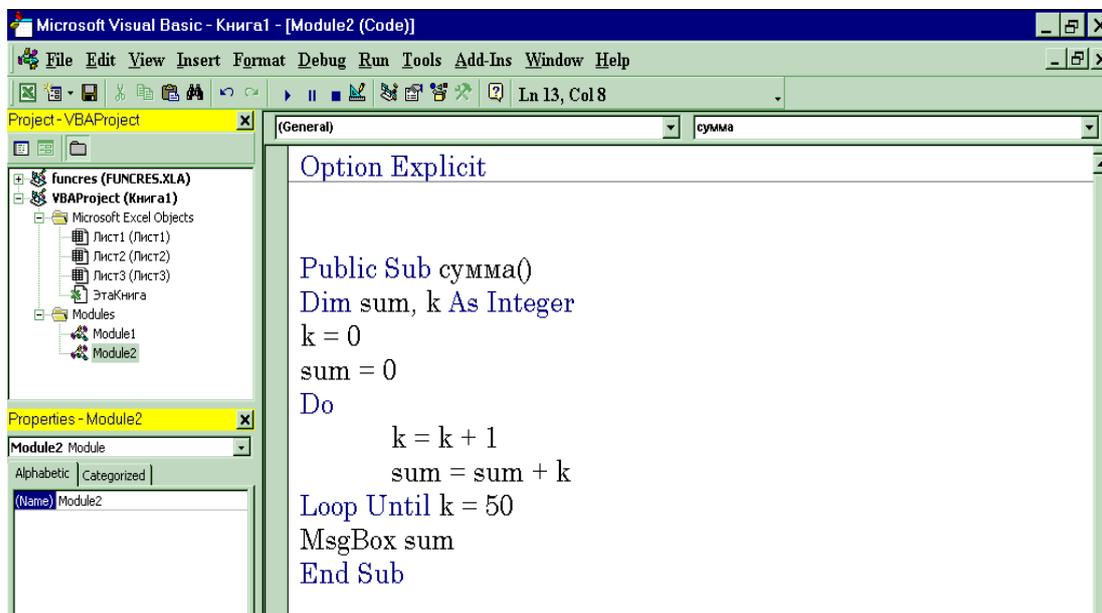


Рисунок 74. Код процедуры Сумма целых чисел

## Контрольные вопросы

1. Алгоритмические языки общего назначения.
2. Алгоритмические языки проблемно-ориентированные.
3. В чем заключается преимущество использования восьмеричной и шестнадцатиричной систем счисления в вычислительной технике?
4. В чем особенности графического способа представления алгоритмов?
5. В чем отличие позиционной системы счисления от непозиционной?
6. В чем признак глобальности сети?
7. В чем состоит авторское право на программные средства и базы данных?
8. В чем состоит идея параллелизма в исполнении программ и процессов?
9. В чем состоит имущественное право на программные средства и базы данных?
10. В чем состоит модульность при структурной разработке алгоритмов?
11. В чем состоит назначение файловой системы ОС?
12. В чем состоит процедура дискретизации непрерывной информации?
13. В чем состоят возможности одной из программ-оболочек электронной почты?
14. В чем состоят принципы пакетной обработки, разделения времени, реального времени?
15. Виды памяти ЭВМ: постоянная, CMOS, оперативная, сверхоперативная, видео, долговременная. Основные характеристики.
16. Внешние устройства ЭВМ. Назначение, основные характеристики.
17. Вспомогательные программы (утилиты). Назначение. Типы.
18. Вычислительная машина Джона фон Неймана. Принципы построения цифровой вычислительной машины.
19. Графический редактор MS Paint. Назначение. Основные возможности. Буфер обмена.
20. Двоичная система счисления. Таблица истинности для операций логического сложения, умножения и отрицания.

