



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Колледж экономики, управления и права

**Методические указания
по организации практических занятий
по Информатике**

Специальность

8.02.01 Экономика и бухгалтерский учёт (по отраслям)

40.02.04 Юриспруденция

38.02.02 Страхование дело по отраслям

Ростов-на-Дону
2021

Методические указания по **УПВ.03.У Информатика** разработаны с учетом ФГОС СПО специальности 09.02.07 Информационные системы и программирование, предназначены для студентов и преподавателей колледжа.

Методические указания определяют этапы выполнения работы на практическом занятии, содержат рекомендации по выполнению заданий и образцы решения задач, а также список рекомендуемой литературы.

Составитель (автор): Д.А. Морозюк преподаватель колледжа ЭУП

Рассмотрены на заседании предметной (цикловой) комиссии специальностей «Общеобразовательных дисциплин»

Протокол № 7 от 28 мая 2021 г.

Председатель П(Ц)К специальности  Т.В. Войлова

и одобрены решением учебно-методического совета колледжа.

Протокол № 6 от 31 мая 2021 г.

Председатель учебно-методического совета колледжа
С.В.Шинакова


личная подпись

Рекомендованы к практическому применению в образовательном процессе.

СОДЕРЖАНИЕ

Практическая работа № 1. Решение задач на определение количества информации	4
Практическая работа № 2. Кодирование информации	8
Практическая работа № 3. Запись и перевод чисел в различные системы счисления	15
Практическая работа № 4. Арифметические действия в позиционных системах счисления	24
Практическая работа № 5. Основные операции алгебры логики	31
Практическая работа № 6. Построение таблиц истинности	36
Практическая работа № 7. Контрольное решение задач	41
Практическая работа № 8. Операционная система. Графический интерфейс пользователя. ПО внешних устройств. Подключение внешних устройств к ПК	42
Практическая работа № 9. Файл как единица хранения информации на компьютере. Защита информации, антивирусная защита	52
Практическая работа № 10. Форматирование электронного документа	58
Практическая работа № 11. Внедрение объектов в электронный документ	63
Практическая работа № 12. Создание и форматирование электронных таблиц	67
Практическая работа № 13. Работа с формулами и функциями в электронных таблицах	71
Практическая работа № 14. Построение диаграмм	76
Практическая работа № 15. Создание базы данных	87
Практическая работа № 16. Создание запросов, отчетов и форм в базе данных	95
Практическая работа № 17-19. Разработка веб-сайта	108
Практическая работа № 20. Построение алгоритма на псевдокоде	115
Практическая работа № 21-22. Построение блок-схем линейных алгоритмов	119
Практическая работа № 23-24. Построение блок-схем разветвляющихся алгоритмов	125
Практическая работа № 25-26. Построение блок-схем циклических алгоритмов	130
Практическая работа № 27. Контрольное решение задач	134
Практическая работа № 28-29. Составление линейных программ	143
Практическая работа № 30. Составление программ обработки текстовых данных	148
Практическая работа № 31-32. Составление разветвляющихся программ	152
Практическая работа № 33-34. Составление циклических программ	159
Практическая работа № 35-36. Составление программ с массивами	174
Практическая работа № 37. Составление программ, использующих подпрограммы	185

Практическая работа № 1. Решение задач на определение количества информации

Цель: изучить основные подходы к понятиям информация и измерение информации.

Теоретический материал

1. Содержательный подход

Для равновероятных событий используется формула Хартли $I = \log_2 N$, где N – количество возможных равновероятных событий, I – количество информации.

Пример 1. В барабане для розыгрыша лотереи находится 32 шара. Сколько информации содержит сообщение о первом выпавшем номере (например, выпал номер 15)?

Решение:

т.к. вытаскивание любого из шаров равновероятно, то количество информации вычисляется по формуле $2^I = N$, где I – количество информации, а N – количество шаров. Тогда $2^I = 32$, отсюда $I = 5$ бит.

Ответ: 5 бит.

Пример 2. Какой объем информации содержит сообщение, уменьшающее неопределенность знаний в 4 раза?

Решение:

1 бит – это количество информации, уменьшающее неопределенность знаний в 2 раза.

1 бит – это количество информации, уменьшающее неопределенность знаний в 2^I раз, отсюда, $4 = 2^I$, $I = 2$ бит.

Ответ: 2 бит.

Пример 3. При угадывании целого числа в некотором диапазоне было получено 8 бит информации. Сколько чисел содержал этот диапазон?

Решение:

$i = 8$ бит

$N = 28$

$N = 256$

Ответ: 256 чисел в диапазоне.

Для реализации Примера 4 – Примера 5 необходимо использовать Калькулятор Windows.

Пример 4.

Какое количество информации несет сообщение: «Встреча назначена на сентябрь».

Решение:

Поскольку появление в сообщении месяца сентябрь равновероятно из 12 месяцев, то количество информации определяется по формуле: $2^I = N$, где I – количество информации, N – количество месяцев. Отсюда: $2^I = 12$, $I = \log_2 12 \approx 3.584962501$ бит.

Пример 5.

Какое количество информации несет сообщение о том, что встреча назначена на 23 октября в 15.00?

Решение:

Поскольку появление в сообщении определенного числа месяца, определенного месяца и определенного часа равновероятно из общего числа дней в месяце, общего числа месяцев, общего числа часов, то количество информации определяется по формуле: $2^I = N$, где I – количество информации, $N = 31 * 12 * 24$ – (количество дней в месяце) * (количество месяцев) * (количество часов в сутках). Отсюда: $2^I = 31 * 12 * 24$, $I = \log_2 (31 * 12 * 24) \approx 13.12412131$ бит.

2. Вероятностный подход

Формула Шеннона

Так как наступление каждого из N событий имеет одинаковую вероятность P , то $P=1/N$. Если событий 6, то вероятность появления одного события равно $1/6$, если событий 100, то вероятность равна $0,01 \Rightarrow N=1/P$.

Американский ученый Клод Шеннон, что формулу Хартли можно записать иначе: $I = \log_2(1/P) = \log_2 P^{-1} = -\log_2 P$, так как $p < 1$, то $\log_2 P < 0$, а $I > 0$ (минус на минус дает плюс)

В 1948 год он предложил другую формулу определения количества информации, учитывая возможную неодинаковую вероятность событий в наборе.

Для событий с различными вероятностями количество информации определяется по **формуле Шеннона**: $I = -\sum_{i=1}^N P_i \log_2 P_i$, где I – количество информации, N – количество возможных не равновероятных событий, P_i – вероятности отдельных событий.

При решении задач необходимо подчеркнуть, что при равновероятных событиях мы получаем большее количество информации, чем при неравно вероятных событиях.

Пример 6.

В корзине лежат 8 черных шаров и 24 белых. Сколько информации несет сообщение о том, что достали черный шар?

Решение:

Вероятность достать черный шар равна $p_ч = 8/(24+8) = 0.25$

Количество информации в сообщении о том, что достали черный шар, вычисляется по формуле: $I_ч = \log_2 1/p_ч$. Отсюда $I = \log_2(1/0.25) = 2$ бита.

3. Алфавитный подход к измерению информации

$N = 2^i$ $i = \log_2 N$ $I = K * i$	N - полное количество символов в алфавите i - количество информации, которое несет каждый символ K - размер текста I - размер информации, содержащейся в тексте
--	--

Пример 7. Алфавит племени Мульти состоит из 8 букв. Какое количество информации несет одна буква этого алфавита?

Решение:

Количество информации, которое несет каждый символ алфавита мощностью N равно: $I = \log_2 N$, $N = 2^i$, $8 = 2^3$, $I = 3$ (бит)

Пример 8. Информационное сообщение объемом 1,5 Кбайта содержит 3072 символа. Сколько символов содержит алфавит, при помощи которого было записано это сообщение?

Решение:

Объем сообщения в битах: $1,5 * 1024 * 8 = 12288$ (бит)

Количество информации которое несет каждый символ в алфавите: $12288/3072 = 4$ (бит)

Мощность алфавита: $N = 2^i$, $N = 2^4 = 16$ (символов)

Задание для практической работы

Вариант 1

1. В рулетке общее количество лунок равно 128. Какое количество информации мы получаем в зрительном сообщении об остановке шарика в одной из лунок?
2. При угадывании целого числа в некотором диапазоне было получено 6 бит информации. Сколько чисел содержит этот диапазон?
3. В коробке лежат 7 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?
4. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?
5. В коробке лежат 64 цветных карандаша. Сообщение о том, что достали белый карандаш, несет 4 бита информации. Сколько белых карандашей было в корзине?
6. В непрозрачном мешочке хранятся 10 белых, 20 красных, 30 синих и 40 зеленых шариков. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика?
7. Для записи текста использовался 256-символьный алфавит. Каждая страница содержит 30 строк по 70 символов в строке. Какой объем информации содержат 5 страниц текста?

Вариант 2

1. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке номер 3. Сколько информации получили школьники из этого сообщения?
2. При угадывании целого числа в диапазоне от 1 до N было получено 7 бит информации. Чему равно N ?
3. При игре в кости используется кубик с шестью гранями. Сколько бит информации получает игрок при каждом бросании кубика?
4. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?
5. В коробке имеется 50 шаров. Из них 40 белых и 10 черных. Сколько информации в сообщении о том, что вытащили белый (черный) шар?
6. В классе 30 человек. За контрольную работу по математике получено 6 пятерок, 15 четверок, 8 троек и 1 двойка. Какое количество информации в сообщении о том, что Иванов получил четверку?
7. Сообщение занимает 3 страницы по 25 строк. В каждой строке записано по 60 символов. Сколько символов в использованном алфавите, если все сообщение содержит 1125 байтов?

Вариант 3

1. В корзине лежат 8 шаров. Все шары разного цвета. Сколько информации несет сообщение о том, что из корзины достали красный шар?
2. Сообщение о том, что ваш друг живет на 10 этаже, несет 4 бита информации. Сколько этажей в доме?
3. В коробке лежат 10 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?
4. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?
5. Сережа – лучший ученик в классе. Сколько информации в сообщении о том, что за контрольную работу по математике Сережа получит "5", если за несколько лет учебы он получил по математике 100 оценок, из них: 60 пятерок, 30 четверок, 8 троек и 2 двойки.

6. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика, если в непрозрачном мешочке хранятся: 30 белых, 30 красных, 30 синих и 10 зеленых шариков.

7. Для записи сообщения использовался 64-х символьный алфавит. Каждая страница содержит 30 строк. Все сообщение содержит 8775 байтов информации и занимает 6 страниц. Сколько символов в строке?

Вариант 4

1. Была получена телеграмма: «Встречайте, вагон 7». Известно, что в составе поезда 16 вагонов. Какое количество информации было получено?

2. Сообщение о том, что Петя живет во втором подъезде, несет 3 бита информации. Сколько подъездов в доме?

3. В коробке лежат 15 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?

4. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?

5. В пруду живут 8000 карасей, 2000 щук и 40 000 пескарей. Сколько информации в сообщении о том, что рыбак поймал в этом пруду пескаря (карася, щуку)?

6. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика, если в непрозрачном мешочке хранятся: 25 белых, 25 красных, 25 синих и 25 зеленых шариков.

7. Для записи текста использовался 256-символьный алфавит. Каждая страница содержит 30 строк по 70 символов в строке. Какой объем информации содержат 5 страниц текста?

Вопросы для самоконтроля

1. Какие бываютходы к измерению количества информации?

Практическая работа № 2. Кодирование информации

Цель: Получить навыки кодирования и декодирования информации

Теоретический материал

Теория кодирования – это раздел теории информации, изучающий способы отождествления сообщений с отражающими их сигналами. Задачей теории кодирования является согласование источника информации с каналом связи.

Объектом кодирования служит как дискретная, так и непрерывная информация, которая поступает к потребителю через источник информации. Понятие кодирование означает преобразование информации в форму, удобную для передачи по определенному каналу связи.

Обратная операция – **декодирование** – заключается в восстановлении принятого сообщения из закодированного вида в общепринятый, доступный для потребителя.

В теории кодирования существует ряд направлений:

- статическое или эффективное кодирование;
- помехоустойчивое кодирование;
- корректирующие коды;
- циклические коды;
- арифметические коды.

С появлением управляющих систем, в частности ЭВМ, роль кодирования существенно возросла и изменилась, так как без кодирования невозможна передача информации. В последнее время в связи с развитием телекоммуникационных систем и широким использованием вычислительной техники для обработки и хранения информации возникла новая область знаний – информационная безопасность.

Кодированием называют универсальный способ отображения информации при ее хранении, обработке и передаче в виде системы соответствий между сигналами и элементами сообщений, при помощи которых эти элементы можно зафиксировать.

Код – это правило однозначного преобразования сообщения из одной символической формы представления сообщения в другую, обычно без каких-либо потерь информации.

Если все кодовые слова имеют одинаковую длину, то код называется равномерным, или блочным.

Под абстрактным алфавитом будем понимать упорядоченное дискретное множество символов.

Алфавитное кодирование. Алфавитное, т.е. побуквенное, кодирование можно задать таблицей кодов. Фактически кодом преобразования является некоторая подстановка.

Тогда $\sigma = \begin{pmatrix} \alpha_1 & \dots & \alpha_n \\ \beta_1 & \dots & \beta_n \end{pmatrix}$, где $\alpha_i \in$ алфавиту А, $\beta_i \in$ множеству слов, составленных в алфавите В.

Множество кодов букв называется множеством элементарных кодов. Алфавитное кодирование можно использовать для любого множества сообщений.

Компьютерная обработка данных основана на применении двоичного кода. Этот универсальный способ кодирования годится для любых данных, независимо от их происхождения и содержания.

Кодирование текста

Тексты – это последовательности символов, входящих некоторый алфавит. Кодирование текста сводится к двоичному кодированию алфавита, на основе которого он построен. Чаще всего применяется байтовое кодирование алфавита. В этом случае максимальная мощность алфавита составляет 256 символов. Такой алфавит может содержать два набора буквенных символов (например, русский и латинский), цифры, знаки препинания и математические знаки, пробел и небольшое число дополнительных символов. Примером такого алфавита является код **ASCII**.

Однако, ограниченный набор из 256 кодов символов сегодня уже не удовлетворяет возросшие потребности международного общения. Все большее распространение получает универсальная система 16-разрядного кодирования символов **UNICODE**.

Мощность алфавита в системе кодирования **UNICODE** составляет $2^{16}=65\ 536$ разных кодов, из которых 63 484 кода соответствуют символам большинства алфавитов, а оставшиеся 2048 кодов разделены пополам и образуют таблицу размером 1024 столбцов x 1024 строк. В этой таблице более миллиона ячеек, в которых можно разместить еще более миллиона различных символов. Это символы «мертвых» языков, а также символы, не имеющие лексического содержания, указатели, знаки и т.п. Для записи этих дополнительных символов необходима пара 16-разрядных слов (16 разрядов для номера строки и 16 разрядов для номера столбца).

Таким образом, система **UNICODE** является универсальной системой кодирования всех символов национальных письменных систем и обладает возможностью существенного расширения.

Кодирование изображений

Рисунки, картинки, фотографии кодируются в **растровом формате**. В этом виде каждое изображение представляет собой прямоугольную таблицу, состоящую из цветowych точек. Цвет и яркость каждой отдельной точки выражаются в числовой форме, что позволяет использовать двоичный код для представления графических данных.

Черно-белые изображения принято представлять в градациях серого цвета, для этого используется модель **GreyScale**. Если яркость точки кодируется одним байтом, можно использовать 256 различных серых тонов. Такая точность согласуется с восприимчивостью человеческого глаза и возможностями полиграфической техники.

При кодировании цветных изображений применяют принцип декомпозиции цвета на составляющие, для этого используют модель **RGB**. Цветное изображение на экране получается путем смешивания трех базовых цветов: красного (Red, R), синего (Blue, B) и зеленого (Green, G).

Каждый пиксель на экране состоит из трех близко расположенных элементов, светящихся этими цветами.

Цветные дисплеи, использующие такой принцип называются **RGB -мониторами**.

Код цвета пикселя содержит информацию о доле каждого базового цвета.



Схема цветообразования

Если все три составляющих имеют одинаковую интенсивность (яркость), то из их сочетаний можно получить 8 различных цветов:

R	G	B	цвет
0	0	0	 черный
0	0	1	 синий
0	1	0	 зеленый
0	1	1	 голубой
1	0	0	 красный
1	0	1	 розовый
1	1	0	 коричневый
1	1	1	белый

Формирование цветов при глубине цвета 24 бита:

цвет	интенсивность R	интенсивность G	интенсивность B
черный	00000000	00000000	00000000
красный	11111111	00000000	00000000
зеленый	00000000	11111111	00000000
синий	00000000	00000000	11111111
голубой	00000000	11111111	11111111
желтый	11111111	11111111	00000000
белый	11111111	11111111	11111111

Чем больше глубина цвета, тем шире диапазон доступных цветов и тем точнее их представление в оцифрованном изображении. Пиксель с битовой глубиной, равной единице, имеет лишь 2 (в первой степени) возможных состояния – два цвета: черный или белый. Пиксель с битовой глубиной в 8 единиц имеет 28 или 256 возможных цветовых значений. Пиксель же с битовой глубиной в 24 единицы имеет 224 степени) или 16,7 миллионов возможных значений. Считается, что 24-битные изображения, содержащие 16,7 миллионов цветов, достаточно точно передают краски окружающего нас мира. Как правило, битовое разрешение задается в диапазоне от 1 до 48 бит/пиксель.

При печати на бумаге используется несколько иная цветовая модель: если монитор испускал свет, оттенок получался в результате сложения цветов, то краски - поглощают свет, цвета вычитаются. Поэтому в качестве основных используют голубую (Cyan, C), пурпурную (Magenta, M) и желтую (Yellow, Y) краски. Кроме того, из-за не идеальности красителей, к ним обычно добавляют четвертую -- черную (black, K). Для хранения информации о каждой краске и в этом случае чаще всего используется 1 байт. Такая система кодирования носит название **СМУК**.

Более грубое представление цвета использует меньшее число разрядов. Например, кодирование цветной графики 16-разрядными числами носит название **High Color**. В этом случае каждому цвету отводят пять разрядов.

Кодирование звука и видео

Приемы работы со звуковой информацией пришли в компьютерную технику позже всего. Аналитический метод кодирования, применимый к любым звуковым сигналам основан на аналогово-цифровом преобразовании. Исходный аналоговый сигнал представляют как последовательность цифровых сигналов, записанных в двоичном коде. Разрядность преобразования определяет объем данных, соответствующих отдельному цифровому сигналу. При воспроизведении звука выполняют обратное цифро-аналоговое преобразование.

Этот метод кодирования содержит погрешность, так что воспроизводимый сигнал несколько отличается от оригинала.

Метод кодирования на основе табличного синтеза применим только к музыкальным произведениям. В заранее подготовленных таблицах хранятся образцы (**сэмплы**) звуков различных музыкальных инструментов. Числовые коды определяют инструмент, ноту и продолжительность звучания.

При кодировании видеосигнала требуется записать последовательность изображений (кадров) и звук (звуковая дорожка). Формат видеозаписи позволяет включить оба потока данных в одну цифровую последовательность.

Пример 1. Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Я УМЕЮ КОДИРОВАТЬ ИНФОРМАЦИЮ".

Решение: Для того, чтобы легко закодировать фразу, заменяя каждую букву ее порядковым номером в алфавите необходимо составить таблицу соответствия "Буква - её порядковый номер":

№	Буква	№	Буква	№	Буква	№	Буква	№	Буква
1	А	8	Ж	15	Н	22	Ф	29	Ы
2	Б	9	З	16	О	23	Х	30	Ь
3	В	10	И	17	П	24	Ц	31	Э
4	Г	11	Й	18	Р	25	Ч	32	Ю
5	Д	12	К	19	С	26	Ш	33	Я
6	Е	13	Л	20	Т	27	Щ	-	расстояние между буквами
7	Ё	14	М	21	У	28	Ъ	-	пробел

Теперь можно легко зашифровать фразу. Она будет выглядеть так: "33 21-14-6-32 12-16-5-10-18-16-3-1-20-30 10-15-22-16-18-14-1-24-10-32"

Пример 2. Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца):

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Й	Э	Ю	Я	_	.	,
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: ТЕЛЕГРАММА.

Решение: 20 05 12 05 03 17 00 13 13 00

Задание для практической работы

Вариант 1

1. Дана кодовая таблица азбуки Морзе

А	•-	Л	•-••	Ц	-•-•
Б	-•••	М	--	Ч	----•
В	•--	Н	-•	Ш	-----
Г	--•	О	----	Щ	--•-
Д	-••	П	•--•	Ъ	•--•-•
Е	•	Р	•-•	Ы	-•--
Ж	•••-	С	•••	Ь	-••-
З	--••	Т	-	Э	••-••
И	••	У	••-	Ю	••--
Й	•---	Ф	••-•	Я	•-•-
К	-•-	Х	••••		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): --

-----••••-----••••-----

2. Закодируйте с помощью азбуки Морзе слова: информатика, данные, алгоритм.

3. Закодируйте с помощью азбуки Морзе свое имя и фамилию.

4. Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Э	Ю	Я	-	.	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: Я УМЕЮ РАБОТАТЬ С ИНФОРМАЦИЕЙ! А ТЫ?

5. Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Я УМЕЮ КОДИРОВАТЬ ИНФОРМАЦИЮ". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

Вариант 2

1. Дана кодовая таблица азбуки Морзе

А	•-	Л	•-••	Ц	-•-•
Б	-•••	М	--	Ч	----•
В	•--	Н	-•	Ш	-----
Г	--•	О	----	Щ	--•-
Д	-••	П	•--•	Ъ	•--•-•
Е	•	Р	•-•	Ы	-•--
Ж	•••-	С	•••	Ь	-••-
З	--••	Т	-	Э	••-••

И	••	У	••–	Ю	••––
Й	•–––	Ф	••–•	Я	•–•–
К	–•–	Х	••••		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): - • - • - - - - -

- Закодируйте с помощью азбуки Морзе слова: информатика, данные, алгоритм.
- Закодируйте с помощью азбуки Морзе свое имя и фамилию.
- Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Э	Ю	Я	–	•	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: "Таблицы маршрутизации - электронные базы данных".

5. Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Скрипт - набор команд какого-либо языка". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

Вариант 3

1. Дана кодовая таблица азбуки Морзе

А	•–	Л	•–••	Ц	–•••
Б	–•••	М	––	Ч	–•••
В	•––	Н	–•	Ш	–•••
Г	–••	О	–••	Щ	–••–
Д	–••	П	•–••	Ъ	•–•••
Е	•	Р	•–•	Ы	–••–
Ж	•••–	С	•••	Ь	–••–
З	–•••	Т	–	Э	••–••
И	••	У	••–	Ю	••––
Й	•–––	Ф	••–•	Я	•–•–
К	–•–	Х	••••		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): • - - • - - - - -

- Закодируйте с помощью азбуки Морзе слова: флаг, имя, кодирование.
- Закодируйте с помощью азбуки Морзе свое имя и фамилию.
- Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С

2	Т	У	Ф	Х	Ц	Ч	Ш	Ш	Ъ
3	Ы	Ь	Э	Ю	Я	_	.	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: "УЧЕНИЕ ДА ТРУД К СЛАВЕ ВЕДУТ"

5. Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "ЮЗЕР - ПОЛЬЗОВАТЕЛЬ КОМПЬЮТЕРА". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

Вопросы для самоконтроля

1. Что изучает теория кодирования?
2. Объясните термины: кодирование и декодирование.
3. Что такое код?
4. В чем заключается алфавитное кодирование?
5. Как кодируются тексты?
6. Как кодируются изображения?
7. Как кодируются звук и видео?

Практическая работа № 3. Запись и перевод чисел в различные системы счисления

Цель: изучить основные алгоритмы перехода из одной системы счисления в другую, сформировать умения переводить числа из одной системы счисления в другую, работать с числами разных систем счисления – позиционных и непозиционных

Теоретический материал

Система счисления – совокупность правил наименования и изображения чисел с помощью набора символов, называемых цифрами.

Количество цифр (знаков), используемых для представления чисел называют *основанием системы счисления*.



Древнегреческая,
кириллическая,
римская

Десятичная, двоичная и т.д.

Римская система записи чисел

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Например, число **CCXXXII** складывается из двух сотен, трех десятков и двух единиц и равно **232**.

В римских числах цифры записываются слева направо в порядке убывания. В таком случае их значения складываются. Если слева записана меньшая цифра, а справа – большая, то их значения вычитаются.

Например:

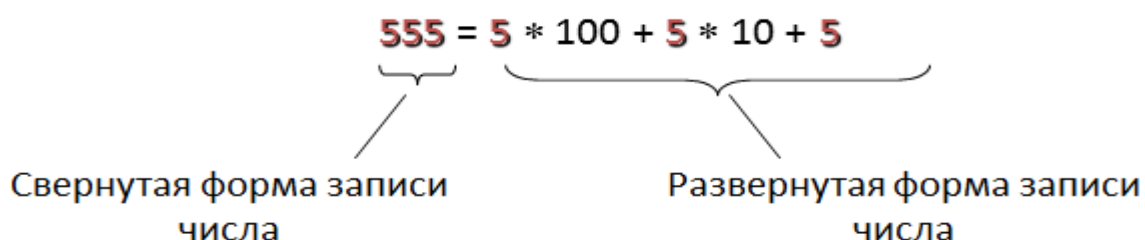
$$VI = 5 + 1 = 6, IV = 5 - 1 = 4.$$

$$MCMXCVII = 1000 + (-100 + 1000) + (-10 + 100) + 5 + 1 + 1 = 1997$$

Позиционная система счисления

Развернутая форма записи числа

В числе 555 первая пятерка обозначает пять сотен, вторая – пять десятков, третья – пять единиц. Одна и та же цифра в зависимости от позиции в записи числа обозначает разные величины.



Всякое десятичное число можно представить как сумму произведений составляющих его цифр на соответствующие степени десятки. То же самое относится и к десятичным дробям.

$$\begin{array}{l}
 10^0 = 1 \\
 10^1 = 10 \qquad 10^{-1} = 0,1 \\
 10^2 = 100 \qquad 10^{-2} = 0,01 \\
 10^3 = 1000 \qquad 10^{-3} = 0,001 \text{ и т.д.}
 \end{array}$$

Например:

$$26,387 = 2 * 10^1 + 6 * 10^0 + 3 * 10^{-1} + 8 * 10^{-2} + 7 * 10^{-3} .$$

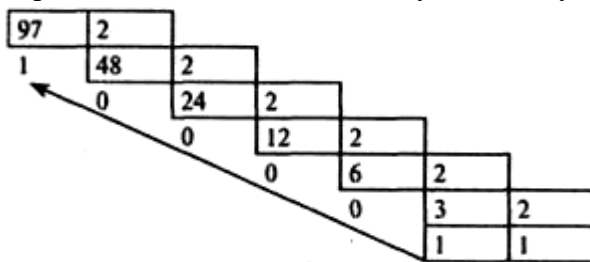
Алгоритмы перевода чисел из одной системы счисления в другую:

1. Алгоритм перевода целых чисел из десятичной системы счисления в любую другую.

1. Последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получится частное, меньше делителя.
2. Полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.
3. Составить число в новой системе счисления, записывая его, начиная с последнего остатка.

Пример 1.

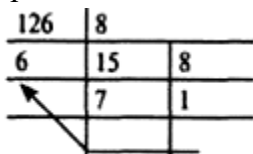
Перевести число 97_{10} в двоичную систему счисления.



Получаем $97_{10} = 1100001_2$

Пример 2.

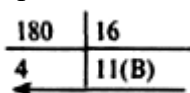
Перевести число 126_{10} в восьмеричную систему счисления.



Получаем $126_{10} = 176_8$

Пример 3.

Перевести число 180_{10} в шестнадцатеричную систему счисления.



2. Алгоритм перевода правильных дробей из десятичной системы счисления в любую другую

1. Последовательно умножаем данное число и получаемые дробные части произведения на основание новой системы счисления до тех пор, пока

Дробная часть произведения не станет равной нулю или будет достигнута требуемая точность представления числа.

2. Полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.

3. Составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Пример 4.

Перевести число $0,65625_{10}$ в восьмеричную систему счисления.

0,	65625
	• 8
5	25000
	• 8
2	00000

Получаем: $0,65625_{10} = 0,52_8$

Пример 5.

Перевести число $0,65625_{10}$ в шестнадцатеричную систему счисления.

0,	65625
	• 16
10 (A)	50000
	• 16
8	00000

Получаем: $0,65625_{10} = 0,A8_{16}$.

Пример 6.

Перевести число $0,625_{10}$ в двоичную систему счисления.

0,	625
	• 2
0	1250
	• 2
0	2500
	• 2
0	5000
	• 2
1	0000

Получаем: $0,625_{10} = 0,0001_2$

Пример 7.

Перевести число $0,9_{10}$ в двоичную систему счисления.

0,	9
	• 2
1	8
	• 2
1	6
	• 2
1	2
	• 2
0	4
	• 2
0	8
	• 2
1	6

Этот процесс можно продолжать бесконечно. В этом случае деление продолжаем до тех пор, пока не получим нужную точность представления.

Получаем: $0,9_{10} = 0,111001_2$ с точностью до семи значащих цифр после запятой.

3. Перевод произвольных чисел

Перевод произвольных чисел, т.е. содержащих целую и дробную часть, осуществляется в два этапа. Отдельно переводится целая часть, отдельно $\sim\sim$ дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой.

Пример 8.

Перевести число 18,34 из десятичной в двоичную систему счисления.

Переводим целую часть:

18	2		
0	9	2	
	1	4	2
		0	1

Переведем дробную часть.

0,	34
	*2
0	68
	*2
1	36
	*2
0	72
	*2
1	44

Получаем: $18,34_{10} = 1010,0101_2$

Пример 9.

Перевести число 124,26 в шестнадцатеричную систему счисления.

Переводим целую часть:

124	16
12	7
(C)	

Переведем дробную часть

0,	26
	*16
4	16
	*16
2	56
	*16
8	96
	*16
15	36
(F)	

Получаем: $124,26_{10} = 7C,428F_{16}$.

С помощью метода триад и тетрад вы научитесь легко переводить числа между СС основанием которых является степень двойки. Это двоичная, четверичная, восьмеричная и шестнадцатеричная СС.

Для этого нам с вами нужно научиться **строить таблицу триад и тетрад**. Сделать это очень просто. Таблица заполняется столбиками, поочередно.

Первый столбик – Выписываем 8 нулей, затем 8 единиц.

Второй столбик – уменьшаем количество 0 и 1 в два раза, т.е. записываем 4 нуля и 4 единицы, повторяем это дважды.

Третий столбик – еще в два раза уменьшаем нули и единицы. Чередуем 2 нуля и 2 единицы до конца столбика.

Четвертый столбик – чередуем нули и единицы.

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

ТАБЛИЦА ДИАД

ТАБЛИЦА ТРИАД

ТАБЛИЦА ТЕТРАД

Таким образом мы с вами построили таблицу **тетрад** (нужна для шестнадцатеричной СС), но внутри нее есть таблица **триад** (для восьмеричной СС). А внутри таблицы триад, есть таблица диад (для четверичной СС).

Для вашего удобства давайте запишем таблицу и выпишем значения для десятичной СС и шестнадцатеричной СС.

Таблица тетрад				16 СС	10 СС
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	10
1	0	1	1	B	11
1	1	0	0	C	12
1	1	0	1	D	13
1	1	1	0	E	14
1	1	1	1	F	15

Как пользоваться этой таблицей?

Давайте представим, что у нас с вами есть некоторое шестнадцатеричное число **CE5**. Нам нужно перевести это число в двоичную СС. Мы могли бы воспользоваться тем что уже умеем, т.е. перевести это число в десятичную СС, а потом в двоичную. Но это долго, можно запутаться и т.д. Поэтому мы сделаем с вами все намного проще. Смотрим по таблице:

Таблица тетрад	16 СС	10 СС
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	A	10
1 0 1 1	B	11
1 1 0 0	C	12
1 1 0 1	D	13
1 1 1 0	E	14
1 1 1 1	F	15

CE5₁₆

1100 1110 0101

CE5₁₆ = 110011100101₂

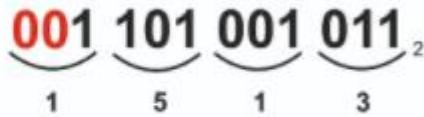
C = 1100, E = 1110, 5 = 0101, отсюда следует что CE5 = 110011100101.

Каждая шестнадцатеричная цифра является тетрадой в двоичной системе счисления. Каждая восьмеричная цифра является триадой в двоичной системе счисления. Каждая четвертичная цифра является диадой в двоичной системе счисления

Рассмотрим пример, когда у нас есть двоичное число

Число 1101001011 в двоичной СС, нам нужно перевести в восьмеричную СС. Нам нужно разделить это число на триады, начиная справа. Получится 1 101 002 011 и добавим к единице два незначащих нуля.

Таблица тетрад				16 СС	10 СС
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	10
1	0	1	1	B	11
1	1	0	0	C	12
1	1	0	1	D	13
1	1	1	0	E	14
1	1	1	1	F	15



$1101001011_2 = 1513_8$

Задание для практической работы

Задание 1.

В некоторой системе счисления цифры имеют форму различных геометрических фигур. Приведены примеры некоторых чисел, записанных в этой системе счисления:

$$\begin{array}{l} \bigcirc \triangle - 4 \\ \triangle \bigcirc - 6 \\ \square \bigcirc \square - 19 \end{array} \quad \begin{array}{l} \square \square \square - 190 \\ \nabla \square \nabla - 1900 \end{array}$$

Какому числу соответствует следующая запись:

$$\nabla \square \nabla \square \square \triangle \bigcirc \bigcirc ?$$

Задание 2.

Выполните действия и запишите результат римскими цифрами:

$$\text{XXII} - \text{V}; \quad \text{CV} - \text{LIII} + \text{XIX};$$

$$\text{MCM} + \text{VIII}; \quad \text{XX} : \text{V};$$

$$\text{X} * \text{IV}; \quad \text{LXVI} : \text{XI};$$

$$\text{XXIV} * \text{VII}.$$

Задание 3.

Записать числа в развернутой форме:

$$285 \quad 3864 \quad 893,735 \quad 34,07$$

Задание 4. Переведите число 2004_{10} в:

- А) двоичную систему счисления;
- Б) восьмеричную систему счисления;
- В) шестнадцатеричную;

Задание 5. Переведите числа из десятичной системы счисления:

- А) 0, 141 в пятеричную систему счисления с точностью до пяти знаков.
- Б) 0, 675 в троичную систему счисления с точностью до пяти знаков.
- В) 0, 2004 в восьмеричную систему счисления с точностью до пяти знаков.
- Г) 0,7982 в двоичную систему счисления с точностью до шести знаков.

Задание 6. Перевести числа в десятичную систему счисления.

- a) 1101_2
- b) $0,123_5$
- c) $16,4_8$
- d) 7688_9
- e) $432,1_5$
- f) 120_3
- g) $102,31_4$

Задание 7. Представьте в десятичной системе счисления число $101,1$, считая записанным в системах счисления от двоичной до девятеричной.

Задание 8. В коробке лежит 31_8 шар. Среди них 12_8 красных и 17_8 желтых. Докажите, что здесь нет ошибки.

Задание 9.

В классе 1111_2 девочек и 1010_2 мальчиков. Сколько учеников в классе?

Задание 10.

Используя приложение Калькулятор операционной системы Windows, заполните таблицу.

Dec	Oct	Bin	Hex
1011			
	1011		
		1011	
			1011

Вопросы для самоконтроля

1. Что такое система счисления?
2. Чем отличаются позиционные системы счисления от непозиционных, в чем их преимущества?
3. Привести примеры позиционных и непозиционных систем счисления.
4. А. С. Пушкин родился в MDCCXCIX году?
5. Что такое основание системы счисления?
6. В какой стране впервые стали использоваться специальные обозначения для 5,10,100,1000,1000000?

Практическая работа № 4. Арифметические действия в позиционных системах счисления

Цель: научиться производить арифметические операции (сложение, вычитание и умножение) с числами в различных системах счисления

Теоретический материал

Арифметические операции во всех позиционных системах счисления выполняются по одним и тем же хорошо известным правилам.

Правила выполнения арифметических операций в десятичной системе хорошо известны – это сложение, вычитание, умножение столбиком и деление уголком. Эти правила применимы и ко всем другим позиционным системам счисления. Только таблицами сложения и умножения надо пользоваться особыми для каждой системы.

Таблицы сложения в любой позиционной системе счисления легко составить, используя **правило счета:**

Если сумма складываемых цифр больше или равна основанию системы счисления, то единица переносится в следующий слева разряд.

Таблица сложения в двоичной системе:

+	0	1
0	0	1
1	1	10

Таблица сложения в восьмеричной системе:

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица сложения в шестнадцатеричной системе:

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Пример:

1) Сложим числа 15 и 6 в различных системах счисления.

Решение. Переведем числа 15 и 6 в двоичную и восьмеричную системы счисления и выполним сложение, используя таблицы сложения (см. выше).

Десятичная:

$$15_{10} + 6_{10}$$

$$\begin{array}{r} 1 \\ 15 \\ + 6 \\ \hline 21 \end{array}$$

$\left. \begin{array}{l} \\ \end{array} \right\} 5+6=11=10+1$
 $\left. \begin{array}{l} \\ \\ \end{array} \right\} 1+1=2$

Двоичная:

$$11111_2 + 110_2$$

$$\begin{array}{r} 111 \\ + 1111 \\ \hline 0110 \\ \hline 10101 \end{array}$$

$\left. \begin{array}{l} \\ \end{array} \right\} 1+0=1$
 $\left. \begin{array}{l} \\ \\ \end{array} \right\} 1+1=2=10$
 $\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} 1+1+1=3=11$
 $\left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} 1+1=2=10$

Восьмеричная:

$$17_8 + 6_8$$

$$\begin{array}{r} 1 \\ 17 \\ + 6 \\ \hline 25 \end{array}$$

$\left. \begin{array}{l} \\ \end{array} \right\} 7+6=13=8+5$
 $\left. \begin{array}{l} \\ \\ \end{array} \right\} 1+1=2$

Ответ: $15+6=21_{10}=10101_2=25_8$

2) Вычислим сумму чисел 438 и 5616. Результат представим в восьмеричной системе счисления.

Решение: переведем число 5616 в восьмеричную систему счисления, используя поразрядный способ перевода разложением на тэтрады и триады:

$$\begin{array}{rcccc}
 56_{16} = & 5 & 6 & a_2 = & 1 & 010 & 110 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 x = & 101 & 0110 & a_8 = & 1 & 2 & 6
 \end{array}$$

Пользуясь правилами сложения в восьмеричной системе счисления, получаем:

$$\begin{array}{r}
 1 \\
 43_8 \\
 + 126_8 \\
 \hline
 171_8 \\
 \quad \quad \quad \downarrow \\
 \quad \quad \quad 3 + 6 = 9 = 8 + 1
 \end{array}$$

Ответ: $43_8 + 56_{16} = 171_8$

Вычитание осуществляется по тем же правилам, что и в десятичной системе счисления. При вычитании из меньшего числа большего производится заем из старшего разряда.

Пример:

Вычислим разность $X - Y$ двоичных чисел, если $X = 1010100_2$ и $Y = 1000010_2$. Результат представим в двоичном виде.

Решение:

$$\begin{array}{r}
 1010100 \\
 - 1000010 \\
 \hline
 10010 \\
 \begin{array}{l}
 \quad \quad \quad \downarrow \\
 \quad \quad \quad 0 - 0 = 0 \\
 \quad \quad \quad 10 - 1 = 1 \\
 \quad \quad \quad 0 - 0 = 0 \\
 \quad \quad \quad 0 - 0 = 0 \\
 \quad \quad \quad 1 - 0 = 1 \\
 \quad \quad \quad 0 - 0 = 0 \\
 \quad \quad \quad 1 - 1 = 0
 \end{array}
 \end{array}$$

Ответ: 10010_2

Замечание. Если вам трудно складывать или вычитать в системах счисления, отличных от десятичной, можете перевести числа в десятичную систему счисления, выполнить арифметические действия, а затем результат перевести в требуемую в ответе систему счисления.

Выполняя **умножение** многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Таблица умножения в двоичной системе:

*	0	1
0	0	0
1	0	1

Таблица умножения в восьмеричной системе:

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	11	25	34	43	52	61

Умножение многозначных чисел в различных позиционных системах счисления происходит по обычной схеме, применяемой в десятичной системе счисления, с последовательным умножением множимого на очередную цифру множителя.

Пример:

Перемножим числа 15 и 12.

Десятичная

$$15_{10} \cdot 12_{10}$$

$$\begin{array}{r} 15 \\ \times 12 \\ \hline + 30 \\ 15 \\ \hline 180 \end{array}$$

Двоичная

$$1111_2 \cdot 1100_2$$

$$\begin{array}{r} 1111 \\ \times 1100 \\ \hline + 1111 \\ 1111 \\ \hline 10110100 \end{array}$$

Восьмеричная:

$$17_8 \cdot 14_8$$

$$\begin{array}{r} 17 \\ \times 14 \\ \hline + 74 \\ 17 \\ \hline 264 \end{array}$$

Ответ: $15 \cdot 12 = 180_{10} = 10110100_2 = 264_8$

Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления. Следует только грамотно пользоваться теми цифрами, которые входят в алфавит используемой системы счисления.

двоичная
система

$$\begin{array}{r} 100011 \quad | \quad 1110 \\ - 1110 \quad | \quad 10,1 \\ \hline 1110 \\ - 1110 \\ \hline 0 \\ 110 \\ \hline 0 \end{array}$$

Ответ: $10,1_2$

восьмеричная
система

$$\begin{array}{r} 13351 \quad | \quad 163 \\ - 1262 \quad | \quad 63 \\ \hline 531 \\ - 531 \\ \hline 0 \end{array}$$

Ответ: 63_8

Задание для практической работы

Вариант 1

1. а) $1\ 101\ 100\ 000(2) + 10\ 110\ 110(2)$;

б) $101\ 110\ 111(2) + 1\ 000\ 100\ 001(2)$;

в) $1\ 001\ 000\ 111,01(2) + 100\ 001\ 101,101(2)$;

г) $271,34(8) + 1\ 566,2(8)$;

д) $65,2(16) + 3CA,8(16)$.

2. а) $1\ 011\ 001\ 001(2) - 1\ 000\ 111\ 011(2)$;

б) $1\ 110\ 000\ 110(2) - 101\ 111\ 101(2)$;

в) $101\ 010\ 000,101\ 11(2) - 11\ 001\ 100,01(2)$;

г) $731,6(8) - 622,6(8)$; д) $22D,1(I6) - 123,8(16)$.

3. а) $1\ 011\ 001(2) * 1\ 011\ 011(2)$;

б) $723,1(8) * 50,2(8)$;

в) $69,4(16) * A, B(16)$.

Вариант 2

1. а) $1\ 010\ 101(2) + 10\ 000\ 101(2)$;

б) $1\ 111\ 011\ 101(2) + 101\ 101\ 000(2)$;

в) $100\ 100\ 111,001(2) + 100\ 111\ 010,101(2)$;

г) $607,54(8) + 1620,2(8)$;

д) $3BF, A(16) + 313D(16)$.

2. а) $1\ 001\ 000\ 011(2) - 10\ 110\ 111(2)$;

б) $111\ 011\ 100(2) - 10\ 010\ 100(2)$;

в) $1\ 100\ 110\ 110,001\ 1(2) - 11\ 111\ 110,01(2)$;

г) $1\ 360,14(8) - 1\ 216,4(8)$; д) $33B,6(16) - 11B,4(16)$.

3. а) $11\ 001(2) * 1\ 011\ 100(2)$;

б) $451,2(8) * 5,24(8)$;

в) $2B, A(16) * 36, 6(16)$.

Вариант 3

1. а) $100\ 101\ 011(2) + 111\ 010\ 011(2)$;
б) $1\ 001\ 101\ 110(2) + 1\ 101\ 100\ 111(2)$;
в) $1\ 010\ 000\ 100,1(2) + 11011110,001(2)$;
г) $674,34(8) + 1\ 205,2(8)$;
д) $2FE,6(16), + 3B,4(16)$.

2. а) $1\ 100\ 110\ 010(2) - 1\ 001\ 101\ 101(2)$;
б) $1\ 110\ 001\ 100(2) - 10\ 001\ 111(2)$;
в) $111\ 001\ 010,01(2) - 1\ 110\ 001,001(2)$;
г) $641,6(8) - 273,04(8)$;
д) $3CE, B8(16) - 39A, B8(16)$.

3. а) $1\ 010\ 101(2) * 1\ 011\ 001(2)$;
б) $1702,2(8) * 64,2(8)$;
в) $7,4(16) * 1D,4(16)$.

Вариант 4

1. а) $101\ 111\ 111(2) + 1\ 101\ 110\ 110(2)$;
б) $10\ 111\ 110(2) + 100\ 011\ 100(2)$;
в) $1\ 101\ 100\ 011,011\ 1(2) + 1100011,01(2)$;
г) $666,2(8) + 1\ 234,24(8)$;
д) $346,4(16) + 3F2,6(16)$.

2. а) $1\ 101\ 010\ 110(2) - 110\ 011\ 110(2)$;
б) $1\ 010\ 001\ 111(2) - 1\ 001\ 001\ 110(2)$;
в) $1\ 111\ 100\ 100,110\ 11(2) - 101\ 110\ 111,011(2)$;
г) $1\ 437,24(8) - 473,4(8)$;
д) $24A,4(16) - B3,8(16)$.

3. а) $101\ 011(2) * 100\ 111(2)$;
б) $1\ 732,4(8) * 34,5(8)$;
в) $36,4(16) * A, A(16)$.

Вариант 5

1. а) $1\ 100\ 011\ 010(2) + 11\ 101\ 100(2)$;
б) $10\ 111\ 010(2) + 1\ 010\ 110\ 100(2)$;
в) $1\ 000\ 110\ 111,011(2) + 1\ 110\ 001\ 111,001(2)$;
г) $1\ 745,5(8) + 1\ 473,2(8)$;
д) $24D,5(16) + 141,4(16)$.

2. а) $1\ 100\ 101\ 010(2) - 110\ 110\ 010(2)$;

- б) $110\ 110\ 100(2) - 110\ 010\ 100(2)$;
- в) $1101111\ 111,1(2) - 1\ 100\ 111\ 110,101\ 1(2)$;
- г) $1\ 431,26(8) - 1\ 040,3(8)$;
- д) $22C,6(16) - 54,2(16)$.
- 3. а) $1\ 001\ 001(2) * 11\ 001(2)$;
- б) $245,04(8) - 112,2(8)$;
- в) $4B,2(16) * 3C,3(16)$.

Вариант 6

- 1. а) $1\ 000\ 011\ 101(2) + 101\ 000\ 010(2)$;
- б) $100\ 000\ 001(2) + 1\ 000\ 101\ 001(2)$;
- в) $101\ 111\ 011,01(2) + 1\ 000\ 100,101(2)$;
- г) $1\ 532,14(8) + 730,16(8)$;
- д) $BB,4(16) + 2F0,6(16)$.

- 2. а) $1\ 000\ 101\ 110(2) - 1\ 111\ 111(2)$;
- б) $1\ 011\ 101\ 000(2) - 1\ 001\ 000\ 000(2)$;
- в) $1\ 000\ 101\ 001,1(2) - 1\ 111\ 101,1(2)$;
- г) $1\ 265,2(8) - 610,2(8)$; д) $409,D(16) - 270,4(16)$.