21. Для чего необходимо формализовать понятие алгоритма?
22. Для чего создаются локальные сети ЭВМ?
23. Запись информации в ЭВМ. Позиционные системы счисления.
24. Инструментальное программное обеспечение ЭВМ. Состав. Назначение.
25. Как ведется поиск информации в сети Internet?
26. Как выбрать модель задачи?
27. Как вызвать контекстное меню и каковы его возможности?
28. Как определяется понятие энтропии?
29. Как осуществляется защита авторских и имущественных прав?
30. Какая форма представления информации - непрерывная или дискретная - приемлема для компьютеров и почему?
31. Какие бывают конфигурации ЛС?
32. Какие виды линий (каналов) используются для связи компьютеров в локальных сетях?
33. Какие виды обмена информацией приняты в электронной почте?
34. Какие виды работ поддерживает сеть Internet?
35. Какие имеются сравнительные преимущества и недостатки у компиляторов и интерпретаторов?
36. Какие конфигурации ЛС используются в компьютерных классах Вашего вуза?
37. Какие методы доступа от компьютера к компьютеру используются в локальных сетях?
38. Какие операции над файлами обеспечиваются операционными системами?
39. Какие определения информатики вы знаете?
40. Какие принципы Ч. Бэббиджа заложили в основу идеи об автоматических цифровых вычислительных машинах?
41. Какие сетевые технологии называются клиент-серверными?
42. Какие существуют наиболее известные информационные технологии?
43. Какие типы моделей существуют?

44. Какие функции выполняют меню Пуск? Поиск?
45. Какие этические проблемы существуют, по вашему мнению, в современной информатике?
46. Какими нормативными актами регулируются отношения в сфере информатики?
47. Каков электронный адрес вашего вуза и что означают его составляющие?
48. Какова общая структура современной информатики?
49. Каково содержание понятий: процесс, ресурс, виртуализация, прерывание?
50. Каковы возможные подходы к определению понятия алгоритм?
51. Каковы основные алгоритмические структуры?
52. Каковы основные социальные последствия информатизации общества?
53. Каковы основные этапы проектирования и разработки программы?
54. Каковы особенности определения количества информации, связанной с появлением различных знаков в сообщениях?
55. Каковы совокупные признаки ЭВМ 1-, 2-, 3-, 4-го поколений?
56. Каковы специфические функции локальной сети учебного назначения?
57. Каковы способы перевода чисел из одной системы счисления в другую?
58. Каковы функции операционной системы?
59. Какое место занимает информатика в системе наук?
60. Классификация вычислительных машин. Поколения ЭВМ.
61. Кто (что) может быть исполнителем алгоритма?
62. Моделирование. Основные понятия и определения. Моделирование как метод исследования информационных систем.
63. Моделирование. Построение моделей вычислительных задач.
64. Моделирование. Построение моделей функциональных процессов.
65. Назовите методологии проектирования и разработки программ.
66. Назовите минимальный состав системы программирования, необходимый для разработки программы.

67. Назовите основные этапы трансляции программы.
68. Общая структурная схема современного компьютера. Назначение основных элементов и их краткие характеристики.
69. Операционная система MS Windows. Основные объекты MS Windows.
70. Операционные системы. Оболочки операционных систем. Назначение.
71. Опишите интерфейс оболочки ОС Norton Commander.
72. Охарактеризуйте базовые структуры алгоритмов.
73. Охарактеризуйте виды компьютерных преступлений.
74. Охарактеризуйте команды операционной системы MS DOS.
75. Охарактеризуйте место операционных систем среди других видов программного обеспечения.
76. Охарактеризуйте основные ступеньки эволюции операционных систем.
77. Охарактеризуйте стиль работы с помощью оболочки Windows.
78. Охарактеризуйте функции основных компонент операционных систем.
79. По каким показателям ЭВМ относят к тому или иному поколению?
80. Понятие алгоритма. Основные программные конструкции.
81. Предмет информатики. Информационное пространство. Основные свойства информации.
82. Прикладное программное обеспечение ЭВМ. Состав. Назначение.
83. Программное обеспечение ЭВМ. Назначение. Общая классификация.
84. Решение каких задач требует использования массивов?
85. Системное программное обеспечение ЭВМ. Состав. Назначение.
86. Современные алгоритмические языки и системы программирования. Виды. Назначение. Области применения.
87. Специальные технологии MS Windows: Drag and Drop, OLE, Plug and Play. Буфер обмена.
88. Таблицы сложения и умножения в шестнадцатиричной системе?
89. Текстовый редактор MS Word. Вставка таблиц. Вычисляемые таблицы.

90. Текстовый редактор MS Word. Гипертекст и гиперссылки. Шаблоны документов.
91. Текстовый редактор MS Word. Назначение. Основные возможности.
92. Текстовый редактор MS Word. Параметры документа. Форматирование документа.
93. Текстовый редактор MS Word. Работа с элементами графики и псевдографики. Режимы вставок.
94. Текстовый редактор MS Word. Стили. Создание оглавления документа. Сноски, ссылки.
95. Технологии MS Windows.
96. Файловая структура. Логические диски. Полное имя файла.
97. Чем определяются свойства алгоритмов «дискретность», «определенность», «понятность», «результативность», «массовость»?
98. Что имеют в виду, говоря о быстродействии ЭВМ?
99. Что называют логическими данными?
100. Что общего и в чем различие информатики и кибернетики?
101. Что означает понятие «модель» в научном познании?
102. Что означает хорошо сформулированная постановка задачи?
103. Что принято понимать под «информационным обществом»?
104. Что такое «информационная модель»?
105. Что такое «модем» и каковы его функции?
106. Что такое «объект» с точки зрения информационного моделирования?
107. Что такое алгоритмический язык?
108. Что такое гипертекст, гипермедиа?
109. Что такое гипертекст?
110. Что такое графические редакторы? Для чего они нужны?
111. Что такое интегральная схема? большая интегральная схема?
112. Что такое интегрированные системы? Для чего они нужны?

113. Что такое каталог (директорий, папка)? Для чего каталоги служат?
114. Что такое код?
115. Что такое компьютерная сеть?
116. Что такое массив?
117. Что такое мультимедиа?
118. Что такое нисходящее проектирование программ?
119. Что такое сервер, рабочая станция?
120. Что такое СУБД? Для чего они нужны?
121. Что такое текстовые редакторы? Для чего они нужны?
122. Что такое тестирование программы?
123. Что такое файл? Какие структуры файлов поддерживаются различными ОС?
124. Что такое экспертные системы? Для чего они нужны?
125. Что такое электронные таблицы? Для чего они нужны?
126. Электронные таблицы MS Excel. Мастер функций. Логические функции: ЕСЛИ, И.
127. Электронные таблицы MS Excel. Мастер функций. Математические функции: СУММА, СУММЕСЛИ, СТЕПЕНЬ.
128. Электронные таблицы MS Excel. Мастер функций. Статистические функции: СРЗНАЧ, МАКС, МИН, РАНГ, СЧЕТЕСЛИ.
129. Электронные таблицы MS Excel. Назначение. Основные возможности. Производство вычислений. Фильтрация данных.
130. Электронные таблицы MS Excel. Относительная и абсолютная адресация. Подбор параметра. Построение диаграмм.
131. Электронные таблицы MS Excel. Создание и работа с таблицами.

## Словарь терминов и определений

**Автоматизированное рабочее место (АРМ)** – средства, обеспечивающие автоматизацию многообразных функций и непосредственного доступа к ресурсам ЭВМ, размещенным на рабочем месте пользователя.

**Алгоритм** – последовательность конечного числа шагов, приводящая к намеченной цели. Каждый шаг не должен требовать никакой человеческой изобретательности для своей реализации.

**Архивное копирование** – процесс копирования файлов для бессрочного или длительного хранения на архивных носителях.

**База данных (БД)** – специально организованная совокупность взаимосвязанных данных, отражающих состояние выделенной предметной области в реальной действительности и предназначенной для совместного использования при решении задач многими пользователями.

**База знаний (БЗ)** представляет собой совокупность знаний по данной предметной области, почерпнутых из публикаций, а также введенных в процессе взаимодействия эксперта (или нескольких экспертов) с экспертной системой.

**Базовая информационная технология** – это совокупность аппаратных средств, предназначенных для организации процесса преобразования данных (информации, знаний), их связи и передачи данных.

**Банк данных** – это универсальная база данных, обслуживающая любые запросы прикладных программ вместе с соответствующим программным обеспечением.

**Воспроизведение информации** – это процесс, при котором ранее записанная на носителе информация считывается устройством воспроизведения.

**Гипертекстовая технология** – это технология преобразования текста из линейной формы в иерархическую.

**Диалог** (диалоговый режим) – интерактивная связь пользователя с ЭВМ через терминал, с которого возможен ввод команд, влияющих на порядок работы программного обеспечения.