- 3. а) $111\ 010(2) * 1\ 100\ 000(2)$;
- б) $1\ 005,5(8) * 63,3(8)$;
- в) $4A,3(16) * F,6(16)$.

Вопросы для самоконтроля

1. Правила выполнения арифметических операций в разных СС

Практическая работа № 5. Основные операции алгебры логики

Цель: изучить основные операции алгебры логики. Научиться строить таблицы истинности основных операций алгебры логики

Теоретический материал

Логика – наука о законах и формах мышления

Высказывание (суждение) – некоторое предложение, которое может быть истинно (верно) или ложно

Утверждение – суждение, которое требуется доказать или опровергнуть

Рассуждение – цепочка высказываний или утверждений, определенным образом связанных друг с другом

Умозаключение – логическая операция, в результате которой из одного или нескольких данных суждений получается (выводится) новое суждение

Логическое выражение – запись или устное утверждение, в которое, наряду с постоянными, обязательно входят переменные величины (объекты). В зависимости от значений этих переменных логическое выражение может принимать одно из двух возможных значений: ИСТИНА (логическая 1) или ЛОЖЬ (логический 0)

Сложное логическое выражение – логическое выражение, составленное из одного или нескольких простых (или сложных) логических выражений, связанных с помощью логических операций.

Алгебра логики – это наука об общих правилах и законах действий над логическими переменными и высказываниями.

Операции алгебры логики:

1. Самой простой логической операцией является операция **НЕ**, по-другому ее часто называют **отрицанием**, дополнением или **инверсией** и обозначают NOT ().

Если A – истинно, то \bar{A} – ложно и наоборот. Результат отрицания всегда противоположен значению аргумента. Логическая операция НЕ является унарной, т.е. действие выполняется над одним операндом. Таблица истинности:

A	\bar{A}
0	1
1	0

2. Логическое **И** еще часто называют **конъюнкцией**, или **логическим умножением**. Операция И (обозначается «И», «and», «&», $A \cdot B$) имеет результат «истина» только в том случае, если оба ее операнда истинны.

Таблица истинности $F = A \wedge B$:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

3. Логическое ИЛИ – дизъюнкция, или логическое сложение. Операция ИЛИ (обозначается «ИЛИ», «or», $A+B$, $A \vee B$) дает «истину», если значение «истина» имеет хотя бы один из операндов. Разумеется, в случае, когда справедливы оба аргумента одновременно, результат по-прежнему истинный.

Таблица истинности $F = A \vee B$:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Операции И, ИЛИ, НЕ образуют полную систему логических операций, из которой можно построить сколь угодно сложное логическое выражение. В вычислительной технике также часто используется операции импликация и эквивалентность.

4. Логическое следование: импликация – связывает два простых логических выражения, из которых первое является условием (A), а второе (B) – следствием из этого условия. Результатом импликации является ЛОЖЬ только тогда, когда условие A истинно, а следствие B ложно. Обозначается символом «следовательно» и выражается словами ЕСЛИ ... , ТО ...

Таблица истинности $F = A \rightarrow B$:

A	B	F
0	0	1
0	1	1
1	0	0
1	1	1

5. Логическая равнозначность: эквивалентность – определяет результат сравнения двух простых логических выражений A и B. Результатом эквивалентности является новое логическое выражение, которое будет истинным тогда и только тогда, когда оба исходных выражения одновременно истинны или ложны. Обозначается символом «эквивалентности». **Таблица истинности $F = A \leftrightarrow B$:**

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Порядок выполнения логических операций в сложном логическом выражении:

1 инверсия \rightarrow 2 Конъюнкция \rightarrow 3 Дизъюнкция \rightarrow 4 Импликация \rightarrow 5 Эквивалентность

Для изменения указанного порядка выполнения операций используются круглые скобки.

6. Штрих Шеффера, $A|B$ или антиконъюнкция, по определению это отрицание конъюнкции $F = A | B = \overline{A \wedge B}$:

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

7. Стрелка Пирса, $A \downarrow B$ или антидизъюнкция, по определению $F = A \downarrow B = \overline{A \vee B}$:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

8. Сумма по модулю два, $A \oplus B$ или антиэквивалентность, по определению $F = A \oplus B = \overline{A \leftrightarrow B}$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Задания практической работы

1 Напишите следующие высказывания в виде логических выражений.

Вариант 1

- 1 Число 17 нечетное и двузначное.
- 2 Если Маша – сестра Саши, то Саша – брат Маши.
- 3 Голова думает тогда и только тогда, когда язык отдыхает.

Вариант 2

- 1 На уроке физики ученики выполняли лабораторную работу и сообщали результаты исследований учителю.
- 2 Ты можешь купить в магазине продукты, если у тебя есть деньги.
- 3 При замерзании воды выделяется тепло.

Вариант 3

- 1 Неверно, что корова – хищное животное.
- 2 На уроке информатике необходимо соблюдать правила техники безопасности.
- 3 Если на улице дождь, то асфальт мокрый.

Вариант 4

- 1 Если компьютер включен, то можно на нем работать.
- 2 Катя любит писать сочинения или решать задачи.
- 3 Тише едешь – дальше будешь.

Вариант 5

1 Если число делится на 2, то оно – четное.

2 Земля движется по круговой или эллиптической орбите.

3 Водительские права можно получить тогда и только тогда, когда тебе исполнится 18 лет.

2 Даны высказывания, определить истинность высказываний:

Вариант 1

Даны высказывания: $A = \{3+3=7\}$, $B = \{3+3=6\}$.

Определить истинность высказываний:

$A, B, A \& B, A, B, A \vee B, A \rightarrow B, A \leftrightarrow B$.

Вариант 2

Даны высказывания: $A = \{2+3=5\}$, $B = \{2*2=4\}$.

Определить истинность высказываний:

$A, B, A \& B, A, B, A \vee B, A \rightarrow B, A \leftrightarrow B$.

Вариант 3

Даны высказывания: $A = \{5*5=25\}$, $B = \{5+5=11\}$.

Определить истинность высказываний:

$A, B, A \& B, A, B, A \vee B, A \rightarrow B, A \leftrightarrow B$.

Вариант 4

Даны высказывания: $A = \{7+3=10\}$, $B = \{7-3=4\}$.

Определить истинность высказываний:

$A, B, A \& B, A, B, A \vee B, A \rightarrow B, A \leftrightarrow B$.

Вариант 5

Даны высказывания: $A = \{10-3=7\}$, $B = \{10-7=3\}$.

Определить истинность высказываний:

$A, B, A \& B, A, B, A \vee B, A \rightarrow B, A \leftrightarrow B$.

3 Постройте отрицание для высказываний:

Вариант 1

Все ребята умеют плавать.

Вариант 2

Невозможно создать вечный двигатель.

Вариант 3

Каждый человек – художник.

Вариант 4

Человек все может.

Вариант 5

Сегодня в театре идет опера «Евгений Онегин».

4 Найти значение логических выражений.

Вариант 1

$F = (1 \vee 1) \vee (1 \vee 0)$

Вариант 2

$F = (0 \& 0) \& (1 \& 1)$

Вариант 3

$F = (0 \& 1) \vee (0 \& 1)$

Вариант 4

$F = (0 \vee 0) \& (1 \& 1)$

Вариант 5

$F = (1 \vee 0) \& (0 \& 1)$

Контрольные вопросы:

- 1 Запишите алгоритм построения таблицы истинности.
- 2 Запишите следующее высказывание в виде логических выражений: «Я поеду к бабушке и, если встречу там друзей, то интересно проведу время».
- 3 Запишите приоритет выполнения логических операций.

Практическая работа № 6. Построение таблиц истинности

Цель: освоить алгоритм построения таблиц истинности для логических функций; научиться определять и анализировать функции проводимости переключательных схем.

Теоретический материал

Часть 1. Построение таблиц истинности для логических функций

Логическая функция - это функция, в которой переменные принимают только два значения: **логическая единица** или **логический ноль**. Истинность или ложность сложных суждений представляет собой функцию истинности или ложности простых. Эту функцию называют **булевой функцией суждений** $f(a, b)$.

Любая логическая функция может быть задана с помощью таблицы истинности, в левой части которой записывается набор аргументов, а в правой части - соответствующие значения логической функции. При построении таблицы истинности необходимо учитывать порядок выполнения логических операций.

Порядок выполнения логических операций в сложном логическом выражении:

1. инверсия;
2. конъюнкция;
3. дизъюнкция;
4. импликация;
5. эквивалентность.

Для изменения указанного порядка выполнения операций используются скобки.

На основе анализа логической связи между высказываниями делается логический вывод. Для получения логического вывода составляется таблица *истинности*, в которой записывают все возможные комбинации каждого простого высказывания.

Алгоритм построения таблиц истинности для сложных выражений:

1. Определить количество строк:

количество строк = 2^n + строка для заголовка,

n - количество простых высказываний.

2. Определить количество столбцов:

количество столбцов = количество переменных + количество логических операций;

- определить количество переменных (простых выражений);
- определить количество логических операций и последовательность их выполнения.

3. Заполнить столбцы результатами выполнения логических операций в обозначенной последовательности с учетом таблиц истинности основных логических операций.

Логические операции

Инверсия (логическое отрицание)

Операция, выражаемая словом "не", называется *логическим отрицанием (инверсией)* делает истинное выражение ложным и, наоборот, ложное – истинным. Обозначается « $\bar{\quad}$ ».

Обозначение: НЕ, $\bar{O}A$, \bar{A} , NOT A

Таблица истинности для логического выражения A имеет вид

A	\bar{A}
1	0
0	1

Конъюнкция (логическое умножение)

Операция, выражаемая связкой "и", называется *логическим умножением (конъюнкцией)* и обозначается "U" (может также обозначаться знаками «?» (точка) или &). Высказывание $A \cup B$ истинно тогда и только тогда, когда оба высказывания A и B истинны.

Обозначение: A и B , $A \cup B$, $A \cdot B$, $A \text{ AND } B$

Таблица истинности для логических переменных A и B

A	B	$A \cup B$
1	1	1
1	0	0
0	1	0
0	0	0

Дизъюнкция (логическое сложение)

Операция, выражаемая связкой "или" (в неисключающем смысле этого слова), называется *логическим сложением (дизъюнкцией)* и обозначается знаком U (или +). Высказывание $A \cup B$ ложно тогда и только тогда, когда оба высказывания A и B ложны.

Обозначение: A ИЛИ B , $A \cup B$, $A + B$, $A \text{ OR } B$

Таблица истинности для логических переменных A и B

A	B	$A \cup B$
1	1	1
1	0	1
0	1	1
0	0	0

В алгебре логики любую логическую функцию можно выразить через основные логические операции, записать ее в виде логического выражения и упростить ее, применяя законы логики и свойства логических операций. По формуле логической функции легко рассчитать ее таблицу истинности. Необходимо только учитывать порядок выполнения логических операций (приоритет) и скобки. Операции в логическом выражении выполняются слева направо с учетом скобок.

Приоритет выполнения логических операций:

- инверсия,
- конъюнкция,
- дизъюнкция

Задание 1

Составить таблицу истинности для заданного логического выражения. Вид выражения выбрать из табл. 1 в соответствии с номером варианта.

Варианты для задания 1

Таблица 1

Варианты для задания 1

Вариант 1	Вариант 6	Вариант11
$(x + \bar{y}) (x \sim yz)$	$((x \vee \bar{y})z) \rightarrow ((x \sim z) + y)$	$((x \sim z) + y) \cdot (x yz)$
Вариант 2	Вариант 7	Вариант 12
$\bar{x} \rightarrow (z \sim (y + x\bar{z}))$	$(x \vee \bar{y})z \rightarrow ((x \downarrow y) z)$	$\bar{x} \rightarrow (\bar{z} \sim (y + xz))$
Вариант 3	Вариант 8	Вариант 13
$(xz \rightarrow y) (xy + xz)$	$(x + (yz)) (xy)$	$(x \sim (\bar{y} + z)) (xy)$
Вариант 4	Вариант 9	Вариант 14
$(\bar{x} \vee y) \downarrow \bar{z} \rightarrow ((x + y) z)$	$(x + y\bar{z}) \rightarrow (z \sim (y \downarrow (x \vee \bar{z})))$	$(x\bar{y}) (x \sim yz)$
Вариант 5	Вариант 10	Вариант 15
$(x \sim \bar{y}) (x \downarrow (\bar{y}z + \bar{x}))$	$((x \vee \bar{y})z) \rightarrow (x \sim y)$	$((x \downarrow \bar{y})z) \rightarrow ((x \sim z) + y)$

Пример выполнения задания

Составим таблицу истинности для выражения $(x \sim z) | ((x \blacksquare y) \sim (y \blacksquare z))$. Заданное выражение содержит три логические переменные. Столбцы 1, 2, 3 таблицы истинности (табл. 1.12) содержат возможные комбинации значений переменных x, y, z . Таблица истинности будет содержать 8 строк по количеству возможных различных комбинаций значений переменных.

Таблица истинности с промежуточными результатами вычислений

1	2	3	4	5	6	7	8
x	y	z	$\sim z$	$\cdot y$	$\cdot z$	$(x \cdot y) \sim (y \cdot z)$	$(x \sim z) ((x y) \sim (yz))$
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	1
1	0	0	0	0	0	1	1
1	0	1	1	0	0	1	0
1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	0

Таблица 1.12

Таблица истинности с промежуточными результатами вычислений

Столбцы 4, 5, 6, 7 содержат результаты промежуточных вычислений, в столбце 8 находятся значения логического выражения для каждой комбинации значений переменных.

Задание 2

Построить функциональную схему для заданной логической функции F . Вид выражения выбрать из табл. 1.13 в соответствии с номером варианта.

Варианты для задания 2

Таблица 1.13

Вариант 1	Вариант 6	Вариант11
$F = A + B + A \blacksquare B$	$F = A \vee (B + C) + AB$	$F = C + AB + (AB + A)$

Вариант 2	Вариант 7	Вариант 12
$F = (A \vee C) A + (B \vee C)$	$F = (A \vee B)(A + B)$	$F = (B \vee C) + A \vee AB$
Вариант 3	Вариант 8	Вариант 13
$f = AB + c + A$	$F = AB \vee AC + C$	$F = A \vee B + CA$
Вариант 4	Вариант 9	Вариант 14
$F = A \vee B + A \blacksquare B$	$F = (AB + C) \vee BC$	$F = A + B(A \vee C)$
Вариант 5	Вариант 10	Вариант 15
$F = CA \vee (A + B)$	$F = AC + C \vee B$	$F = A(B + C) + A \vee B$

Пример выполнения задания

Допустим, задана логическая функция: $F = BA + \bar{B}A + C\bar{B}$. Построение схемы удобнее выполнять, начиная с ее выхода.

Первый этап. Функция F рассматривается как результат операции «ИЛИ» (логическое сложение), примененной к функциям BA , $\bar{B}A$ и $C\bar{B}$. Используем логический элемент «ИЛИ» и получаем первый фрагмент схемы (рис. 1.8).

Второй этап. К входам элемента «ИЛИ» подключаются логические элементы «И», входными переменными которых являются A , B , C и их инверсии (рис.1.9).

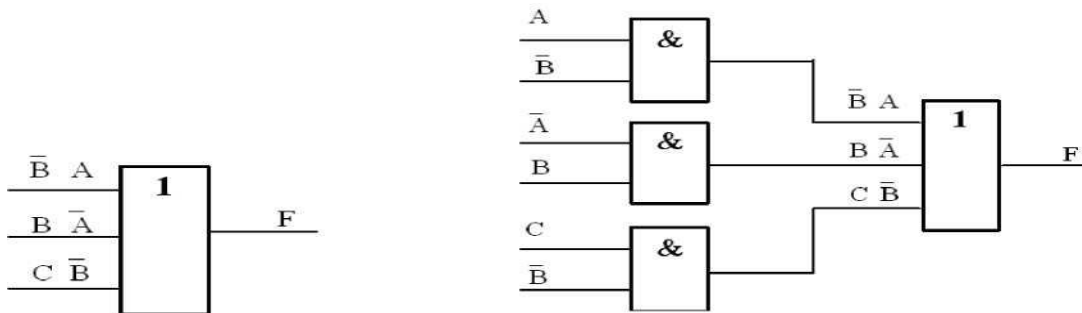


Рис. 1.8. Первый этап построения

Третий этап. Для получения инверсий для A и B на соответствующих входах ставим инверторы и соединяем одноименные входные переменные в одну линию (рис. 1.10). Построение схемы закончено.

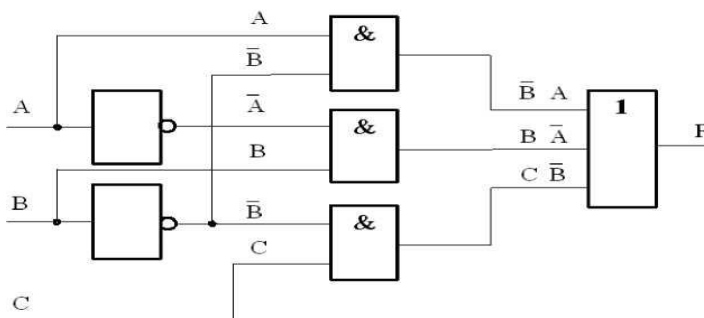


Рис. 1.10. Результат построения схемы

Вопросы для самоконтроля:

1. Каков порядок выполнения логических операций?
2. По какому правилу выполняется операция отрицание?
3. По какому правилу выполняется операция конъюнкция?
4. По какому правилу выполняется операция дизъюнкция?
5. По какому правилу выполняется операция импликация?
6. По какому правилу выполняется операция эквиваленция?
7. Что называют таблицей истинности?

8. Каков алгоритм построения таблицы истинности?
9. Какие высказывания называют: тождественно истинными, тождественно ложными, равносильными?

Практическая работа № 7. Контрольное решение задач

Цель: освоить алгоритм построения таблиц истинности для логических функций, научиться определять и анализировать функции проводимости переключательных схем.

Задания:

Задание 1

- | | |
|---|---|
| 1 $F = A \vee \bar{B} \vee (\bar{A} \vee C)$ | 16 $F = A \leftrightarrow C \vee B \rightarrow A$ |
| 2 $F = A \rightarrow \bar{B} \vee C$ | 17 $F = A \leftrightarrow \bar{C} \vee B \rightarrow \bar{A}$ |
| 3 $F = B \vee (\bar{A} \leftrightarrow C)$ | 18 $F = (A \leftrightarrow C) \vee (B \rightarrow A)$ |
| 4 $F = \bar{B} \vee (A \leftrightarrow C)$ | 19 $F = A \leftrightarrow C \vee (B \rightarrow \bar{A})$ |
| 5 $F = A \wedge B \rightarrow \bar{B} \wedge C$ | 20 $F = A \leftrightarrow (C \vee B \rightarrow A)$ |
| 6 $F = A \wedge B \leftrightarrow \bar{B} \vee C$ | 21 $F = (\bar{A} \leftrightarrow C) \vee B \rightarrow A$ |
| 7 $F = (A \vee \bar{B}) \vee (\bar{A} \rightarrow C)$ | 22 $F = \bar{A} \leftrightarrow (C \vee \bar{B} \rightarrow A)$ |
| 8 $F = (A \rightarrow \bar{B}) \vee C$ | 23 $F = A \wedge (B \rightarrow \bar{C}) \wedge C$ |
| 9 $F = B \vee C \leftrightarrow \bar{A} \vee \bar{C}$ | 24 $F = A \wedge (B \leftrightarrow A) \vee C$ |
| 10 $F = \bar{B} \vee (A \wedge C \rightarrow B)$ | 25 $F = (C \vee \bar{B}) \vee (\bar{A} \vee C)$ |
| 11 $F = A \vee B \rightarrow \bar{B} \vee C$ | 26 $F = A \rightarrow \bar{B} \vee (C \rightarrow \bar{B})$ |
| 12 $F = A \wedge B \leftrightarrow \bar{B} \vee C$ | 27 $F = (A \wedge B \rightarrow \bar{B}) \wedge (C \vee \bar{A})$ |
| 13 $F = A \rightarrow \bar{B} \vee (\bar{A} \vee C)$ | 28 $F = \bar{B} \vee (A \leftrightarrow C) \wedge C$ |
| 14 $F = A \wedge B \rightarrow \bar{B} \vee C$ | 29 $F = A \wedge B \rightarrow \bar{B} \wedge C$ |
| 15 $F = B \vee (\bar{A} \leftrightarrow C) \wedge A$ | 30 $F = A \wedge B \leftrightarrow \bar{B} \vee C$ |

Задание 2

- | | |
|--|---|
| 1 $(A \leftrightarrow B) \vee \bar{A} \bar{B} \vee C$ | 16 $B \vee (A \leftrightarrow CB) \vee \bar{A} \bar{C}$ |
| 2 $(A \rightarrow B) \vee \bar{A} \bar{C} \vee BC$ | 17 $(AC \rightarrow B) \vee \bar{A} \bar{B} \bar{C}$ |
| 3 $(AC \rightarrow B) \vee \bar{A} \bar{C}$ | 18 $(\bar{A} \leftrightarrow C) \wedge (BC \rightarrow AB)$ |
| 4 $\bar{A} \bar{B} \vee (A \leftrightarrow C) \bar{B}$ | 19 $(B \rightarrow C) \vee (B \rightarrow AC)$ |
| 5 $(\bar{A} \rightarrow B) \wedge (\bar{A} \bar{C} \vee BC)$ | 20 $(AB \rightarrow C) \vee A \vee \bar{A} C$ |
| 6 $(A \leftrightarrow C) \vee \bar{A} \bar{B} \vee AC$ | 21 $(A \leftrightarrow C) \vee (\bar{A} \bar{B} \rightarrow C)$ |
| 7 $(A \leftrightarrow C) \vee \bar{A} \bar{B} \vee BC$ | 22 $(\bar{A} \bar{B} \rightarrow \bar{C}) \vee \bar{A} \bar{B} C$ |
| 8 $(C \leftrightarrow B) \vee \bar{A} \bar{C} \vee BC$ | 23 $(AB \rightarrow C) \vee \bar{A} \bar{C}$ |
| 9 $(BC \rightarrow A) \vee \bar{A} \bar{C}$ | 24 $(\bar{A} \rightarrow BC) \wedge (A \leftrightarrow C)$ |
| 10 $(AB \rightarrow C) \vee \bar{A} \bar{C}$ | 25 $(\bar{A} \leftrightarrow B) \vee (A \rightarrow BC)$ |
| 11 $(\bar{A} \rightarrow C) \wedge (\bar{B} \bar{C} \vee AB)$ | 26 $(\bar{A} \rightarrow \bar{B}) \wedge (\bar{C} \bar{A} \rightarrow B)$ |
| 12 $(\bar{A} \leftrightarrow B) \wedge (A \rightarrow BC)$ | 27 $(A \rightarrow \bar{B} C) \vee \bar{A} \bar{B} \vee BC$ |
| 13 $(B \rightarrow C) \vee \bar{A} \bar{B} \vee \bar{A} C$ | 28 $(A \rightarrow C) \vee \bar{A} \bar{B} \vee BC$ |
| 14 $(A \rightarrow \bar{B} C) \vee \bar{A} \bar{B} \vee \bar{B} \bar{C}$ | 29 $(\bar{A} \rightarrow \bar{B}) \wedge (\bar{B} \bar{A} \rightarrow C)$ |
| 15 $(AC \rightarrow \bar{B}) \vee \bar{B} \bar{C}$ | 30 $(AB \rightarrow \bar{C}) \vee \bar{A} \bar{B} C$ |

Практическая работа № 8. Операционная система. Графический интерфейс пользователя. ПО внешних устройств. Подключение внешних устройств к ПК

Цель: Изучить основные понятия операционной системы и файловой системы. Приобрести умение выполнять создание, копирование, перемещение и удаление файлов и папок. выработать практические навыки обмена файлами между пользователями локальной компьютерной сети.

- закрепить знания о программном обеспечении;
- изучить методы работы с программным обеспечением.
- научиться устанавливать и деинсталлировать программы.
- научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью;

Теоретический материал

Операционная система (ОС) предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.

Основными функциями ОС являются:

1. загрузка программ в оперативную память (ОП) и управление ходом их выполнения;
2. обеспечение операций по обмену данными между выполняющейся программой и внешними устройствами;
3. обслуживание нестандартных ситуаций в ходе выполнения программы (например, если пользователь по каким-то причинам желает прекратить выполнение программы);
4. удаление выполненной программы из ОП и освобождение места для загрузки новой программы;
5. организация хранения программ и данных на внешних носителях;
6. организация взаимодействия пользователя и операционных систем – прием и выполнение команд пользователя;
7. выполнение различных вспомогательных функций, таких как форматирование дисковых устройств, копирование информации с одного дискового устройства на другое, проверка качества рабочих поверхностей дисковых носителей, служба времени (системные часы);
8. обеспечение защиты данных; и некоторые другие функции.

ОС – комплекс программных средств, обеспечивающих интерфейс нескольких видов:

- ⇒ интерфейс пользователя (взаимодействие между пользователем и программно-аппаратными средствами);
- ⇒ аппаратно-программный интерфейс (взаимодействие между аппаратными и программными средствами);
- ⇒ программный интерфейс (взаимодействие между разными видами ПО).

Операционная система может находиться на одном из внешних запоминающих устройств (гибком, жестком или лазерном диске). А образующие базовую систему ввода/вывода части ОС записываются в постоянную память (ПЗУ). Диск, на котором размещена ОС, называется системным диском. Загрузка ОС в ОП осуществляется с системного диска при включении ПК.

Организация файловой системы - одна из самых важных функций любой операционной системы. Вся информация на дисках организуется в виде файлов. Файл (от англ. file – картотека, архив)– это именованная область диска для постоянного хранения информации (программ, данных для их работы, текстов, рисунков и т. д.). Каждый файл имеет имя и тип (расширение), которые записываются через точку: имя.тип.

Для каждого файла, кроме имени и расширения, ОС хранит информацию о размере файла, дате и времени его создания или последней модификации, и несколько величин, называемых

атрибутами. Атрибуты – дополнительные параметры, определяющие свойства файлов: Read Only (Только для чтения); Hidden (Скрытый); System (Системный); Archive (Архивный).

Файлы по любому общему признаку, выбранному пользователем, объединяются в каталоги (папки). Каталог (папка, folder) – место на диске, в котором хранятся сведения о файлах: их имена и их атрибуты. На логическом уровне каталоги – это элементы иерархической структуры, необходимые для обеспечения удобного доступа к файлам, особенно, если файлов на диске слишком много. Каждый каталог имеет свое имя, задаваемое пользователем при его создании. Каталог может быть вложенным, т. е. находится в каталоге более высокого уровня. Корневой каталог является самым верхним уровнем вложенности иерархической структуры и организуется на диске ОС при форматировании диска. Корневой каталог обозначается обратной косой чертой (backslash) \ и указывается после имени физических ВЗУ: C:\, D:\ - имена корневых каталогов разделов жесткого диска; E:\ - корневой каталог

лазерного диска; A:\, B:\ - корневые каталоги гибких магнитных дисков.

Основным недостатком файлов MSDOS является их небольшой размер. Это недостаток устранен в операционных системах Windows 9x за счет введения нового понятия – «длинного» имени файла. **Длинные имена файлов** обладают следующими свойствами:

- они могут содержать до 255 символов включая расширение;
- они могут содержать пробелы;
- они могут содержать несколько точек;
- в имени файла нельзя ставить следующие символы: / - косую черту (слэш); \ - обратную косую черту (бекслэш); ? - знак вопроса; * - звездочку; > - знак больше; < - знак меньше; : - двоеточие; " - кавычку; | - вертикальную черту

В отличие от имен файлов, в именах **папок (директорий, каталогов)** расширение обычно не ставится, так как в этом нет особой необходимости.

В пределах одной папки могут находиться сколько угодно файлов, но имена файлов вместе с расширениями должны быть уникальными, то есть не должны повторяться. Зато не запрещено иметь в одной папке несколько файлов с одинаковыми именами, но разными типами: письмо.txt, письмо.doc

В процессе работы с файлами возникает необходимость создавать новые файлы, заменять одни файлы другими, перемещать их с одного места на другое, переименовывать, удалять.

Windows является наиболее популярной операционной системой с графическим интерфейсом и обеспечивает возможность многозадачности - одновременной работы нескольких приложений.

Простой и удобный интерфейс операционных систем семейства Windows обеспечивает естественность общения пользователя с компьютером.

Интерфейс пользователя - способ представления информации на экране, связующее звено между пользователем и компьютером.

После загрузки операционной системы Windows вся поверхность монитора - это **Рабочий стол**. На Рабочем столе как на поверхности обычного письменного стола - всё самое нужное для работы: значки, папки, Мой компьютер, ярлыки, корзина. В нижней части Рабочего стола располагается **панель задач**, на ней находится кнопка Start (**Пуск**), с которой начинается и заканчивается работа, а также ярлыки всех загруженных в оперативную память программ, между которыми можно переключаться. **Главное системное меню** “всплывает” после щелчка мыши кнопки Пуск. С его помощью можно выполнить различные операции: запустить программу, открыть документ, вызвать панель управления для настройки компьютера, провести поиск файла или папки, получить справку и т.д. Одно из основных понятий графического интерфейса ОС Windows - это окно.

Проводник – служебная программа Windows, предназначенная для навигации по файловой структуре компьютера и ее обслуживанию. Цель навигации состоит в обеспечении доступа к нужной папке и ее содержимому. Запуск проводника может осуществляться одним из способов:

- а) Пуск - Программы - Проводник

- b) Контекстное меню кнопки Пуск - Проводник
- c) Контекстное меню стандартной папки Мой компьютер Проводник

Окно Проводника разделено на две области: левую, называемую панелью папок, и правую, называемую панелью содержимого (рис. 1.) В левой панели показана структура папок.

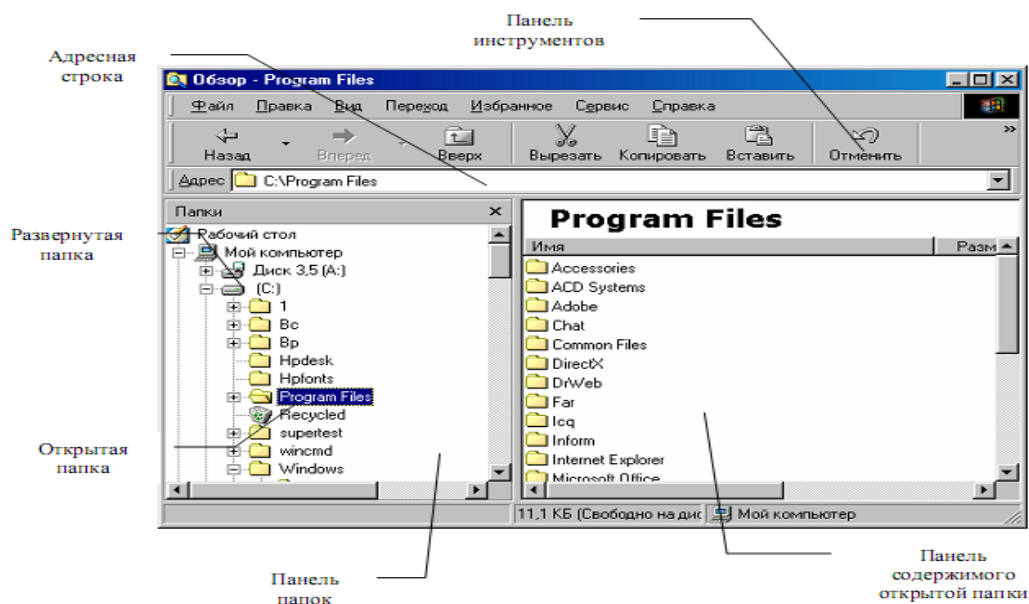


Рис. 1. Окно программы «Проводник»

Каждая папка может быть раскрыта щелчком левой кнопки мыши на ее значке, при этом ее содержимое отображается на правой панели, закрывается папка автоматически при раскрытии любой другой папки. Одна из папок в левой панели раскрыта всегда. Если папка имеет вложенные папки, то на левой панели рядом с папкой отображается узел, отмеченный знаком «+», с помощью щелчка на котором папку можно развернуть, при этом значок узла меняется на «-». Сворачиваются папки также при помощи щелчка, выполненном на узле. Изменение вида отображаемых в правой части окна объектов (в виде крупных или мелких значков, в виде списка или таблицы) выполняется с помощью команд пункта меню Вид.

Абоненты сети- объекты, генерирующие или потребляющие информацию.

Абонентами сети могут быть отдельные ЭВМ, промышленные роботы, станки с ЧПУ (станки с числовым программным управлением) и т.д. Любой абонент сети подключён к станции.

Станция- аппаратура, которая выполняет функции, связанные с передачей и приёмом информации.

Для организации взаимодействия абонентов и станции необходима физическая передающая среда.

Физическая передающая среда- линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

Одной из основных характеристик линий или каналов связи является скорость передачи данных (пропускная способность).

Скорость передачи данных- количество бит информации, передаваемой за единицу времени.

Обычно скорость передачи данных измеряется в битах в секунду (бит/с) и кратных единицах Кбит/с и Мбит/с.

Соотношения между единицами измерения: 1 Кбит/с =1024 бит/с; 1 Мбит/с =1024 Кбит/с; 1 Гбит/с =1024 Мбит/с.

Топология локальных сетей. Под топологией компьютерной сети обычно понимают физическое расположение компьютеров сети относительно друг друга и способ соединения их линиями.

Топология определяет требования к оборудованию, тип используемого кабеля, методы управления обменом, надежность работы, возможность расширения сети. Существует три основных вида топологии сети: шина, звезда и кольцо.

Информационный объект – это совокупность логически связанной информации.

Для хранения и передачи электронных информационных объектов используют съемные цифровые носители. К ним относятся:

- **съемный жесткий диск** – устройство хранения информации, основанное на принципе магнитной записи, информация записывается на жесткие (алюминиевые или стеклянные) пластины, покрытые слоем ферромагнитного материала,
- **дискета** – портативный носитель информации, используемый для многократной записи и хранения данных, представляющий собой помещенный в защитный пластиковый корпус гибкий магнитный диск, покрытый ферромагнитным слоем,
- **компакт-диск** – оптический носитель информации в виде пластикового диска с отверстием в центре, процесс записи и считывания информации которого осуществляется при помощи лазера (CD-ROM и DVD-диск - предназначенный только для чтения; CD-RW и DVD-RW информация может записываться многократно),
- **карта памяти или флеш-карта** – компактное электронное запоминающее устройство, используемое для хранения цифровой информации (они широко используются в электронных устройствах, включая цифровые фотоаппараты, сотовые телефоны, ноутбуки, MP3-плееры и игровые консоли),
- **USB-флеш-накопитель (сленг. флэшка)** – запоминающее устройство, использующее в качестве носителя флеш-память и подключаемое к компьютеру или иному считывающему устройству по интерфейсу USB.

Все программы и данные хранятся в долговременной (внешней) памяти компьютера в виде файлов.

Файл – это определенное количество информации (программа или данные), имеющее имя и хранящееся в долговременной (внешней) памяти.

Имя файла состоит из двух частей, разделенных точкой: собственно имя файла и расширение, определяющее его тип (программа, данные и т. д.). Собственно имя файлу дает пользователь, а тип файла обычно задается программой автоматически при его создании.

Тип файла	Расширение
Исполняемые программы	exe, com
Текстовые файлы	txt, rtf, doc
Графические файлы	bmp, gif, jpg, png, pds и др.
Web-страницы	htm, html
Звуковые файлы	wav, mp3, midi, kar, ogg
Видеофайлы	avi, mpeg

В операционной системе Windows имя файла может иметь до 255 символов, причем допускается использование русского алфавита, разрешается использовать пробелы и другие ранее запрещенные символы, за исключением следующих девяти: \:*\?"<>|. В имени файла

можно использовать несколько точек. Расширением имени считаются все символы, стоящие за последней точкой.

Роль расширения имени файла чисто информационная, а не командная. Если файлу с рисунком присвоить расширение имени TXT, то содержимое файла от этого не превратится в текст. Его можно просмотреть в программе, предназначенной для работы с текстами, но ничего вразумительного такой просмотр не даст.

Атрибуты файла устанавливаются для каждого файла и указывают системе, какие операции можно производить с файлами. Существует четыре атрибута:

- только чтение (R);
- архивный (A);
- скрытый (H);
- системный (S).

Атрибут файла «Только чтение».

Данный атрибут указывает, что файл нельзя изменять. Все попытки изменить файл с атрибутом «только чтение», удалить его или переименовать завершатся неудачно.

Атрибут файла «Скрытый».

Файл с таким атрибутом не отображается в папке. Атрибут можно применять также и к целым папкам. Надо помнить, что в системе предусмотрена возможность отображения скрытых файлов, для этого достаточно в меню Проводника Сервис – Свойства папки – вкладка Вид – Показывать скрытые файлы и папки.

Атрибут файла «Архивный».

Такой атрибут имеют практически все файлы, его включение/отключение практически не имеет никакого смысла. Использовался атрибут программами резервного копирования для определения изменений в файле.

Атрибут файла «Системный».

Этот атрибут устанавливается для файлов, необходимых операционной системе для стабильной работы. Фактически он делает файл скрытым и только для чтения. Самостоятельно выставить системный атрибут для файла невозможно.

Для изменения атрибутов файла необходимо открыть окно его свойств и включить соответствующие опции.

Существуют также дополнительные атрибуты, к ним относятся атрибуты индексирования и архивации, а также атрибуты сжатия и шифрования.

При передаче и хранении различных файлов необходимо учитывать объем этих файлов. Если объем слишком велик, можно создать архив файлов с помощью программ архиваторов (7-zip, WinRAR, WinZip).

Архивация – это сжатие файлов, то есть уменьшение их размера.

При создании архивов исполняемые программы, текстовые файлы, графические файлы, Web-страницы, звуковые файлы, видео файлы сжимаются по-разному.

Запись информации.

Запись информации - это способ фиксирования информации на материальном носителе.

Способы записи информации на компакт-диски:

1) **с помощью специальных программ записи** (Nero, CDBurnerXP, Burn4Free, CD DVD Burning и др.);

2) **через задачи для записи CD** (помещаем нужные объекты на диск с помощью перетаскивания или копирования, выбираем в задачах записи CD «записать файлы на компакт-диск»).

Способы записи информации на остальные съемные цифровые носители:

- 1) **копирование** (выделяем нужные объекты, нажимаем правой кнопкой мыши, в появившемся контекстном меню выбираем «копировать»; через контекстное меню правой кнопки мыши, выбирая «вставить», вставляем объекты на нужный цифровой носитель);
- 2) **перетаскивание** (выделяем нужные объекты, нажимаем левую кнопку мыши, удерживая её, перетаскиваем документы на нужный цифровой носитель).

Задания к практической работе

Часть 1

Задание 1. Создайте на диске D: в папке Студенты, папку Ваша группа, а в ней папку Ваша фамилия и в ней три папки База данных, Информатика, Презентации; в папке Информатика создайте три папки Тексты, Рисунки, таблицы (рис. 2) с использованием программы Мой компьютер.

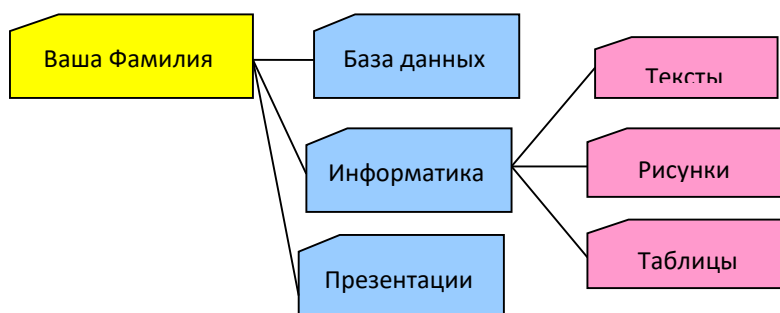


Рис. 2

Порядок работы

1. Откройте окно Мой компьютер. Установите стиль просмотра Крупные значки (Вид - Крупные значки).
2. Перейдите на диск D: в папку Студенты, создайте папку Ваша группа (Файл – Создать – Папка).
3. Откройте папку Ваша группа и создайте в ней папку Ваша фамилия (Файл – Создать – Папка)
4. Затем откройте папку Ваша фамилия создайте папки База данных, Информатика и Презентации.
5. Откройте папку Информатика и нажмите на панели содержимого папки правую клавишу мыши, затем в контекстном меню выберите Создать – Папку и присвойте имя папки Тексты и нажмите клавишу Enter. Затем создайте папки Рисунки и Таблицы.
6. Закройте окно Мой компьютер (Файл - Закрывать)

Задание 2. Создайте рисунок в стандартном приложении WindowsPaint и сохраните его на диск D: в папке Студенты, Ваша группа, Ваша фамилия, Информатика, Рисунки под именем Рисунок_Фамилия.jpg

Порядок работы

1. Откройте программу Paint (Пуск - Программы - Стандартные).
2. Задайте ширину рисунка, равную 300 точек и высоту – 200 точек (Рисунок→Атрибуты).
3. Используя различные инструменты и используя различные цвета создайте рисунок, где должны присутствовать объекты, нарисованные с использованием инструментов эллипс, линия, распылитель, карандаш, прямоугольник. Для заливки используйте инструмент Заливка (выбор основного цвета – щелчок левой клавиши, фонового – правой). Для коррекции использовать инструмент Ластик.

4. Сохраните изображение под именем Рисунок_Фамилия.jpg (Файл – Сохранить как - D:\Студенты\Ваша группа\Ваша фамилия\Информатика имя файла Рисунок_Фамилия, тип файла jpg)

5. Закройте окно Paint(Файл - Выход)

Задание 3. Создайте текст в стандартном приложении Windows Блокнот и сохраните его на диск D: в папке Студенты, Ваша группа, Ваша фамилия, Информатика, Тексты под именем Блокнот_Фамилия.txt

Порядок работы

1. Запустите текстовый редактор Блокнот (Пуск - Программы - Стандартные).
2. Вставьте текущую дату (Правка→Время и дата). Введите с клавиатуры свою фамилию имя и отчество и группу. Нажмите клавишу Enter. Напишите фразу: Практическая работа . Тема: «Операционная система. Графический интерфейс пользователя. Нажмите клавишу Enter.

3. Скопируйте с помощью Буфера обмена (Правка→Копировать и Правка→Вставить) данную фразу 5 раз.

4. Сохраните созданный документ под именем Блокнот_Фамилия.txt (Файл – Сохранить как - D:\Студенты\Ваша группа\Ваша фамилия\Информатика имя файла Блокнот_Фамилия, тип файла txt)

5. Закройте окно Блокнот (Файл - Выход)

Задание 4. В программе Проводник выполнить копирование файла Рисунок_Фамилия.jpg из папки Рисунки в папку Презентации; перемещение папки Тексты в папку Ваша фамилия; удалите (удалениенапок выполнять только в присутствии преподавателя) папку Информатика.

Порядок работы

1. Запустите программу Проводник (Пуск – Программы – (Стандартные) - Проводник).
2. На панели папок (в левой части Проводника) откройте папку **Рисунки**, при этом содержимое папки Рисунки отобразится в правой части Проводника. В правой части выделить файл Рисунок_Фамилия.jpg для копирования и перетащить влево левой клавишей мыши в папку **Презентации** при нажатой клавише Ctrl.

3. На панели папок (в левой части Проводника) откройте папку **Информатика**, при этом содержимое папки **Информатика** отобразится в правой части Проводника. В правой части выделить Папку **Тексты** для перемещения и перетащить влево левой клавишей мыши в папку **Ваша фамилия**.

4. Для удаления папки **Информатика** необходимо в правой части Проводника выделить папку **Информатика** (предварительно в левой части Проводника открыть папку **Ваша фамилия**) для удаления, нажать правую клавишу мыши и выбрать удаление папки.

часть 2

Задание №1. Определение общих ресурсов компьютера. Для этого:

- В операционной системе Windows найти на рабочем столе значок Сеть.
- Открыть папку, где будут видны все компьютеры, которые подключены в одну сеть.
- В данном окне появятся все компьютеры, которые подключены к сети. Открыть один из них. Посмотреть ресурсы компьютера, которыми можно воспользоваться. Такие ресурсы называются общими.

Задание № 2.

Предоставить доступ для пользователей локальной сети к папке на своем компьютере, подключенном к локальной сети. Для этого:

- В операционной системе Windows открыть окно папки Компьютер и на одном из

- дисков С: или D: создать свою папку. Назвать ее номером своей группы.
- Щелкнуть правой кнопкой мыши по значку папки и в контекстном меню папки выбрать команду Общий доступ.
- В появившемся диалоговом окне Дополнительный общий доступ установить флажок Открыть общий доступ к этой папке.
- Если все правильно сделано, то на диске (у вашей папки) появится значок, который показывает, что папка является общей.

Задание №3. Осуществить проверку возможности доступа к ресурсам компьютеров, подключенных к локальной сети. Для этого:

- Щелкнуть по значку Сеть, в окне появится список компьютеров, подключенных к локальной сети (смотри задание 1.)
- Открыть свой компьютер и внимательно посмотреть: какие из ресурсов доступны пользователям. Если название Вашей папки есть в перечне, то все сделано правильно.

Задание №4. Максимальная скорость передачи данных в локальной сети 100 Мбит/с. Сколько страниц текста можно передать за 1 сек, если 1 страница текста содержит 50 строк и на каждой строке - 70 символов?

Задание 5

Письменно ответьте на вопросы:

- 1 Носители информации - это
- 2 Пакет магнитных дисков, надетых на общую ось это –
- 3 Что является оптическим носителем информации?
- 4 В целях сохранения информации гибкие магнитные диски необходимо оберегать от ...
- 5 Что было самым первым носителем магнитной записи?
- 6 Материальная среда, используемая для записи и хранения информации
- 7 Какой компьютерный носитель информации вышел из массового употребления?
- 8 На ЭВМ первого поколения сменным носителем информации для устройств внешней памяти была ...
- 9 Что участвует в записи и чтении CD и DVD- дисков?
- 10 В какой системе счисления хранится информация в компьютере?

Задание 6

Сделайте вывод о проделанной работе

Вопросы для самоконтроля:

1. Для чего предназначена операционная система?
2. Каковы функции операционной системы?
3. Какие операционные системы различают по числу обрабатываемых задач? Что такое задача?
4. Какие операционные системы различают по типу интерфейса?
5. Приведите пример операционной системы с интерфейсом командной строки.
6. Приведите пример операционной системы с графическим интерфейсом.
7. Как операционная система управляет работой периферийных устройств компьютера?
8. Что такое драйвер?
9. Что такое файл?
10. Какова структура имени файла?
11. Какие расширения могут иметь программные приложения?
12. Какие расширения могут иметь текстовые файлы?
13. Какие расширения могут иметь графические файлы?

14. Какие расширения могут иметь видеофайлы?
15. Что такое каталог (папка)?
16. Как обозначается корневой каталог?
17. Как открыть и закрыть папку?
18. Что представляет собой программное приложение?
19. Что такое документ?
20. Что составляет основу работы пользователя с операционной системой Windows?
21. Перечислите элементы графического интерфейса Windows?
22. Что представляет собой Рабочий стол в Windows?
23. Что такое меню? Какие виды меню вы знаете?
24. Опишите правила работы с буфером обмена.
25. Для чего предназначен Проводник?