**Документ** – совокупность взаимосвязанных показателей, рассматриваемых с точки зрения формы и содержания.

**Жизненный цикл** – это период создания и использования информационной системы, охватывающий ее различные состояния, начиная с момента возникновения необходимости в данной информационной системе и заканчивая моментом ее полного выхода из эксплуатации.

**Знания** – совокупность представлений и понятий человека о предметах, явлениях и законах действительности, формируемых в результате целенаправленного педагогического процесса, самообразования и жизненного опыта.

**Интернет-технологии** – множество способов, методов, правил и протоколов для передачи данных по сети Интернет.

**Интерфейс** – граница раздела двух систем, устройств или программ; совокупность средств и правил, обеспечивающих взаимодействие устройств, программ.

**Информационная база** - совокупность информационных массивов, организованных соответствующим образом и размещенных на машинных носителях.

**Информационная технология (ИТ)** – это система методов и способов сбора, накопления, хранения, поиска, обработки, анализа, выдачи данных, информации и знаний на основе применения аппаратных и программных средств в соответствии с требованиями, предъявляемыми пользователями. Целью любой информационной технологии является получение нужной информации требуемого качества на заданном носителе.

**Информационная услуга** – услуга по предоставлению в распоряжение пользователя необходимых ему информационных продуктов или информации.

**Информационное обеспечение АИС** - совокупность проектных решений по объемам, размещению, формам организации информации (единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков), циркулирующей в организации, а также методология построения баз данных.

**Информация** – сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления.

**Источник информации** – субъект, который порождает информацию.

**Канал передачи информации** – совокупность технических средств, обеспечивающих передачу сигналов от источника к потребителю.

**Канал связи** – совокупность технических средств передачи сигналов от источника к потребителю, организованная в соответствии со спецификой передаваемых сообщений.

**Количество информации** – мера снятия неопределенности одной случайной величины в результате наблюдения за другой.

**Корпоративная информационная система** – информационная система масштаба предприятия, осуществляющая информационную поддержку производственных, административных и управленческих процессов (бизнес-процессов), формирующих продукцию или услуги предприятия.

**Лингвистическое обеспечение АИС** – совокупность языков общения (языковых средств) персонала информационной системы и пользователей с программным, математическим и информационным обеспечением, а также совокупность терминов, используемых в информационной системе.

**Математическое обеспечение АИС** – совокупность математических методов, моделей, алгоритмов обработки информации, используемых при решении задач в

информационной системе (функциональных и автоматизации проектирования информационных систем).

**Меню** – список функций и ссылок, доступных пользователю информационной системы.

**Методическое и организационное обеспечение АИС** – совокупность методов, средств и документов, регламентирующих взаимодействие персонала информационной системы с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

**Мониторинг** – разработка системы показателей, определяющих возникновение той или иной проблемы, и механизмов их отслеживания.

**Мультимедиа** – специальные технологии, позволяющие с помощью программного обеспечения и технических средств обрабатывать на компьютере обычную информацию (текст и графику), а также звук, фотографию, анимацию (движущиеся образы) и видео.

**Навигация** – набор инструментов и индикаторов для упрощения процесса перемещения пользователя по разделам информационной системы.

**Обеспечивающие информационные технологии** – это технологии обработки информации, которые могут использоваться как инструментарий в различных предметных областях для решения различных задач.

**Объект управления** – это непосредственный исполнитель, обеспечивающий выдачу информации о своем состоянии и состоянии внешней среды, восприятие информационных воздействий от управляющего объекта и осуществление управляющих действий на основе полученной информации.

**Отображение информации** – это представление информации, т.е. генерация сигналов на основе исходных данных, а также правил и алгоритмов их преобразования в форме, приемлемой для непосредственного восприятия человеком.

**Пакет прикладных программ** – совокупность программных средств, имеющих четко выраженную модульную структуру и стандартное средство связи с помощью управляющей программы.

**Подсистема** – часть системы, выделенная по определенному признаку, обладающая некоторой самостоятельностью и допускающая разложение на элементы в рамках данного рассмотрения.

**Пользователь** – лицо или группа лиц, взаимодействующих с информационной системой в процессе ее создания и функционирования.

**Правовое обеспечение АИС** – совокупность правовых норм, регламентирующих создание, юридический статус и эксплуатацию информационных систем. С помощью правового обеспечения в первую очередь регламентируется порядок получения, преобразования и использования информации для укрепления законности работы предприятия.