Дополнительная часть

26. Где располагается операционная система чаще всего?
27. Какая часть операционной системы хранится в постоянном запоминающем устройстве?
28. Что представляет собой WIMP-интерфейс операционной системы?
29. Что представляет собой SILK-интерфейс операционной системы?
30. Чем отличаются стандартные драйверы от загружаемых?
31. Что представляет собой шаблон? Какие символы используют в шаблонах? Приведите примеры.
32. Что такое FAT?
33. Какая информация содержится в FAT?
34. Опишите иерархическую структуру организации каталогов.
35. Какой каталог называется корневым?
36. Чем отличается корневой каталог от подкаталога?
37. Чем отличается имя файла от имени папки?
38. Что такое буфер обмена?
39. Что такое значок (пиктограмма)?
40. Чем отличается ярлык от пиктограммы?
41. Чем отличается счетчик от поля ввода?
42. Что представляет собой список в Windows?
43. Чем отличается кнопка выбора от флажка?
44. Как выполнить создание новой папки в папке Мой компьютер?
45. Как выполнить копирование файлов (папок) в программе Проводник?
46. Как выполнить перемещение файлов (папок) в программе Проводник?
47. Указать основное назначение компьютерной сети.
48. Указать объект, который является абонентом сети.
49. Указать основную характеристику каналов связи.
50. Что такое локальная сеть, глобальная сеть?
51. Что понимается под топологией локальной сети?
52. Какие существуют виды топологии локальной сети?
53. Охарактеризуйте кратко топологию «шина», «звезда», «кольцо».
54. Что такое протокол обмена?

Практическая работа № 9. Файл как единица хранения информации на компьютере. Защита информации, антивирусная защита

Цели:

- 1) - помочь учащимся закрепить представление о файлах и файловых системах, именах файлов, пути к файлам, типах файлов и их связи с программами и расширениями имен файлов, дать основные понятия, необходимые для работы на компьютере;
- 2) - воспитание информационной культуры учащихся, внимательности, аккуратности, дисциплинированности, усидчивости;
- 3) - развитие познавательных интересов, навыков работы с мышью и клавиатурой, самоконтроля, умения конспектировать
- 4) выработать практические навыки работы с антивирусными программами, навыки правильной работы с компьютером.
- 5) изучить классификацию вирусов, способы их распространения, способы борьбы с ними; изучить классификацию и назначение антивирусных программ.

Методические указания

Вирусы. Антивирусное программное обеспечение

Компьютерный вирус - программа способная самопроизвольно внедряться и внедрять свои копии в другие программы, файлы, системные области компьютера и в вычислительные сети, с целью создания всевозможных помех работе на компьютере.

Признаки заражения:

- o прекращение работы или неправильная работа ранее функционировавших программ
- o медленная работа компьютера
- o невозможность загрузки ОС
- o исчезновение файлов и каталогов или искажение их содержимого
- o изменение размеров файлов и их времени модификации
- o уменьшение размера оперативной памяти
- o непредусмотренные сообщения, изображения и звуковые сигналы
- o частые сбои и зависания компьютера и др.

Понятие вируса.

Официальное появление *первого компьютерного вируса* датируется 1981 годом, задолго до выхода первой версии Microsoft Windows. Этот вирус, замаскированный под компьютерную игру, атаковал наиболее популярный компьютер того времени – Apple II. Распространялся он с черепашей скоростью (с помощью дискет).

Согласно подсчетам экспертов, объем *malware* (общепринятое название всех видов вредоносных программ) возрастает более чем на 15 % в год. Согласно данным компании Sophos, разработчика антивирусных программ, каждый день появляются примерно 30 новых вирусов, а перечень активных вирусов пополняется 10 тыс. новых наименований в год.

Вирус – это часть программного кода, которая тиражируется путем добавления в другой объект, обычно незаметно и без разрешения пользователя.

Часть 1

Задание 1

Предложите варианты имен и типов для перечисленных ниже файлов.

Содержание

Имя

Тип

Полное имя файла

Фото моей семьи

Рецепт яблочного пирога

Буклет Моя школа

Открытое письмо Биллу Гейтсу

BillG

doc

BillG.doc

Семейный альбом Моя родословная

Репродукция картины Малевича

Чёрный квадрат

Петиция директору школы об увеличении количества уроков информатики

Реферат по истории

Реклама концерта рок-группы

Статья в журнал

Информатика и образование

Задание 2

Предложите варианты программ, открывающих файлы с тем или иным расширением.

Расширение имени файла

Программа

ТХТ

DOC

RTF

BMP

ARJ

HTML

Задание 3

Выпиши в один столбик правильные имена файлов, а во второй правильные имена каталогов:


Письмо.18, letter.txt, WinWord, письмо.doc, Школа?234, Мои документы, роза.bmp, stop12.exe, 1С, red.com

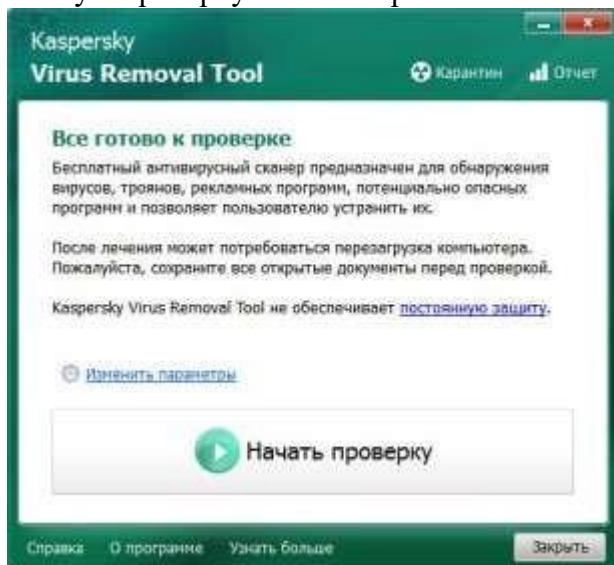
Часть 2

Задание 1. Обновить через Интернет антивирусную программу, установленную на Вашем компьютере. Выполнить проверку папки «Мои документы» на вирусы. Дать характеристику этой программы.

Задание № 2

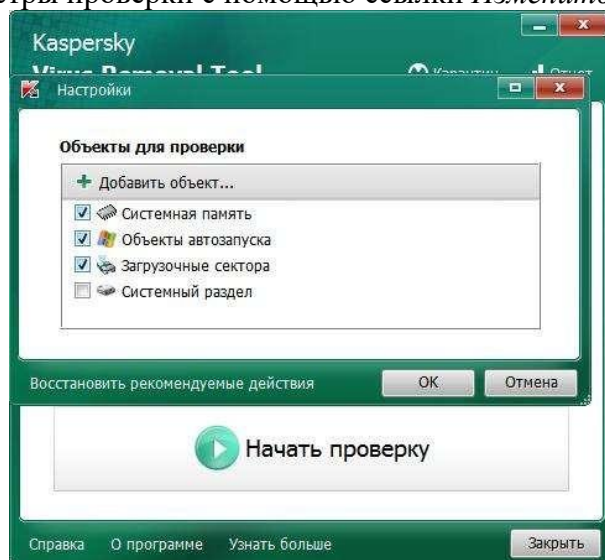
1.1. Запустите утилиту Антивируса Касперского **Kaspersky Virus Removal Tool**

(программа находится в папке материалов PZ-8 и обозначена иконкой ) и выполнить минимальную проверку компьютера.



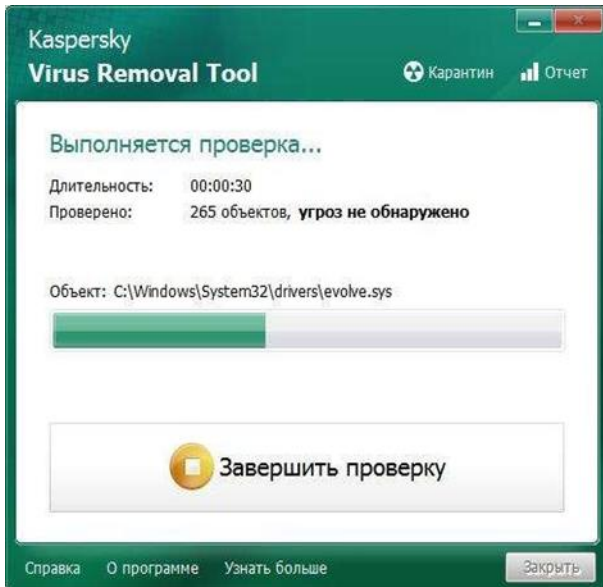
Сделайте скриншот и вставьте его в протокол выполнения работы.

1.2 Проверьте параметры проверки с помощью ссылки *Изменить параметры*



При указанных параметрах указаны начальные объекты проверки.
Сделайте скриншот стандартных параметров проверки и вставьте его в протокол выполнения работы.

1.3 Запустите проверку и дождитесь результатов.



Сделайте скриншот результатов проверки и вставьте его в протокол выполнения работы.

2 Измените параметры проверки – назначьте проверку папки с распакованными заданиями практического задания №8 (PZ-8)



Сделайте скриншот настройки параметров проверки и вставьте его в протокол выполнения работы.

2.1 Выполните проверку указанной папки и сделайте скриншот результатов проверки и вставьте его в протокол выполнения работы.

2.2 При окончании поиска изучите отчет поиска.

Сделайте скриншот отчета и карантина проверки и вставьте его в протокол выполнения работы.

Задание: 3-5

Используя таблицу вариантов вопросов найти ответы и записать их в отчет.

ФИО	Задание 3	Задание 4	Задание 5
Буханова Ольга	1.	30	AVG AntiVirus Free 2017
Вишневская Наталья	2.	29	AppRemover 3.1.24.1

Войтов Алексей	3.	28	Avira Free Antivirus
Габлеева Татьяна	4.	27	Comodo Cleaning Essentials 2.5
Гущина Екатерина	5.	26	McAfee Cloud AV 0.5.157.3 Beta
Ефремова Людмила	6.	25	Panda Free Antivirus 18.01.00
Женевская Анастасия	7.	24	Microsoft Security Essentials RU 4.10.209.0 Final
Жорова Алёна	8.	23	BitDefender Antivirus Free Edition 2017
Журавлёва Вероника	9.	22	Kaspersky Free Anti-Virus 2017
Илюкевич Иван	10.	21	Avast! Free Antivirus 2017
Карасёв Владимир	11.	20	360 Total Security
Кильдюшева Валерия	12.	19	Ad-Aware Free Antivirus+
Коржуев Илья	13.	18	Avetix Antivirus Free 5.0.159

Перечень вопросов

1. Охарактеризуйте информацию и ее свойства.
2. Что является предметом и объектом защиты информации?
3. Приведите классификацию конфиденциальной информации.
4. Охарактеризуйте свойства достоверности и своевременности информации.
5. Что понимается под угрозой информации?
6. Назовите разновидности угроз информации.
7. Приведите классификацию угроз информации.
8. Какие основные направления и методы реализации угроз Вам известны?
9. Поясните классификацию злоумышленников.
10. Охарактеризуйте причины и виды утечки информации.
11. Назовите и приведите примеры каналов утечки информации.
12. Классификация автоматизированных систем по уровню защищенности.
13. Назначение межсетевых экранов.
14. Классы защищенности межсетевых экранов.
15. Перечислите уровни информационной безопасности.
16. Что понимается под политикой безопасности?
17. Перечислите способы кодирования информации.
18. Методы шифрования информации.
19. Принцип работы электронной цифровой подписи.
20. Назовите существующие системы защиты от несанкционированного доступа к информации.
21. Перечислите каналы утечки информации.
22. Что такое не санкционированный доступ к компьютерным системам?
23. Принцип работы Закладки типа «Программного червя».
24. Принципиальное отличие компьютерного вируса от троянской программы.
25. Способы проникновения вируса на компьютер.
26. Характер воздействия компьютерных вирусов

27. Перечислите основные классы компьютерных вирусов.
28. Когда произошла первая эпидемия «компьютерного вируса»,
29. Перечислите названия безопасных программных утилит антивирусных программ.
30. Перечислите группы антивирусных программ

Задание № 6

При выполнении задания опишите особенности данной программы, технологии ее работы. Сайт загрузки (разработчика) данной программы. Название антивирусной программы выбрать согласно своему варианту .

Содержание отчета

В начале файла отчета должны быть указаны:

Протокол выполнение работы

Задание 1 Запустить утилиту Антивируса Касперского (Kaspersky Virus Removal Tool) и выпонить минимальную проверку компьютера.скриншот начала работы программы

- скриншот стандартных параметров проверки.
- скриншот результатов проверки

Задание 2 Выполнить проверку папки PZ-8

- скриншот настройки параметров проверки папки PZ-8 (на рабочем столе)
- скриншот отчета проверки
- скриншот карантина проверки

Практическая работа № 10. Форматирование электронного документа

Цель: изучение интерфейса текстового процессора MSWord, знакомство с панелями инструментов, изучение основных приемов создания и редактирования документов, основных приемов работы с абзацами.

Задание:

Для получения практических навыков работы в текстовом процессоре *MICROSOFT WORD* выполните следующее задание:

1) Создайте новый документ в папке *Практические студентов*. Для этого откройте необходимую папку, нажмите правой кнопкой мыши (п.к.м.) и в контекстном меню выберите команду *Создать – Документ MicrosoftWord*.

2) Наберите ниже приведенный текст, делая абзацные переходы (нажатие клавиши *Enter*) в конце смысловых предложений и отформатируйте его.


Текст к заданию:

Закон РФ «О персональных данных»

Настоящим Федеральным законом регулируются отношения, связанные с обработкой персональных данных, осуществляемой федеральными органами государственной власти, органами государственной власти субъектов Российской Федерации, иными государственными органами (далее - государственные органы), органами местного самоуправления, иными муниципальными органами (далее - муниципальные органы), юридическими лицами и физическими лицами с использованием средств автоматизации, в том числе в информационно-телекоммуникационных сетях, или без использования таких средств, если обработка персональных данных без использования таких средств соответствует характеру действий (операций), совершаемых с персональными данными с использованием средств автоматизации, то есть позволяет осуществлять в соответствии с заданным алгоритмом поиск персональных данных, зафиксированных на материальном носителе и содержащихся в картотеках или иных систематизированных собраниях персональных данных, и (или) доступ к таким персональным данным

Для форматирования текста выполните следующие действия:

1. Оформите заголовок текста следующим образом: шрифт – полужирный, курсивный *Times New Roman*, выравнивание *По центру*, размер 16 пт, вразрядку с интервалом 4 пт. Используйте команды на панели инструментов на вкладке **ГЛАВНАЯ** – *Шрифт*

 (Рисунок 1), *Шрифт – Дополнительно – Интервал – Разреженный* (Рисунок 2), предварительно выделив текст;

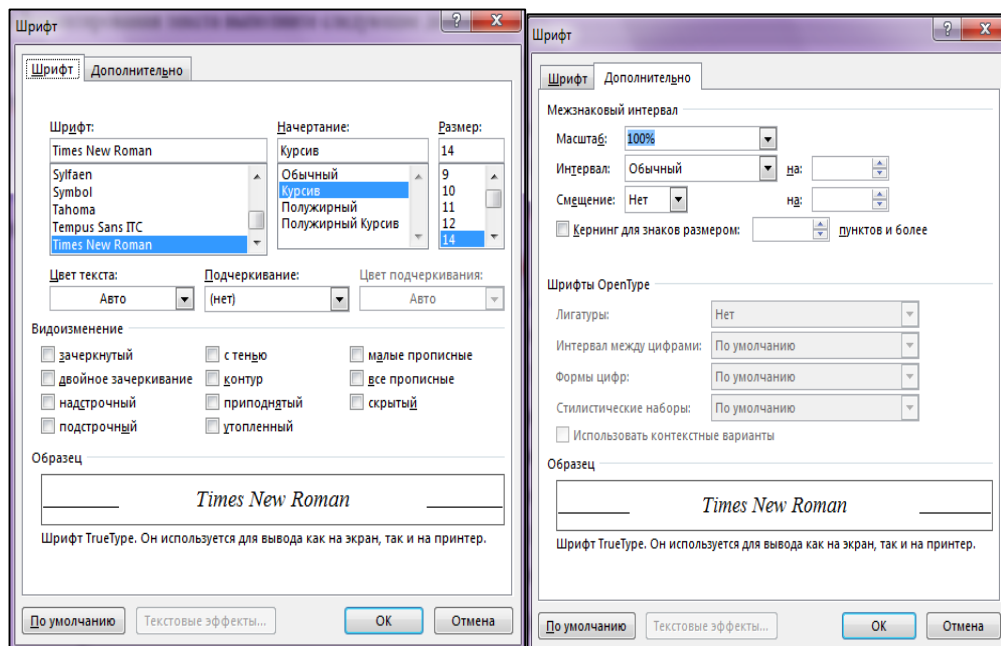
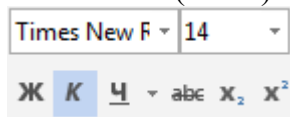


Рисунок 1

Рисунок 2

Текст оформите следующим образом, установите: отступ первой строки (Красная строка) (Вид/Линейка); выравнивание текста *По ширине*, шрифт – *Arial*, полужирный курсив, размер 12 пт. Для этого предварительно выделите текст и щелкните последовательно левой кнопкой мыши (л.к.м.) на кнопках панели инструментов *Ж*, *К*, *Размер шрифта - 12*;



2. Наведите указатель мыши на кнопку «Междустрочный интервал» и установите интервал равный – «1,15»

3. Попробуйте заключить последнее словосочетание в рамку. Для этого предварительно выделив текст, выполните последовательно команды *Граница* - *Границы и заливка*. В диалоговом окне *Границы* установите параметры: *Ширина*: 2,25 пт, *Цвет*: авто. В диалоговом окне *Заливка* установите параметры: *Узор*: 20%.

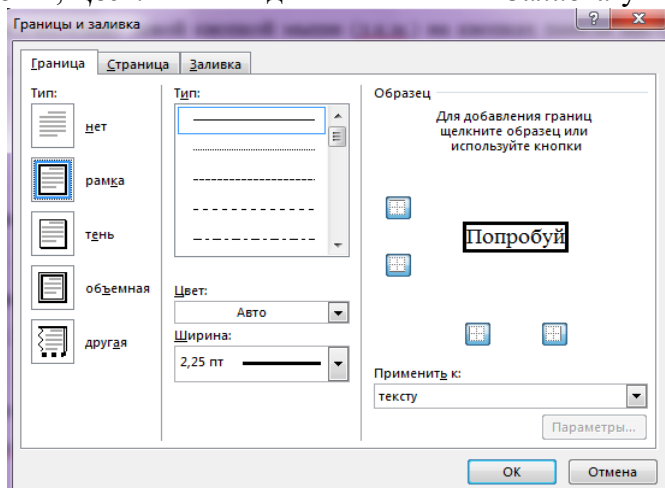


Рисунок 3

1. Задайте установки режима сохранения, выполнив команду *Файл - Параметры*, вкладка *Сохранение*, поле *Автосохранение каждые 10 мин* (Рисунок 4).

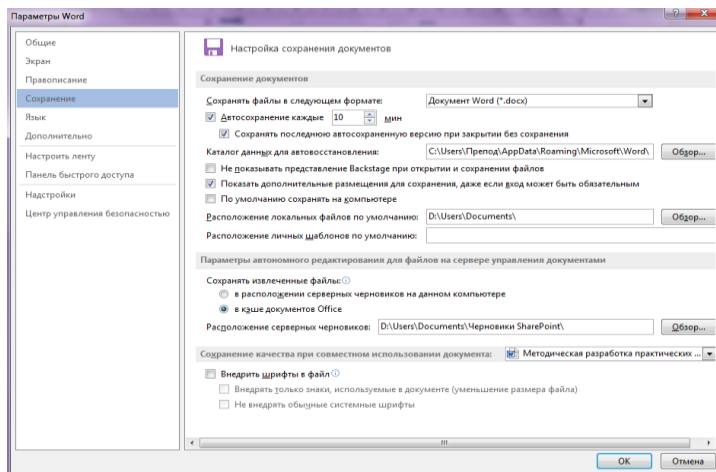


Рисунок 4

3. Сохраните файл под именем *Практическая работа №1 (Фамилия Имя)* в папке *Работы студентов* в формате *Word*, последовательно выполнив команды *Файл - Сохранить как*.

4. Создайте элемент автозамены: *персональные данные*, используя аббревиатуру *пд*. Для этого выполните следующие действия: нажмите на вкладку *ФАЙЛ*, выберите команду *Параметры – Правписание – кнопка Параметры автозамены (рисунок 7)* и в окне *Автозамена* выберите *заменить пд на персональные данные*. Нажмите кнопку *Добавить*, затем выберите новую команду из списка и нажмите *Ок*.

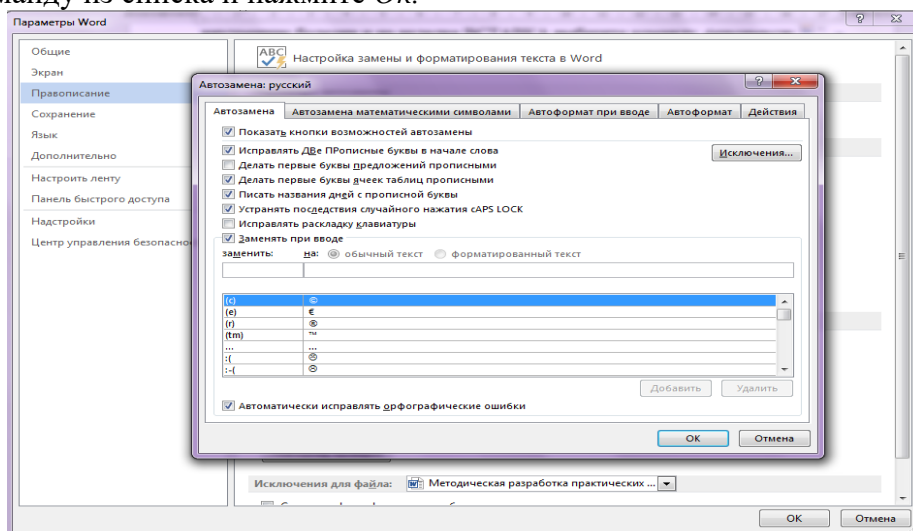
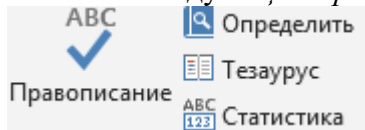
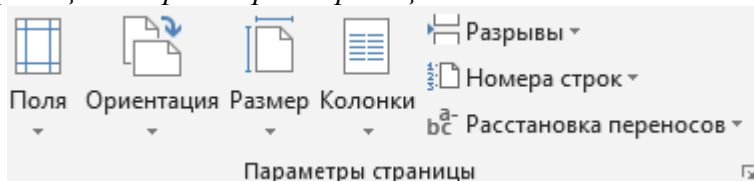


Рисунок 7

5. Предварительно установив курсор в начало текста, проверьте орфографию текста, выполнив команду *Рецензирование - Правписание*.



6. Настройте параметры страницы с помощью последовательности команд *Разметка страницы - Параметры страницы*.



В окне *Параметры страницы (рисунок 8)* выберите вкладку *Поля* и установите следующие значения: *Верхнее - 2см, Нижнее - 2см, Левое - 3см, Правое - 3см*. На вкладке *Источник бумаги* в группе *От края до колонтитула* установите *верхнего - 1см, нижнего - 1см*. В поле *вертикальное выравнивание*, выберите *По верхнему краю*, и далее в поле *Применить* укажите *Ко всему документу*.

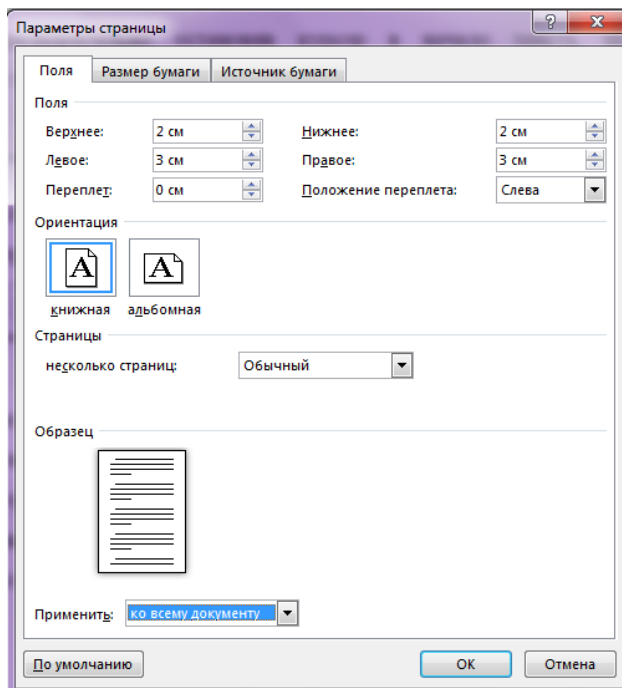
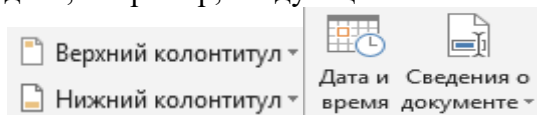


Рисунок 8.

7. На вкладке ВСТАВКА сформируйте верхний колонтитул (текст по центру, с нижним подчеркиванием) и нижний колонтитул (дата, время). В качестве текста верхнего колонтитула введите, например, следующий текст: «Закон РФ «О персональных данных»».



Задание практической работы

*Перейдите на следующую страницу с помощью комбинации клавиш **CTRL+ENTER** и введите текст:

Набрать приведенный ниже текст, соблюдая оформление, и установив параметры страницы: верхнее поле – 1,5 см, нижнее – 1,5 см, левое – 2 см, правое – 1,5 см, ориентация – альбомная. Параметры абзаца: выравнивание – по ширине, отступ первой строки – 1,25 см, интервал перед – 0 пт, после – 6 пт. Основной текст – шрифт Таhоmа, размер 12. Заголовок – выравнивание по центру, полужирный, размер 14.

История создания и развития компьютерной техники.

Контрольные вопросы:

Бытующее мнение о том, что компьютеры — изобретение XX столетия, не совсем верно. С тех пор как человек появился на земле и занялся производством и торговлей, он стал нуждаться в системе подсчетов и вычислений. Более тысячи лет эти операции выполнялись либо вручную, либо с помощью устройств вроде абака — счетной доски с камешками, разделенной на полосы. Человеческая мысль не стоит на месте, и вслед за абаком появились счеты, сумматор Паскаля (1642), арифмометр Лейбница (1673) и прочие полезные изобретения, без которых не было бы компьютеров в том виде, в каком мы сейчас привыкли их видеть...

В 1801 году француз Жозеф Жаккард совершил поистине революционное открытие в области производства ткани - изобрел ткацкий станок для тканей со сложной фактурой, читавший инструкции с карточек. Отверстия, пробитые в карточке, составляли цепь последовательных команд: какие нити использовать и что с ними делать. Изобретение было гениальным, но современники Жаккарда не оценили его по достоинству; сам изобретатель сильно пострадал от французских ткачих, утверждавших, что машина Жаккарда отнимет у них рабочие места.

В 1834 году британский изобретатель Чарльз Бэббидж разработал проект аналитической машины, которая должна производить вычисления, используя ту же идею хранения инструкций на карточках, названных впоследствии перфокартами.

Машина, по сути, представляла собой гигантский арифмометр с программным управлением, памятью и арифметическим устройством и таким образом явилась прообразом первого компьютера. Гениальный ученый долгое время пытался уговорить британское правительство профинансировать его проект, но не добился успеха.

1. Что такое редактирование текста?
2. Что такое форматирование текста?
3. Как включить линейку?
4. Как настроить: междустрочный интервал;
отступ правой строки (красная строка);
разреженный интервал (например 4 пт);
выравнивание и какие виды выравнивания существуют;
параметры страницы;
параметры автосохранения;
текст в рамку/границу, и установить узор к (примеру 20%)
параметры автозамены.
5. Как вставить нижний и верхний колонтитул?

Практическая работа № 11. Внедрение объектов в электронный документ

Цель: научить структурировать данные с помощью таблиц, научить добавлять таблицы в текстовый документ; познакомить с основными приемами работы с таблицами.

План урока:

1. Организационный момент.
2. Актуализация знаний по теме «Вставка рисунков в текст».
3. Обучающая практическая работа.
4. Самостоятельная работа.
5. Подведение итогов.

Ход урока

I. Организационный момент.

Постановка темы и цели урока.

II. Актуализация знаний по теме «Вставка рисунков в текст».

Опрос:

Как можно добавить рисунок в текст?

Как изменить размер рисунка, не нарушая пропорций?

Что означает термин «обтекание» рисунка текстом?

Что означает термин «рисунок в тексте»?

III. Обучающая практическая работа.

Таблицы в текстовом процессоре используются для упорядочения и более наглядного представления данных. Таблица Microsoft Word состоит из строк и столбцов. Клетка на пересечении строки столбца называется ячейкой. Ячейки могут содержать числа, текст, рисунки.

Большинство команд работы с таблицами находятся в верхнем меню **Работа с таблицами** (разделы: **Макет** и **Конструктор**), которое появляется на ленте, если курсор расположен внутри таблицы. Некоторые команды продублированы на ленте **Главная** и в контекстном меню.

Практическая работа

Задание. Создайте и оформите таблицу по образцу (рис.1).

Рисунки расположены в папке *Страны*

ВЕЛИКОБРИТАНИЯ		
		
Европа <i>Великобритания</i> Столица: Лондон Площадь: 244,1 тыс. км ² Население: 10658 тыс. чел.		

Рис. 1

Комментарии.

Рассмотрим создание и редактирование таблиц на примере создания таблицы *Великобритания*. Первое, на что следует обратить внимание, это сколько столбцов и строк лежит в основе таблицы. В нашем случае - три строки и три столбца.

1. На ленте **Вставка** в разделе **Таблицы** выберите команду **Таблица**.

2. Перемещая указатель мыши, выберите три строки и три столбца (рис. 2). Также можно воспользоваться командой **Вставить таблицу** из этого же меню.

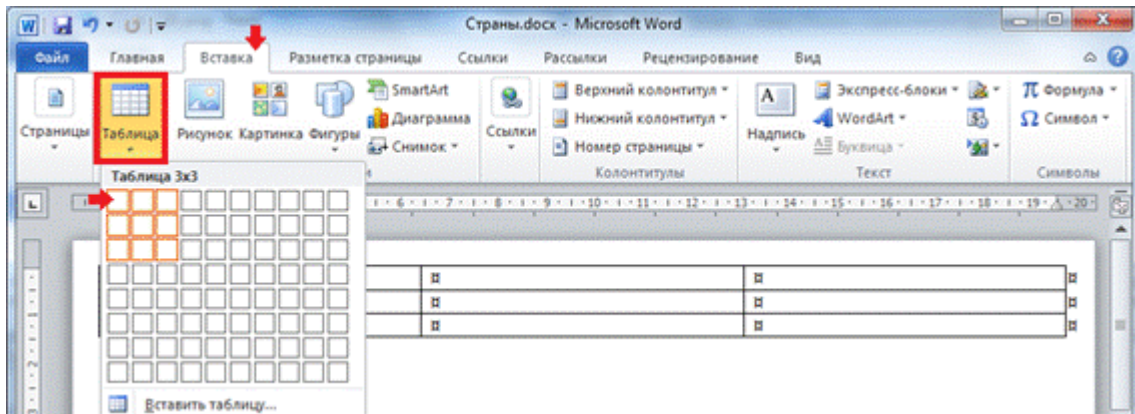


Рис. 2

3. Объедините строчки в первом столбце. Для этого выделите ячейки первого столбца и на ленте **Работа с таблицами: Макет** в разделе **Объединение** выберите команду **Объединить ячейки** (рис. 3).

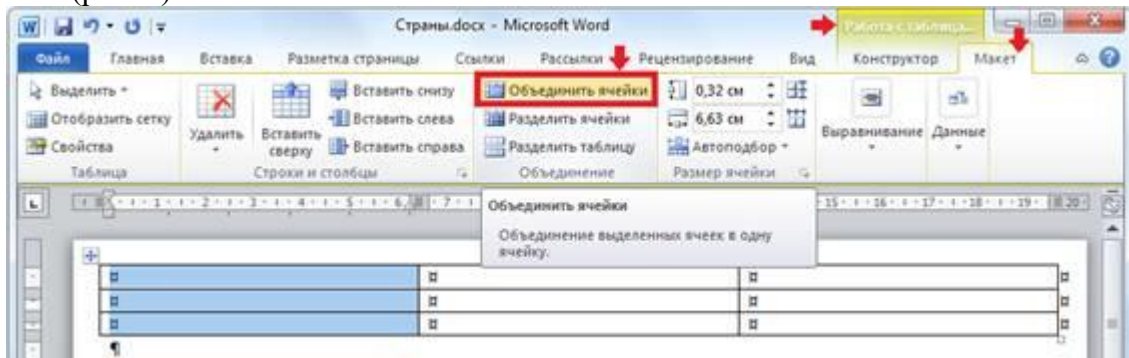


Рис. 3

4. Аналогично объедините вторую и третью строку во втором столбце; а также первые ячейки во втором и третьем столбцах (рис. 4).

5. Заполните верхнюю ячейку таблицы, напечатав по центру *Великобритания* (рис. 4).

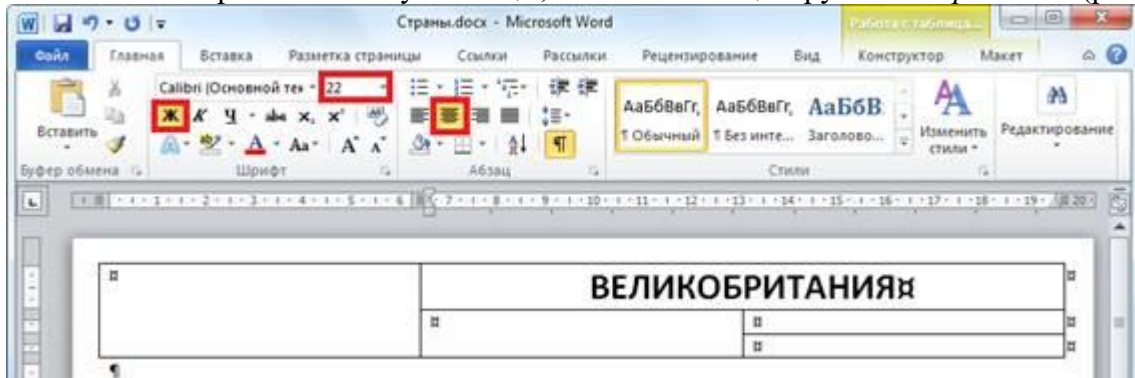


Рис. 4

6. Отформатируйте текст в ячейках таблицы по центру как по горизонтали, так и по вертикали: на ленте **Работа с таблицами: Макет** в разделе **Выравнивание** выберите пиктограмму **Выровнять по центру** (рис. 5).

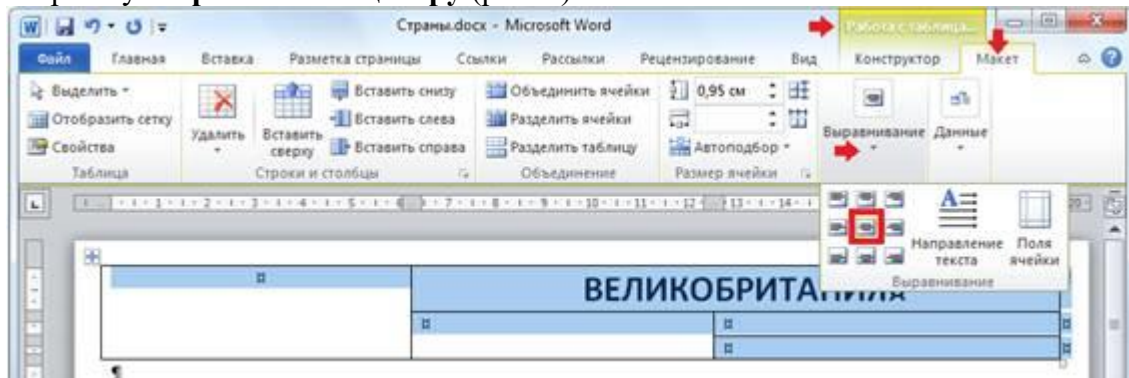


Рис. 5

7. Добавьте в первый столбец рисунок из файла: на ленте **Вставка** в разделе **Иллюстрации** выберите пиктограмму **Рисунок** (рис. 6).

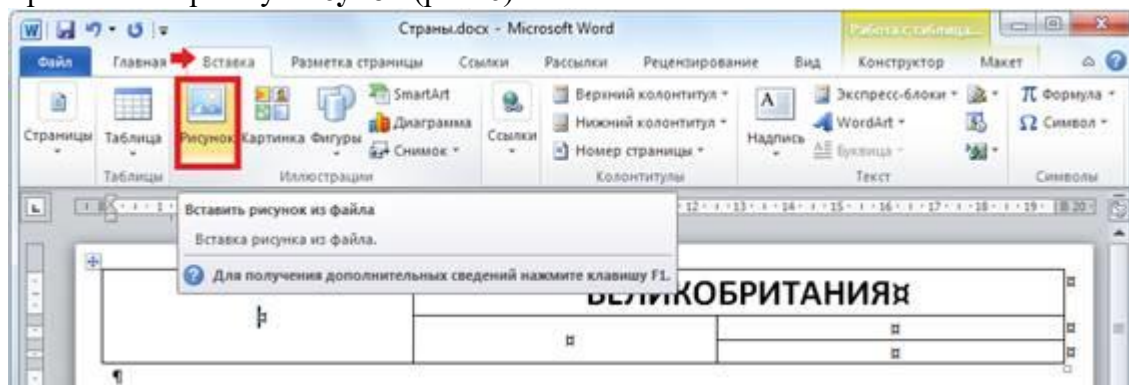


Рис. 6

8. Подкорректируйте ширину первого столбца, сделав ее меньше. Установите указатель мыши на границе между столбцами (рис. 7). Перетащите указатель мыши влево или вправо, чтобы изменить ширину столбца.

Несмотря на то, что высоту строк можно изменить аналогичным образом, старайтесь не использовать этот способ. Изменив высоту «вручную» один раз, вы навсегда теряете возможность автоматического подбора высоты ячейки в зависимости от ее содержимого. Следовательно, при каждом редактировании содержимого ячеек, вам придется следить и регулировать высоту ячейки самому.

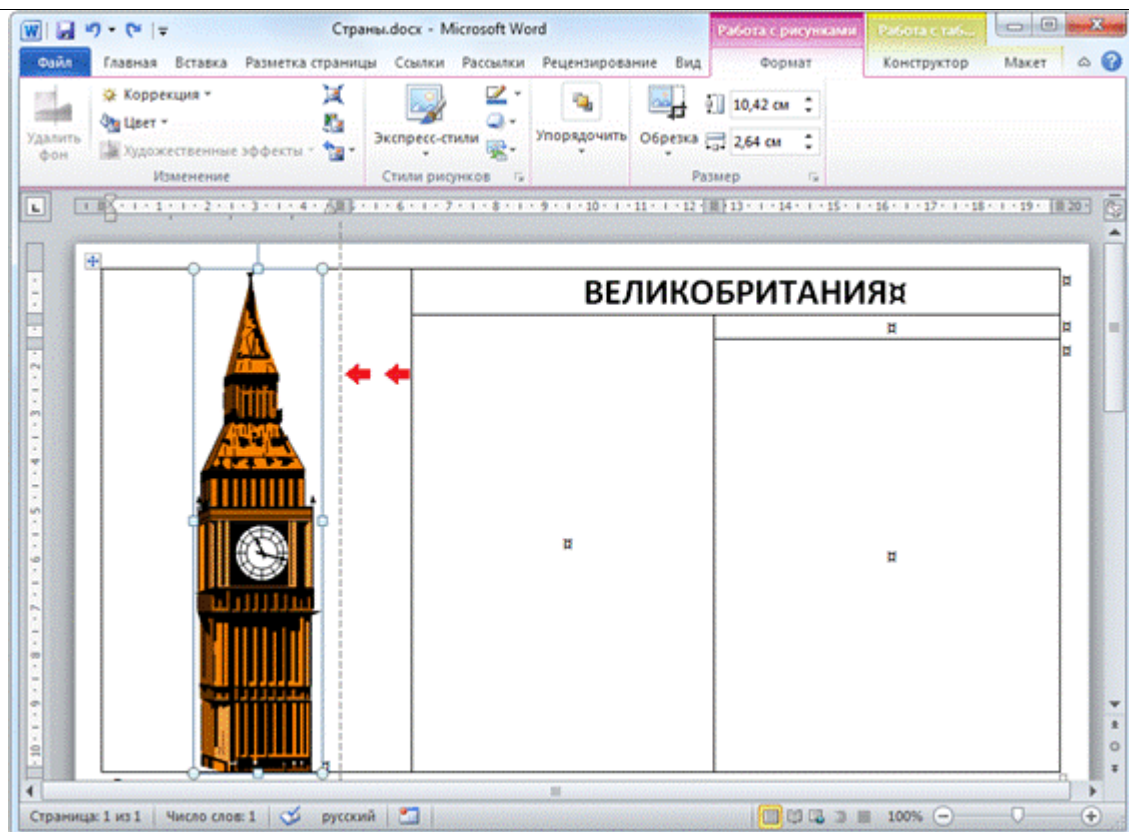


Рис. 7

9. Заполните данными остальные ячейки таблицы (рис. 8).

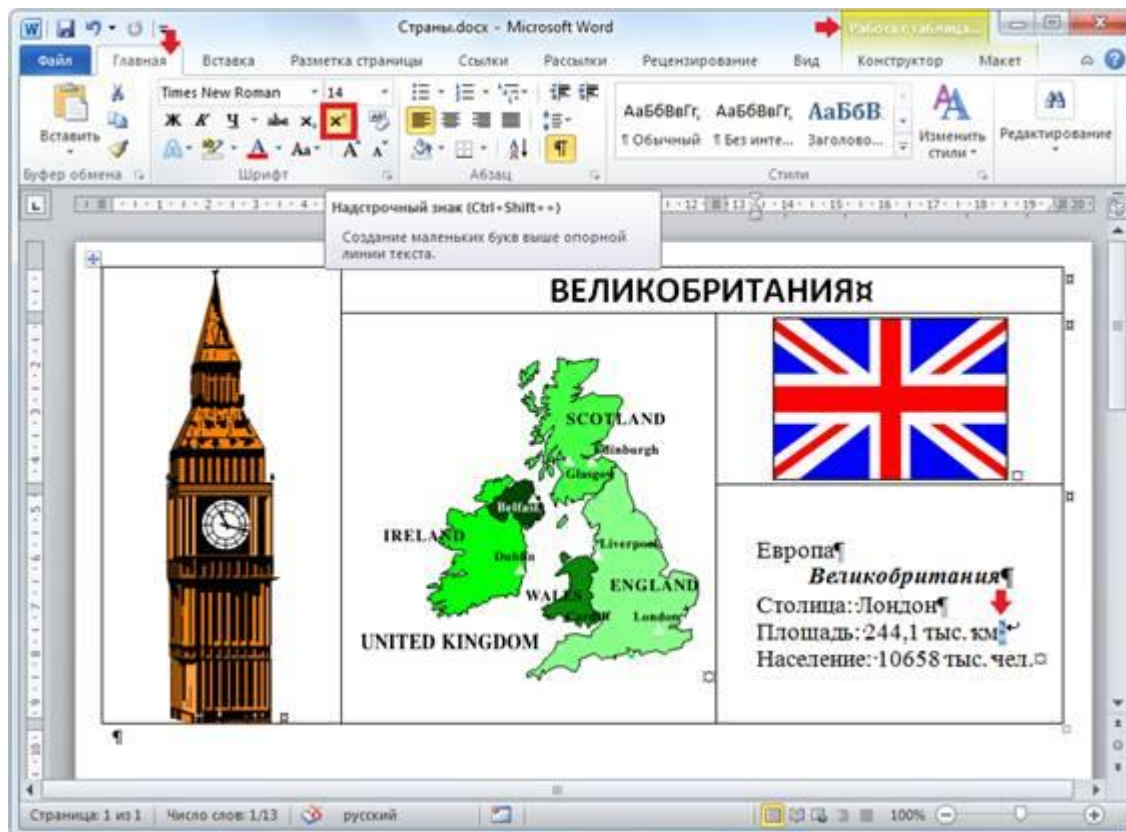


Рис. 8

V. Подведение итогов.

Контрольные вопросы:

- Как добавить в текстовый документ таблицу?
- Как объединить несколько ячеек по вертикали или горизонтали?
- Как изменять размер таблицы?
- Как изменить ширину столбцов?
- Почему не рекомендуется изменять высоту строк вручную?
- Какие способ форматирования текста внутри ячеек вы знаете?

Практическая работа № 12. Создание и форматирование электронных таблиц

Цель: Получение практических навыков по созданию, редактированию и форматированию таблиц. Формирование навыков работы по оформлению (форматированию) таблиц, построению и редактированию диаграмм в среде MS EXCEL

Задание 1:

Средствами табличного процессора EXCEL создайте **Таблицу1** на основе ниже приведённого сценария.

1. Запустите табличный процессор EXCEL.
2. Установите курсор в ячейку **A1** (щелчком мыши по ячейке) и введите текст: **Выручка от реализации книжной продукции.**
3. Введите таблицу согласно образцу, представленному в **таблице1.**

Таблица 1

Выручка от реализации книжной продукции								
	A	B	C	D	E	F	G	H
1	Выручка от	реализации книжной продукции						
2	Осень 2014							
3	Объём продаж	июнь	Июль	август	сентябрь	Октябрь	Итого	Процент реализации
4	До 100 пачек	1050	2100	4250	2250	1950		
5	До 200 пачек	330	6160	13530	11200	1100		
6	> 200 пачек		1000	6000	4000			
7	Сумма							
8								

4. Рассчитайте сумму выручки от реализации книжной продукции в июне месяце одним из двух способов:

- 1 способ: установите курсор в ячейку **B7**, введите в ячейку **B7** формулу (формула начинается со знака =, занесение имён ячеек в формулу осуществляем щелчками мыши по соответствующим ячейкам): **=B4+B5+B6**. Нажмите клавишу **Enter**. В ячейке появится сумма диапазона ячеек **B4:B6**;


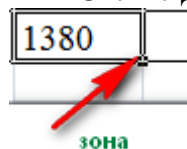
- 2 способ: выделите диапазон ячеек **B4:B7** и нажмите кнопку **Автосумма**  (Вкладка **Главная, Группа Редактирование**). В ячейке **B7** появится сумма диапазона ячеек **B4:B6**. Сравните полученный результат со значением ячейки **B7** в **таблице 2.**

Таблица 2

=СУММ(B4:B6)								
	A	B	C	D	E	F	G	H
1	Выручка от реализации книжной продукции							
2	Осень 2014							
3	Объём продаж	июнь	Июль	август	сентябрь	Октябрь	Итого	Процент реализации
4	До 100 пачек	1050	2100	4250	2250	1950		
5	До 200 пачек	330	6160	13530	11200	1100		
6	> 200 пачек		1000	6000	4000			
7	Сумма	1380	9260	23780	17450	3050		
8								

5. Распространите операцию суммирования на диапазон **C7:F7**:

- Скопируйте содержимое ячейки **B7** в ячейки **C7:F7**. Для этого схватите ячейку **B7** за



правый нижний угол (**зону автозаполнения**) **автозаполнения** и не отпуская кнопку мыши перетащите её в ячейку **F7**. Очистите диапазон **C7:F7**.


6. Убедитесь в правильности выполненной операции:

- выделите ячейку **B7**. В строке формул должно отобразиться выражение: **=СУММ(B4:B6)**;

7. Подсчитайте суммарную выручку от реализации книжной продукции (столбец **Итого**).

Для этого:

- выделите в строке ячейки **B4:G4**;

- нажмите кнопку  ;

- в ячейке **G4** появится сумма диапазона **B4:F4**.

8. Подсчитайте суммы в остальных ячейках столбца **Итого**. Для этого: схватите ячейку **G4** за правый нижний угол (**зону автозаполнения**) и, не отпуская кнопку мыши, протащите её до ячейки **G7**. В ячейках **G5**, **G6**, **G7** появятся суммарная выручка от реализации книжной продукции.


9. Подсчитайте **Процент реализации**, полученной от продажи партий товара. Для этого:

- выделите ячейку **H4**;

- введите формулу **=G4/G7**;

- нажмите клавишу **Enter**;

- выделите ячейку **H4**;

- нажмите кнопку  (**Вкладка Главная, Группа Число**). В ячейке **H4** появится значение доли выручки в процентах.

10. *Рассчитайте долю выручки для других строк таблицы, используя автозаполнение.*

В результате автозаполнения в ячейках **H5**, **H6** и **H7** появится сообщение **#ДЕЛ/0!** (деление на ноль). Такой результат связан с тем, что в знаменатель формулы введён **относительный** адрес ячейки, который в результате копирования будет смещаться относительно ячейки **G7** (**G8**, **G9**, **G10** – пустые ячейки). Измените **относительный** адрес ячейки **G7** на **абсолютный** – **\$G\$7**, это приведёт к получению правильного результата счёта. Еще раз попробуйте рассчитать доли выручки в процентах. Для этого:

- очистите диапазон **H4:H7**;

- выделите ячейку **H4**;

- введите формулу **=G4/\$G\$7**;

- нажмите клавишу **Enter**;

- рассчитайте долю выручки для других строк таблицы, используя автозаполнение.

В результате в ячейках диапазона **H4:H7** появится доля выручки в процентах.

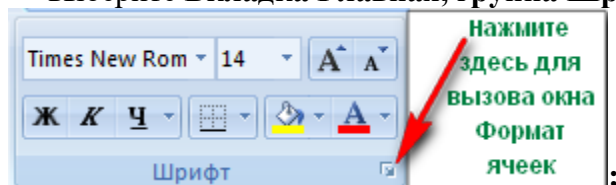
Задание 2:

Средствами табличного процессора **MS Excel** измените оформление таблицы, размещенной в файле и постройте диаграмму на основе данных указанной таблицы. Последовательность Ваших действий описана в ниже приведённом сценарии.

1. Измените шрифт заголовка таблицы. Для этого:

- выделите заголовок таблицы;

- выберите **Вкладка Главная, группа Шрифт**

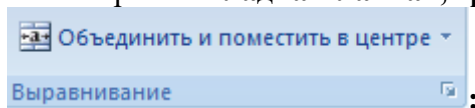


- установите параметры: тип шрифта (например, **Arial Cyr**), стиль (например, **полужирный**), размер (например **16**), цвет и т. д.;

2. Задайте выравнивание заголовка по **центру**. Для этого:

- выделите диапазон ячеек, в пределах которых будет размещён заголовок (обычно заголовок центрируется по ширине таблицы);

- выберите **Вкладка Главная, группа Выравнивание**




- нажмите кнопку **Объединить и поместить в центре**.

3. Измените цвет фона шапки таблицы (совокупность наименований столбцов):

- выделите диапазон ячеек **A3:H3**;

- выберите **Вкладка Главная, группа Шрифт** (используйте кнопку **Цвет заливки**  для изменения цвета фона).

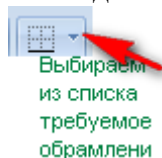
4. Измените цвет текста шапки таблицы с помощью кнопки **Цвет текста** 

5. Уберите линии сетки. Для этого:

6. Нажмите **Вкладку/Вид**, снимаем галочку с пункта **Сетка**

7. Выполните процедуру обрамления таблицы (границы). Для этого:

- выделите нужные строки (например, всю таблицу);



- выберите **Вкладка Главная, группа Шрифт** ;

- установите параметры: границы (например, все границы), вид линии (толщина линии), цвет линии (из палитры).

8. Измените высоту строки: Для этого:

- установите указатель мыши на границу двух любых строк (в зоне заголовков строк),



указатель мыши превратится в двунаправленную стрелку, нажмите левую клавишу мыши и, не отпуская её, протащите границу вверх для уменьшения высоты строки и вниз для увеличения высоты строки, отпустите мышь.

9. На основе данных электронной таблицы **Выручка от реализации книжной продукции** постройте диаграмму. Для этого:

- выделите диапазон ячеек **A3:F6**;

- выберите **Вкладка/Вставка/группа Диаграммы** нажмите кнопку **Гистограмма**.

10. Ведите заголовок диаграммы. Например: **Выручка от реализации книжной продукции**.

11. Отредактируйте полученную диаграмму, которая состоит из:

- области диаграммы;
- области построения;
- области рядов данных (пункт 16);
- легенды (пункт 17);
- заголовка.

12. Отредактируйте область всей диаграммы. Для этого:

- выделите область диаграммы щелчком мыши по ней. Обратите внимание на изменённый вид **Ленты**. Вам стала доступна **Вкладка Работа с диаграммами** и дополнительная **Вкладка Конструктор**.

13. Отредактируйте **область построения**. Для этого:

- выделите область построения щелчком мыши;
- перейдите на **Вкладку Макет**;

- нажмите на **Кнопку Область построения**, выберите **Дополнительные параметры построения**, переберите предложенные варианты оформления **Области построения**.

14. Отредактируйте **область рядов** данных. Для этого:

- выделите один из столбцов щелчком мыши по нему. Одновременно будут выделены все остальные столбцы, относящиеся к одному ряду;
- вызовите контекстное меню области ряда данных, щелкнув правой кнопкой мыши по выделенному столбцу;
- окрасьте столбцы (залейте) одного ряда в выбранный Вами цвет и измените тип границы столбца;
- аналогичным образом окрасьте все остальные группы столбцов.

15. Отредактируйте **легенду**. Для этого:

- выделите легенду, в которой приведены обозначения, используемые для выделения различных групп столбцов;
- измените шрифт и цвет фона в легенде.

16. Отредактируйте заголовок. Для этого:

- Нажмите на **Кнопку Название диаграммы**, выберите способ размещения **Над диаграммой**;
- щелкните мышью по строке формул; наберите новый текст заголовка в строке формул и нажмите клавишу **Enter**; текст появится внутри области диаграммы;
- укажите на рамку, окружающую появившийся текст заголовка, нажмите мышью и перетащите заголовок на нужное место;
- измените цвет текста с помощью кнопки **Цвет шрифта**.

17. Поместите диаграмму рядом с таблицей (справа или внизу). Для этого:

- щёлкните мышью по области диаграммы и, не отпуская кнопки мыши, перетащите её в нужное место.

18. Постройте объёмную круговую диаграмму. Для этого:

- выделите в электронной таблице область **A3:F4**;
- выберите **Вкладка Вставка, группа Диаграммы** нажмите кнопку **Круговая, Объёмная круговая**;
- щелчком мыши выделите область рядов данных диаграммы, вызовите контекстное меню, выберите **Поворот объёмной фигуры** и поверните объёмную диаграмму на нужный угол;
- выделите область рядов данных диаграммы. С помощью контекстного меню окрасьте всю диаграмму в выбранный Вами цвет;
- выделите передний сектор диаграммы установите цвет точки данных. Например: красный;
- выделите передний сектор диаграммы и, удерживая нажатой левую клавишу мыши, выдвиньте этот сектор вперед, вырезав его из круга.

19. Сохраните созданную таблицу в папке **Отчеты**.

Контрольные вопросы

- 1 Что такое электронная таблица и каково ее назначение?
- 2 Как обозначаются столбцы и строки в MS EXCEL?
- 3 Как изменить ширину столбца (высоту строки)?
- 4 С какими типами данных работает MS EXCEL?
- 5 Какое расширение имеет файл книги электронной таблицы?
- 6 Из чего состоит рабочая книга?
- 7 Как ввести данные в ячейку? Как отредактировать данные в ячейке?

Практическая работа № 13. Работа с формулами и функциями в электронных таблицах

Цель. Освоение основных приемов работы с формулами и функциями в табличном процессоре MS Excel.

Порядок практической работы.

1. Ввод формул вручную.

- Формула всегда начинается со знака «=». Знак «=» вводится с клавиатуры.
- С клавиатуры так же вводятся числа и математические знаки + (сложение), - (вычитание), * (умножение), / (деление), ^ (возведение в степень), % (процент, пример: «=3 %» — преобразуется в 0,03; «=37*8 %» — нашли 8 % от 37). То есть если мы дописываем после числа знак «%», то число делится на 100.
- Скобками определяется порядок действий.

Вычислим значение выражений:

Вычисления будем выполнять на Листе1

1. $35*2,5-16$ (результат в ячейке A1)
2. $(5,6+0,4)/10$ (результат в ячейке A2)
3. $\frac{1,1+12,9}{12-103}$ наберите на клавиатуре кнопки $- 1 5 / (1 , 1 + 1 2 , 9)$ (результат в ячейке A3)
4. $\frac{1,5+4}{3,2-10}$ наберите на клавиатуре кнопки $(1 2 - 1 0 3) / (1 , 5 + 4)$ (результат в ячейке A4)
5. -4 наберите на клавиатуре кнопки $(3 , 2 - 1 0) / (- 4)$ (результат в ячейке A5)
6. $5^6 ; 19^2$ чтобы поставить знак ^ надо переключить клавиатуру на английский язык и, удерживая кнопку Shift, нажать цифру 6 (результат в ячейке A7 и ячейке A8)
 - В формулах так же могут содержаться адреса ячеек (в вычислении участвует содержимое ячейки). Чтобы адрес ячейки отобразился в формуле необходимо нужную ячейку (блок ячеек) щёлкнуть левой кнопкой мыши или прописать адрес этой ячейки (блок ячеек)

Подготовим на Листе2 таблицу для вычислений.

Найдём сумму чисел, которые находятся в A2, B2, C2, в A3, B3, C3 и т.д.

	A	B	C	D	E
1	число1	число2	число3	сумма	произведение
2	0,03	-10	2,11		
3	6	-1,4	3,3		
4	12	6	-2,3		
5	34	-3	8		
6	2,01	-1	0,45		
7	-0,1	1	2		
8	10	4	-4,1		
9	19	3	0,06		
10	0,25	6	0,01		

Чтобы вычислить сумму чисел, которые находятся в ячейках A2, B2, C2 надо в ячейке D2 набрать с клавиатуры знак «=», затем щёлкнуть ячейку A2, знак «+» с клавиатуры, затем щёлкнуть ячейку B2, знак «+» с клавиатуры, затем щёлкнуть ячейку C2 и кнопку Enter.

Принцип относительной адресации.


Принцип относительной адресации обозначает следующее:

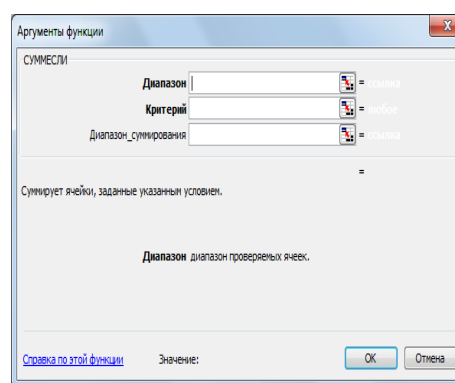
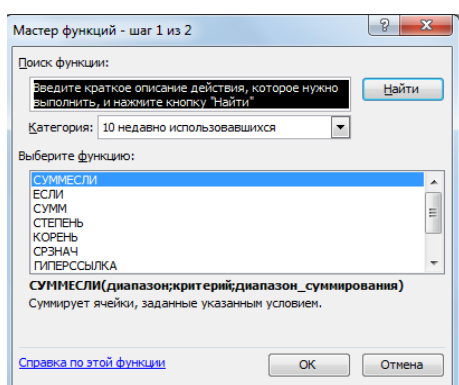
адреса ячеек, используемые в формулах, определены относительно места расположения формулы.

Поэтому, если в ячейках D3, D4, ..., D10 надо вычислить сумму чисел, находящихся соответственно в ячейках A3, B3, C3 и A4, B4, C4 и т.д. формулу $= A2 + B2 + C2$, которая находится в ячейке D2 можно скопировать (протянуть) в ячейки D3, D4, ..., D10 .

Аналогичным образом, вычислите произведение чисел, которые находятся в ячейках A2, B2, C2... A10, B10, C10.

2. Ввод формул с помощью Мастера функций.

Кнопка **ВСТАВКА ФУНКЦИЙ**  находится на строке формул. **МАСТЕР ФУНКЦИЙ** состоит из 2 шагов – 2 диалоговых окон.



1 шаг – выбор функции:

В Excelе более 150 встроенных функций для обработки данных. Для удобства поиска все функции разбиты на категории, внутри каждой категории они отсортированы в алфавитном порядке.


Кроме этого есть две категории – “10 недавно использовавшихся” и “Полный алфавитный перечень”, в котором все встроенные функции располагаются в алфавитном порядке. Назначение каждой функции подробно описывается.

2 шаг – выбор аргументов:

Аргументами могут быть числа или тем, что в итоге преобразуется в число. Если вместо одного из них вписать текст, Excel выдаст ошибку.

Так же аргументами являются и адреса ячеек. Диапазон можно выделить левой кнопкой мыши. Если необходимо указать диапазон ячеек, то первый и последний адреса разделяются двоеточием, например A12:C20.


Порядок работы с функциями

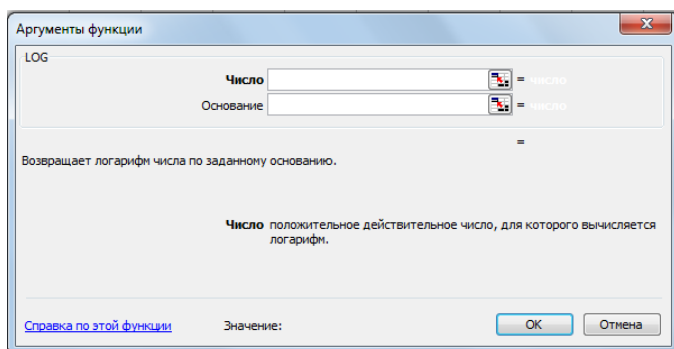
- Сделаем активной ячейку, в которую хотим поместить результат.
- Выбираем команду Вставка – Функция или нажимаем кнопку 
- В первом появившемся окне Мастера функций определяем категорию и название конкретной функции.
- Во втором окне необходимо определить аргументы для функции. Для этого щелчком кнопки выделяем нужный диапазон ячеек или прописываем аргументы с клавиатуры.
- Затем для завершения работы нажимаем клавишу <OK>. В исходной ячейке окажется результат вычисления.

Вычислим значение выражений с помощью Мастера функций (вернёмся на Лист1):

1. Вычислите $\log_5 25$ (результат в ячейке C1)

Для того что бы найти данный логарифм, надо выполнить данные действия:

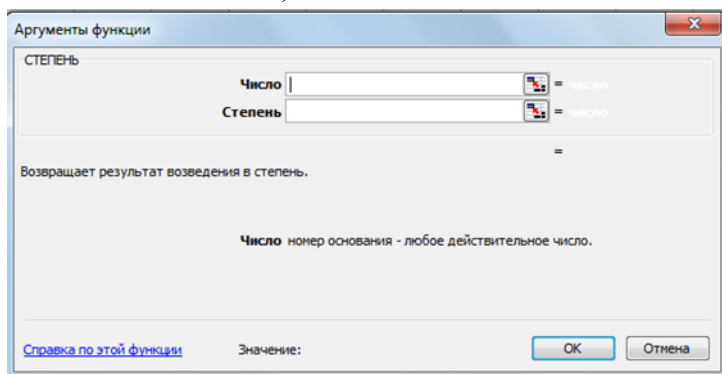
- Сделаем активной ячейку, в которую хотим поместить результат.
- Выбираем команду Вставка – Функция или нажимаем кнопку 
- В 1 появившемся окне МАСТЕРА ФУНКЦИЙ определяем категорию математические и функцию LOG, затем кнопку ОК.
- Во 2 окне Аргументы функции прописываем **число 25, основание 5**, затем кнопку ОК.



2. Вычислите $\log_3 80$ (результат в ячейке C2)

3. Вычислите 17^2 (результат в ячейке C3)

Выбираем математическую функцию - СТЕПЕНЬ, которая возвращает результат возведения в степень: Число - 17, Степень – 2



4. Вычислите $\sqrt{144}$ (результат в ячейке C4)

Выбираем математическую функцию - КОРЕНЬ, которая возвращает значение квадратного корня.

5. Вычислите $\sqrt[3]{27}$ (результат в ячейке C5)

Чтобы вычислить, корень n-ой степени надо выбрать функцию СТЕПЕНЬ и представить корень

в виде степени с рациональным показателем $\sqrt[3]{27} = 27^{\frac{1}{3}}$

Число - 27, Степень – 1/3

6. Вычислите $\log_{\frac{1}{3}} 81 + 11^3$ (результат в ячейке C6)

7. Найдите среднее значение чисел 354, 564, 98, -45, 122, 200, -10 с помощью Мастера функций. Данные числа перед вычислением внесите в ячейки E1....E7, в ячейку E8 введите текст «Среднее значение», а в ячейку E9 внесите результат.

Выбираем статистическую функцию СРЗНАЧ (среднее значение), которая возвращает среднее арифметическое (сумма чисел, делённая на количество чисел). Во втором шаге, при выборе аргументов укажите нужный диапазон чисел.

Простейшие статистические функции МИН, МАКС

8. Среди чисел, находящихся в ячейках E1...E7, найдите максимальное и минимальное значение и запишите результаты в ячейки в ячейки соответственно в E10 и E11.

Выберите статистическую функцию **МИН** и при выборе аргументов выделите нужный диапазон ячеек (E1:E7)

Затем выберите функцию **МАКС** и выделите тот же диапазон.


Вернёмся к таблице на Листе2 и продолжим вычисления:

1. Вычислим **сумму** чисел1, чисел2 и чисел3.

В ячейках A11, B11, C11 введем текст «сумма чисел1», «сумма чисел2» и «сумма чисел3»

В ячейку A12 введем формулу с помощью Мастера функций:

Для того надо выполнить данные действия:

- Сделаем активной ячейку A12.
- Выбираем команду Вставка – Функция или нажимаем кнопку 
- В **1** появившемся окне МАСТЕРА ФУНКЦИЙ определяем категорию - математические или 10 недавно использовавшихся. Выбираем функцию **СУММ**, затем кнопку ОК.
- Во **2** окне Аргументы функции при выборе аргументов выделяем нужный диапазон A2 : A10, затем кнопку ОК.

Формулу, которая находится в ячейке A12, скопировать в ячейки B12, C12, используя маркер автозаполнения.

	А	В	С
1	число1	число2	число3
2	0,03	-10	2,11
3	6	-1,4	3,3
4	12	6	-2,3
5	34	-3	8
6	2,01	-1	0,45
7	-0,1	1	2
8	10	4	-4,1
9	19	3	0,06
10	0,25	6	0,01
11	сумма чисел1	сумма чисел2	сумма чисел3
12			
13	произведение чисел1	произведение чисел2	произведение чисел3
14			


2. Аналогично выполнить нахождение **произведения** чисел 1, 2, 3 и записать результаты в ячейках A14, B14, C14.

3. Вычислить **сумму** только положительных чисел1 (диапазон A2:A2 из приготовленной таблицы).

Если необходимо не просто просуммировать все значения в диапазоне, а включить в сумму только те, которые удовлетворяют определённому условию, надо воспользоваться математической функцией **СУММЕСЛИ**.

В ячейку A15 введем текст «сумма положительных чисел1». В ячейку A16 введем формулу:

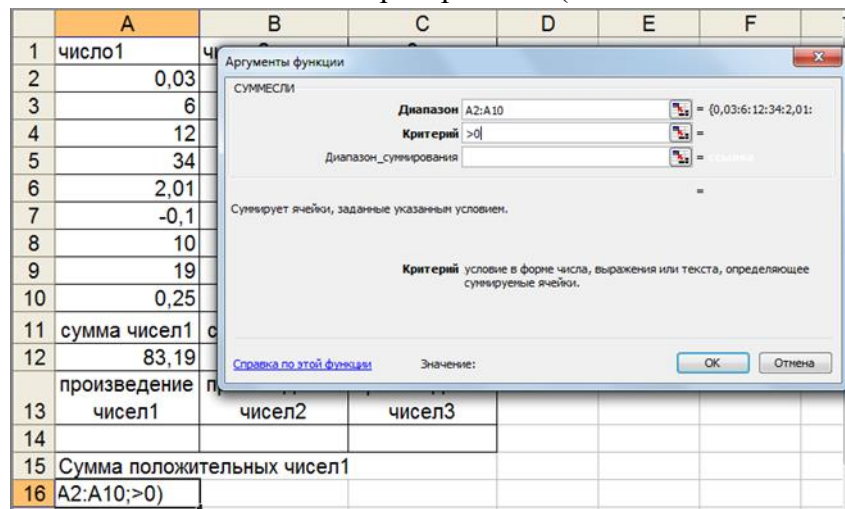
Для того надо выполнить данные действия:

- Сделаем активной ячейку A16.
- Выбираем команду Вставка – Функция или нажимаем кнопку 

- В 1 появившемся окне МАСТЕРА ФУНКЦИЙ определяем категорию - математические или 10 недавно использовавшихся. Выбираем функцию **СУММЕСЛИ**, затем кнопку ОК.
- Во 2 окне Аргументы функции при выборе аргументов:

Выделяем нужный диапазон A2 : A10, каждая ячейка из этого диапазона проверяется на соответствие условию, указанному во втором аргументе.

Указываем необходимый критерий – >0(положительные числа), затем кнопку ОК.



Критерии в данной функции:

операции сравнения <(меньше) , <=(меньше, либо равно) , >(больше), >=(больше, либо равно), =(равно) , <>(неравно) (например, >100 — суммировать все числа, большие 100), также можно использовать текстовые значения (например, "яблоки" — суммировать все значения, находящиеся напротив текста "яблоки") и числовые (например, 300 — суммировать значения в ячейках, значения в которых 300).

Контрольные вопросы:

- 1 Каково назначение формул в *MS EXCEL*? Что может входить в формулу?
- 2 Что отображается в ячейке после введения в нее формулы? Как увидеть формулу?
- 3 Что такое абсолютный и относительный адрес ячейки, входящей в формулу?
- 4 Как отобразить числа с символом денежной единицы, процента?
- 5 Какие способы автозаполнения таблицы вы знаете?
- 6 Как выполняется копирование формулы, распространение на соседние ячейки (автозаполнение) формулы?
- 7 Как разграничить таблицу?
- 8 Как задать цвет ячейкам и цвет текста?
- 9 Как защитить ячейки от изменений в них?
- 10 В чем суть автоматического перерасчета в *MS EXCEL*?
- 11 Что происходит во время копирования формул в *MS EXCEL*?
- 12 Что такое диапазон ячеек?
- 13 Как выделить смежные и несмежные диапазоны ячеек?
- 14 Как добавить в таблицу строки (столбцы)?
- 15 Как удалить ненужные строки (столбцы)?

Практическая работа № 14. Построение диаграмм

Цель: Формирование навыков работы по оформлению (форматированию) таблиц, построению и редактированию диаграмм в среде MS EXCEL

Методические указания

Работа с диаграммами

Основные термины

Диаграмма представляет собой графический способ представления табличных данных. При этом используется следующая терминология.

- **Серия данных** – группа данных расположенная внутри одной строки таблицы.
- **Имя серии** – имя строки таблицы, содержимое которой образует данную серию.
- **Легенда** – набор всех имен серий данной таблицы.
- **Категория** – группа значений, расположенных в одном столбце таблицы.

Элементы диаграммы

Для осознанной работы с диаграммами необходимо знать название и назначение ее основных элементов.



1. **Область диаграммы.** Область размещения диаграммы и всех ее элементов.

2. **Область построения диаграммы.** На плоских диаграммах – ограниченная осями область, содержащая все ряды данных. На объемных диаграммах – ограниченная осями область, содержащая все ряды данных, имена категорий, подписи делений и названия осей.

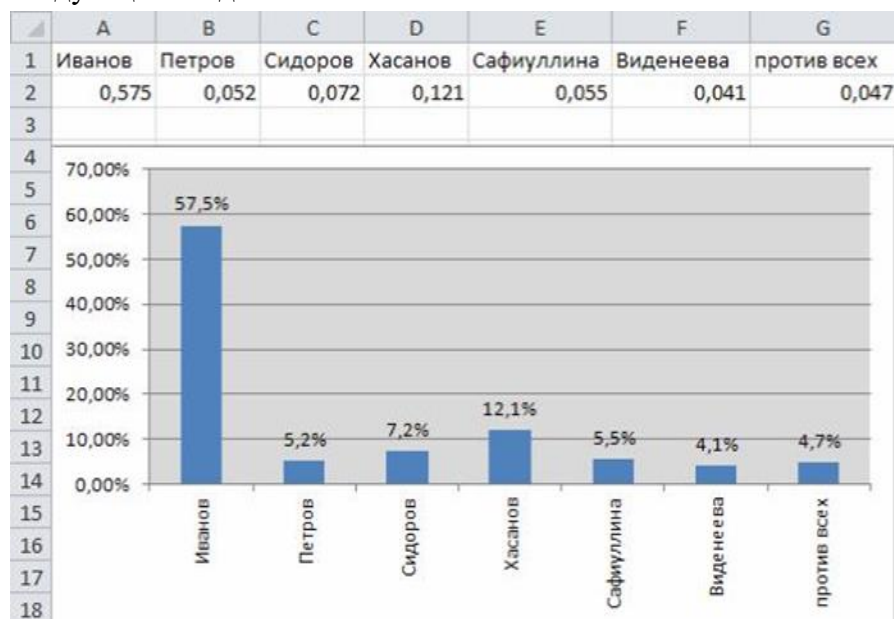
3. **Точки данных для ряда данных,** отмеченные на диаграмме. **Точки данных** – отдельные значения, отображаемые на диаграмме в виде полос, столбцов, линий, секторов, точек или других объектов, называемых маркерами данных. Маркеры данных одного цвета образуют ряд данных. **Ряд данных** – Набор связанных между собой элементов данных, отображаемых на диаграмме. Каждому ряду данных на диаграмме соответствует отдельный

цвет или способ обозначения, указанный на легенде диаграммы. Диаграммы всех типов, кроме круговой, могут содержать несколько рядов данных.

4. **Ось категорий.** Горизонтальная ось, вдоль нее строятся категории.
5. **Ось значений.** Вертикальная ось, вдоль нее строятся данные.
6. **Легенда диаграммы.** Рамка, в которой определяются узоры или цвета рядов или категорий данных на диаграмме.
7. **Названия на диаграммах.** Описательный текст, располагающийся вдоль осей или по центру в верхней части диаграммы.
8. **Подписи данных.** Подпись с дополнительными сведениями о маркере данных, предоставляющем одну точку данных или значение ячейки листа.

Создание диаграмм

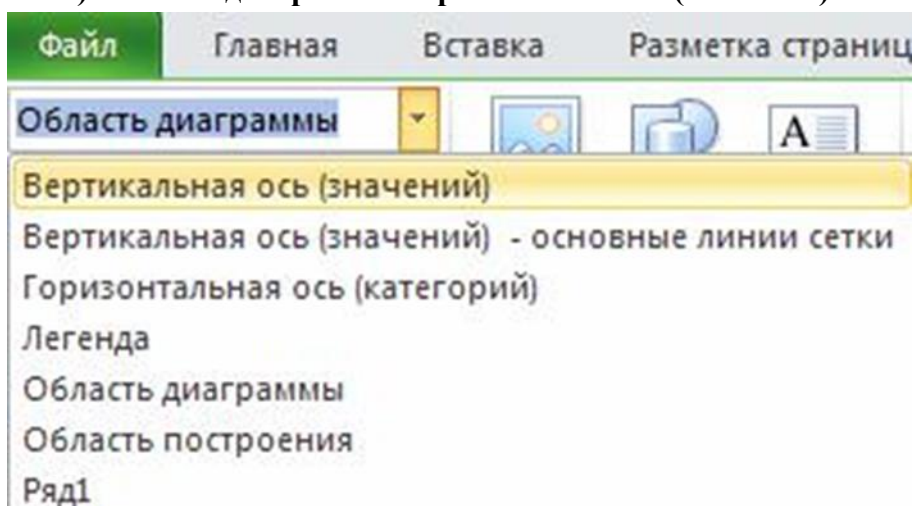
В Microsoft Excel 2010 можно построить диаграмму следующим способом: необходимо предварительно создать в документе таблицу данных, выделить ее, и после этого запускать вставку диаграмм. Диаграмма связана с данными, на основе которых она создана, и обновляется автоматически при изменении данных. Диаграмму можно создать на отдельном листе или поместить в качестве внедренного объекта на лист с данными. Создадим таблицу и диаграмму следующего вида.



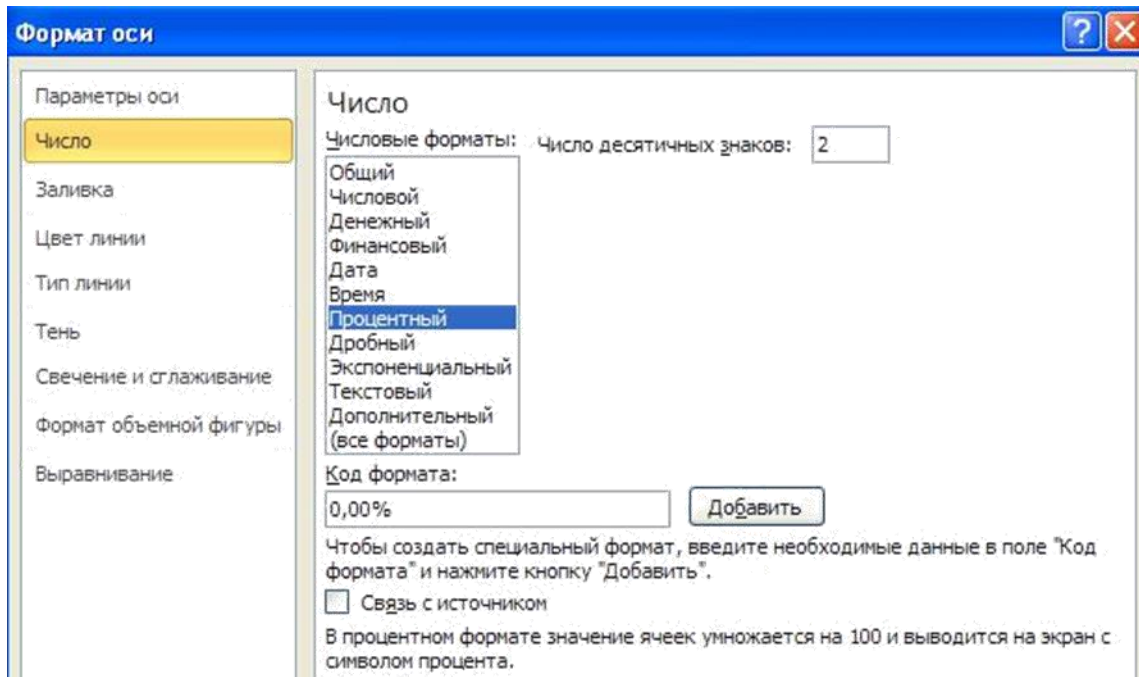
1. Создайте новую книгу. Для этого нажмите на кнопку **Файл**, выполните команду: **Создать/Новая книга** и нажмите **Создать**.
2. Заполните таблицу, т.е. ячейки **A1-G1, A2-G2**.
3. Выделите таблицу и выполните команду **Вставка/(Диаграммы) Гисто-**
грамма.
4. В списке гистограмм выбираем пункт **Гистограмма с группировкой**.



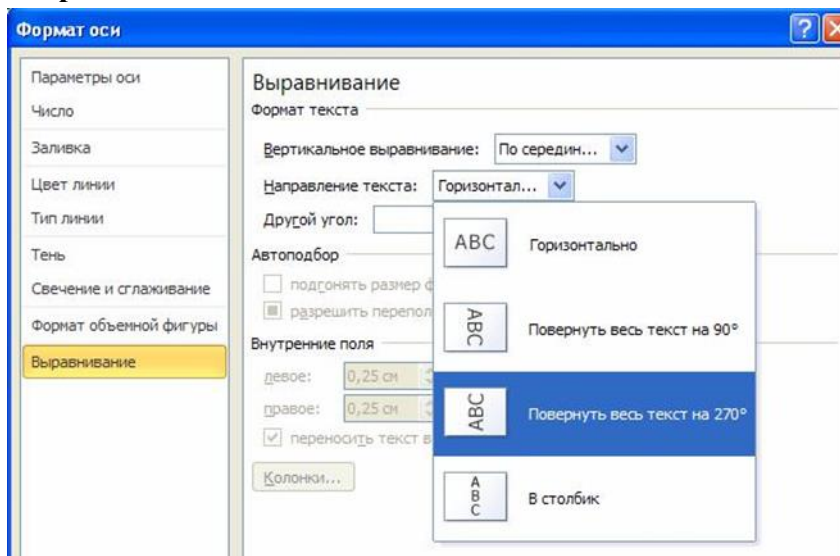
5. Выделите диаграмму, затем выполните команду: **Макет/(Текущий фрагмент) Область диаграммы/Вертикальная ось (значений)**.



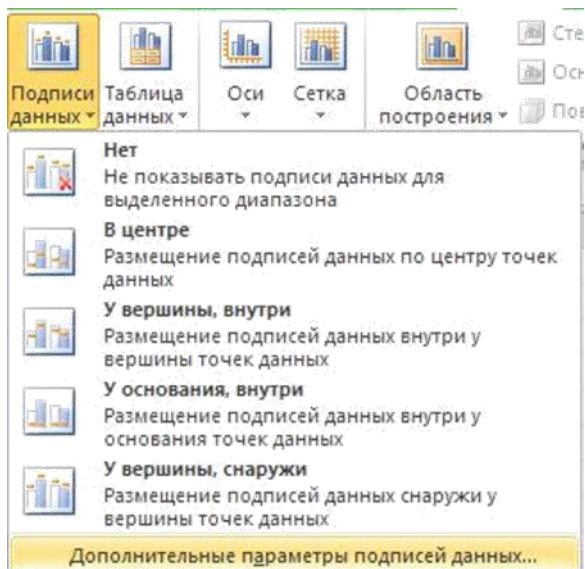
6. Далее выполните команду: **Макет/(Текущий фрагмент)Формат выделенного объекта**.
7. В появившемся окне **Формат оси** выберем вкладку **Число**.
8. Далее выбираем **Числовые форматы: Процентный, число десятичных знаков: 2**. Нажмите **Заккрыть**.



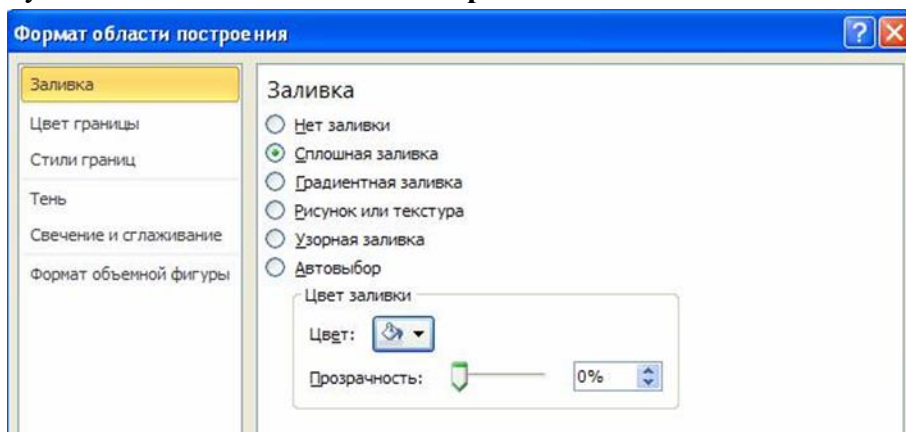
9. Выбираем **Горизонтальную ось (категорий)**, аналогично с пунктом 5.
10. Далее выполните команду: **Макет/(Текущий фрагмент)Формат выделенного объекта**.
11. В появившемся окне **Формат оси** выберем вкладку **Выравнивание**. В строке **Направление текста** в выпадающем списке выбираем пункт **По-вернуть весь текст на 270**. Нажимаем **Заккрыть**.



12. Выбираем **Ряд1**, аналогично с пунктом 5.
13. Далее выполните команду: **Макет/(Подписи) Подписи данных**.



14. В выпадающем списке выберите пункт **Дополнительные параметры подписей данных**.
15. В открывшемся окне, по аналогии с предыдущими пунктами, выбираем также **Числовой формат: процентный, число десятичных знаков: 1**.
16. Во вкладке **Параметры подписи** установите следующие параметры: **Включить в подписи: значения** и **Положение подписи: у вершины, снаружи**. Нажимаем **Заккрыть**.
17. Аналогично предыдущим пунктам 5 и 6 выберите **Область построения**. В открывшемся окне **Формат области построения** выбираем: **Заливка/сплошная заливка**. В строке **Цвет** выбираем нужный нам цвет. Нажимаем **Заккрыть**.

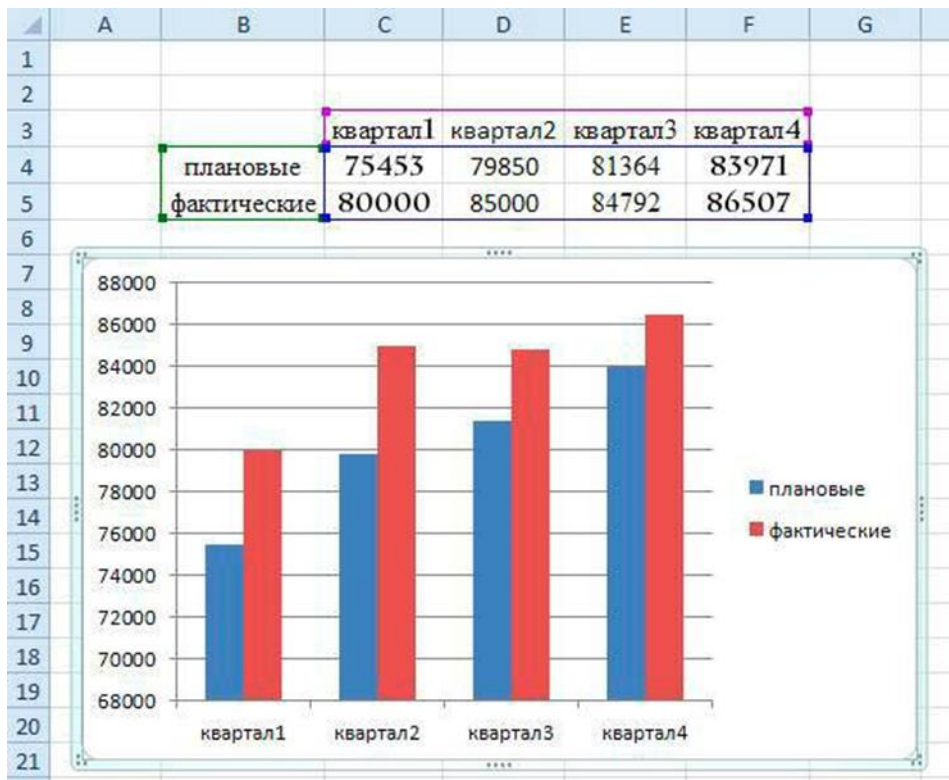


18. При построении диаграммы справа у нас есть надпись: **Ряд1**. Это **Легенда**, уберем ее выполнив команду: **Макет/(подписи)Легенда/Нет**.
19. Расположите диаграмму под таблицей и установите ей соответствующий размер.

Использование рисунков в диаграммах

Создание таблицы данных и диаграммы

С помощью меню **Диаграмма** можно корректировать и изменять любые параметры и свойства диаграмм: изменять тип диаграмм, задавать цвета и стили линий диаграммы, форматировать диаграмму, изменять формат построения диаграммы и т.д. С помощью мастера диаграмм можно также строить графики различных функций. Сначала создать таблицу и диаграмму следующего вида.



Создадим таблицу данных.

	A	B	C	D	E	F
1						
2						
3			квартал1	квартал2	квартал3	квартал4
4		плановые	75453	79850	81364	83971
5		фактические	80000	85000	84792	86507

1. Поместите курсор мыши в рабочую область программы. Наберите в ячейке **C3** имя поля: **квартал1**.
2. Выделите ячейку **C3**, далее нажмите на черный квадратик, находящийся в правом нижнем углу ячейки, затем, не отпуская, кнопку мыши перетащите курсор до ячейки **F3**. У вас должно получиться следующее:

	C	D	E	F	G
		квартал1	квартал2	квартал3	квартал4

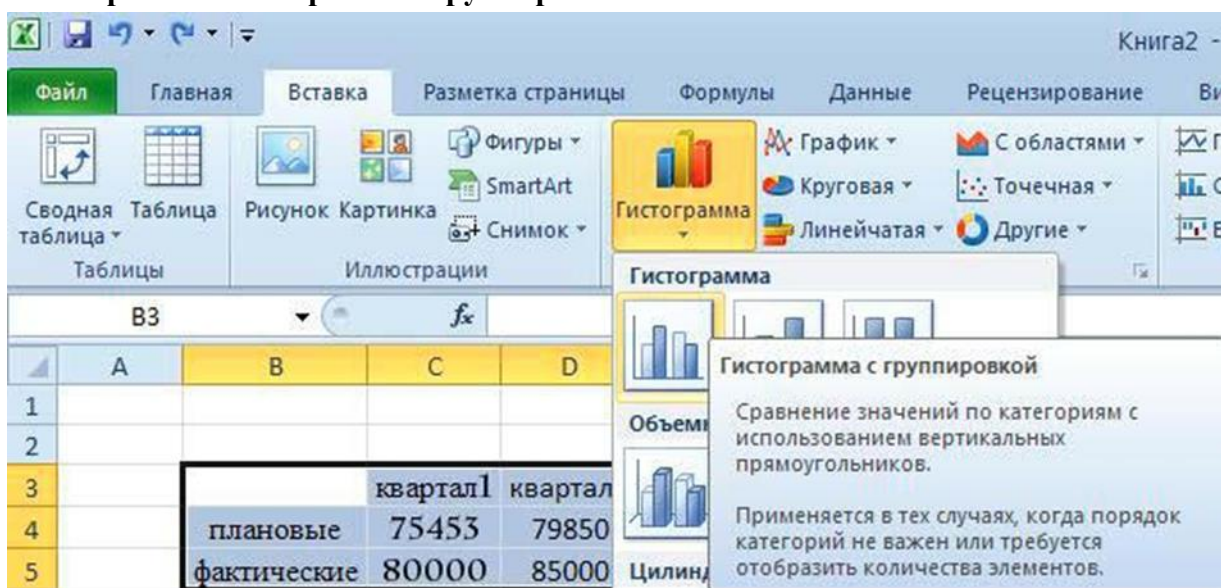
3. В ячейки **B4** и **B5** введите имена полей: **плановые** и **фактические** соответственно.
4. В диапазон ячеек **C4:F5** ввести числовые данные.

	A	B	C	D	E	F
1						
2						
3			квартал1	квартал2	квартал3	квартал4
4		плановые				
5		фактические				

	A	B	C	D	E	F
1						
2						
3			квартал1	квартал2	квартал3	квартал4
4		плановые	75453	79850	81364	83971
5		фактические	80000	85000	84792	86507

5. Выделите ячейки **B3:F5** и установите шрифт **Charlemagne** 12 пунктов, а затем выровняйте по центру. В итоге мы получим исходную таблицу.

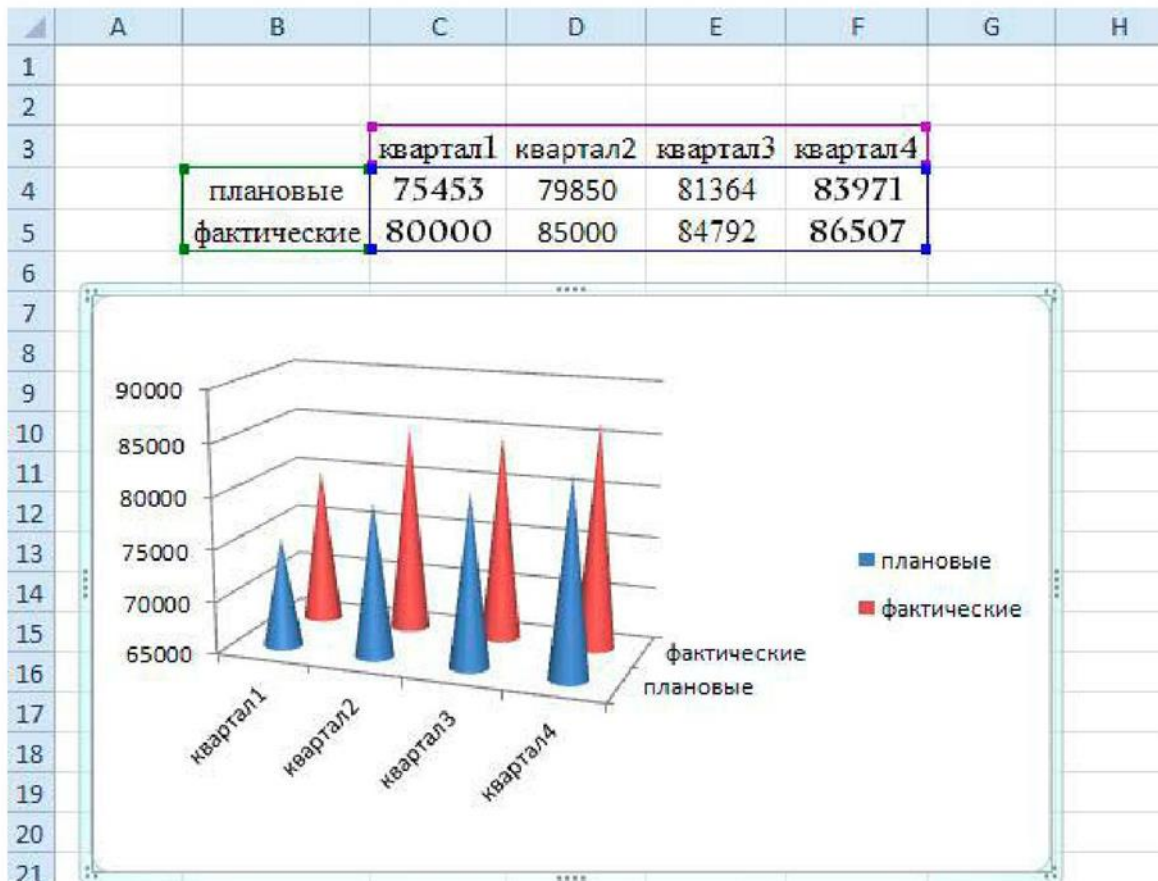
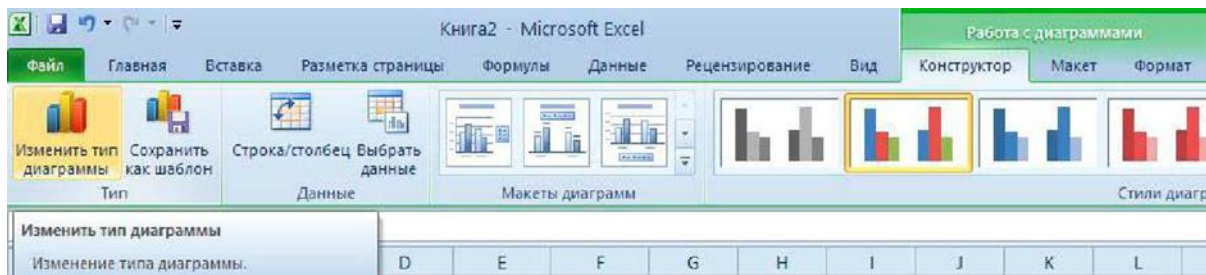
4. Выделить введенные данные, далее выполните команду **Вставка/(Диаграммы) Гистограмма/Гистограмма с группировкой**.