**Предметная область** – часть реальной) мира, представляющая интерес для данного исследования.

**Предметная технология** – это последовательность технологических этапов преобразования первичной информации определенной предметной области в результатную, не зависящая от использования средств вычислительной техники и информационной технологии.

**Программное обеспечение ЛИС** совокупность программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

**Профилактика** – некие затраты для устранения, снижения вероятности или снижения ущерба нештатной ситуации.

**Режим реального времени** – режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними процессами в темпе, соизмеримом со скоростью их протекания.

**Резервное копирование** – это создание копий файлов для быстрого восстановления работоспособности системы при возникновении аварийной ситуации.

**Система** – совокупность взаимосвязанных элементов, образующих единое целое и функционирующих совместно для достижения единой цели.

**Система поддержки принятия решений** – система, обеспечивающая лицо, принимающее решение, необходимыми для принятия решения данными, знаниями, выводами и/или рекомендациями.

**Система управления** – это совокупность управляющего объекта, объекта управления и каналов прямой и обратной связи между ними.

**Система управления базами данных (СУБД)** – совокупность программных и языковых средств предназначенных для создания баз данных, поддержания их в актуальном состоянии и организации доступа к ним различных пользователей в условиях принятой технологии данных.

**Совокупная стоимость владения информационной системой** (от англ. Total Cost of Ownership) – это сумма прямых и косвенных затрат, которые несет владелец системы за период ее жизненного цикла.

**Ссылка** – слово, фраза или набор фраз, с помощью которых осуществляется переход от одного узла к другому.

**Статистическая информация** – совокупность различных сведений, подлежащих фиксации, передаче, преобразованию, хранению и использованию для принятия решений в различных областях.

**Статистическая информационная система** – система автоматизации работ в области статистики, позволяющая собирать, хранить и обрабатывать разнородные массивы данных с использованием единой информационной базы.

**Статистический отчет** – документ, содержащий совокупность показателей, сгруппированных по определенным признакам.

**Текстовый редактор** – программное средство для подготовки текстовых документов.

**Техническое задание** – документ, определяющий цели, требования и основные исходные данные, необходимые для разработки АИС и определения уровня экономической эффективности ее внедрения.

**Техническое обеспечение АИС** – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

**Технологический процесс обработки информации** – строго определенная последовательность взаимосвязанных процедур, выполняемых для преобразования первичной информации с момента ее возникновения до получения требуемого результата.

**Узел гипертекста** – информационный фрагмент, из которого возможен переход к другим информационным фрагментам гипертекста.

**Управление критической ситуацией** означает документирование и регламентирование действий в случае возникновения непредвиденной ситуации.

**Управляющий объект** предназначен для выработки информационных воздействий на основе собранной информации и выдачи их объектам управления. Фактически управляющий объект представляет собой управленческий аппарат системы.

**Функциональные информационные технологии** – это модификация обеспечивающих информационных технологий, при которой реализуется какая-либо из предметных технологий.

**Цифровое изображение** – графическая форма представления данных, предназначенная для зрительного восприятия, которая имеет такие характеристики, как яркость, контрастность, разрешение, цветопередача и т.д.

**Экономическая информационная система** - совокупность информации, экономико-математических методов и моделей, технических, программных, технологических средств и специалистов, предназначенная для сбора, хранения, обработки и выдачи информации об объекте управления и принятия управленческих решений.

**Экспертная система (ЭС)** – это интеллектуальная вычислительная система, в которую включены знания опытных специалистов (экспертов) о некоторой предметной области (финансы, медицина, право, геология, страхование, поиск неисправностей в радиоэлектронной аппаратуре и т. д.) и которая в пределах этой области способна принимать экспертные решения (давать советы, ставить диагноз).

**Электронная таблица (табличный процессор)** – это специальная модель структурирования, представления и обработки произвольной информации, тесно связанная и с текстовыми документами, и с базами данных.