Изменение типа диаграммы.

1. Для того чтобы изменить тип диаграммы, щелкните на диаграмму.

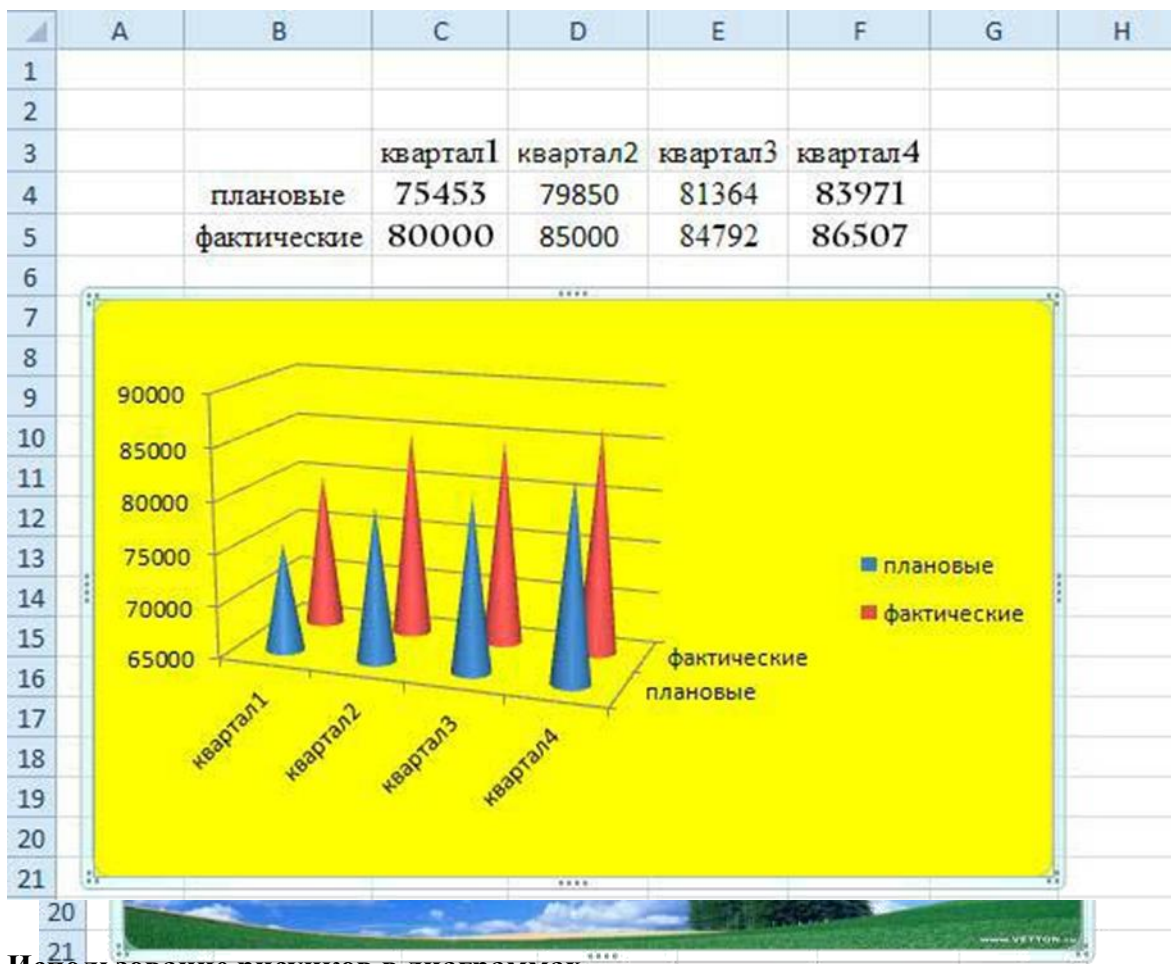
2. Выполните команду **Конструктор / (Тип) Изменить тип диаграммы / Объемная коническая**.



Выбор цвета, границ и заливки допускается в маркерах данных, в области диаграммы, в области построения и в других элементах диаграммы.

Следующая процедура используется для изменения цветов, узора, ширины линии или типа рамки для маркеров данных, области диаграммы, области построения, сетки, осей и подписей делений на плоских и объемных диаграммах:

1. Щелкните левой кнопкой мыши на область диаграммы.
2. При необходимости заливки фигуры выберите **Формат/(Стили фигур)Заливка фигуры** и укажите нужный вид во вкладках **Градиентная** или **Текстура**. (В нашем примере **Формат/(Стили фигур)Заливка фигуры/Желтый цвет**)
3. При необходимости изменения вида диаграммы пользуйтесь вкладками **Конструктор**, **Макет** или **Формат**.

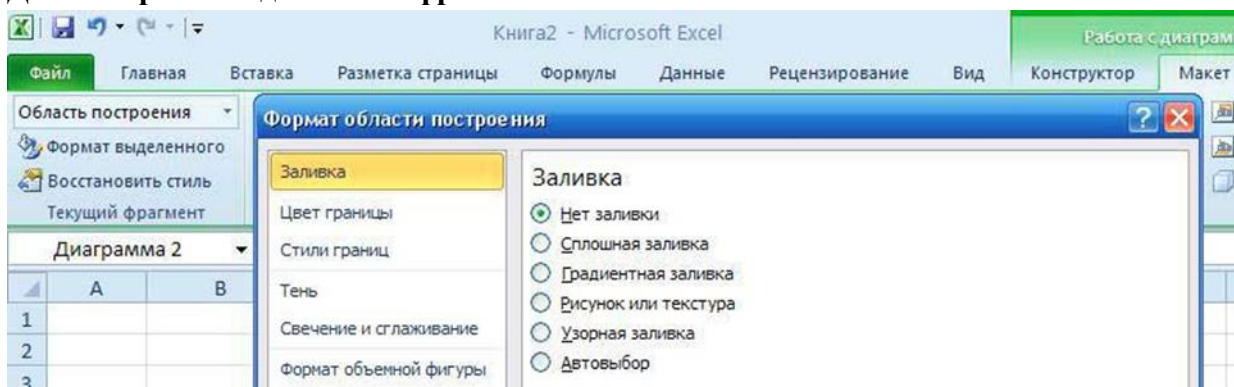


Использование рисунков в диаграммах.

Следующая процедура используется для добавления рисунка – например, точечного – к определенным типам маркеров данных, области диаграммы, области построения или легенде на плоских и объемных диаграммах.

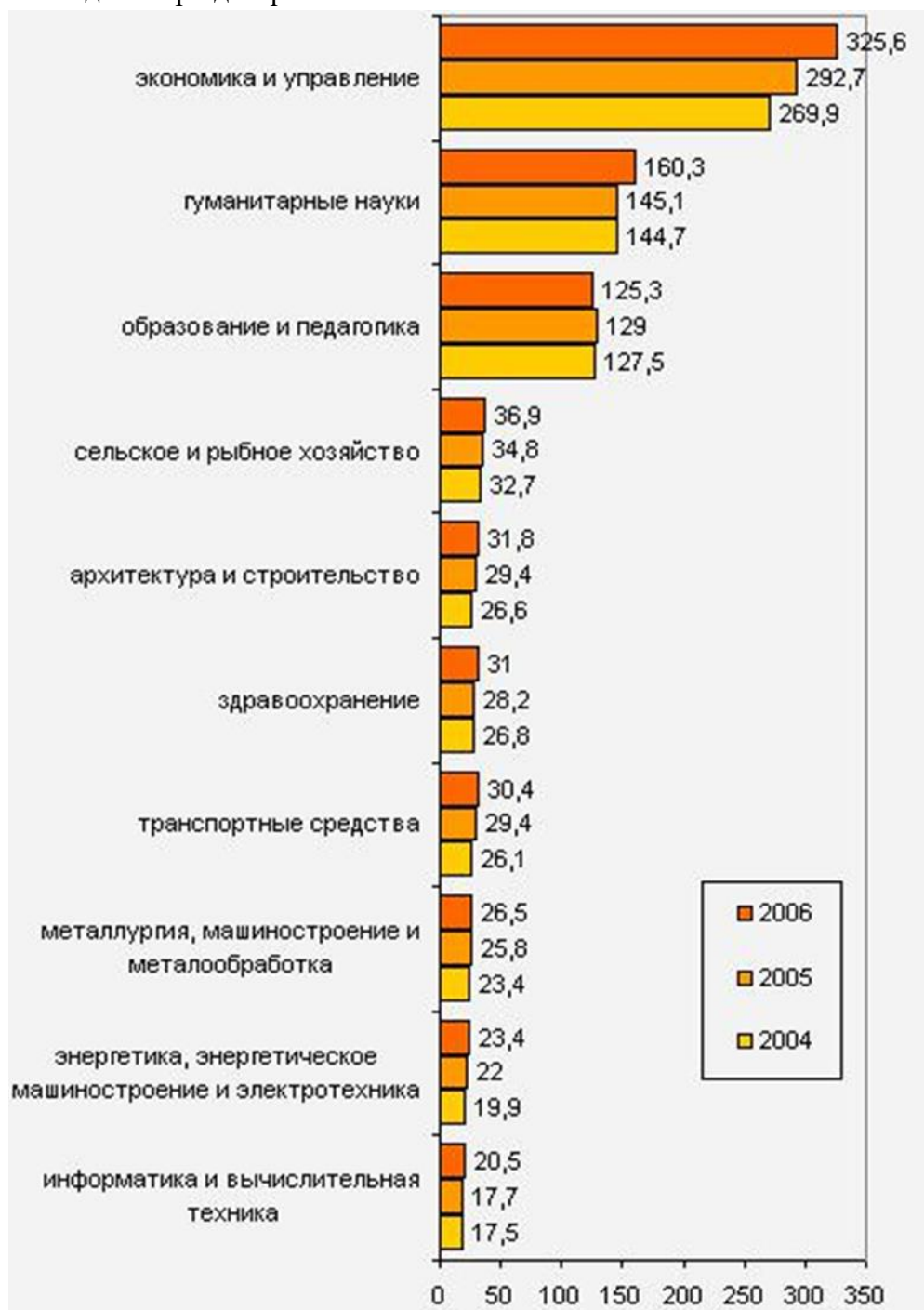
Для того чтобы у вас получилось также, выполните следующие команды.

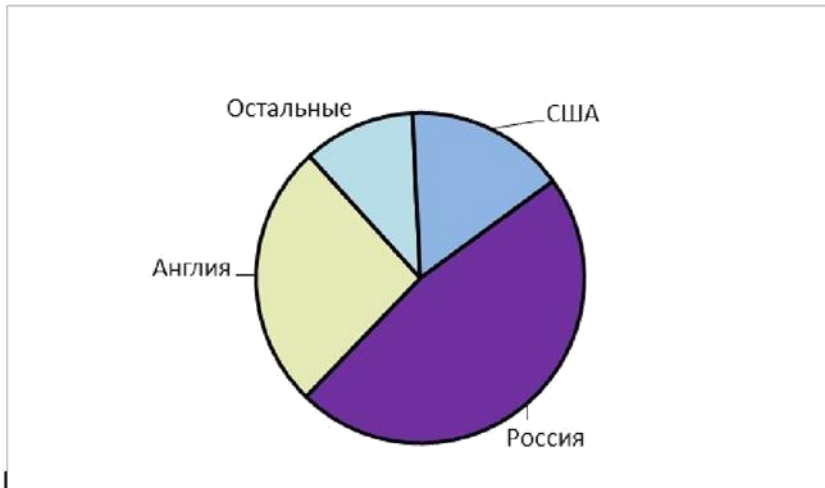
1. Щелкните левой кнопкой мыши на область диаграммы.
2. Выполните команду **Формат/(Стили фигур) Заливка фигуры/Рисунок**.
3. В появившемся окне **Вставка рисунка** выберите папку, где находится нужный рисунок, а затем дважды щелкните его.
4. Далее вы полните команду **Макет/Текущий фрагмент**. Выберите **Область построения**. Далее **Формат выделенного фрагмента/Нет заливки**.



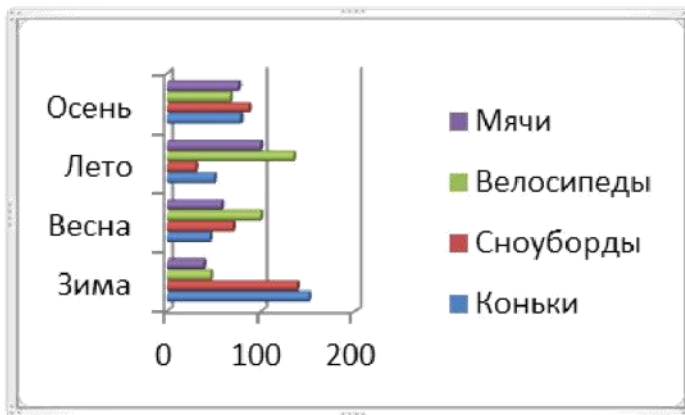
Задания практической работы

1. Создайте три диаграммы.





Объем продаж в спортивном магазине



Контрольные вопросы:

- 1 Каково назначение диаграмм? Что такое легенда, категория, ряд данных?
- 2 Какие типы диаграмм вы знаете?
- 3 Какие элементы (области) диаграммы вы знаете?
- 4 Как построить диаграмму?
- 5 Каково назначение Мастера диаграмм?
- 6 Как редактировать диаграмму?
- 7 Как добавить (удалить) из диаграммы ряды данных (категорию)?
- 8 Какие действия с графическими объектами можно производить в MS EXCEL?

Практическая работа № 15. Создание базы данных

Цель: формирование навыков создания баз данных, а также таблиц баз данных в режиме Конструктора; отработка навыков определения типов данных. научиться находить в базе данных нужную информацию, используя Фильтр по выделенному, Расширенный фильтр; научиться делать сортировку информации. научиться создавать таблицу с полем OLE; создавать связи между таблицами; закрепить знания о связях с отношениями «один к одному», «один со многими».

Задание 1

Создать базу данных «Географическая». В ней создать таблицы «Атлас мира», «Климат», «Население» по образцам.

Методические указания

А) Базы данных

Хранение информации - одна из важнейших функций компьютера. Одним из распространенных средств хранения являются базы данных. *База данных* - файл специального формата, содержащий информацию, структурированную заданным образом. Большинство баз данных имеет *табличную структуру*. В базах данных столбцы называются *полями*, а строки - *записями*.

Поля - это основные элементы структуры базы данных. Они обладают *свойствами*, от которых зависит, какие типы данных можно вносить в поле, а какие - нет, а также что можно делать с данными, содержащимися в поле.

Очевидным уникальным свойством любого поля является его имя. Разумеется, одна база данных не может иметь двух полей с одинаковым именем, поскольку компьютер запутается в их содержимом.

1) СУБД MS Access

Системы управления базами данных (СУБД) - это программные средства, с помощью которых можно создавать базы данных, наполнять их и работать с ними. С помощью MSAccessобычные пользователи получили удобное средство для создания и эксплуатации достаточно мощных баз данных без необходимости что-либо программировать.

Исходное окно Accessотличается простотой и лаконичностью. Шесть вкладок этого окна представляют шесть объектов, с которыми работает программа.

Таблицы - это основные объекты базы данных. В них хранятся данные.

Запросы - это специальные структуры, предназначенные для обработки данных базы. С помощью запросов данные упорядочивают, фильтруют, отбирают, изменяют, объединяют, то есть обрабатывают.

Формы - это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся.

Отчеты - это формы «наоборот». С их помощью данные выдают на принтер в удобном и наглядном виде.

С организационной точки зрения в работе с любой базой данных есть два режима: проектировочный и эксплуатационный (пользовательский). Создатель базы имеет право создавать в ней новые объекты (например, таблицы), задавать их структуру, менять свойства полей, устанавливать необходимые связи. Он работает со структурой базы и имеет полный доступ к базе.



Создание таблиц

Создание базы начинается с создания первой таблицы. Есть несколько способов создания новой таблицы, отличающихся уровнем автоматизации. Самый «автоматичный» способ состоит в импорте таблиц из другой базы. В зависимости от обстоятельств из импортируемой таблицы может поступить структура полей, их названия и свойства, а также и содержание базы. Если что-то импортируется не совсем так, как надо, необходимые правки (например, в свойства полей) вносят вручную.

Опытные разработчики пользуются Мастером таблиц. Это программа, ускоряющая создание структуры таблицы. Мастер задает ряд вопросов и, руководствуясь полученными ответами, создает структуру таблицы автоматически. Несмотря на то, что этот режим служит для упрощения работы, начинающим пользоваться им не рекомендуется, поскольку, не владея всей терминологией, легко запутаться в вопросах и ответах. Первые таблицы стоит попробовать создать вручную.

Наиболее универсальный ручной метод предоставляет пункт Конструктор. В этом режиме можно самостоятельно задать имена полей, выбрать их тип и настроить свойства.

Практические задания

1. Запустить Access (*Пуск/Программы/Microsoft Office/ Microsoft Access*).
2. В появившемся диалоговом окне выбрать *Новая база данных* и кликнуть по кнопке *ОК* или *Создать файл/Новая база данных*. На экране появится окно *Файл новой базы данных*.
3. Выбрать диск и папку, где будет сохраняться файл БД. В поле *Имя файла* ввести «Географическая. Имя». Кликнуть по кнопке *Создать*, после чего на экране появится окно базы данных.
4. Для создания таблицы выбрать вкладку *Таблицы* и кликнуть по кнопке *Создать*.
5. В окне *Новая таблица* выбрать пункт *Конструктор* и кликнуть по кнопке *ОК*. На экране появится окно таблицы.
6. **Начинаем конструировать таблицу.** Для этого в графе *Имя поля* необходимо ввести имена полей (заголовки таблицы). Необходимо учитывать, что поле «Площадь» имеет числовой тип данных, а все остальные поля – текстовый. Ввести в таблицу имена полей и их типы по образцу (см. рис. 1).
7. Поле «Страна» сделайте ключевым: поставьте маркер таблицы на нужное поле и нажмите *Ключевое поле*  на Панели инструментов.
8. Сохранить таблицу (нажать *Сохранить* на панели инструментов). При этом записать в окне *Сохранение* имя «Атлас мира».
9. Кликнуть по кнопке *Вид* на панели инструментов: откроется *Режим таблицы* .
10. Начиная заполнять таблицу *Атлас мира* (см. рис. 2) **по одной строке**.
11. Заполнить таблицу. Сохранить ее.
12. Создать аналогично таблицы «Климат» (см. рис. 3) и «Население» (см. рис. 4), начиная выполнять с пункта 4.
13. Поле «Страна» в таблице «Население» сделать ключевым.

Атлас мира : таблица		
	Имя поля	Тип данных
	Страна	Текстовый
	Столица	Текстовый
	Часть света	Текстовый
	Площадь, кв км	Числовой
	Денежная единица	Текстовый
	Рельеф	Текстовый
	Тип экономики	Текстовый

Рис. 1

Атлас мира : таблица							
	Страна	Столица	Часть света	Площадь, кв км	Денежная единица	Рельеф	Тип экономики
▶	+ Израиль	Тель-Авив	Азия	20770	шекель	равнинный	индустриально-аграрная
	+ Австрия	Вена	Европа	83900	Евро	горный	сфера услуг
	+ Южная Корея	Сеул	Азия	99400	вона	горно-равнинный	индустриальная
	+ Италия	Рим	Европа	301300	Евро	горный	индустриальная
	+ Япония	Токио	Азия	377873	иена	горный	индустриальная
	+ Египет	Каир	Африка	1001450	фунт	горно-равнинный	аграрная
	+ Боливия	Ла-Пас	Америка	1098600	боливиано	горный	аграрная
	+ Аргентина	Буенос-Айрес	Америка	2766890	песо	горно-равнинный	аграрная
	+ Индия	Нью-Дели	Азия	3287600	рупия	горно-равнинный	аграрная
	+ Австралия	Канберра	Австралия	7686850	доллар	горно-равнинный	индустриальная
*				0			

Рис. 2

Климат

Страна	Регион	Средняя температура июля	Средняя температура января	Среднегодовой уровень осадков, мм
Австралия		22	34	500
Австрия		20	-3	900
Аргентина	юг	1	9	200
Аргентина	север	18	28	500
Боливия		18	28	2000
Египет		35	25	100
Израиль		30	15	500
Индия		28	20	2500
Италия	север	20	3	1500
Италия	юг	27	11	1000
Южная Корея		26	0	1300
Япония	север	19	-10	1200
Япония	юг	28	16	1500

Рис. 3

Население

Страна	Численность ч	Плотность чел/кв км	Религия	Национальность
Австралия	19731000	2	христианство, ислам, иудаизм	английские потомки
Австрия	8100000	96	христианство	австрийцы
Аргентина	38740000	14	христианство	испанцы, индейцы, метисы
Боливия	8300000	7	христианство	индейцы, метисы
Египет	70712000	70	ислам	египтяне
Израиль	6029000	290	иудаизм, христианство	евреи
Индия	1045845000	318	индуизм	индийцы
Италия	57600000	191	христианство	итальянцы
Южная Корея	48289000	485	буддизм, христианство	корейцы
Япония	126700000	335	буддизм, конфуцианство	японцы

Рис. 4

Задание 2

I. В таблице «Атлас мира» географической базы данных отсортировать:

- страны в порядке, обратном алфавитному;
- столицы по алфавиту и в порядке, обратном алфавитному;
- найти страну с наибольшей и наименьшей площадью.

II. В этой же таблице найти:

- все страны, расположенные в Европе (Азии, Америке);

- страны, имеющие горный (горно-равнинный рельеф);
- индустриальные (аграрные) страны;
- страны, в которых значение площади заканчивается цифрами 00;
- страны, в которых значение площади заканчивается цифрами 00 и название начинается на букву «И»;
- страны с денежной единицей «евро» и «иена».

III. В таблице «Климат», используя сортировку:

- найти страну с наибольшей и наименьшей летней температурой;
- найти страну с наибольшей и наименьшей зимней температурой;
- найти страну с наибольшим и наименьшим среднегодовым уровнем осадков.

IV. В таблице «Климат», используя фильтрацию данных, найти:

- страны со средней температурой июля 20 °С; со средней температурой января 28 °С;
- страны, в которых среднегодовой уровень осадков 500;
- страны, в которых значение средней температуры января заканчивается цифрой 0;
- страны, в которых значение среднегодового уровня осадков заканчивается цифрами 00, а название начинается на «Ав».

V. В таблице «Население», используя сортировку:


- найти страну с наибольшей и наименьшей численностью населения;
- найти страну с наибольшей и наименьшей плотностью населения;
- отсортировать поле «Религия» по возрастанию.



VI. В таблице «Население», используя фильтрацию данных, найти:

- страны, жители которых христиане;
- страны, жители которых исповедуют христианство и ислам;
- страны, в которых проживают австрийцы и корейцы;
- страны, в которых значение плотности населения заканчивается цифрой 5;
- страны, жители которых христиане, а название страны начинается на букву «А».


Порядок выполнения

1. Запустить *Microsoft Access*. Открыть свою географическую базу данных, в ней – таблицу «Атлас мира».

2. Для выполнения заданий пункта I использовать пиктограммы на панели инструментов *Сортировка по возрастанию* и *Сортировка по убыванию* .

3. При выполнении пункта II для поиска стран, расположенных в Европе, установить курсор на какой-либо записи, которая содержит данный текст (поле «Континент») и нажать кнопку *Фильтр по выделенному*  на панели инструментов. Для отмены необходимо нажать кнопку *Удалить фильтр* .

4. По рельефу и типу экономики отфильтровать аналогично пункту 3.

5. Для поиска стран, в которых значение площади заканчивается цифрами 00, выбрать в строке меню команду *Записи/Фильтр/Расширенный фильтр*. Перенести поле «Площадь» и указать условие отбора записей *00. Выбрать команду *Фильтр/Применить фильтр* или кнопку на панели инструментов  .

6. Для следующих заданий условия выбора будут соответственно *00 в поле «Площадь» и И* в поле «Страна»; «евро» или «иена» в поле «Денежная единица».

7. Задания III, IV, V, VI выполнять аналогично заданиям I и II.

Задание 3

I. В своей географической базе данных создать таблицу «О государстве» (см. рис. 1) по образцу. В качестве ключевого задать поле «Страна».

II. Создать связи между таблицами «Атлас мира», «Климат», «Население», «О государстве» географической базы данных. При этом «Атлас мира» сделать главной таблицей, а таблицы «Климат», «Население», «О государстве» - подчиненными. Обеспечить целостность данных, а также одновременное обновление и удаление данных во всех подчиненных таблицах при их изменении в главной таблице.

III. На схеме данных рассмотреть полученные связи. Определить типы отношений.

О государстве

О государстве : таблица				
	Страна	Государственное устройство	Государственный язык	Флаг
+	Австралия	конституционная монархия	английский	
+	Австрия	республика	немецкий	
+	Аргентина	республика	испанский	
+	Боливия	республика	испанский	
+	Египет	республика	арабский	
+	Израиль	парламенская демократия	иврит, арабский	
+	Индия	республика	хинди, английский	
+	Италия	республика	итальянский	
▶ +	Южная Корея	республика	корейский	
+	Япония	конституционная монархия	японский	Файл снимка

Порядок выполнения

I. 1. Запустить *Access* (*Пуск/Программы/Microsoft Office/ Microsoft Access*). Открыть свою географическую базу данных.

2. Создать таблицу «О государстве». Для вставки рисунка в поле «Флаг» выполнить следующие действия:

- ✓ установить курсор в поле «Флаг» (тип данных – поле объекта *OLE*);
- ✓ выполнить команду *Объект* из меню *Вставка*;
- ✓ выбрать пункт *Рисунок Paintbrush* (или *Точечный рисунок*);
- ✓ нарисовать флаг и закрыть графический редактор.

Флаги государств



Австралия



Австрия



Аргентина



Боливия



Египет



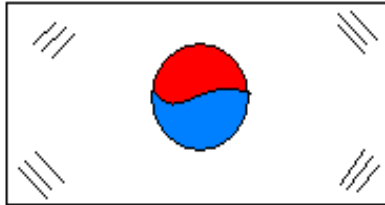
Израиль



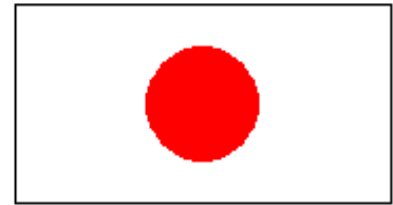
Индия




Италия



Южная Корея



Япония

II. 3. Открыть окно *Схема данных*, кликнув на кнопке на панели инструментов или командой *Сервис/Схема данных* .

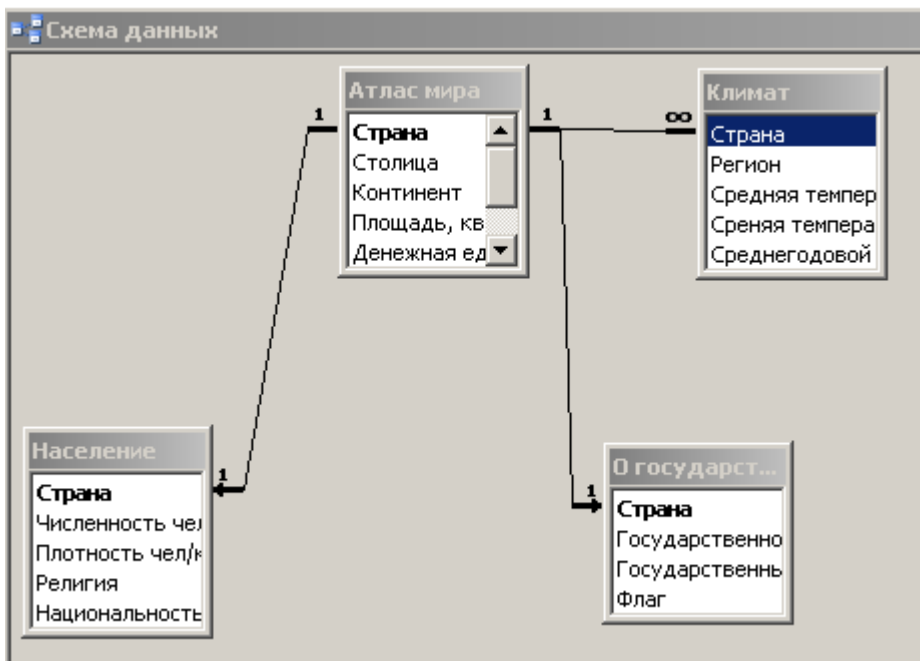
4. В меню *Связи/Добавление таблицы* выбрать требующиеся таблицы. Для включения в структуру межтабличных связей нажать кнопку *Добавить*. Закрыть окно *Добавление таблицы*.

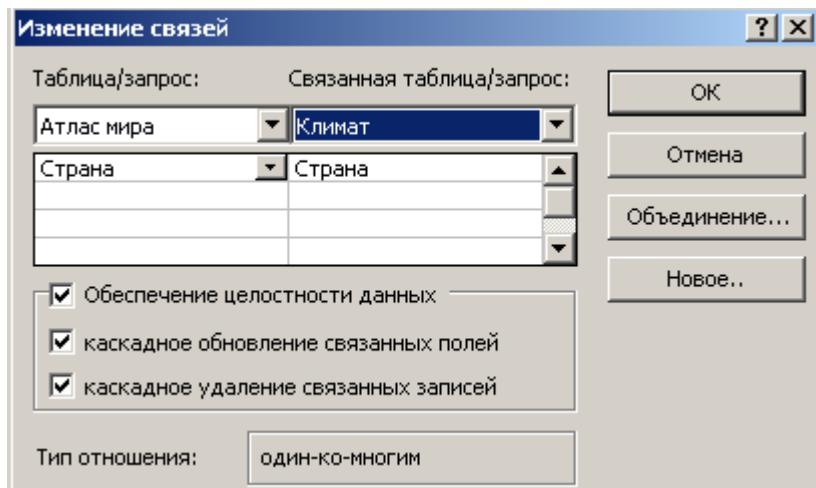
5. Связь между полями установить, перетягивая имя поля из одной таблицы в другую на соответствующее связанное поле (поле «Страна» из таблицы «Атлас мира» на соответствующие поля других таблиц).

6. В диалоговом окне *Смена связей* задать свойства создаваемой связи следующим образом:

- ✓ включить флажок *Обеспечение условия целостности данных*;
- ✓ включить флажок *Каскадное обновление связанных полей*;
- ✓ включить флажок *Каскадное удаление связанных записей*.

Нажать кнопку *Создать*.





III. 7. После создания всех необходимых связей выполнить пункт III.

8. Закрыть «Схему данных», ответив положительно на вопрос о сохранении.

9. Проверить условие целостности данных, обновление связанных полей, удаление связанных записей. Для этого в таблице «Атлас мира» исправить название одной из стран, сохранить. Открыть связанные таблицы, убедиться в обновлении связей.

Практическая работа № 16. Создание запросов, отчетов и форм в базе данных

Цель: научиться создавать запросы, отчеты и формы в БД с помощью MS Access

Методические указания

Задание 1


В своей географической базе данных на основе таблицы «Атлас мира» создать запрос, добавив к нему название страны, столицы, части света.

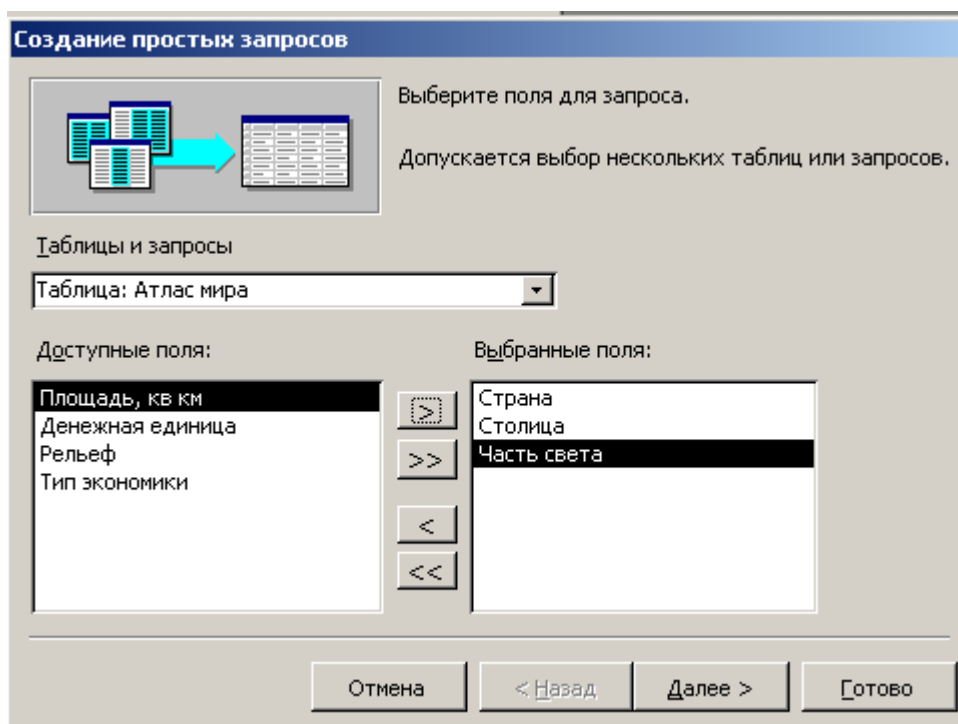
Запросы

В результате работы запроса из общей выходной базы формируется *результатирующая таблица*, которая содержит часть общей информации, которая отвечает запросу.

Важной частью запросов является то, что при создании результирующей таблицы можно не только выбрать информацию из базы, но и обрабатывать ее. После работы запроса данные могут упорядочиваться (сортироваться), фильтроваться (отсеиваться), объединяться, разделяться, изменяться и при этом никаких изменений в базовых таблицах может не происходить.

Порядок выполнения

1. Открыть свою географическую базу данных.
2. В окне базы данных выбрать закладку *Запросы* и кликнуть по кнопке *Создать*.
3. В окне *Новый запрос* выбрать в списке *Простой запрос/ОК*.
4. Из списка выбрать таблицу «Атлас мира», на основе которой будет создан запрос.
5. С помощью кнопки  выбрать из таблицы поля «Страна», «Столица», «Часть света». Нажать *Далее*.
6. Ввести имя *Запрос по странам* и кликнуть по кнопке *Готово*.



Задание 2.

I. На основе таблицы «Атлас мира» создать запрос на выборку:

- а. Выбрать страны, которые имеют площадь свыше 1 000 000 км². Запрос назвать «Наибольшие страны».
- б. Выбрать страны, имеющие площадь менее 1 000 000 км². Запрос назвать «Наименьшие страны».

II. На основе таблицы «Климат» создать запрос на выборку:

- а. Выбрать страны, в которых среднегодовой уровень осадков больше 1000 мм. Запрос назвать «Влажные страны».
- б. Выбрать страны, в которых среднегодовой уровень осадков меньше 1000 мм. Запрос назвать «Засушливые страны».
- в. Выбрать страны, в которых средняя температура января превышает 18 °С, а средняя температура июля превышает 25 °С (для стран северного полушария) или средняя температура января превышает 25 °С, а средняя температура июля превышает 18 °С (для стран южного полушария).

Порядок выполнения

1. Открыть свою географическую базу данных.
2. Для создания запроса выбрать закладку *Запросы* и кликнуть по кнопке *Создать*.
3. В диалоговом окне *Новый запрос* выбрать пункт *Конструктор*.
4. Выбрать таблицу, на основе которой будет создан запрос. Выбор осуществляется в диалоговом окне *Добавление таблицы*. Таблицу «Атлас мира» занести в верхнюю половину бланка запроса по образцу, кликнув по кнопке *Добавить*. Закрыть окно *Добавление таблицы*.
5. Строку *Поле* заполнить перетягиванием названий полей из таблиц в верхней части бланка (для заданий I.а. и I.б. этими полями будут «Страна», «Столица», «Часть света», «Площадь»). Каждому полю будущей *результующей таблицы* соответствует один столбик бланка запроса по образцу.
6. В строке *Условия отбора* записать критерий, по которому будут отобраны записи для включения в *результующую таблицу*. По каждому полю можно создать свое условие отбора. Для задания I.а. условие отбора имеет вид $> 1\,000\,000$ по полю «Площадь», для задания I.б. – $< 1\,000\,000$ по полю «Площадь» (см. рис. 1).
7. Запуск запроса выполнить, кликнув по кнопке *Вид*. При запуске образуется *результующая таблица* (см. рис. 2).
8. Сохранить запрос.
9. Задания II.а, б выполнить аналогично пунктам 1 – 8. При выполнении пункта II.в. условие отбора задается на двух полях.

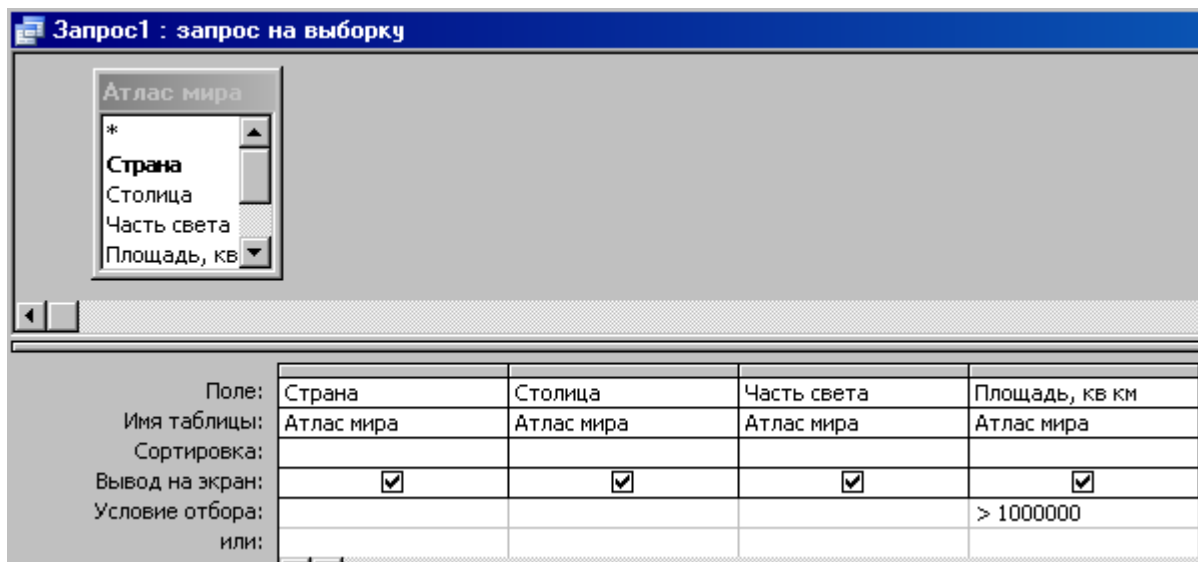


Рис. 1

	Страна	Столица	Часть света	Площадь, кв км
▶	Австралия	Канберра	Австралия	7686850
	Египет	Каир	Африка	1001450
	Аргентина	Буенос-Айрес	Америка	2766890
	Индия	Нью-Дели	Азия	3287600
	Боливия	Ла-Пас	Америка	1098600
*				0

Рис. 2

Задание 3

I. На основе таблицы «Атлас мира»:

- Создать запрос на рельеф стран. Назвать запрос «Рельеф». Вывести страны с горным рельефом.
- Создать запрос на экономику стран. Создать запрос «Экономика». Сначала ввести только индустриальные страны, потом – аграрные.
- Создать запрос с параметром по полю «Часть света». Назвать запрос «Часть света». Вывести страны, расположенные в Азии (Европе, Америке).

II. На основе таблицы «О государстве» создать запрос о государственном устройстве стран, включив поля «Страна», «Государственное устройство», «Государственный язык». Назвать запрос «Устройство». Вывести страны, в которых государственное устройство – республика.

III. На основе таблицы «Население» создать запрос о вероисповедании жителей стран. Назвать запрос «Религия». Вывести страны, где жители исповедуют христианство.

Порядок выполнения

- Открыть свою географическую базу данных.
- Дважды кликнуть по значку *Создание запроса в режиме конструктора*.
- Выбрать таблицу, на основе которой будет создан запрос. Выбор осуществляется в диалоговом окне *Добавление таблицы*. Таблицу занести в верхнюю половину бланка запроса по образцу, кликнув по кнопке *Добавить*. Закрыть окно *Добавление таблицы*.
- Строку *Поле* заполнить, перетягивая названия полей из таблиц в верхней части бланка (для задания I.a этими полями будут «Страна», «Столица», «Рельеф»).

5. В строке *Условие выбора* в поле, по которому осуществляется выбор («Рельеф»), записать команду.

LIKE [...].

В квадратных скобках записать какой-либо текст, обращенный к пользователю, например:

LIKE [Введите рельеф]. (см. рис. 1)

6. После запуска запроса кнопкой *Вид*  откроется диалоговое окно. Ввести параметр – слово *горный* (см. рис. 2).

7. Сохранить запрос. Таблица запроса с параметром может иметь вид рис. 3.

8. Задания **I.б, в, II, III** выполнить аналогично пунктам 2 – 8.

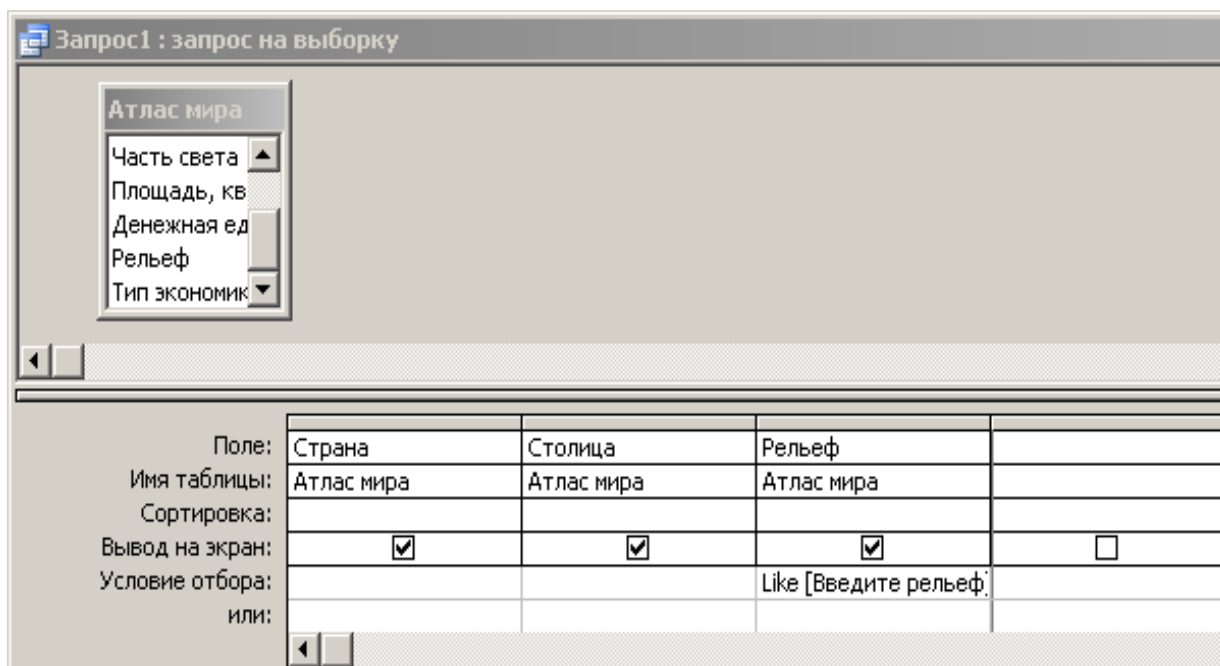


Рис. 1

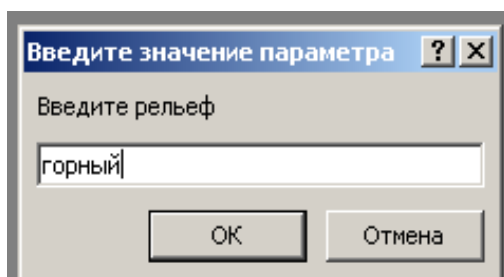


Рис. 2



	Страна	Столица	Рельеф
▶	Австрия	Вена	горный
	Италия	Рим	горный
	Боливия	Ла-Пас	горный
	Япония	Токио	горный
*			

Рис. 3

Обычно разработчик базы данных создает структуру таблиц и запросов, но заполнением таблиц информацией он не занимается. Для этого существуют специальные кадры (малоквалифицированные), которые выполняют функции составителя. Для упрощения их работы разработчик базы может подготовить специальные объекты – *формы*.

Форма представляет собой некоторый электронный бланк, в котором есть поля для введения данных. Составитель вводит данные в эти поля, и данные автоматически заносятся в таблицы базы.

По типу подачи данных формы бывают: *простые, одиночные, подчиненные, связанные*.

Простые формы – формы, созданные на основе одной таблицы или одного запроса.

Одиночные формы – формы, созданные на основе нескольких таблиц или запросов. Поля этих форм равнозначны.

Подчиненной формой называют форму, вставленную в другую форму. Первичная форма называется главной формой, а форма в середине формы называется подчиненной формой.

Связанные формы – простые формы, связанные между собой элементом управления, например, кнопкой.

Задание 4

I. Создать простые формы на основе таблиц «Атлас мира», «Климат», «Население», «О государстве», добавив поля таблиц. Внешний вид форм выбрать в *один столбик, табличный* или *выровненный*.

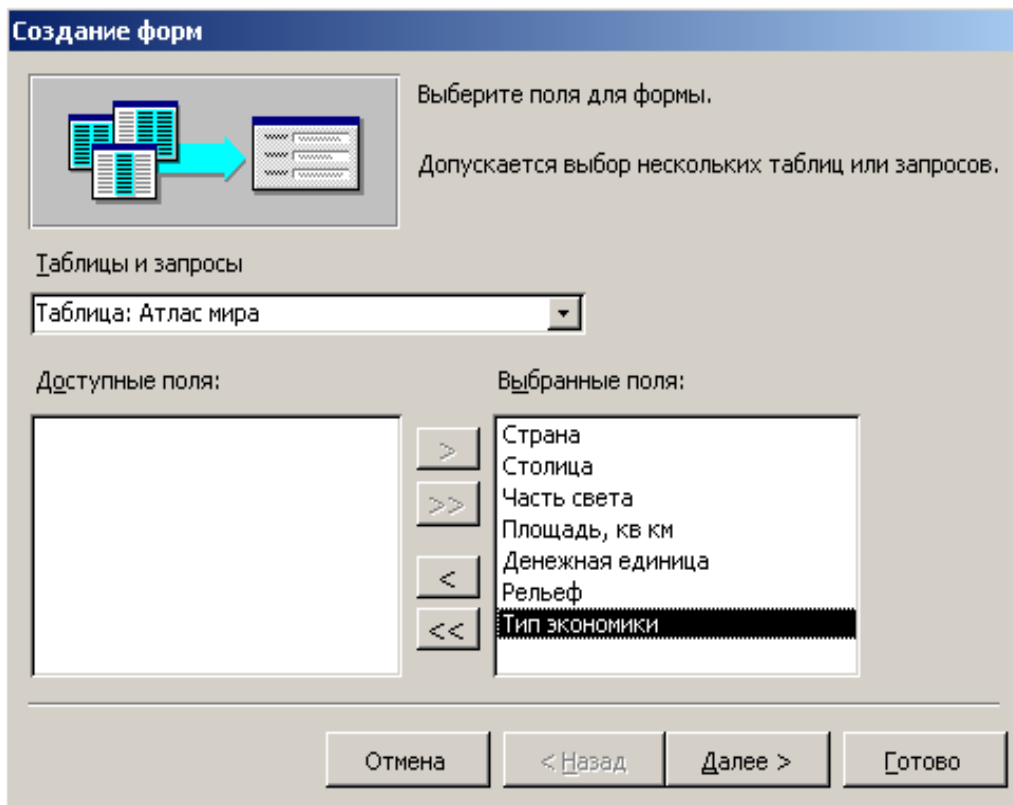
II. Создать одиночную форму «Политическая география», добавив в нее поля «Страна», «Столица» из таблицы «Атлас мира», а также поля «Государственное устройство», «Государственный язык», «Флаг» из таблицы «О государстве».

III. Создать простую форму «Экономическая география», добавив в нее поля «Страна», «Экономика», «Денежная единица» из таблицы «Атлас мира».

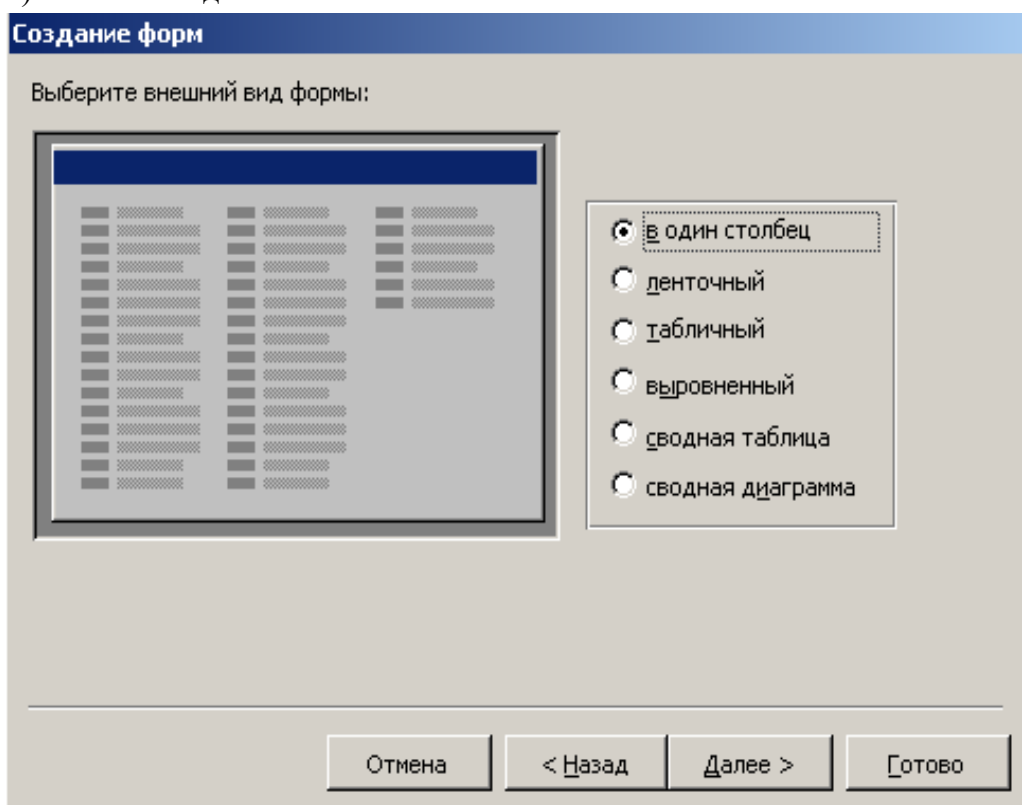
IV. Создать подчиненную форму «Физическая география», добавив в нее поля «Страна», «Часть света», «Площадь», «Рельеф» из таблицы «Атлас мира» и полей «Регион», «Средняя температура июля», «Средняя температура января», «Среднегодовой уровень осадков» из таблицы «Климат». Поля таблицы «Атлас мира» включить в главные формы, поля таблицы «Климат» - в подчиненные.

Порядок выполнения

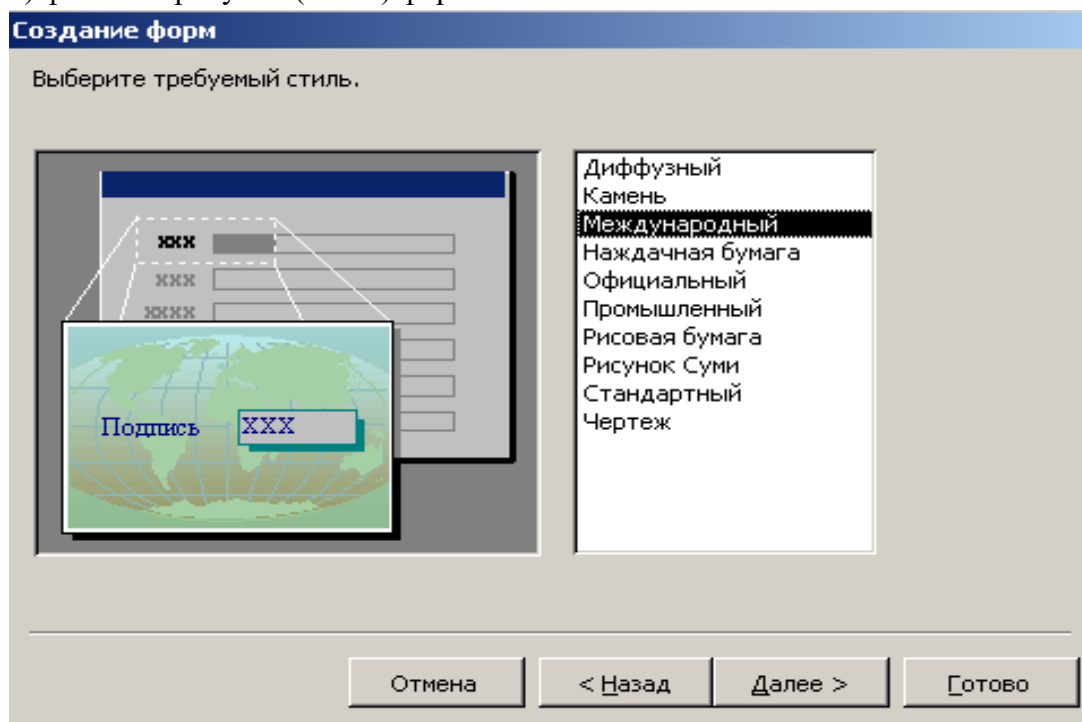
1. Открыть свою географическую базу данных.
2. Выбрать вкладку *Формы* и кликнуть по кнопке *Создать*. В диалоговом окне *Новая форма* выбрать пункт *Мастер форм*.
3. Далее выполнить инструкции *Мастера*:
 - а) выбрать поля:



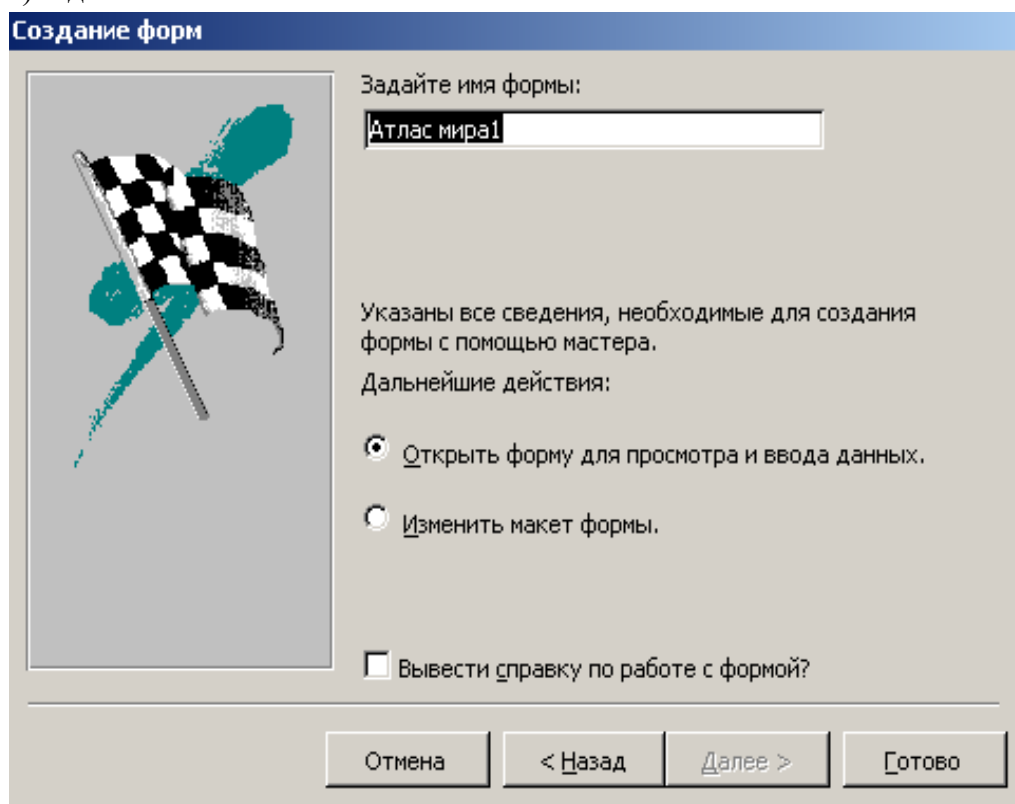
б) внешний вид:



в) фоновый рисунок (стиль) формы:



г) задать имя:



и кликнуть по кнопке *Готово*.

4. Созданная форма имеет вид:

Страна	Израиль
Столица	Тель-Авив
Часть света	Азия
Площадь, кв км	20770
Денежная единица	шекель
Рельеф	равнинный
Тип экономики	индустриально-аграрная

Запись: 1 из 10

Для создания *одиночной формы* при выполнении пункта **II** необходимо на втором шаге *Мастера* в окне *Тип подачи данных* выбрать таблицу «Климат».

Отчеты в БД предназначены для выведения на печать результатов обработки данных. Отчет – это готовый к печати электронный документ. Отчеты могут быть использованы для заполнения бланков документов, например, сертификатов о прохождении внешнего независимого оценивания.

Задание 5

I. На основе запроса «Самые жаркие страны» создать **строчный** автоотчет.

II. Создать отчет «Население» на основе таблицы «Население» **в столбик** с помощью *Мастера форм*

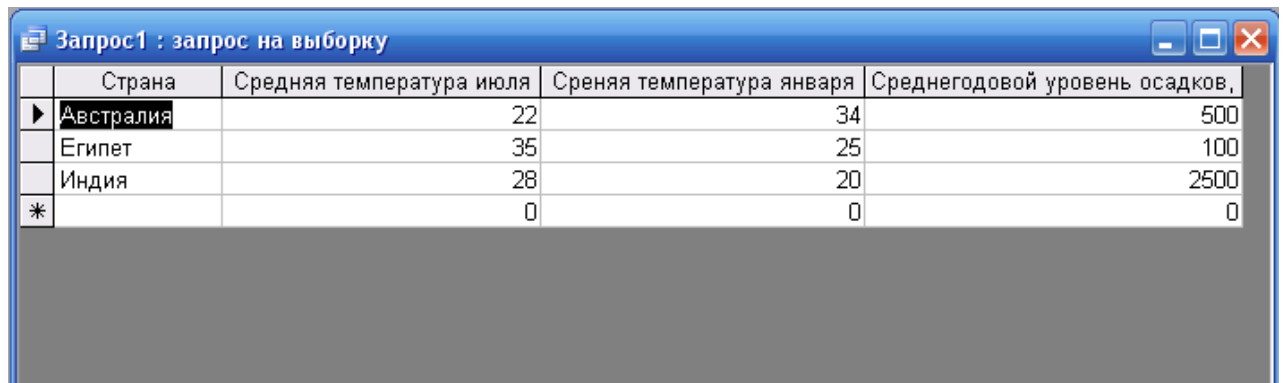
III. Создать отчет «Республики» на основе запроса «Устройство» по образцу.

Порядок выполнения

1. Открыть свою географическую базу данных.
2. Создать **запрос «Самые жаркие страны»** на основе таблицы «Климат».


Использовать поля *Средняя температура июля*, *Средняя температура января*, *Среднегодовой уровень осадков*. В строке *Условия отбора* записать критерий запроса: температура в течение года выше 20 °С.

3. Запуск запроса выполнить, кликнув по кнопке *Вид*. При запуске образуется *результатирующая таблица* (см. рис. 1).

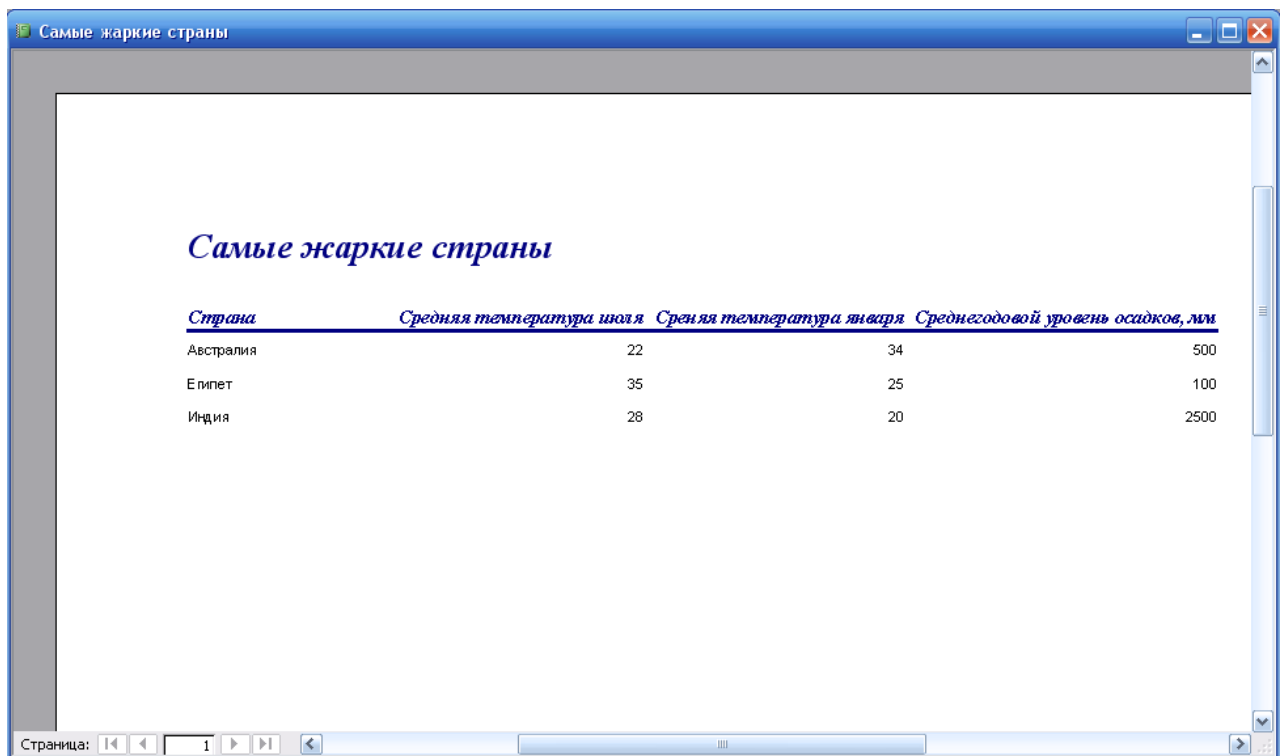


Страна	Средняя температура июля	Средняя температура января	Среднегодовой уровень осадков, мм
▶ Австралия	22	34	500
Египет	35	25	100
Индия	28	20	2500
*	0	0	0

рис. 1

4. В окне **База данных географическая** выбрать вкладку *Отчет* и кликнуть по кнопке *Создать*. В диалоговом окне *Новый отчет* выбрать пункт *Автоотчет: ленточный*, выбрать запрос «Самые жаркие страны», нажать *ОК*. По желанию отредактировать отчет в режиме *Конструктора* .

5. Сохранить отчет (см. рис. 2).



Самые жаркие страны

<u>Страна</u>	<u>Средняя температура июля</u>	<u>Средняя температура января</u>	<u>Среднегодовой уровень осадков, мм</u>
Австралия	22	34	500
Египет	35	25	100
Индия	28	20	2500

Страница: 1

рис. 2

6. **Отчет «Население»** создать таким образом:

- дважды кликнуть по значку *Создание отчета с помощью мастера*;
- выполнить инструкции *Мастера* – выбрать таблицу, поля, **убрать равные группировки** (если они есть), выбрать макет *В столбик*, стиль отчета, задать его имя (см. рис. 3).

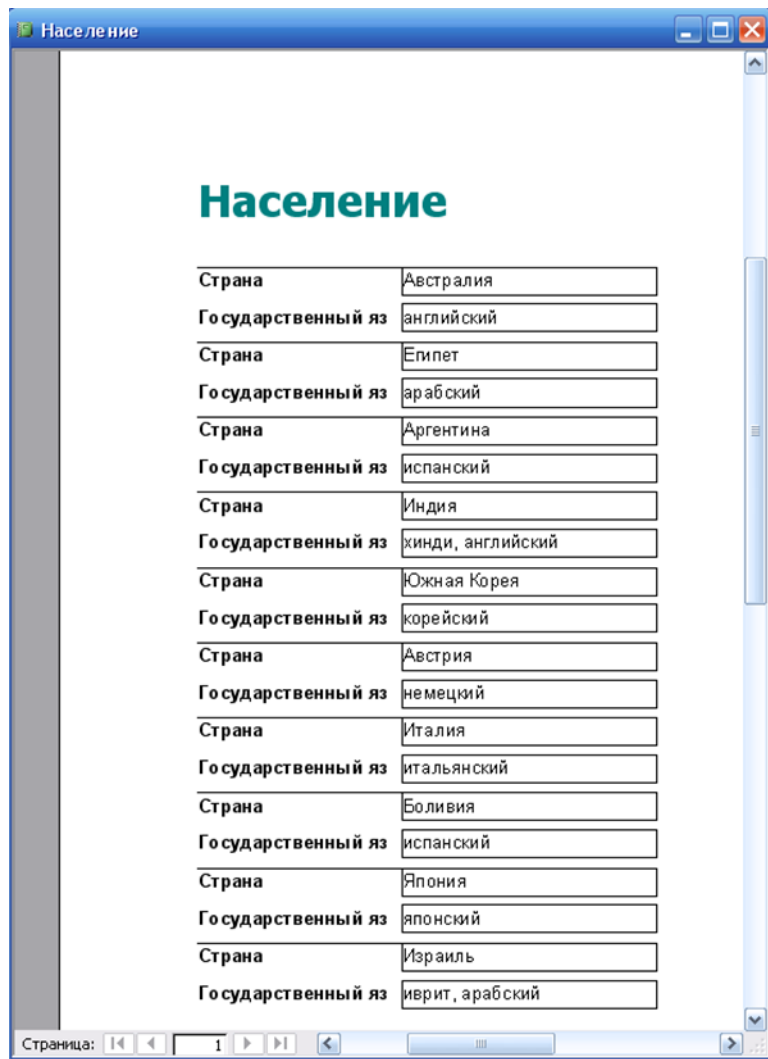


рис. 3

7. Отчет «Республика» также создать с помощью *Мастера отчетов*:

- выбрать отчет «Устройство»;
- задать **равные группировки полей** в следующем порядке: «Государственный строй», «Страна», «Государственный язык», **выделить** поле «Страна»;
- задать макет *Структура 1*;
- выбрать стиль *Полужирный*;

сохранить под именем «Республики» (см. рис. 4)

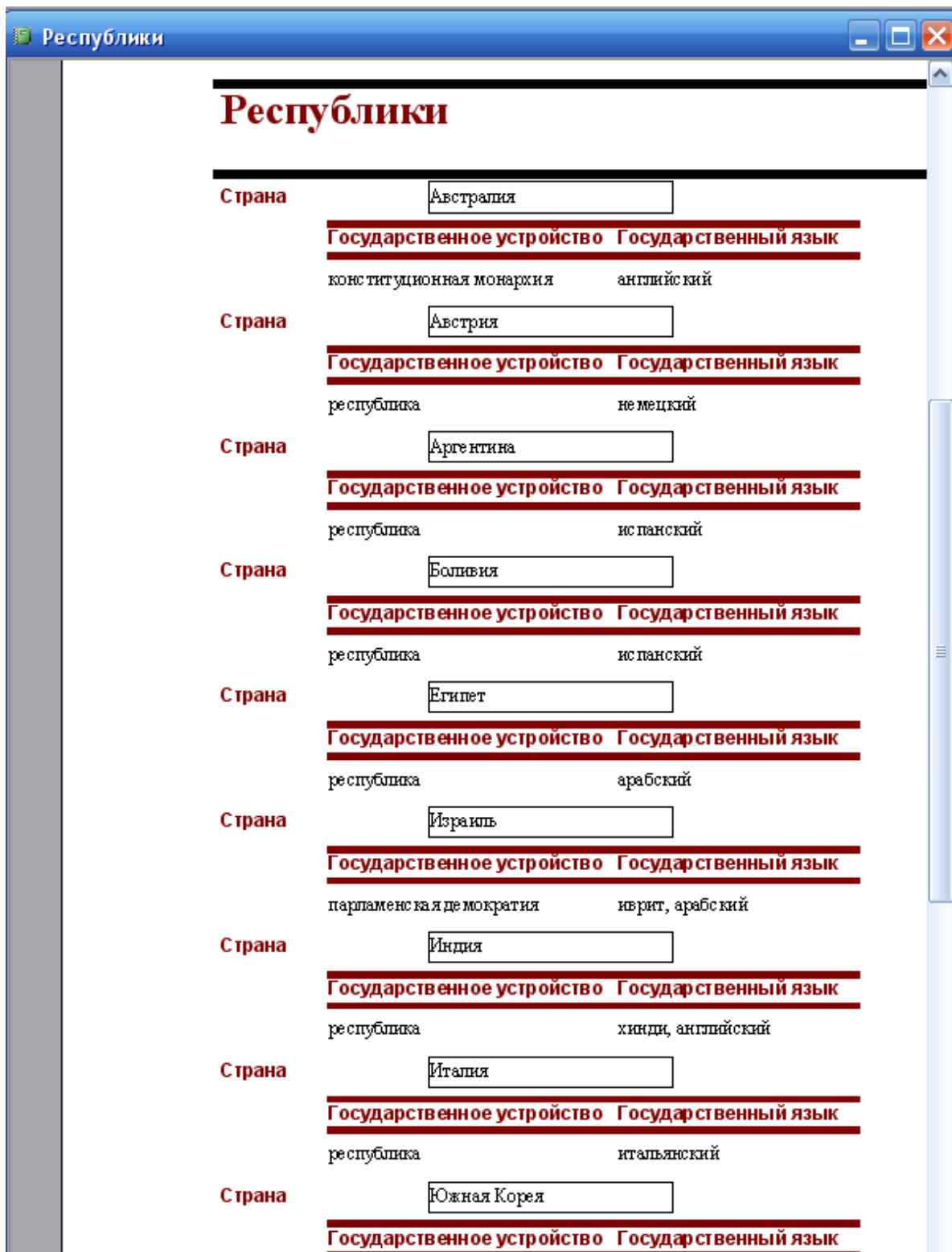


рис. 4

8. Ввести значение параметра – *республика*, нажать *OK*.
9. Отредактировать размеры поля «Государственное устройство» в режиме *Конструктора*.
10. Результат покажите учителю.

Контрольные вопросы

1. База данных – это...
2. Типы баз данных.
3. Опишите табличную базу данных. Пример.
4. Что такое поле базы данных?

5. Что такое запись базы данных?
6. Что такое ключевое поле?
7. Перечислите основные типы данных.
8. Что такое счётчик?
9. Опишите иерархическую базу данных. Пример.
10. Какие объекты называют предками, потомками, близнецами?
11. Опишите сетевую базу данных. Пример.
12. Что такое системы управления базами данных (СУБД)?
13. Какое приложение Microsoft Office является СУБД?
14. Сколько БД может быть открыто одновременно в СУБД Access?
15. Что такое окно БД?
16. Перечислите объекты БД.
17. Опишите объект «таблица».
18. Опишите объект «запрос».
19. Опишите объект «форма».
20. Опишите объект «отчёт».
21. Опишите объект «макрос».
22. Опишите создание БД с помощью конструктора.
23. Опишите создание БД с помощью мастера.
24. Опишите создание формы.
25. Опишите быстрый поиск данных.
26. Опишите поиск данных с помощью фильтров.
27. Опишите создание запроса с помощью конструктора.
28. Опишите создание запроса с помощью мастера.
29. Что такое сортировка записей?
30. Опишите вложенную сортировку с помощью запроса.
31. Каким образом можно осуществлять печать данных?
32. Какой объект целесообразно использовать для красивой печати документов?
33. Опишите создание отчёта с помощью мастера.
34. Какие БД называют реляционными?
35. Какие типы связей между таблицами возможны в реляционных БД?

Практическая работа № 17-19. Разработка веб-сайта

Цель: Научиться создавать Web-сайты с использованием языка разметки текста HTML в простейшем текстовом редакторе Блокнот

Методические указания

Задание. В операционной системе Windows или Linux создать сайт с использованием языка разметки текста HTML в простейшем текстовом редакторе Блокнот. Сайт "Компьютер" должен содержать начальную страницу и страницы "Программы", "Словарь" и "Анкета". Сохранить их в файлах с именами index.htm, software.htm, glossary.htm и anketa.htm в папке сайта.

Начальная страница сайта. Создадим начальную страницу Web-сайта "Компьютер".

1. В операционной системе Windows или Linux запустить простейший текстовый редактор Блокнот.

2. Ввести тэги, определяющие структуру Web-страницы.

Ввести заголовок Web-страницы: "Компьютер".

Ввести заголовок текста, отображаемый в браузере: "Всё о компьютере"

```
<HTML>
<HEAD>
<TITLE>Компьютер</TITLE>
</HEAD>
<BODY>
Всё о компьютере
</BODY>
</HTML>
```

Просмотреть получившуюся Web-страницу в браузере.

На начальной странице сайта обычно размещается текст, кратко описывающий его содержание. Поместим на начальную страницу текст, разбитый на абзацы с различным выравниванием.

3. <P ALIGN="left">На этом сайте вы сможете получить различную информацию о компьютере, его программном обеспечении и ценах на компьютерные комплектующие. </P>

<P ALIGN="right">Терминологический словарь познакомит вас с компьютерными терминами, а также вы сможете заполнить анкету.</P>

Пусть начальная страница сайта "Компьютер" будет содержать центрированный крупный заголовок синего цвета, отделенный горизонтальной линией от двух по-разному выровненных абзацев.

```
4. <FONT COLOR="blue">
  <H1 ALIGN="center">
  Всё о компьютере
  </H1>
</FONT>
<HR>
<P ALIGN="left">На этом сайте...</P>
```

<P ALIGN="right">Терминологический словарь...</P>

На начальной странице сайта "Компьютер" логично разместить изображение компьютера.

5. Для того чтобы рисунок располагался по правому краю текста, тэг вставки изображения должен принять следующий вид:

```
<IMG SRC="computer.gif" ALIGN="right">
```

Рисунок для практики:



Создадим папку сайта "Компьютер" и добавим в сайт пустые страницы "Программы", "Словарь" и "Анкета". Сохраним их в файлах с именами software.htm, glossary.htm и anketa.htm в папке сайта. Такие пустые страницы должны иметь заголовки, но могут пока не иметь содержания.

```
6. <HTML>
  <HEAD>
    <TITLE>Заголовок страницы</TITLE>
  </HEAD>
  <BODY>

  </BODY>
</HTML>
```

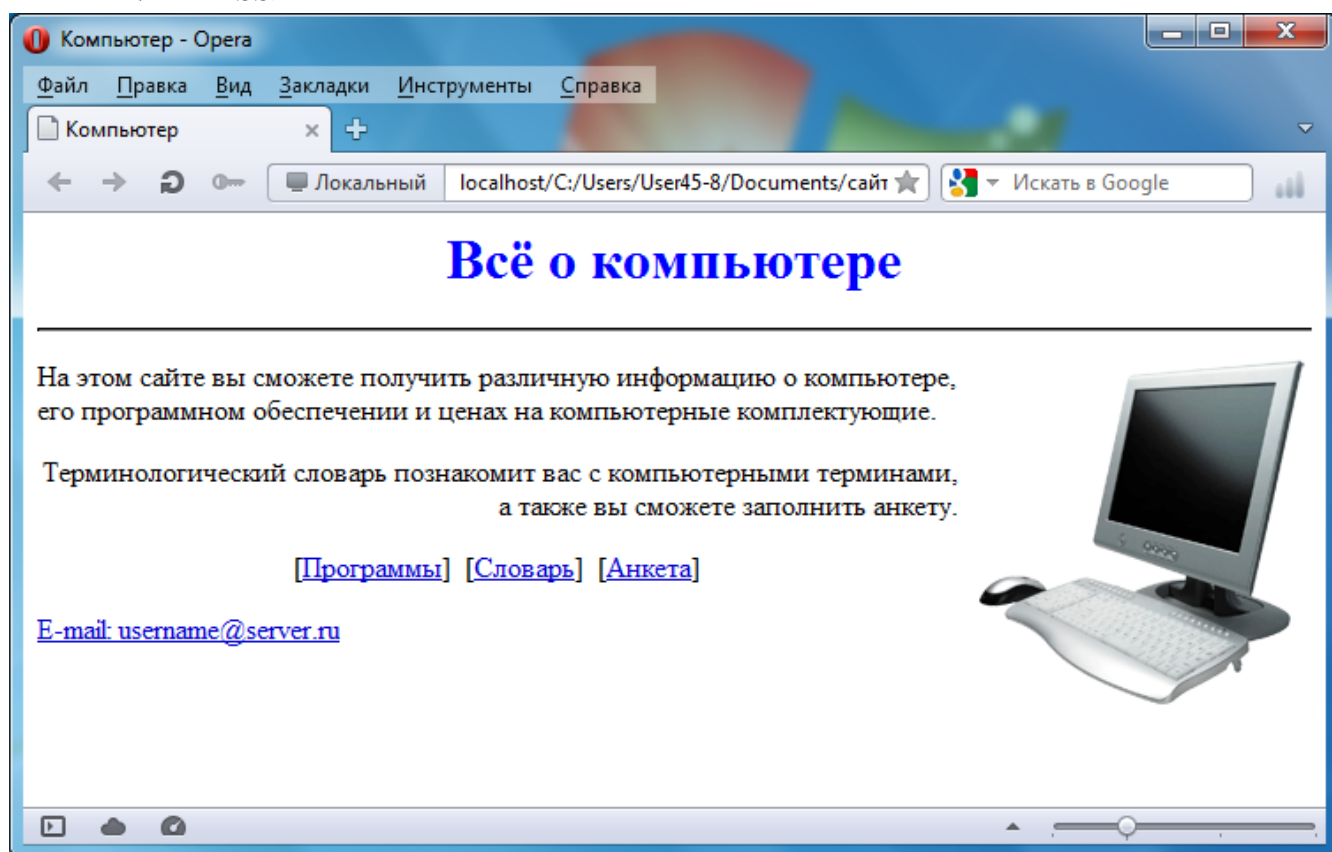
Создадим панель навигации по сайту "Компьютер". На начальной странице сайта разместим указатели гиперссылок на страницы сайта. В качестве указателей гиперссылок удобнее всего выбрать названием страниц, на которые осуществляется переход.

Разместим указатели гиперссылок внизу страницы в новом абзаце в одну строку, разделив их пробелами (). Такое размещение гиперссылок часто называют панелью навигации.

Созданная начальная страница Web-сайта "Компьютер" содержит заголовок, изображение компьютера, два абзаца текста, панель навигации и ссылку на адрес электронной почты.

```
7. <P ALIGN="center">
  [<A HREF="software.htm">Программы</A>] &nbsp; [<A
  HREF="glossary.htm">Словарь</A>] &nbsp; [<A HREF="anketa.htm">Анкета</A>]
  </P>
  <ADDRESS>
```

E-mail: username@server.ru
</ADDRESS>

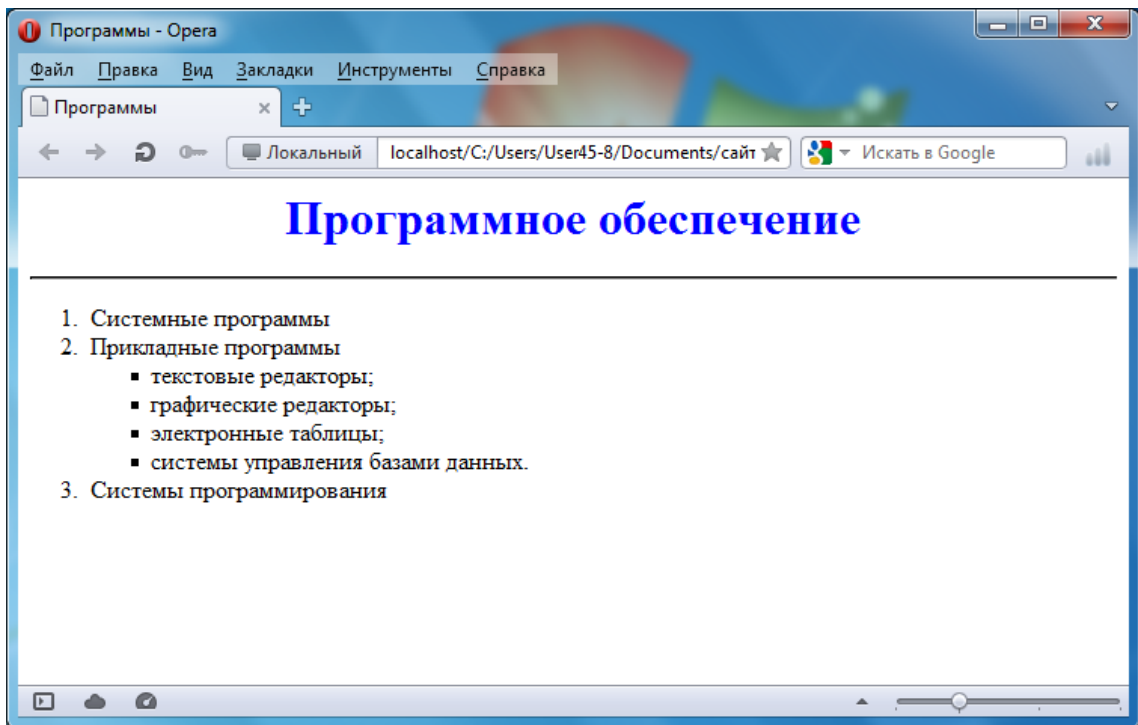


Web-страницы "Программы". Web-страницу "Программы" мы представим в виде нумерованных и маркированных списков.

```
8. <html>
  <head>
    <title>Программы</title>
  </head>

  <body>
    <H1 ALIGN="center">
      <FONT COLOR="blue">
        Программное обеспечение
      </FONT>
    </H1>
    <HR>
    <OL>
      <LI> Системные программы
      <LI> Прикладные программы
      <UL TYPE="square">
        <LI> текстовые редакторы;
        <LI> графические редакторы;
        <LI> электронные таблицы;
        <LI> системы управления базами данных.
      </UL>
      <LI> Системы программирования
```

```
</OL>
</body>
</html>
```



Web-страница "Словарь". Web-страницу "Словарь" мы представим в виде словаря компьютерных терминов.

9. <DL>

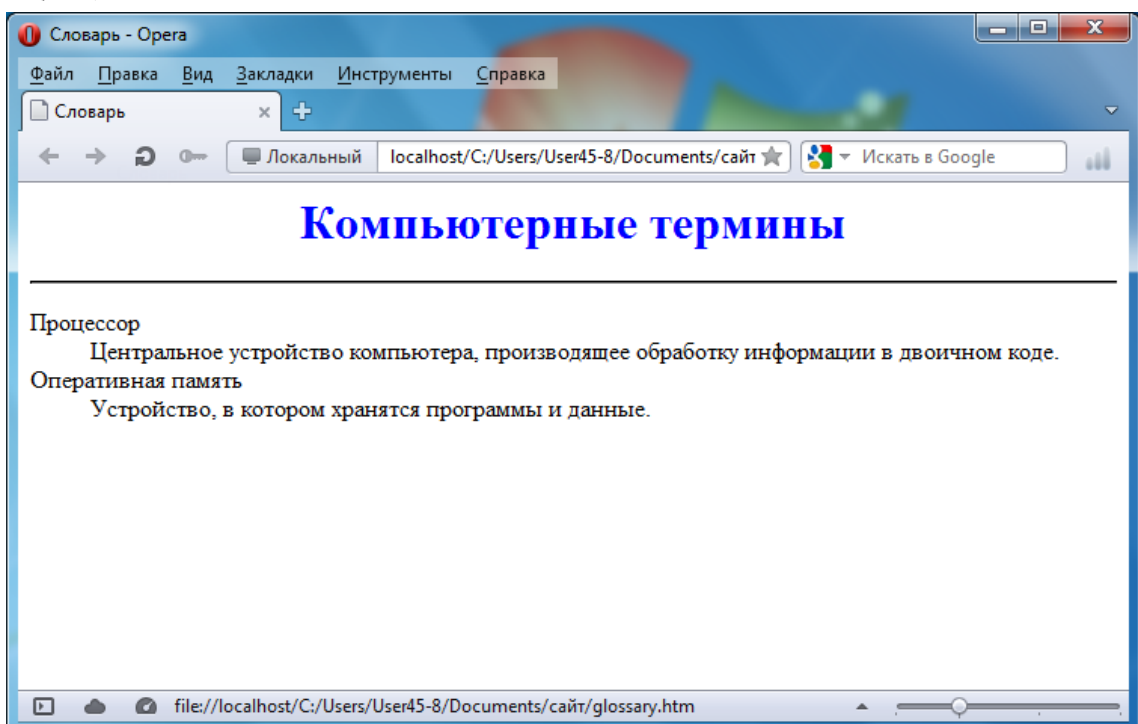
<DT>Процессор

<DD>Центральное устройство компьютера, производящее обработку информации в двоичном коде.

<DT>Оперативная память

<DD>Устройство, в котором хранятся программы и данные.

</DL>



Интерактивная Web-страница "Анкета". Интерактивная Web-страница "Анкета" содержит форму, которая заключается в контейнере <FORM></FORM>. В первую очередь выясним имя посетителя нашего сайта и его электронный адрес, чтобы иметь возможность ответить ему на замечания и поблагодарить за посещение сайта.

10. <FORM>

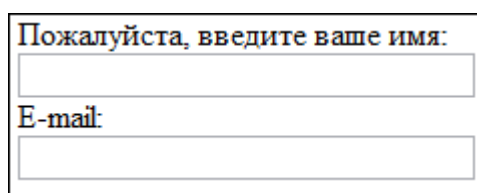
Пожалуйста, введите ваше имя:

<INPUT TYPE="text" NAME="ФИО" SIZE=30>

E-mail:

<INPUT TYPE="text" NAME="e-mail" SIZE=30>

</FORM>



The screenshot shows a rectangular box containing two text input fields. The first field is preceded by the text "Пожалуйста, введите ваше имя:" and the second field is preceded by "E-mail:". Both fields are empty and have a standard rectangular border.

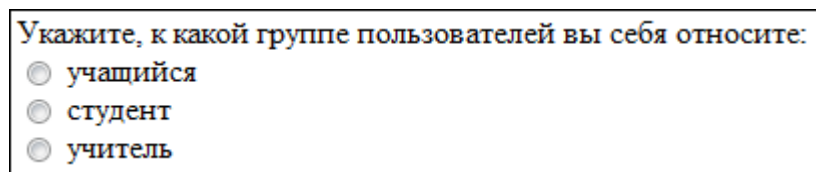
Вставим в HTML-код группу переключателей, в которой устанавливается, к какой группе пользователей относится посетитель.

11. Укажите, к какой группе пользователей вы себя относите:

<INPUT TYPE="radio" NAME="group" VALUE="учащийся"> учащийся

<INPUT TYPE="radio" NAME="group" VALUE="студент"> студент

<INPUT TYPE="radio" NAME="group" VALUE="учитель"> учитель



The screenshot shows a rectangular box containing a heading "Укажите, к какой группе пользователей вы себя относите:" followed by three radio button options: "учащийся", "студент", and "учитель". Each option is preceded by a radio button symbol.

Вставим в HTML-код группу флажков, которые выявляют наиболее популярные сервисы Интернета.

12. Какие из сервисов Интернета вы используете наиболее часто:

<INPUT TYPE="checkbox" NAME="box1" VALUE="WWW"> WWW

<INPUT TYPE="checkbox" NAME="box2" VALUE="e-mail"> e-mail

<INPUT TYPE="checkbox" NAME="box3" VALUE="FTP"> FTP

Какие из сервисов Интернета вы используете наиболее часто:

WWW

e-mail

FTP

Вставим в HTML-код раскрывающийся список, содержащий наиболее популярные браузеры.

13. Какой браузер вы используете наиболее часто:

```
<BR>
<SELECT NAME="Браузер">
<OPTION SELECTED>Internet Explorer
<OPTION SELECTED>Google Chrome
<OPTION SELECTED>Opera
<OPTION SELECTED>Mozilla
</SELECT>
<BR>
```

Какой браузер вы используете наиболее часто:

Google Chrome

Internet Explorer

Google Chrome

Opera

Mozilla

Вставим в HTML-код текстовую область, в которой посетитель сайта может высказать свои замечания и предложения.

14. Какую ещё информацию вы хотели бы видеть на сайте?

```
<BR>
<TEXTAREA NAME="Ваши предложения" ROWS=4 COLS=30>
</TEXTAREA>
<BR>
```

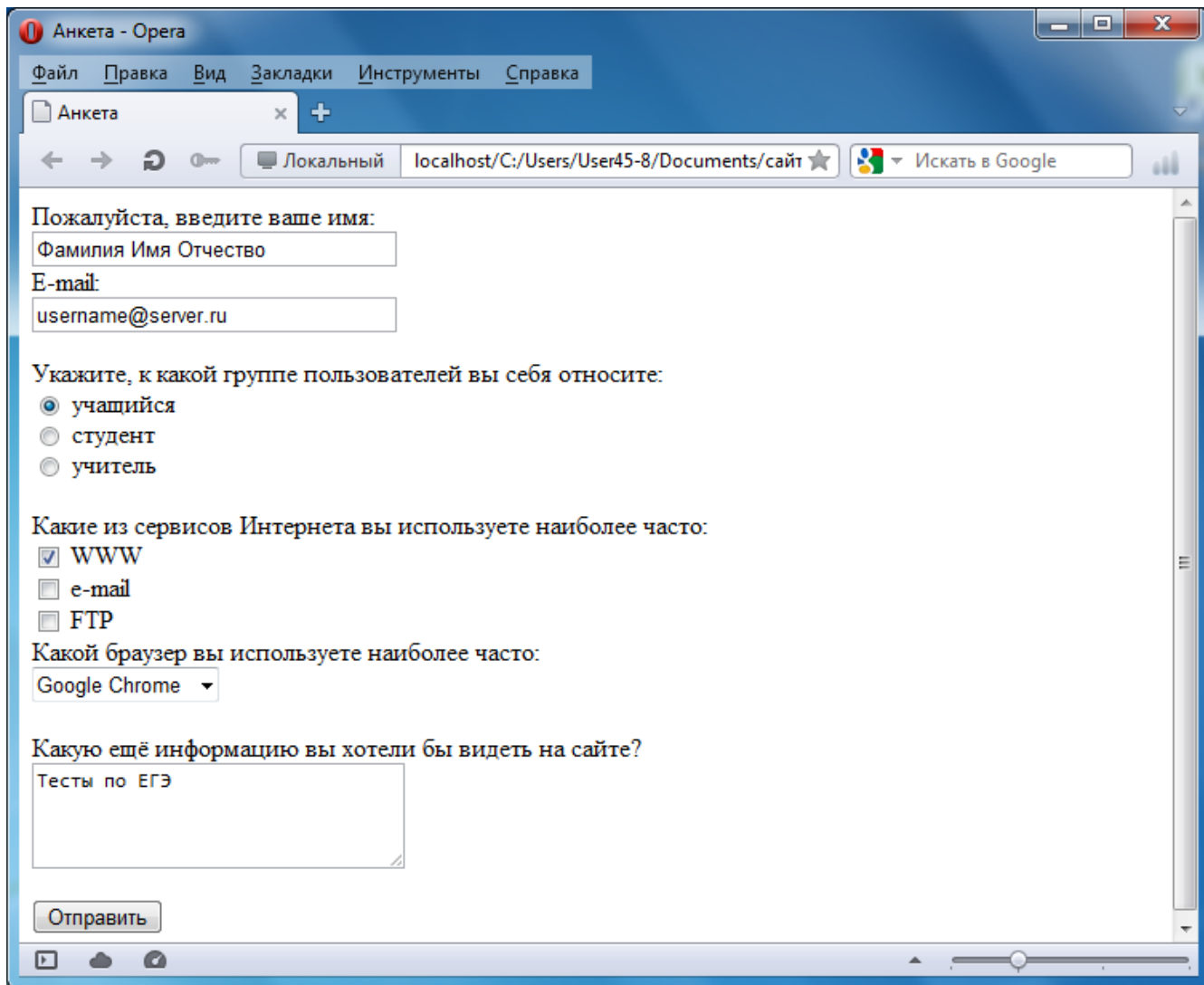
Какую ещё информацию вы хотели бы видеть на сайте?

Чтобы данные из интерактивной формы были отправлены по указанному адресу электронной почты или на сервер, необходимо указать этот адрес и создать кнопку Отправить.

```
15. <FORM ACTION="mailto:ugrinovich@methodist.ru" METHOD="POST"
ENCTYPE="text/plain">
<INPUT TYPE="submit" VALUE="Отправить">
```

Отправить

На данном этапе должна получиться такая страница:



16. После открытия в браузере Web-страницы "Анкета" и внесения данных в поля формы необходимо щёлкнуть на кнопке Отправить. Данные будут отправлены по указанному адресу электронной почты.

Через несколько секунд по электронной почте придёт сообщение, в котором будут указаны имена полей формы и введённые пользователем значения.