## Библиографический список

1. Артамонов Ю.Н., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Технология программирования. Учебно-методический комплекс для направления 230200.62 "Информационные системы". – Московский городской университет управления Правительства Москвы. Москва, 2009.
2. Гапоненко В.Ф., Пронькин Н.Н. и др. Основы теории управления. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
3. Гапоненко В.Ф., Пронькин Н.Н. и др. Проектирование информационных систем в управлении. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
4. Глуценко В.М., Гапоненко В.Ф., Елизаров В.С., Пронькин Н.Н. Компьютерные технологии в экологии и природопользовании. Учебно-методический комплекс для направления 020800.62 "Экология и природопользование". – Московский городской университет управления Правительства Москвы. Москва, 2010.
5. Глуценко В.М., Гапоненко В.Ф., Елизаров В.С., Пронькин Н.Н. Экоинформатика: основы и перспективы развития. Учебно-методический комплекс для направления 020800.62 "Экология и природопользование". – Московский городской университет управления Правительства Москвы. Москва, 2010.
6. Глуценко В.М., Елизаров В.С., Каманин И.О., Пронькин Н.Н. Информатика. Учебно-методический комплекс для направления 230200.62 "Информационные системы". – Московский городской университет управления Правительства Москвы. Москва, 2009.
7. Глуценко В.М., Елизаров В.С., Каманин И.О., Пронькин Н.Н. Современные офисные технологии. Учебно-методический комплекс для направления 230200.62 "Информационные системы". – Московский городской университет управления Правительства Москвы. Москва, 2009.
8. Глуценко В.М., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Информационные технологии в управленческой деятельности (Операционная система Windows, электронные таблицы Excel). Учебно-методический комплекс для государственных гражданских служащих г. Москвы, обучающихся по образовательной программе повышения квалификации. – М: Московский городской университет управления Правительства Москвы, 2011. – 48 с.
9. Глуценко В.М., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Информационные технологии в управленческой деятельности (Операционная система Windows, текстовый редактор Word). Учебно-методический комплекс для государственных гражданских служащих г. Москвы, обучающихся по образовательной программе повышения квалификации. – М: Московский городской университет управления Правительства Москвы, 2011. – 52 с.
10. Глуценко В.М., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Информационные технологии в управленческой деятельности. Учебное пособие для государ-

ственных гражданских служащих г. Москвы, обучающихся по образовательной программе повышения квалификации. – Московский городской университет управления Правительства Москвы. Москва, 2010.

11. Глущенко В.М., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Информационно-справочные системы. Учебно-методический комплекс для студентов специальности 100103.65 "Социально-культурный сервис и туризм". – Московский городской университет управления Правительства Москвы. Москва, 2009.
12. Глущенко В.М., Елизаров В.С., Новиков А.Н., Пронькин Н.Н. Теория систем и системный анализ. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
13. Глущенко В.М., Малышев М.Н., Новиков А.Н., Пронькин Н.Н. Информационная безопасность и защита информации. Учебно-методический комплекс для направления 230200.62 «Информационные системы». – М: Московский городской университет управления Правительства Москвы, 2011. – 72 с.
14. Глущенко В.М., Пронькин Н.Н. и др. Информационное обеспечение размещения государственных и муниципальных закупок. Пособие для государственных гражданских служащих г. Москвы, обучающихся по образовательной программе профессиональной переподготовки "Управление государственными и муниципальными заказами" / В.М. Глущенко [и др.]. Москва, 2011.
15. Глущенко В.М., Пронькин Н.Н. и др. Информационное обеспечение размещения государственных и муниципальных закупок. Учебно-методический комплекс для государственных гражданских служащих г. Москвы, обучающихся по образовательной программе профессиональной переподготовки "Управление государственными и муниципальными заказами" / В.М. Глущенко [и др.]. Москва, 2011.
16. Глущенко В.М., Пронькин Н.Н. и др. Информационные системы и технологии. Учебник. – Московский городской университет управления Правительства Москвы. Москва, 2012.
17. Глущенко В.М., Пронькин Н.Н. и др. Моделирование систем. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
18. Глущенко В.М., Пронькин Н.Н. и др. Теория информационных процессов и систем. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
19. Глущенко В.М., Пронькин Н.Н. и др. Экономическая безопасность в сфере государственных и муниципальных закупок. Пособие для государственных гражданских служащих г. Москвы, обучающихся по образовательной программе профессиональной переподготовки "Управление государственными и муниципальными заказами" / В. М. Глущенко [и др.]. Москва, 2011.
20. Глущенко В.М., Пронькин Н.Н. и др. Экономическая безопасность в сфере государственных и муниципальных закупок. Учебно-методический комплекс для государственных гражданских служащих г. Москвы, обучающихся по образова-

тельной программе профессиональной переподготовки "Управление государственными и муниципальными заказами" / В. М. Глуценко [и др.]. Москва, 2011.