Практическая работа № 20. Построение алгоритма на псевдокоде

Цель: Получить практические навыки составления алгоритмов на псевдокоде

Методические указания

Задание: на основе полученного задания составить словесно-формульное описание алгоритма задачи, описание на псевдокоде

Таблица распределения заданий по вариантам

Номер по списку	Задание	Номер по списку	Задание
1.	3, 5	13.	3, 5
2.	4, 6	14.	4, 6
3.	5, 7	15.	5, 7
4.	6, 8	16.	6, 8
5.	7, 9	17.	7, 9
6.	8, 10	18.	8, 10
7.	9, 11	19.	9, 11
8.	10, 12	20.	10, 12
9.	11, 1	21.	11, 1
10.	12, 2	22.	12, 2
11.	1, 3	23.	1, 3
12.	2, 4	24.	2, 4

Базовые конструкции структурного программирования

Существует несколько способов описания алгоритмов:

- словесно-формульное описание алгоритма, т. е. описание алгоритма с помощью слов и формул;
- графическое описание алгоритма, т. е. описание с помощью специальных графических схем алгоритмов – блок-схемы;
- способ, использующий псевдокоды. Псевдокоды – это интерпретация шагов алгоритма на обычном языке, которая описывает действие команды. Псевдокод используется в листингах, чтобы показать общую структуру программы, не применяя реальных операторов языка программирования;
- запись алгоритма на одном из языков программирования (Basic, C++, Pascal, и т. п.).

1. Типы алгоритмов

В зависимости от особенностей своего построения алгоритмы делятся на три основные группы:

- линейные;
- разветвляющиеся;
- циклические.

Разнообразие алгоритмов определяется тем, что любой алгоритм распадается на части, фрагменты и каждый фрагмент представляет собой алгоритм одного из трех указанных. Поэтому важно знать структуру каждого из алгоритмов и принципы их составления.

Линейные алгоритмы

Линейным называется алгоритм, в котором все этапы решения задачи выполняются строго

последовательно.

То есть линейный алгоритм выполняется в естественном порядке его написания и не содержит разветвлений и повторений.

Псевдокод

нач

ввод переменные
действие

вывод

кн

Алгоритмы ветвящейся структуры

Алгоритмом ветвящейся структуры будем называть такой алгоритм, в котором выбирается один из нескольких возможных путей (вариантов) вычислительного процесса.

Каждый подобный путь называется ветвью алгоритма.

Признаком разветвляющегося алгоритма является наличие операций условного перехода, когда происходит проверка истинности некоторого логического выражения (проверяемое условие) и в зависимости от истинности или ложности проверяемого условия для выполнения выбирается та или иная ветвь алгоритма. Алгоритм предполагает выполнение Действия 1, если записанное условие истинно (выполняется), и выполнение Действия 2, если условие ложно (не выполняется).

В частном случае может отсутствовать один из блоков — «Действие 1» или «Действие 2».

Псевдокод

если

условие

то

действие 1

иначе

действие 2

кв

Циклический алгоритм

Реализует повторение некоторых действий. Иными словами циклические алгоритмы включают в себя циклы.

Циклом называется последовательность действий, выполняемых многократно, каждый раз при новых значениях параметров.

Примером циклических алгоритмов может служить алгоритм покраски забора. Действительно, рассмотрим этот алгоритм в словесно-формульном виде:

Шаг I. Подготовить исходные данные (забор, краску, кисть).

Шаг II. Подойти к забору.

Шаг III. Обмакнуть кисть в краску.

Шаг IV. Нанести краску кистью на поверхность забора.

Шаг V. Если забор еще не весь окрашен, то повторить алгоритм начиная с пункта (шаг III).

Существует несколько видов циклических инструкций, с помощью которых можно

организовать циклы:

1. Инструкция «Цикл с параметром» (цикл с заданным количеством повторений).
2. Инструкция «Цикл с предусловием» (цикл — «пока»).

Псевдокод

пока условие, повторять

нц

действие

кц

3. Инструкция «Цикл с постусловием» (цикл — «до»).

Псевдокод

повторять

действие

до условие

Задания

Блок 1

- 1 Алгоритм поиска суммы минимального и максимального из трех введенных чисел.
- 2 Даны два числа. Меньшее заменить их полусуммой, а большее - удвоенным произведением. Составить алгоритм.
- 3 Найти меньшее из двух целых чисел.
- 4 Найти большее из двух целых чисел.
- 5 Каждая бактерия делится на две в течение одной минуты. В начальный момент времени имеется одна бактерия. Составить алгоритм нахождения числа бактерий на 10 минуте.
- 6 Даны a , b , c . Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами равен c .
- 7 Дана сторона квадрата a . Найти его периметр $P = 4 \cdot a$.
- 8 Дана сторона квадрата a . Найти его площадь $S = a^2$.
- 9 Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.
- 10 Дана длина ребра куба a . Найти объем куба $V = a^3$ и площадь его поверхности $S = 6 \cdot a^2$.
- 11 Даны два числа a и b . Найти их среднее арифметическое: $(a + b)/2$.
- 12 Даны переменные A , B , C . Изменить их значения, переместив содержимое A в B , B — в C , C — в A , и вывести новые значения переменных A , B , C .

Блок 2

1. $F(x) = \begin{cases} x^2 - 3x + 9, & \text{если } x \leq 3; \\ \frac{1}{x^3 + 6}, & \text{если } x > 3. \end{cases}$	7. $F(x) = \begin{cases} x^2, & \text{если } 0 \leq x \leq 3; \\ 4, & \text{если } x > 3 \text{ или } x < 0. \end{cases}$
2. $F(x) = \begin{cases} -x^2 + 3x + 9, & \text{если } x \geq 3; \\ \frac{1}{x^3 - 6}, & \text{если } x < 3. \end{cases}$	8. $F(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 2; \\ \frac{1}{x^2 + 4x + 5}, & \text{если } x > 2. \end{cases}$
3. $F(x) = \begin{cases} 9, & \text{если } x \leq -3; \\ \frac{1}{x^2 + 1}, & \text{если } x > -3. \end{cases}$	9. $F(x) = \begin{cases} x^2 - x, & \text{если } 0 \leq x \leq 1; \\ x^2 - \sin \pi x^2, & \text{если } x > 1 \text{ или } x < 0. \end{cases}$

$4. F(x) = \begin{cases} 0, & \text{если } x \leq 1; \\ \frac{1}{x+6}, & \text{если } x > 1. \end{cases}$	$10. F(x) = \begin{cases} -x^2 + x - 9, & \text{если } x \geq 8; \\ \frac{1}{x^4 - 6}, & \text{если } x < 8. \end{cases}$
$5. F(x) = \begin{cases} -3x + 9, & \text{если } x \leq 7; \\ \frac{1}{x-7}, & \text{если } x > 7. \end{cases}$	$11. F(x) = \begin{cases} 4x^2 + 2x - 19, & \text{если } x \geq -3, 5; \\ -\frac{2x}{-4x+1}, & \text{если } x < 3, 5. \end{cases}$
$6. F(x) = \begin{cases} 3x - 9, & \text{если } x \leq 7; \\ \frac{1}{x^2 - 4}, & \text{если } x > 7. \end{cases}$	$12. F(x) = \begin{cases} -x^2 + 3x + 9, & \text{если } x \leq 3; \\ \frac{x}{x^2 + 1}, & \text{если } x > 3. \end{cases}$

Пример 1. Даны два числа а и в. Найти их среднее арифметическое: $(a + b)/2$.

Псевдокод:

алг Среднее арифметическое

цел а, в

вещ. SA

нач ввод а,в

SA=(a+b)/2

вывод SA

кон

Практическая работа № 21-22. Построение блок-схем линейных алгоритмов

Цель: научиться составлять алгоритмы графическим способом (блок-схем)

Методические указания

Этапы решения задачи на ЭВМ. Работа по решению любой задачи с использованием компьютера включает в себя шесть этапов

1. постановка задачи
2. формализация задачи
3. построение алгоритма
4. составление программы на языке программирования
5. отладка и тестирование программы
6. проведение расчетов и анализ полученных результатов

Часто эту последовательность называют технологической цепочкой решения задачи на ЭВМ.

На этапе постановки задачи следует четко определить, что дано и что требуется найти. Важно описать полный набор исходных данных, необходимых для решения задачи. На этапе формализации чаще всего задача переводится на язык математических формул, уравнений и отношений. Если решение задачи требует математического описания какого-то реального объекта, явления или процесса, то ее формализация равносильна получению соответствующей математической модели.

Третий этап – это построение алгоритма. Опытные программисты часто сразу пишут программы на определенном языке, не прибегая к каким-либо специальным средствам описания алгоритмов (блок-схемам, псевдокодам), однако в учебных целях полезно сначала использовать эти средства, а затем переводить полученный алгоритм на язык программирования.

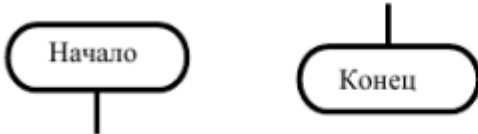
Алгоритм – это последовательность команд управления каким-либо исполнителем. В школьном курсе информатики с понятием алгоритма и методами построения алгоритмов ученики знакомятся на примерах учебных исполнителей: Робота, Черепахи, Чертежника и др. Эти исполнители ничего не вычисляют. Они создают рисунки на экране, перемещаются в лабиринтах, перетаскивают предметы с места на место.

Данные и величины. Совокупность величин, с которыми работает компьютер, принято называть данными. По отношению к программе различают исходные, окончательные (результаты) и промежуточные данные, которые получают в процессе вычислений.

В каждом языке программирования существует своя концепция и своя система типов данных. Однако в любой язык входит минимально необходимый набор основных типов данных: целые вещественные, логические и символьные. С типом величины связаны три ее свойства: множество допустимых значений, множество допустимых операций, форма внутреннего представления.

Блок-схема – графическое представление алгоритма. Она состоит из функциональных блоков, которые выполняют различные назначения (ввод/вывод, начало/конец, вызов функции и т.д.). Существует несколько основных видов блоков, которые нетрудно запомнить:

Виды и значения алгоритмических блоков

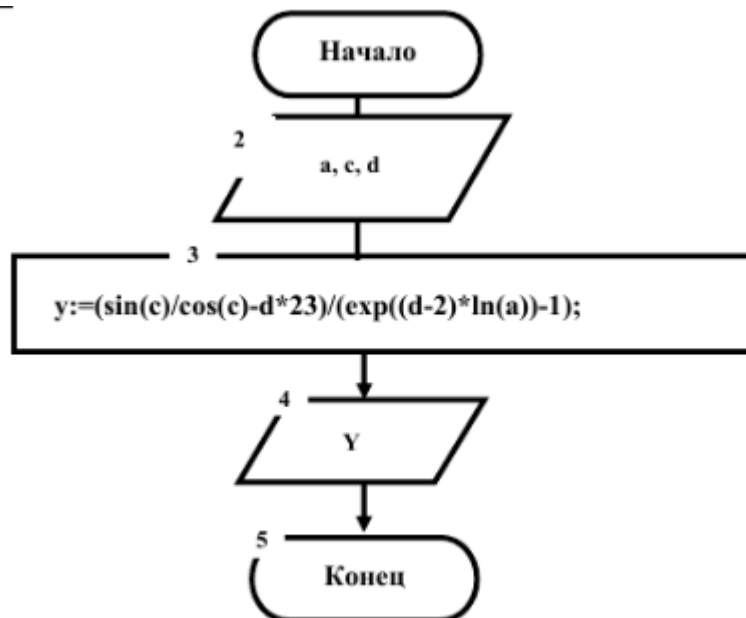
№ п/п	Изображение	Значение
1		Блоки начала и конца программы

2		Блок ввода или вывода информации
3		Блок вычислений
4		Логический блок
5		Блок модификации (начало цикла)
6		Блок подпрограммы
7		Внутристраничные соединительные блоки
8		Межстраничные соединительные блоки
9		Блок комментариев

Линейными называют алгоритмы, в которых операции выполняются последовательно одна за другой, в естественном и единственном порядке следования. В таких алгоритмах все блоки имеют последовательное соединение логической связью передачи информационных потоков.

В них могут использоваться все блоки, за исключением блоков проверки условия и модификации. Линейные алгоритмы, как правило, являются составной частью любого алгоритмического процесса.

Используя общепринятые символы блоков, на рисунке изображена блок-схема линейного алгоритма.



Пример №1: Рассчитать площадь и периметр прямоугольника по двум известным сторонам.

Данная задача не должна представлять особой трудности, так как построена она на хорошо известных всем нам формулах расчета площади и периметра прямоугольника, поэтому заикливаться на выведении этих формул мы не будем.

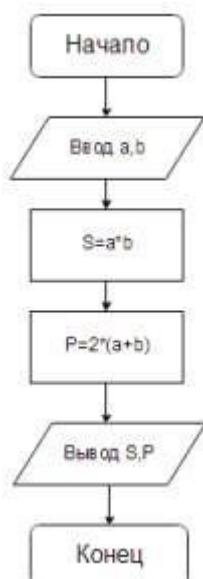
Составим алгоритм решения подобных задач:

1. Прочитать задачу.
2. Выписать известные и неизвестные нам переменные в «дано». (В задаче №1 к известным переменным относятся стороны: a,b ;к неизвестным – площадь S и периметр P)
3. Вспомнить либо составить необходимые формулы. (У нас: $S=a*b$; $P=2*(a+b)$)
4. Составить блок-схему.
5. Записать решение на языке программирования Pascal. Запишем условие в более кратком виде.

Дано: a,b

Найти: S,P

Блок-схема:



Словесное описание алгоритма:

Структура программы, решающей данную задачу, тоже проста:

1. Описание переменных;
2. Ввод значений сторон прямоугольника;
3. Расчет площади прямоугольника;
4. Расчет периметра прямоугольника;
5. Вывод значений площади и периметра;

6. Конец.

ЗАДАНИЕ

Составить словесно-формульный алгоритм и блок-схему для следующих задач:

№	Функция
1	$y = \frac{5x + \sin x^2}{2x + \operatorname{tg} x} + \sin x^2 $
2	$y = \frac{\log_3 x - 10 }{2x - 7} - \cos(3x + 5)$
3	$y = \frac{\cos(x_1 + 5) + x_2}{3x_1 + 6x_2} * \sin x_1^2 $
4	$y = \frac{\cos(3x + 5) + 5^x}{4.35x} + \cos x^2 + \frac{5}{x}$
5	$y = \frac{3x_1 + 2^{x_2}}{\sin(x_1 + x_2)} + \frac{5}{ \sin x_2 } + 6$
6	$y = \frac{\sin^2(x + 3.5) + \lg x}{3x + e^{x+1.35}}$
7	$y = \frac{x_1 + 5.5x_2}{\cos x_1 + \ln x_2} + \operatorname{tg}(x_1 + 4.3)$
8	$y = \frac{3^{x-1.4} + e^x}{4.5 + x} + \operatorname{tg} 3x$
9	$y = \frac{x_1^{x_2} + 3.5 \operatorname{tg} x_1}{3x_1 + 5x_2} + \frac{5x_1}{x_2 + 6} + 5$
10	$y = \frac{\cos^3(x + 2.5) + 1.5x}{x^2 + 3.5} + \frac{3x^2}{8x} + 8.5$

11	$y = \frac{\ln(x+1.3)+5}{1.35x^2+6} + \sqrt{\frac{x+1.3}{(35x)^2+6}}$
12	$y = \frac{2x_1+1.4^{x_2}}{\operatorname{tg}x_1+2x_2} + \sqrt[3]{6x_1+8^{x_2}}$
13	$y = \frac{3x+\operatorname{tg}x}{2.36x+6} + 2x+1.4^x$

№	Функция
14	$t = \frac{2\cos\left(x-\frac{\pi}{6}\right)}{0.5+\sin^2 y} \left(1 + \frac{z^2}{3-z^2/5}\right).$
15	$u = \frac{\sqrt[3]{8+ x-y ^2+1}}{x^2+y^2+2} - e^{ x-y }(\operatorname{tg}^2 z+1)^x.$
16	$v = \frac{1+\sin^2(x+y)}{\left x-\frac{2y}{1+x^2y^2}\right } x^{ y } + \cos^2\left(\operatorname{arctg}\frac{1}{z}\right).$

17	$w = \cos x - \cos y ^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$
18	$\alpha = \ln\left(y^{-\sqrt{ x }}\right) \left(x - \frac{y}{2}\right) + \sin^2 \operatorname{arctg}(z).$
19	$\beta = \sqrt{10}(\sqrt[3]{x} + x^{y+2}) (\arcsin^2 z - x-y).$
20	$\gamma = 5\operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \frac{x+3 x-y +x^2}{ x-y z+x^2}.$
21	$\varphi = \frac{e^{ x-y } x-y ^{x+y}}{\operatorname{arctg}(x)+\operatorname{arctg}(z)} + \sqrt[3]{x^6+\ln^2 y}.$
22	$\psi = \left x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1+(y-x)^2}.$
23	$a = 2^{-x} \sqrt{x+4\sqrt{ y }} \sqrt[3]{e^{x-1/\sin z}}.$

24	$b = y^{\sqrt[3]{ x }} + \cos^3(y) \frac{ x-y \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{ x-y } + \frac{x}{2}}.$
25	$c = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctg}(z) - \frac{\pi}{6} \right)}{ x + \frac{1}{y^2 + 1}}.$
26	$f = \frac{\sqrt[4]{y} + \sqrt[3]{x-1}}{ x-y (\sin^2 z + \operatorname{tg} z)}.$
27	$g = \frac{y^{x+1}}{\sqrt[3]{ y-2 } + 3} + \frac{x + \frac{y}{2}}{2 x+y } (x+1)^{-1/\sin z}.$
28	$h = \frac{x^{y+1} + e^{y-1}}{1+x y-\operatorname{tg} z } (1+ y-x) + \frac{ y-x ^2}{2} - \frac{ y-x ^3}{3}$

Контрольные вопросы

- 1 Что такое алгоритм?
- 2 Что такое линейный алгоритм?
- 3 Какие фигуры используются для составления блок-схем?

Практическая работа № 23-24. Построение блок-схем разветвляющихся алгоритмов

Цель: развить навыки составления структурных схем линейных и разветвляющихся алгоритмов;

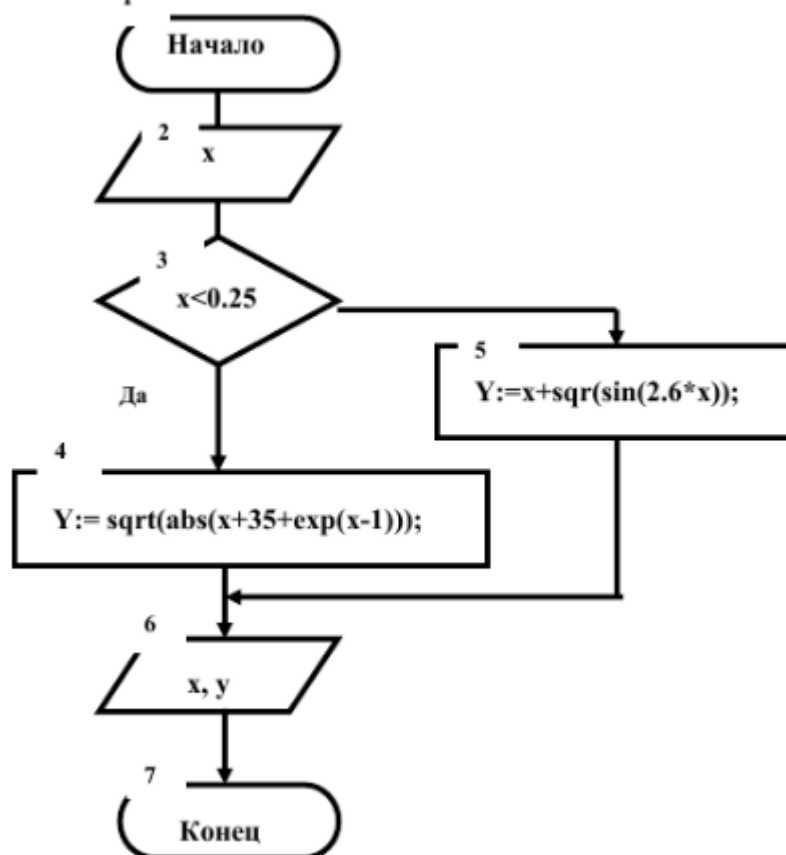
Теоретический материал

Алгоритмы с ветвлением

При составлении схем алгоритмов часто возникает необходимость проведения анализа исходных данных или промежуточных результатов вычислений и определения дальнейшего порядка выполнения вычислительного процесса в зависимости от результатов этого анализа. Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют **разветвляющимися**, или **алгоритмы с ветвлением**. Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений.

Каждое возможное направление вычислений называется **ветвью**. Логическое условие называют **простым**, если разветвляющийся процесс имеет две ветви, и **сложным**, если процесс разветвляется на три и более ветви.

На рисунке приводится пример разветвляющегося алгоритма с простым логическим условием. Для определения ветви, по которой необходимо производить процесс вычисления значения x , достаточно проверить выполнение одного условия. Если условие $x > 3$ не выполняется, то очевидно и без дополнительной проверки, что будет выполнено условие $3 \leq x$.



На рисунке приводится пример сложного алгоритма с ветвлением.

Пример 1. Разработать программу, которая вычисляет значение функции, заданной следующим образом:

$$y = \begin{cases} |x| & \text{при } |x| \leq 1; \\ x^2 & \text{при } 1 < |x| \leq 2; \\ 4 & \text{иначе.} \end{cases}$$

Решение.

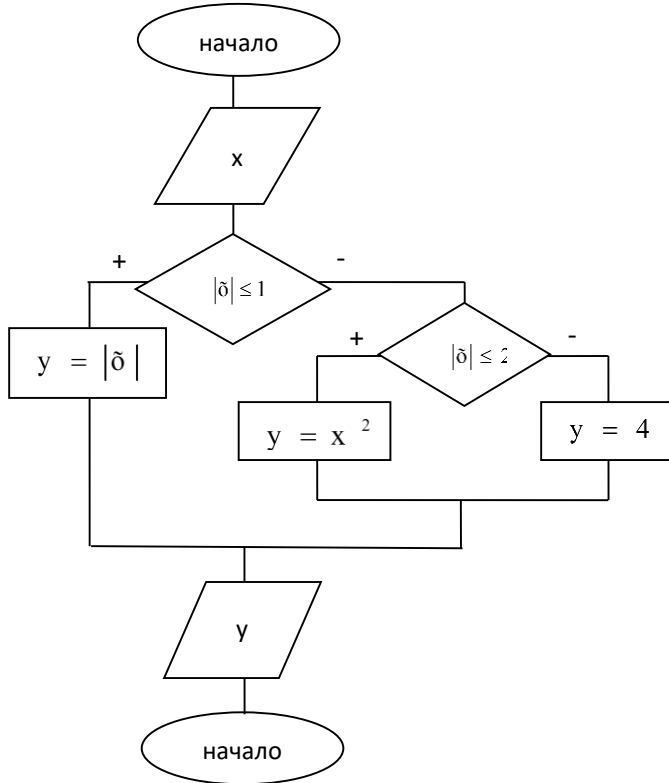
1. Построение математической модели.

Исходные данные: x .

Результат: y .

Вычисления: Если $|\delta| \leq 1$, то $y = |\delta|$, иначе если $|\delta| \leq 2$, то $y = x^2$, иначе $y = 4$.

2. Составление блок – схемы.



Задания практической работы

Задание 1. Рассчитать значение функции $Y(x)$. Значение аргумента вводится с клавиатуры. При вводе значения x , для которого функция не определена, должно выводиться сообщение «Функция не определена».

Каждый выполняет по 2 задачи

1.1	$y = \begin{cases} \sin(x+2), & x < 1,35 \\ x + 3,5 \operatorname{tg} x, & 1,35 \leq x \end{cases}$	1.2	$y = \begin{cases} \sin(x+2), & x < 1,35 \\ x + 3,5 \operatorname{tg} x, & 2 \leq x \leq 4 \\ \sqrt{ 2,56x - 0,35 }, & x > 5 \end{cases}$
2.1	$y = \begin{cases} \operatorname{tg}^2(x+2,5) + e^x, & x < 0,25 \\ 3,5 \sin x + \cos^2 x, & 0,25 \leq x \end{cases}$	2.2	$y = \begin{cases} -\frac{\operatorname{tg}(x ^{2x}) + x^2}{\cos(x+2)}, & x < -5 \\ 12 + 5x, & 0 \leq x < 5 \\ \sqrt{ x^{2-x} - x^2 } * \operatorname{tg} 2x, & x \geq 10 \end{cases}$
3.1	$y = \begin{cases} 3x + 5, & x \leq 1 \\ \sqrt{7x - 6,35}, & 1 \leq x \end{cases}$	3.2	$y = \begin{cases} 5,6(1 + \operatorname{tg} x), & x < 0,5\pi \\ \sin x + 6, & \pi \leq x \leq 3\pi \\ 2,56\sqrt{x^{2x} + 1}, & x > 5\pi \end{cases}$

4.1	$y = \begin{cases} 3^{x+3,5} + \operatorname{tg}(2x), & x < 1,45 \\ \sin^2 x + e^x, & 1,45 \leq x \end{cases}$	4.2	$y = \begin{cases} 3e^{\sin x} * \operatorname{tg} 2x, & x \leq 2 \\ 2,47 \lg x + x^{2x}, & 3 < x < 4 \\ \sqrt{ \cos^2 x + 6} + 4, & x \geq 6 \end{cases}$
5.1	$y = \begin{cases} \cos x + 3, & x < 2,5 \\ x + 1,35, & 2,5 \leq x \end{cases}$	5.2	$y = \begin{cases} 6(\cos^2 x - \sin^2 x), & x < -4 \\ \sqrt{3,2x^2 + \operatorname{tg}^2 x}, & -1 \leq x < 2 \\ e * \cos(2,58x), & x \geq 5 \end{cases}$
6.1	$y = \begin{cases} \log_3(x+2), & x \geq 1 \\ \frac{x}{\sin(x-6)}, & 1 > x \end{cases}$	6.2	$y = \begin{cases} \operatorname{tg}(2x+4,2) - \lg x , & x < 1 \\ \sin x + \sqrt{6x}, & 2 \leq x \leq 5 \\ 6 + \operatorname{arctg}\left(\frac{2x}{1+\sqrt{x}}\right), & x > 7 \end{cases}$
7.1	$y = \begin{cases} \operatorname{tg}(x+3) - \frac{1}{x}, & x < 2,5 \\ x^2 + 6, & 2,5 < x \end{cases}$	7.2	$y = \begin{cases} 2 + x^{2+x}, & x < 0,1\pi \\ \sin^2(x^2 + 0,5), & \pi \leq x \leq 1,5\pi \\ \sin^2(2,45\sqrt{x-1,2}), & x > 2,5\pi \end{cases}$
8.1	$y = \begin{cases} x^3 + 2, & 0 \leq x \leq 2,5 \\ \sqrt[3]{x+3,5} + \sin x^2, & 2,5 < x \end{cases}$	8.2	$y = \begin{cases} \sqrt{1+2,4x^2}, & x < 1 \\ \lg 5,9x, & 2 \leq x \leq 3 \\ \sqrt{x^2+1} + 2x, & x > 5 \end{cases}$
9.1	$y = \begin{cases} x^2 + 3x \sin(x+6), & x \leq 1,3 \\ \sqrt{ x-10,5 + \operatorname{tg}^2 x}, & 1,3 < x \end{cases}$	9.2	$y = \begin{cases} e^{\sin(x)} \operatorname{tg}(2,3+x), & x < -0,1\pi \\ \operatorname{arctg} x, & -1 \leq x \leq 0,5 \\ -1 + \sqrt{23,5x}, & x > 1,5\pi \end{cases}$
10.1	$y = \begin{cases} x^2 + 5,35, & 4 > x \\ \frac{\sqrt{3x-6,6}}{\operatorname{tg} x}, & 4 \leq x \end{cases}$	10.2	$y = \begin{cases} \frac{e}{\operatorname{tg}(2+x)}, & x < 1 \\ \ln((x+2)^2 - x^2), & 2 \leq x \leq 4 \\ \sin^2(x-6), & x > 5 \end{cases}$
11.1	$y = \begin{cases} 2^{x+3} + \sin x, & x > 2 \\ \sin^2(x-6), & x \leq 2 \end{cases}$	11.2	$y = \begin{cases} \ln(5,3x^{3x} - x^2), & x < 1 \\ \frac{0,025}{\operatorname{tg}(2,6+x)}, & 2 \leq x \leq 4 \\ \sin^2(x-6), & x > 5 \end{cases}$

12.1	$y = \begin{cases} \sin(x+1), & x < 3,5 \\ 2,5x+3, & 3,5 \leq x \end{cases}$	12.2	$y = \begin{cases} \frac{e}{\operatorname{tg}(2,9+3x)}, & x < 1 \\ 0,6 \ln(5-x^2), & 2 \leq x \leq 4 \\ \sin 3x + \operatorname{lg}(x+0,3), & x > 6 \end{cases}$
13.1	$y = \begin{cases} x^2+8, & 6 > x \\ \frac{\sqrt{3x-3,6}}{\operatorname{tg}x + \operatorname{lg}(x+2)}, & 6 \leq x \end{cases}$	13.2	$y = \begin{cases} \frac{2,3x+3,56}{\operatorname{tg}(x+1)}, & x < 2 \\ \sin(5,4x^2 - x^{2-x}), & 3 \leq x \leq 4 \\ \operatorname{tg}(x+3) - \frac{1}{x}, & x > 8 \end{cases}$
14.1	$y = \begin{cases} 24x^{2-x}, & 3 > x \\ \frac{\ln x}{\operatorname{tg}x} x^2 + 5,35, & 3 \leq x \end{cases}$	14.2	$y = \begin{cases} 6,25+7x, & x < 2 \\ 5 \ln(2-x^2), & 6 \leq x \leq 8 \\ \frac{25x}{\operatorname{tg}x}, & x > 10 \end{cases}$
15.1	$y = \begin{cases} x^2+6 \sin(x+6), & x \leq 4 \\ \sqrt{ x-10,5 + \operatorname{tg}^2 x}, & 4 < x \end{cases}$	15.2	$y = \begin{cases} \frac{2,56x+2}{\operatorname{tg}(2+x)}, & x < 2 \\ \ln(x^2-1), & 2 \leq x \leq 4 \\ \operatorname{tg}(2,78x+2), & x > 10 \end{cases}$

Задание 2. Составить блок-схему для решения следующей задачи

1. Вводится число от 1 до 4, определяющее пору года. Дать название этой поры года (1 — зима, 2 — весна, 3 — лето, 4 — осень).
2. Вводится число от 1 до 7, определяющее день недели. Дать название этого дня (1 — понедельник, 2 — вторник, ..., 7 — воскресенье).
3. Вводятся числа 12, 1, 2, определяющие зимний месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, 12 — декабрь).
4. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два, ..., 10 — десять).
5. Дано натуральное число A ($A < 20$), определяющее сумму денег в рублях. Дать для этого числа наименование: "рубль", "рубля", "рублей".
6. Дано натуральное число N ($N < 100$), определяющее возраст человека в годах. Дать для этого числа наименование: "год", "года", "лет".
7. Вводится число от 1 до 12, определяющее месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, ..., 12 — декабрь).
8. Вводится число от 1 до 20. Дать название этого числа (1 — один, 2 — два, ..., 20 — двадцать).
9. Вводятся числа 3, 4, 5, определяющие весенний месяц года. Дать название этого месяца года (3 — март, 4 — апрель, 5 — май).
10. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два, ..., 10 — десять).
11. Вводится число от 2 до 10. Вывести сообщение: четное или нечетное введенное число.
12. Вводится число от -10 до 10. Вывести сообщение: введенное число больше 0, меньше 0 или равно 0.

13. Написать программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота», «Воскресенье».
14. Вводится номер месяца (1, 2, ..., 12). Вывести количество дней в указанном месяце.
15. Вводится значение года в укороченной форме (от 0 по 10). Вывести значение года текущего столетия в полном формате (0 – 2000, 1 – 2001 и т.д.).

Контрольные вопросы

- 1 Что такое алгоритм с ветвлением?
- 2 Что такое полная и сокращенная записи условного оператора?
- 3 Что используется в качестве условий в операторе ветвления?
- 4 Какие знаки отношений можно использовать при составлении условий?
- 5 Что такое составное условие?
6. Для чего предназначен оператор выбора?
7. Что такое полная и сокращенная записи оператора выбора?

Практическая работа № 25-26. Построение блок-схем циклических алгоритмов

Цель: развить навыки составления структурных схем циклических алгоритмов;

Теоретический материал

Циклический алгоритм

Алгоритм **циклической** структуры предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины.

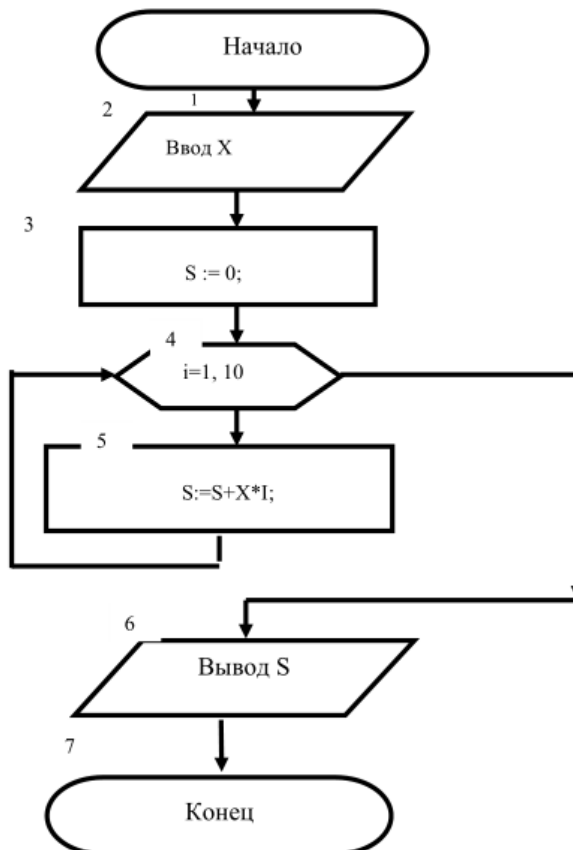
Циклические алгоритмы позволяют существенно сократить объем программы за счет многократного выполнения группы повторяющихся вычислений, так называемого **тела цикла**.

Специально изменяемый по заданному закону параметр, входящий в тело цикла, называется **переменной цикла**. Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания. В качестве переменной цикла используют любые переменные, индексы массивов, аргументы вычисляемых функций и тому подобные величины цикла.

Циклы, в теле которых нет разветвлений и других встроенных в них циклов, называют **простыми**. В противном случае их относят к **сложным**. Циклические алгоритмы разделяют на **детерминированные** и **итерационные**.

Циклы, в которых число повторений заранее известно из исходных данных или определено в ходе решения задачи, называют **детерминированными**. Для организации детерминированных циклов наиболее целесообразно использовать блок модификации, внутри которого указывается переменная цикла, ее начальное и конечное значения. Организовать подобный цикл возможно и при использовании блока проверки условия вместо блока модификации, однако при этом несколько усложняется алгоритм и теряется его рациональность.

На рисунке пример циклического алгоритма с использованием блока модификации. Операция нахождения суммы, при предварительном обнулении значения переменной S (блок 5), повторяется 10 раз в теле цикла.

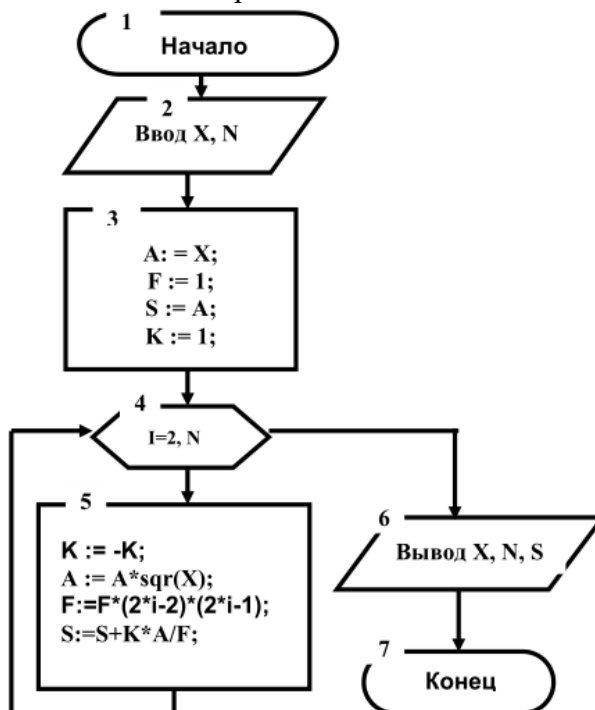


Использована операция присваивания $S := S + X * I$, по которой и осуществляется вычисление суммы путем прибавления к предыдущему значению переменной S всё новых значений элементов ряда.

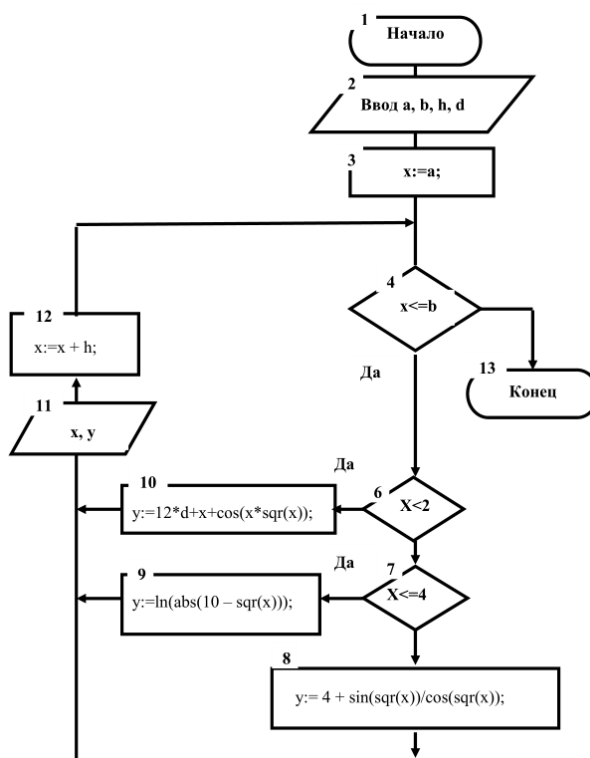
Цикл является детерминированным, и количество его повторений заранее определено (10 раз). В качестве переменной цикла i принято текущее значение членов натурального ряда.

На рис. 8 блок-схема алгоритма для расчета суммы бесконечного ряда. Вычисление суммы прекратить, как только значение очередного элемента ряда станет меньше или равно ϵ ($\epsilon = 0,00001$), и значениями остальных элементов ряда можно пренебречь.

На рисунке для примера организован цикл в виде итерационного, т.к. число повторений заранее неизвестно. В алгоритме выход из цикла или его продолжение определяется выполнением условия $\text{delta} \leq \epsilon$ в блоке 5. Если условие не выполняется, то вычисление суммы продолжается путем прибавления к предыдущему значению суммы (переменная S) значения очередного члена ряда, отслеживаемого переменной цикла i .



На рисунке разработана блок-схема табулирования функции, заданной на отрезке $[a, b]$, где h шаг приращения аргумента x , значение константы d вводится с клавиатуры.



Вводятся исходные данные (**a** и **b** – границы интервала табулирования функции **y**, **h** – приращение аргумента, **d** – постоянная величина), затем задается начальное значение аргумента функции (**x = a**) и вычисляется значение функции **y**, которая задана системой равенств. Значения аргумента **x** и функции **y** выводятся на экран. Для расчета следующего значения функции аргумент **x** получает приращение **h** (блок 12).

После выполнения первой итерации управление передается на начало цикла, где проверяется условие повторения цикла. Если оно выполняется, то расчет функции повторяется, в противном случае происходит выход из цикла.

Задания практической части

Задание 1. Вычислить значение суммы или произведения членов ряда. Значение **x** задать с клавиатуры.

1	$P = \prod_{k=2}^{10} \frac{x + \ln x}{2xk}$	9	$S = \sum_{n=3}^8 \frac{x * \sin nx}{2n + 1}$
2	$S = \sum_{n=1}^8 \frac{x * n}{2n}$	10	$P = \prod_{n=1}^4 \frac{n + 2 + \lg nx}{2n}$
3	$P = \prod_{n=1}^4 \frac{n + \cos nx}{e^n}$	11	$P = \prod_{n=3}^8 \frac{n + \cos(nx + 5)}{2n}$
4	$S = \sum_{n=3}^8 \frac{x * \sin(8x)}{2n + 1}$	12	$P = \prod_{n=1}^4 \frac{2n + 1 + \lg nx}{2n}$
5	$S = \sum_{n=2}^6 \frac{x * \sin nx}{6 + n}$	13	$P = \prod_{n=1}^6 \frac{n + 3 + \cos 2x}{2n - 1}$
6	$P = \prod_{n=3}^8 \frac{2n + 2 + \lg nx}{n + 3 + x}$	14	$P = \prod_{n=3}^9 \frac{n + 2 + nx}{2n + 2}$
7	$S = \sum_{n=3}^8 \frac{x * \sin(x + 5)}{2n + 3}$	15	$P = \prod_{n=1}^4 \frac{2n + 3 + \lg nx}{2n + 4}$
8	$S = \sum_{n=1}^8 \frac{x^2 * n}{2 + n}$	16	$P = \prod_{n=3}^7 \frac{n + 3 + \lg nx}{2 + n}$

Задание 2. Вычислить значение суммы или произведения членов ряда. Значение **x** задать с клавиатуры.

1	$P = \prod_{k=2}^{10} (-1)^{k+1} \frac{x^{2k-1} + \ln x}{2xk * (k + 2)!}$	9	$S = \sum_{n=3}^8 (-1)^n * \frac{x^{2n} * \sin nx}{(2n + 1)!}$
2	$S = \sum_{n=1}^8 (-1)^{n+1} \frac{x^{n+2} * (n + 4)!}{2n}$	10	$P = \prod_{n=1}^4 \frac{n + x^{n+3} + \lg nx}{(2n)!}$

3	$P = \prod_{n=1}^4 (-1)^{n+1} \frac{(n! + \cos nx)}{e^n}$	11	$P = \prod_{n=3}^8 \frac{(n+2)! + nx}{(2n)!}$
4	$S = \sum_{n=3}^8 (-1)^n * \frac{x^{2n} * \sin nx}{(2n+1)!}$	12	$P = \prod_{n=1}^4 \frac{(2n+1)! + \lg nx}{(2n)!}$
5	$S = \sum_{n=2}^6 (-1)^{n+1} \frac{x^{2n} * \sin nx^{n+2}}{(6+n)!}$	13	$P = \prod_{n=1}^6 \frac{(n+3)! + 8x^{2+n}}{(2n-1)!}$
6	$P = \prod_{n=3}^8 \frac{(2n+2)! + \lg nx}{x^{n+4}}$	14	$P = \prod_{n=2}^9 \frac{(n+3)! + nx^{n+1}}{(2n+1)}$
7	$S = \sum_{n=3}^8 (-1)^n * \frac{x^{2n} * \sin nx}{(2n+1)!}$	15	$P = \prod_{n=3}^9 \frac{(n+1)! + 5x^{n+4}}{(n+4)}$
8	$S = \sum_{n=3}^8 (-1)^n * \frac{x^{2n} * \sin nx}{(2n+1)!}$	16	$P = \prod_{n=1}^6 \frac{8x+6}{(3n+1)!}$

Контрольные вопросы

1. Для чего предназначен оператор цикла?
2. Какие существуют виды циклов?
3. В каких случаях применяется цикл ПОКА?
4. Что является телом цикла?
5. Как в теле цикла выполнить несколько операторов?
6. Почему перед выполнением цикла некоторым переменным нужно задавать начальные значения?
7. При каких условиях цикл ПОКА не выполнится ни разу?

Практическая работа № 27. Контрольное решение задач

1. Продолжите предложение:

Алгоритм - ...

Блок-схема - ...

Линейный алгоритм - ...

Команда присваивания - ...

2. Определите значение переменной C после выполнения следующего фрагмента программы.

1 вариант	2 вариант	3 вариант	4 вариант
$a := 5;$ $a := a + 6;$ $b := -a;$ $c := a - 2*b;$	$a := 1;$ $b := 2;$ $a := a + b;$ $b := a - b;$ $c := a - b;$	$a := 7;$ $a := a - 4;$ $b := -a;$ $c := -a + 2*b;$	$a := 4;$ $b := 2;$ $a := a + b;$ $b := a * b;$ $c := 2 * a - b;$
$a := 35;$ $b := (a \text{ div } 10) + 14;$ $a := (b \text{ mod } 10) + 2;$ $c := a + b;$	$a := 29;$ $b := (a \text{ div } 10) + 17;$ $a := (b \text{ mod } 10) + 5;$ $c := a - b;$	$a := 37;$ $b := a \text{ mod } 10;$ $a := a \text{ div } 10;$ $c := 3 * a + b;$	$a := 24;$ $b := (a \text{ mod } 10) * 10;$ $a := a \text{ div } 10 + b;$ $c := 2 * a - b;$

3. Дан алгоритм. Выполните его. Какое значение будет иметь переменная у?

1 вариант	2 вариант	3 вариант	4 вариант
начало	начало	начало	начало
↓	↓	↓	↓
ВВОД a,b,c	ВВОД a,b,c	ВВОД a,b,c	ВВОД a,b,c
↓	↓	↓	↓
$s := \frac{\sqrt{b-1}}{a+c}$	$s := \frac{\sqrt{b}}{a-c}$	$s := b + a + c$	$s := b - ac $
↓	↓	↓	↓
$k := s - 3ac $	$k := 2s^2 - 3ac$	$k := s + ac$	$k := s^2 + a$
↓	↓	↓	↓
$y := 3s + 2k$	$y := 6s + 2k$	$y := sb + k$	$y := s + kc$
↓	↓	↓	↓
ВЫВОД y	ВЫВОД y	ВЫВОД y	ВЫВОД y
↓	↓	↓	↓
конец	конец	конец	конец
$a=1$ $b=65$ $c=3$	$a=3$ $b=16$ $c=1$	$a=3$ $b=2$ $c=1$	$a=2$ $b=5$ $c=4$

4. Составьте блок-схему:

1 вариант	2 вариант	3 вариант	4 вариант
Найдите площадь (S) и периметр (P) прямоугольника по двум его сторонам (A, B).	Зная длины трех сторон (A, B, C). треугольника, вычислить площадь (S) и периметр (P) треугольника.	Заданы длины двух катетов (A, B) в прямоугольном треугольнике. Найти длину гипотенузы (C) и площадь треугольника (S).	Найти длину окружности (L) и площадь круга (S) заданного радиуса (R). В качестве значения π использовать 3.14.
Дана величина A, выражающая объем информации в битах. Перевести A в более крупные единицы измерения информации (байты - B, Кбайты - K, Мбайты - M, Гбайты - G). Составьте блок-схему алгоритма решения поставленной задачи.	Информационный объем текста книги, набранной на компьютере с использованием кодировки Unicode, — A Мбайт. Определить количество символов (K) в тексте книги. Составьте блок-схему алгоритма решения поставленной задачи.	Дана величина A, выражающая объем информации в Гбайтах. Перевести A в более мелкие единицы измерения информации (Мбайты - M, Кбайты - K, байты - B, биты - BI). Составьте блок-схему алгоритма решения поставленной задачи.	В книге A страниц. На каждой странице B строк по D символов в строке. Вычислить информационный объем (V) книги. Перевести V в более крупные единицы измерения информации (байты - B, Кбайты - K). Составьте блок-схему алгоритма решения поставленной задачи.