21. Елизаров В.С., Ковалева Е.Д., Пронькин Н.Н. Математика и информатика. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2009.
22. Елизаров В.С., Малышев М.Н., Пронькин Н.Н. Информационные технологии в социальной работе. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2011.
23. Елизаров В.С., Прудкин В.Е., Пронькин Н.Н. Компьютерная геометрия и графика. Учебно-методический комплекс. – Московский городской университет управления Правительства Москвы. Москва, 2009.
24. Кудрявцев А.С., Пронькин Н.Н. и др. Информационные системы в управлении городским хозяйством. Учебно-методический комплекс для направления 230200.62 "Информационные системы" очной формы обучения высшего профессионального образования. – Московский городской университет управления Правительства Москвы. Москва, 2011.
25. Любина О.Н., Пронькин Н.Н. Программы MICROSOFT WORD И MICROSOFT EXCEL: основные возможности и их использование на государственной службе города Москвы. Учебно-методический комплекс для государственных гражданских служащих города Москвы, обучающихся по образовательным программам повышения квалификации. – Московский городской университет управления Правительства Москвы. Москва, 2011.
26. Новиков А.Н., Пронькин Н.Н. Информатика. Учебно-методический комплекс для студентов специальностей 080109.65 – «Бухгалтерский учет, анализ и аудит» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 88 с.
27. Новиков А.Н., Пронькин Н.Н. Информатика. Учебно-методический комплекс для студентов специальностей 080107.65 – «Налоги и налогообложение» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 88 с.
28. Новиков А.Н., Пронькин Н.Н. Информатика. Учебно-методический комплекс для студентов специальностей 080105.65 «Финансы и кредит» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 88 с.
29. Новиков А.Н., Пронькин Н.Н. Информатика. Учебно-методический комплекс для студентов направления подготовки бакалавров «Экономика» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 88 с.
30. Новиков А.Н., Пронькин Н.Н. Информационная система «Электронная Москва»: учебно-методический комплекс для слушателей образовательной программы профессиональной переподготовки «Кадровая работа на государ-

ственной гражданской службе города Москвы». – М.: Московский городской университет управления Правительства Москвы, 2011. – 44 с.

31. Новиков А.Н., Пронькин Н.Н. Информационные системы в экономике. Учебно-методический комплекс для студентов специальностей 080109.65 – «Бухгалтерский учет, анализ и аудит» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 108 с.
32. Новиков А.Н., Пронькин Н.Н. Информационные системы в экономике. Учебно-методический комплекс для студентов специальностей 080105.65 – «Финансы и кредит» очной формы обучения высшего профессионального образования. – М: Московский городской университет управления Правительства Москвы, 2011. – 108 с.
33. Новиков А.Н., Пронькин Н.Н. Корпоративные информационные системы: учебно-методический комплекс для направления 230200.62 «Информационные системы». – М.: Московский городской университет управления Правительства Москвы, 2011. - 36 с.
34. Новиков А.Н., Пронькин Н.Н. Современные компьютерные технологии в экономической науке: учебно-методический комплекс для направления 080100.68 «Экономика». – М.: Московский городской университет управления Правительства Москвы, 2011. - 36 с.
35. Пронькин Н.Н. Архитектура ЭВМ и систем. Учебно-методический комплекс для направления 230200.62 "Информационные системы". – Московский городской университет управления Правительства Москвы. Москва, 2011.
36. Пронькин Н.Н. Информатика. Учебно-методический комплекс для направления 020800.62 "Экология и природопользование" – Московский городской университет управления Правительства Москвы. Москва, 2011.
37. Пронькин Н.Н. Информатика. Учебно-методический комплекс для специальности 071401.65 " Социально-культурная деятельность ". – Московский городской университет управления Правительства Москвы. Москва, 2012.

ISBN 978-5-907072-88-6



Усл. печ. л. 4,2.

Объем издания 8,2 МВ

Оформление электронного издания:

НОО Профессиональная наука, mail@scipro.ru

Дата размещения: 15.10.2019 г.

URL: <http://scipro.ru/conf/Informatics.pdf>