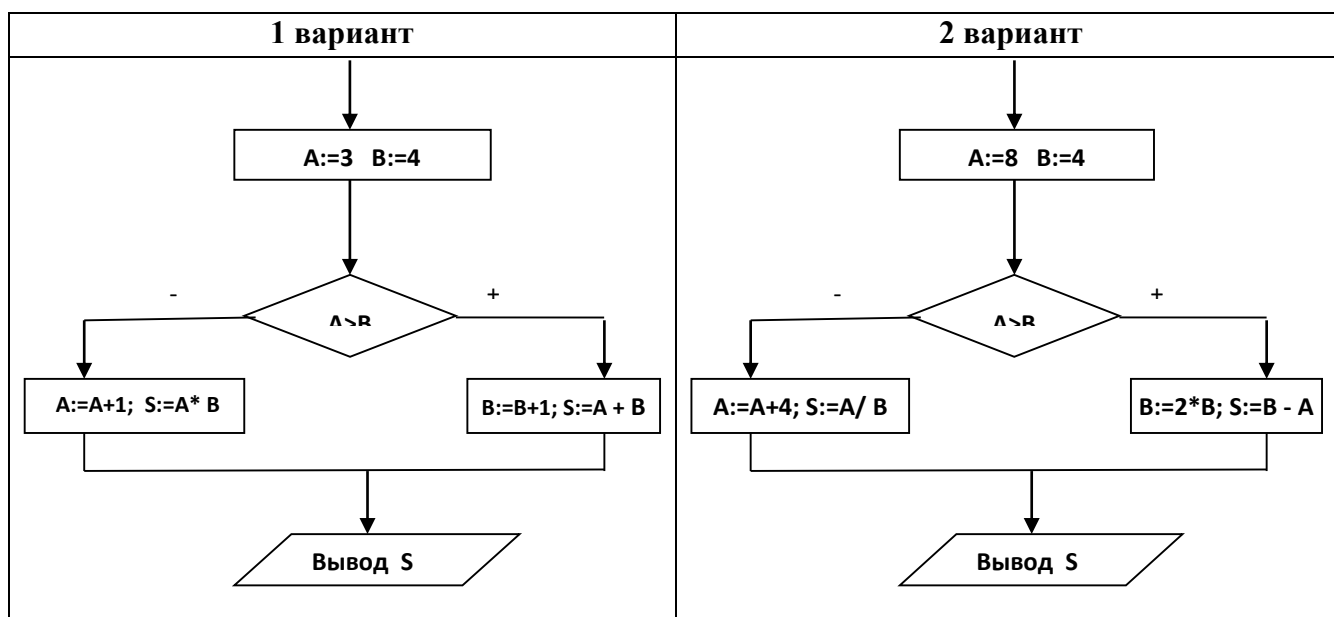
5. Продолжите предложение*:

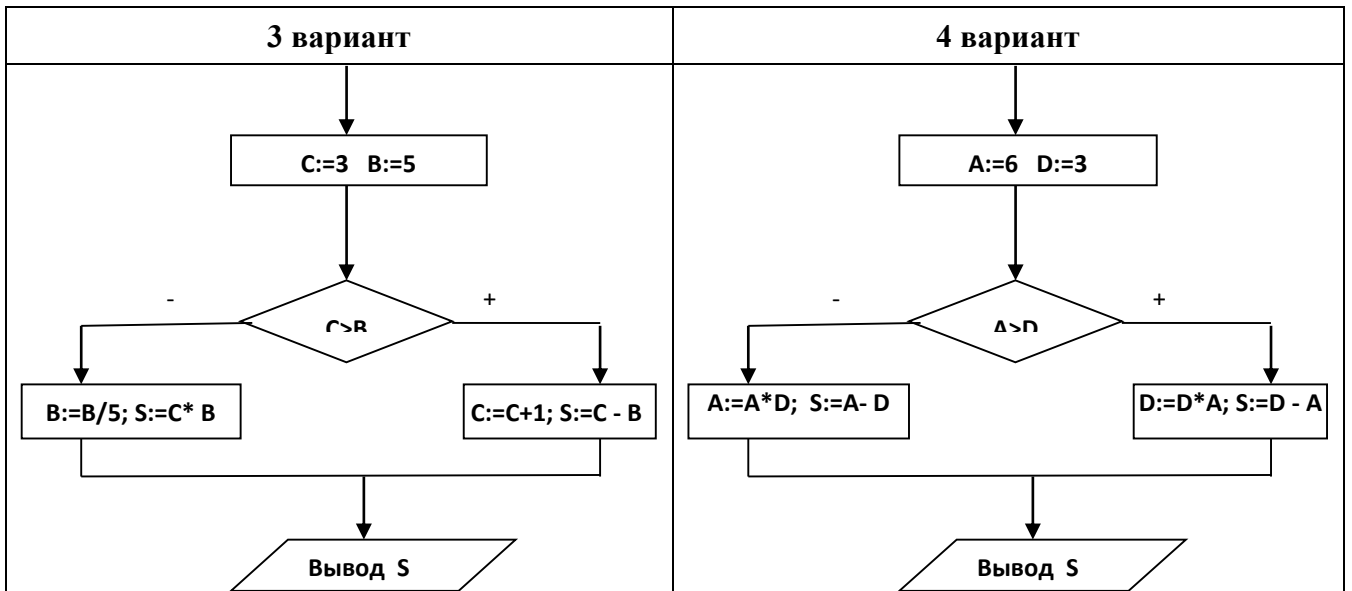
Разветвляющийся алгоритм - ...

Массовость алгоритма - ...

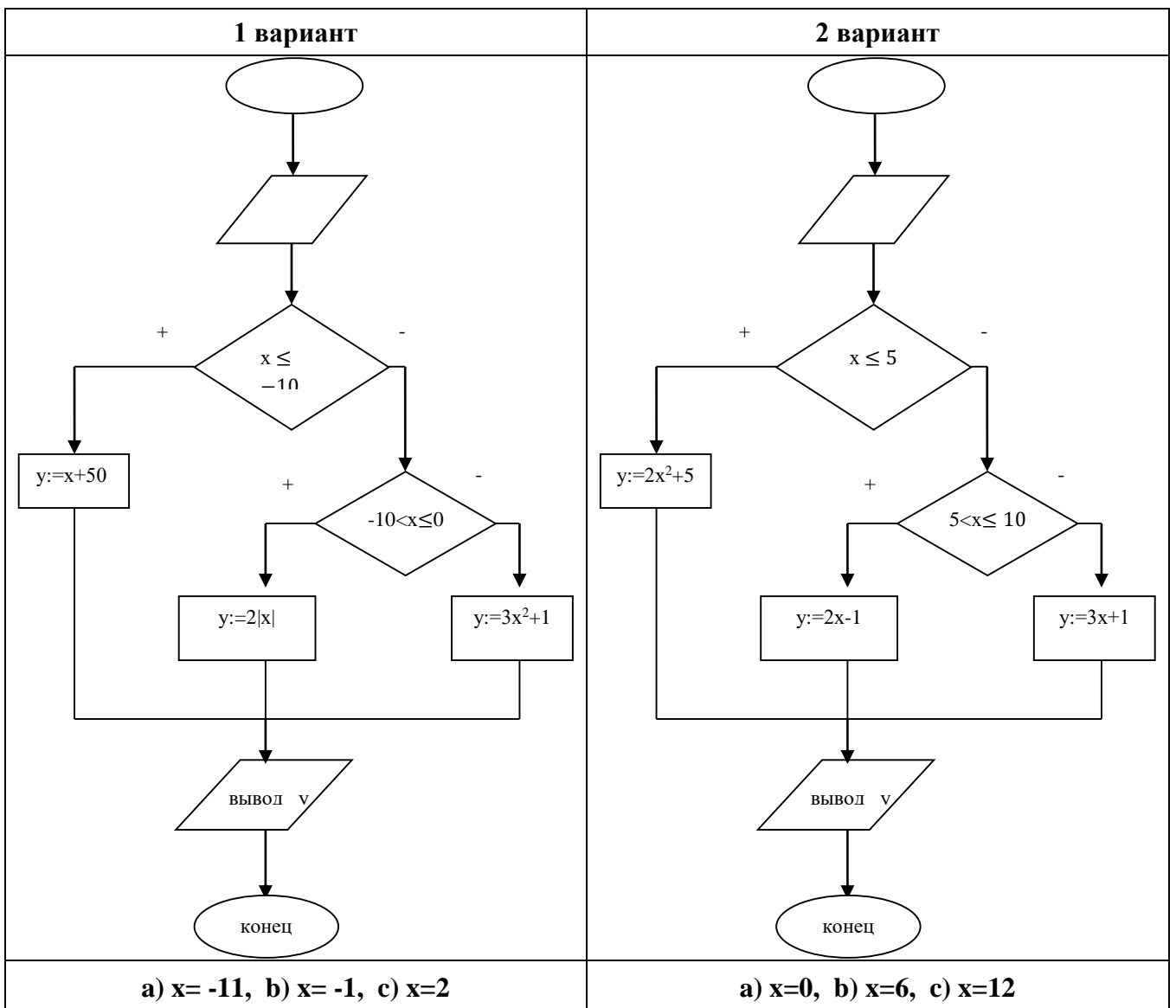
Результативность алгоритма - ...

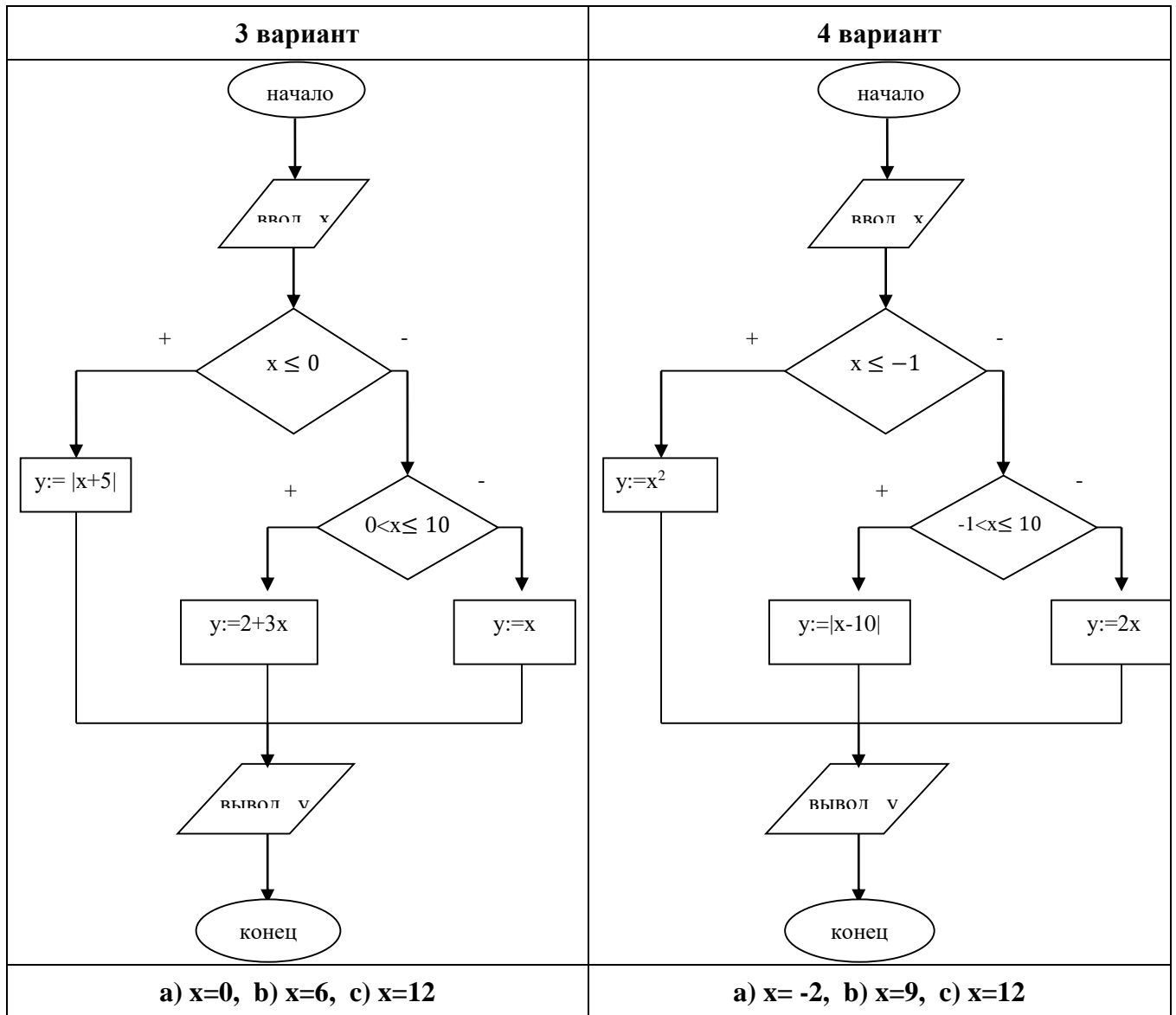
6. Определите, какое значение переменной S будет напечатано в результате выполнения алгоритма*.





7. Определите, какое значение переменной Y будет напечатано в результате выполнения алгоритма. Рассмотрите все три случая: a , b , c .





8. Составить блок-схему

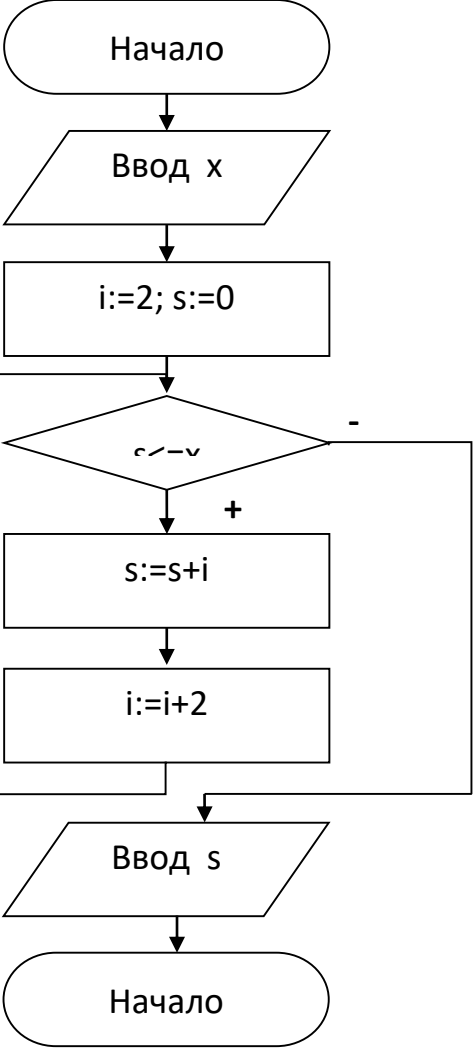
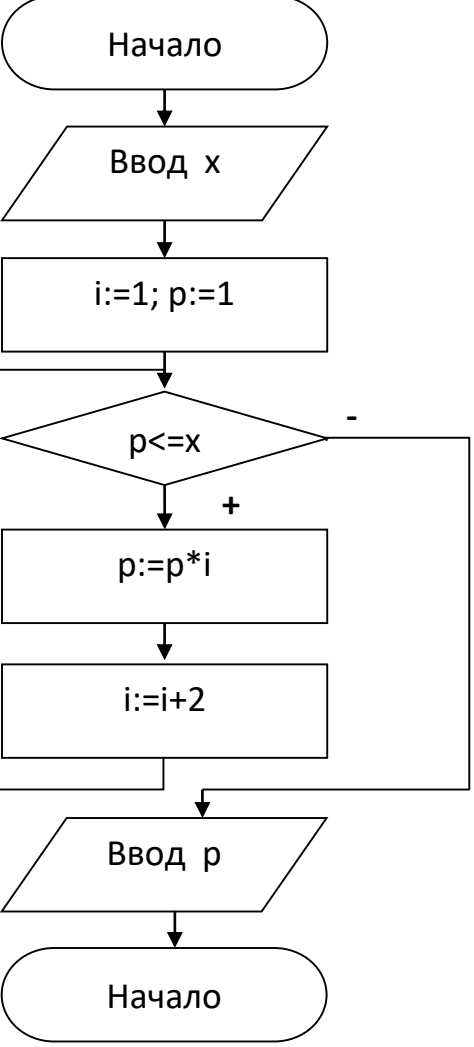
1 вариант*	2 вариант*	3 вариант*	4 вариант*
Составьте блок-схему для проверки, что больше - (a-3) или (2a-10), - для числа a вводимого с клавиатуры	Составьте блок-схему для определения, является ли значение функции $Y = \frac{a+b}{a-b}$ числом положительным, значения чисел a и b вводятся с клавиатуры	Составьте блок-схемы для вычисления значений Y по формуле $Y = \sqrt{x}$ для X, заданного с клавиатуры. Если X число отрицательное, то выведите сообщение, что значения Y не существует.	Составьте блок-схему для проверки, что больше - a^2 или $2a^2$, - для числа a вводимого с клавиатуры
1 вариант	2 вариант	3 вариант	4 вариант
Для четырех произвольных чисел, введенных на клавиатуры, выведите на экран соответствующие результаты:	Для четырех произвольных чисел, введенных с клавиатуры, выведите на экран соответствующие результаты:	Для четырёх произвольных чисел, введенных с клавиатуры, выведите на экран соответствующие результаты:	Для четырех произвольных чисел, введенных с клавиатуры, выведите на экран соответствующие результаты:

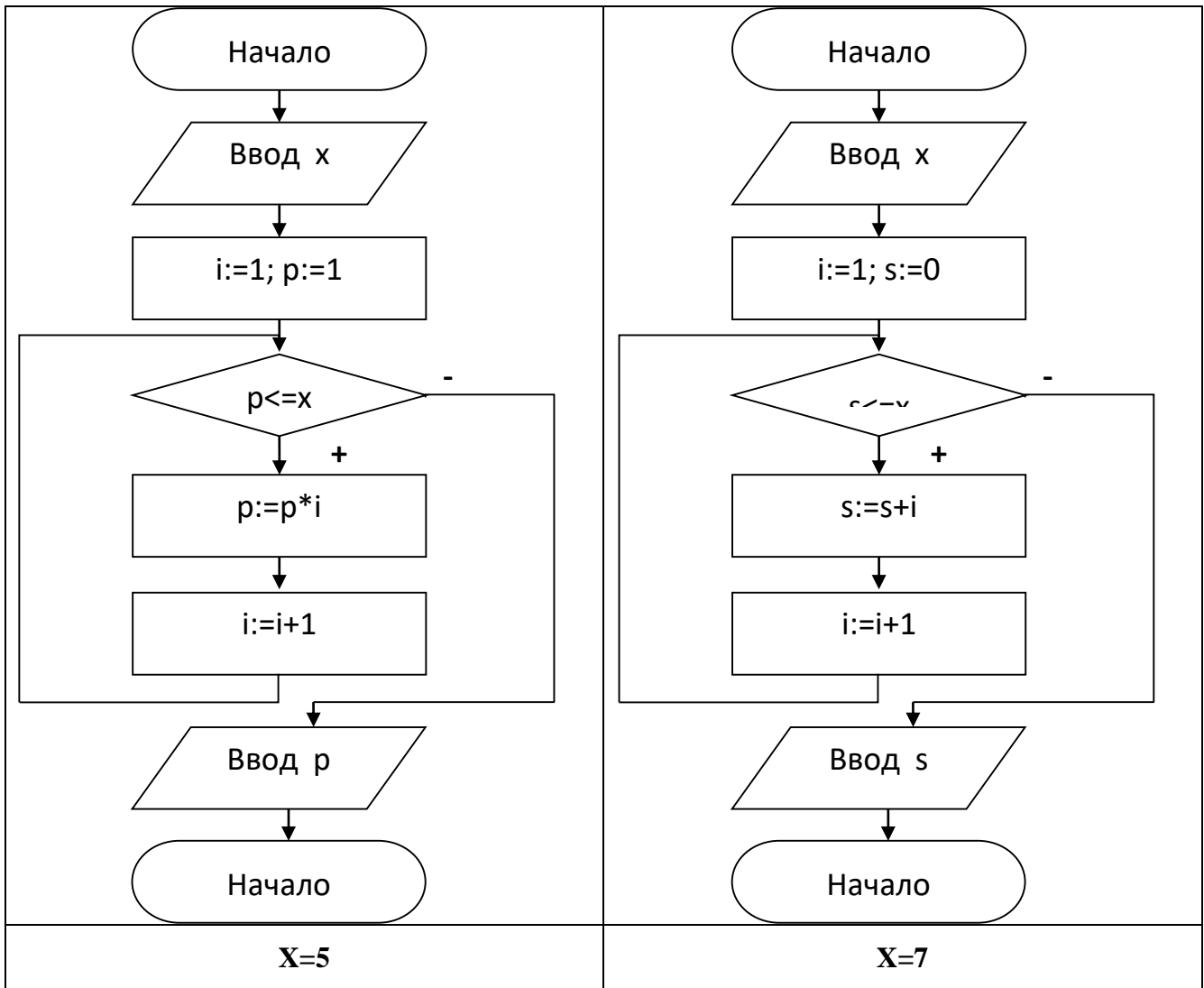
<ul style="list-style-type: none"> • Если произведение второго и четвертого числа четно, то выведите эти два числа и их произведения. • Если сумма первого и третьего числа больше 150, то выведите эту сумму и все четыре числа. • Если выполняются одновременно два перечисленных выше условия, то выведите произведение второго и четвертого чисел и сумму первого и третьего. • Иначе, выведите эти четыре числа без изменения. 	<ul style="list-style-type: none"> • Если произведение этих чисел положительно, то выведите утроенное значение каждого числа. • Если сумма первого и третьего чисел отрицательна, то выведите эти два числа и их сумму. • Если выполняются одновременно два перечисленных выше условия, то выведите четыре числа без изменения. • Иначе, выведите сумму этих четырех чисел, уменьшенную в два раза. 	<ul style="list-style-type: none"> • Если первое число четно, то выведите сумму этих четырех чисел. • Если второе и третье числа положительны, то выведите их среднее арифметическое. • Если выполняются одновременно два перечисленных выше условия, то выведите все четыре числа без изменения. • Иначе, выведите числа, противоположные исходным.. 	<ul style="list-style-type: none"> • Если разность первого и последнего превышает второе, то выведите второе число. • Если сумма второго и третьего меньше четвертого числа, то выведите эту сумму. • Если выполняются одновременно два перечисленных выше условия, то выведите второе число и сумму второго и третьего чисел. • Иначе, выведите числа без изменения
---	---	---	--

9. Продолжите предложение:

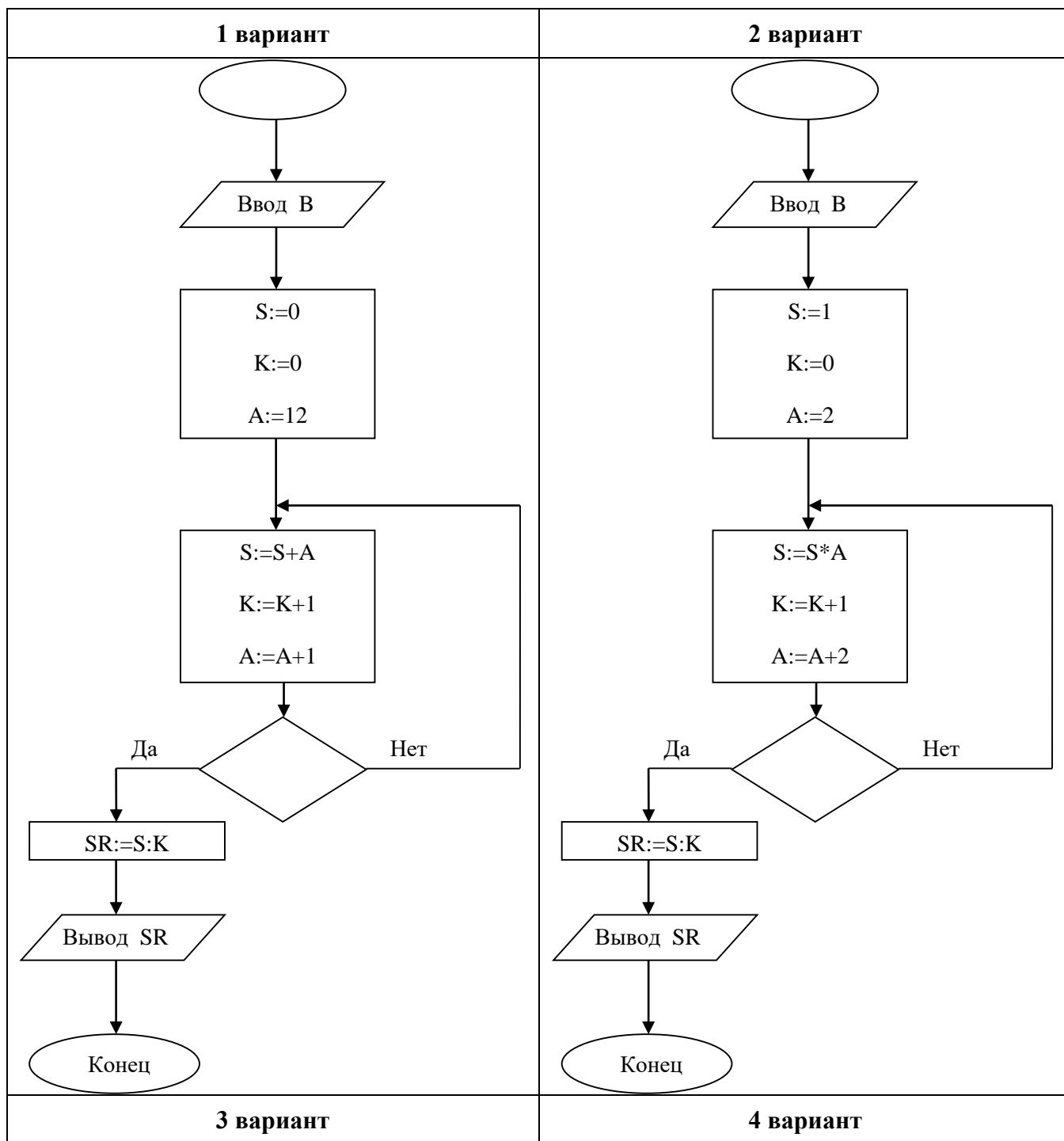
Циклический алгоритм - ...

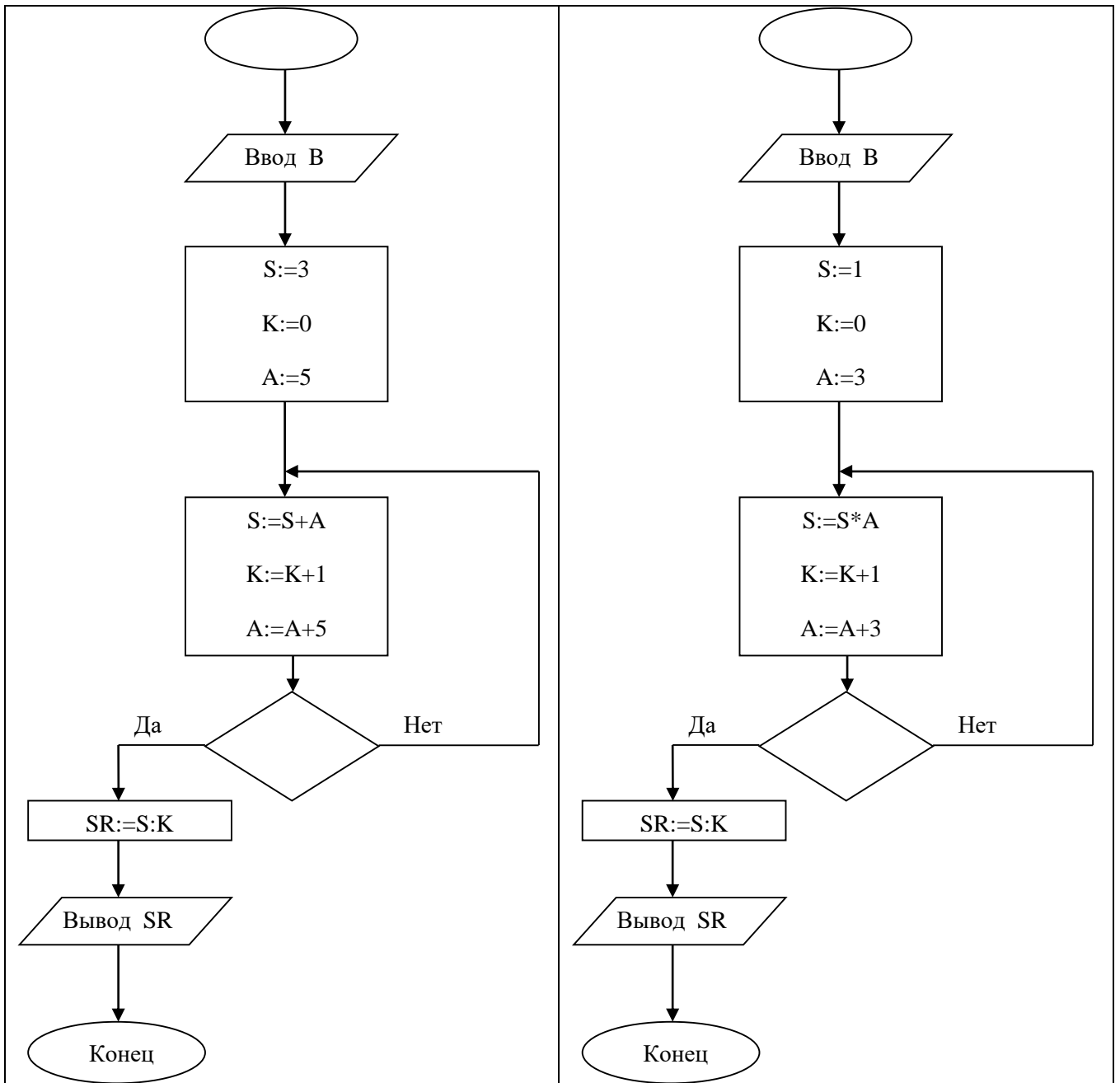
10. Определите, какое значение переменной S/P будет напечатано в результате выполнения алгоритма*. Что находит данный алгоритм.

1 вариант	2 вариант
 <pre> graph TD Start([Начало]) --> Input[/Ввод x/] Input --> Init[i:=2; s:=0] Init --> Loop{с ≤ x} Loop -- "+" --> Sum[s:=s+i] Sum --> Inc[i:=i+2] Inc --> Loop Loop -- "-" --> Output[/Ввод s/] Output --> End([Начало]) </pre>	 <pre> graph TD Start([Начало]) --> Input[/Ввод x/] Input --> Init[i:=1; p:=1] Init --> Loop{p ≤ x} Loop -- "+" --> Mult[p:=p*i] Mult --> Inc[i:=i+2] Inc --> Loop Loop -- "-" --> Output[/Ввод p/] Output --> End([Начало]) </pre>
X=10	X=7
3 вариант	4 вариант



11. Определите, какое значение переменной SR будет напечатано в результате выполнения алгоритма.





12. Составить блок-схему

1 вариант	2 вариант	3 вариант	4 вариант
Вывести значения функции $y = 5 - x^2/2$ на отрезке $[a; b]$ с шагом h .	Вывести кубы нечетных чисел от A до B .	Найти среднее арифметическое чисел на отрезке $[a; b]$.	Вывести квадраты четных чисел от A до B .

Практическая работа № 28-29. Составление линейных программ

Цель: развить навыки составления программ, имеющих линейную структуру на языке программирования Паскаль

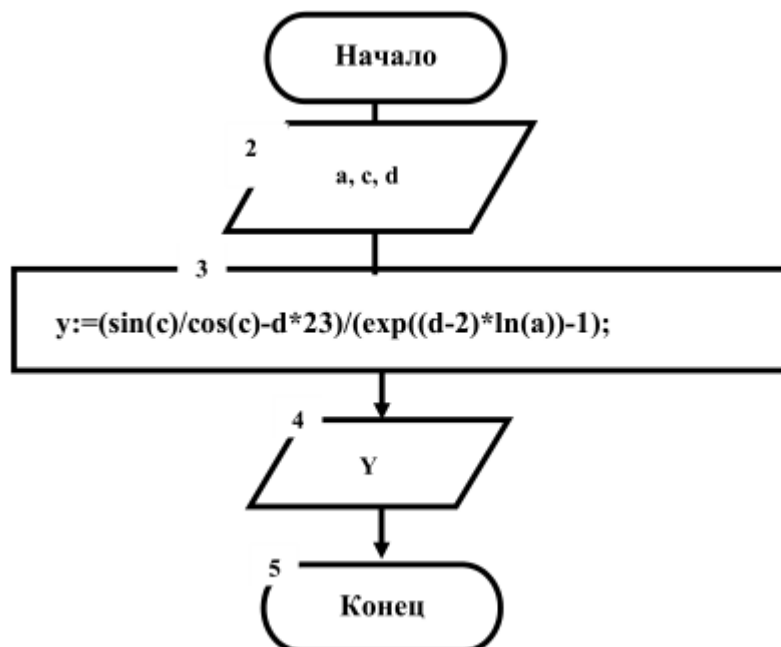
Теоретический материал

Программы с линейной структурой являются простейшими и используются для реализации обычных вычислений по формулам (рисунок). В программах с линейной структурой инструкции выполняются последовательно, одна за другой.

Пример 1. Написать программу вычисления функции $Y(a,c,d)$. Значения a, c, d вводятся с клавиатуры.

$$y = \frac{\operatorname{tg}c - d * 23}{a^{d-2} - 1}$$

Блок-схема алгоритма для решения данной задачи



Текст программы на языке Pascal

```
Program Project1; //заголовок программы
var a,c,d,y: real; //раздел описания переменных
begin //начало раздела операторов
  writeln('Введите значения a, c, d '); //вывод текста на экран
  readln(a,c,d); // ввод значений a,c,d
  //вычисление значения y и присваивание полученного значения
  //переменной y
  y:=(sin(c)/cos(c) - d * 23)/(exp((d - 2) * ln(a)) - 1);
  //вывод полученного значения на экран
  writeln ('Значение функции y= ', y:5:2);
  readln; //задержка экрана для просмотра результата
end. //конец программы
```

Задания практической работы

№	Функция	Проверочное значение у(х) при х = 2 (или у(х1, х2) при х1 = 2 и х2 = 2)
1	$y = \frac{5x + \sin x^2}{2x + \operatorname{tg}x} + \sin x^2 $	5,8496
2	$y = \frac{\log_3 x-10 }{2x-7} - \cos(3x+5)$	-0,6354
3	$y = \frac{\cos(x_1+5) + x_2}{3x_1+6x_2} * \sin x_1^2 $	0,1158
4	$y = \frac{\cos(3x+5) + 5^x}{4.35x} + \cos x^2 + \frac{5}{x}$	6,027716
5	$y = \frac{3x_1 + 2^{x_2}}{\sin(x_1 + x_2)} + \frac{5}{ \sin x_2 } + 6$	-1,7147
6	$y = \frac{\sin^2(x+3.5) + \lg x}{3x + e^{x+1.35}}$	0,0232
7	$y = \frac{x_1 + 5.5x_2}{\cos x_1 + \ln x_2} + \operatorname{tg}(x_1 + 4.3)$	46,9482
8	$y = \frac{3^{x-1.4} + e^x}{4.5 + x} + \operatorname{tg} 3x$	1,1432
9	$y = \frac{x_1^{x_2} + 3.5 \operatorname{tg} x_1}{3x_1 + 5x_2} + \frac{5x_1}{x_2 + 6} + 5$	6,022
10	$y = \frac{\cos^3(x+2.5) + 1.5x}{x^2 + 3.5} + \frac{3x^2}{8x} + 8.5$	9,6488
11	$y = \frac{\ln(x+1.3) + 5}{1.35x^2 + 6} + \sqrt{\frac{x+1.3}{(35x)^2 + 6}}$	0,5693
12	$y = \frac{2x_1 + 1.4^{x_2}}{\operatorname{tg} x_1 + 2x_2} + \sqrt[3]{6x_1 + 8^{x_2}}$	7,5196
13	$y = \frac{3x + \operatorname{tg}x}{2.36x + 6} + 2x + 1.4^x$	6,3159

14	$y = \frac{e^{x-1.3} + \sin x}{x + 3.5} + 2x + 1.4 \sin x$	5,8045
15	$y = \frac{x + 3 \cos(x^2 + 1.5)}{\operatorname{tg} x + 4.56} + \sin x \frac{2 + x}{\sqrt{5^x + 56x}}$	2,0480
16	$y = \frac{x + 3 \cos(x^2 + 1.5)}{\operatorname{tg} x + 4.56} + \sin(2x) + \cos(x^2 + 5)$	0,0694
17	$y = \frac{\cos^2(x_1 + 1.3) + x_2}{2x_1 + e^{x_2}} + \cos(x_2^2 + 1.5) + \frac{x_2}{\cos(x_1^2 + 1.5)}$	3,7921
18	$y = \frac{5x_1 + 1.3^{x_2}}{\cos(x_1 + x_2)} + \cos(x_2^2 + 1.5) + \frac{5 + x_2}{\sqrt{x_1}}$	-12,2259
№	Функция и проверочные данные	
19	$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right).$	
	При $x=14.26, y=-1.22, z=0.035$ $t=0.564849$	
20	$u = \frac{\sqrt[3]{8 + x - y ^2 + 1}}{x^2 + y^2 + 2} - e^{ x-y } (\operatorname{tg}^2 z + 1)^x.$	
	При $x=-4.5, y=0.000075, z=84.5$ $u=-55.6848$	
21	$v = \frac{1 + \sin^2(x + y)}{\left x - \frac{2y}{1 + x^2 y^2}\right } x^{ y } + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right).$	
	При $x=0.0374, y=-0.825, z=16, v=1.0553$	
22	$w = \cos x - \cos y ^{(1+2 \sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$	
	При $x=4000, y=-0.875, z=-0.000475$ $w=1.9873$	
23	$\alpha = \ln\left(y^{-\sqrt{ x }}\right) \left(x - \frac{y}{2}\right) + \sin^2 \operatorname{arctg}(z).$	
	При $x=-15.246, y=0.04642, z=2000.1$ $\alpha=-182.036$	
24	$\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\arcsin^2 z - x - y)$	
	При $x=0.01655, y=-2.75, z=0.15$ $\beta=-40.6307$	
25	$\gamma = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \frac{x + 3 x - y + x^2}{ x - y z + x^2}.$	
	При $x=0.1722, y=6.33, z=0.000325$ $\gamma=-205.3056$	
26	$\varphi = \frac{e^{ x-y } x - y ^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}.$	
	При $x=-0.02235, y=2.23, z=15.221$ $\varphi=39.374$	
27	$\psi = \left x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right + (y - x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}.$	
	При $x=182.5, y=18.225, z=-0.03.298$ $\psi=1.2131$	

28	$a = 2^{-x} \sqrt{x + 4\sqrt{ y }} \sqrt[3]{e^{x-1/\sin z}}$
	При $x=0.03981, y=-1625, z=0.512$ $a=1.26185$.
29	$b = y^{\sqrt[3]{ x }} + \cos^3(y) \frac{ x-y \left(1 + \frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{ x-y } + \frac{x}{2}}$
	При $x=6.251, y=0.827, z=25.001$ $b=0.7121$
30	$c = 2^{(y^x)} + (3^x)^y - \frac{y \left(\arctg(z) - \frac{\pi}{6} \right)}{ x + \frac{1}{y^2 + 1}}$
	При $x=3.251, y=0.325, z=0.0000466$ $c=4.2514$
31	$f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{ x-y (\sin^2 z + \operatorname{tg} z)}$
	При $x=17.421, y=0.010365, z=82800$ $f=0.33056$
32	$g = \frac{y^{x+1}}{\sqrt[3]{ y-2 } + 3} + \frac{x + \frac{y}{2}}{2 x+y } (x+1)^{-1/\sin z}$
	При $x=1.23, y=15.4, z=252$ $g=82.8257$.
33	$h = \frac{x^{y+1} + e^{y-1}}{1 + x y - \operatorname{tg} z } (1 + y-x) + \frac{ y-x ^2}{2} - \frac{ y-x ^3}{3}$
	При $x=2.444, y=0.00869, z=-130$ $h=-0.49871$

Контрольные вопросы

1. Из каких разделов состоит программа на языке Паскаль?
2. Что такое оператор?
3. Какие операторы языка Паскаль вам известны?
4. Зачем нужен оператор присваивания? Какой вид он имеет?
5. Что может быть записано в правой части оператора присваивания?
6. Что такое переменная?
7. Что такое константа?
8. Какие правила применяются для создания имен переменных?
9. Что такое идентификатор?
10. Почему знак умножения всегда выписывают явно (например, пишут $a*t$, а не at)?
11. Как описываются переменные в Паскале?
12. Какие стандартные числовые типы языка Паскаль вам известны?
13. Что вам известно о соответствии типов переменных в языке Паскаль?
14. Какие арифметические операции можно выполнять в Паскале?
15. Что вам известно о приоритете арифметических действий в Паскале?
16. Какие математические функции есть в Паскале?
17. Какая команда служит для ввода данных?
18. Какой формат записи имеет команда ввода?
19. Чем команда ввода READ отличается от READLN?
20. Какая команда служит для вывода данных?
21. Какой формат записи имеет команда вывода?
22. Чем команда вывода WRITE отличается от WRITELN?
23. Как распечатать текст?
24. Что такое форматный вывод?
25. Как напечатать значение переменной на фиксированном количестве позиций?

26. Как напечатать значение переменной с фиксированным количеством знаков после запятой?
27. Как разместить комментарии в программе?

Практическая работа № 30. Составление программ обработки текстовых данных

Цель: Изучить строковый тип `string`. Изучить стандартные функции работы со строками, научиться выполнять стандартные операции со строками.

Теоретический материал

1 Строка представляет собой особую форму одномерного массива символов, которая имеет существенное отличие. Массив символов имеет фиксированную длину (количество элементов), которая определяется при описании.

Строка имеет две разновидности длины:

- Общая длина строки, которая характеризует размер памяти, выделяемый строке при описании;

- Текущая длина строки (всегда меньше или равна общей длине), которая показывает количество смысловых символов строки в каждый конкретный момент времени.

2 Строка в Паскале – упорядоченная последовательность символов. Количество символов в строке называется ее длиной. Длина строки в Паскале может лежать в диапазоне от 0 до 255. Каждый символ строковой величины занимает 1 байт памяти и имеет числовой код в соответствии с таблицей кодов ASCII.

3 Для описания строковых переменных в Паскале существует предопределенный тип **string**. В общем виде описание строковой переменной будет выглядеть следующим образом:

```
Var <имя_переменной>: string[<максимальная длина строки>]
```

4 Операция слияния (сцепления, конкатенации) применяется для соединения нескольких строк в одну, обозначается знаком «+». Операция слияния применима для любых строковых выражений, как констант, так и переменных.

5 Операции отношения позволяют сравнивать строки на отношение равенства (=), неравенства (<>), больше (>), меньше (<), больше или равно (>=), меньше или равно (<=). В результате сравнения двух строк получается логическое значение (true или false). Сравнение строк производится слева направо посимвольно до первого несовпадающего символа, большей считается та строка, в которой первый несовпадающий символ имеет больший код в таблице кодировки. Если строки имеют различную длину, но в общей части символы совпадают, считается, что короткая строка меньше. Строки равны, если они имеют равную длину и соответствующие символы совпадают.

6 Стандартные функции для работы со строками в Паскале

- `Copy (S, poz, n)` выделяет из строки `S`, начиная с позиции `poz`, подстроку из `n` символов. Здесь `S` – любое строковое выражение, `poz`, `n` – целочисленные выражения.

- `Concat (s1, s2,...,sn)` выполняет слияние строк `s1, s2,...,sn` в одну строку.

- `Length(S)` определяет текущую длину строкового выражения `S`. Результат – значение целого типа.

- `Pos(subS, S)` определяет позицию первого вхождения подстроки `subS` в строку `S`. Результат – целое число, равное номеру позиции, где находится первый символ искомой подстроки. Если вхождение подстроки не обнаружено, то результат функции будет равен 0

7 Стандартные процедуры для работы со строками в Паскале:

- `Delete (S, poz, n)` удаляет из строки `S`, начиная с позиции `poz`, подстроку из `n` символов. Здесь `S` – строковая переменная (в данном случае нельзя записать никакое другое строковое выражение, кроме имени строковой переменной, т.к. только с именем переменной связана область памяти, куда будет помещен результат выполнения процедуры); `poz`, `n` – любые целочисленные выражения.

- `Insert(subS, S, poz)` вставляет в строку `S`, начиная с позиции `poz`, подстроку `subS`. Здесь `subS` – любое строковое выражение, `S` – строковая переменная (именно ей будет присвоен результат выполнения процедуры), `poz` – целочисленное выражение.

8 Процедуры преобразования типов в Паскале:

– Str(x, S) преобразует число x в строковый формат. Здесь x – любое числовое выражение, S – строковая переменная. В процедуре есть возможность задавать формат числа x. Например, str(x: 8: 3, S), где 8 – общее число знаков в числе x, а 3 – число знаков после запятой.

– Val(S, x, kod) преобразует строку символов S в число x. Здесь S – строковое выражение, x – числовая переменная (именно туда будет помещен результат), kod – целочисленная переменная (типа integer), которая равна номеру позиции в строке S, начиная с которой произошла ошибка преобразования, если преобразование прошло без ошибок, то переменная kod равна 0

Пример 1. Дана строка. Определить, сколько раз в нее входит группа букв ешь.

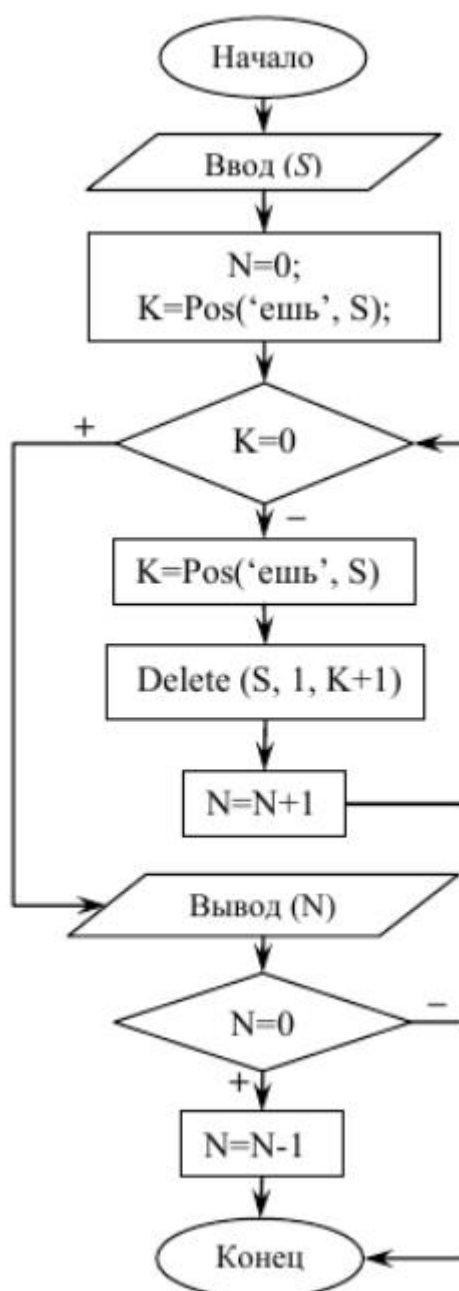
Решение:

1 Алгоритм задачи:

а) для поиска первого вхождения сочетания «ешь» используем функцию Pos, она покажет первый символ k;

б) дальше надо отрезать от строки символы с первого по k+1 и снова воспользоваться указанной функцией;

в) выполнять эти действия, пока значение Pos не станет нулевым, но когда это произошло, N все равно увеличивается на 1, поэтому печатаем N – 1



2 Программа на языке Паскаль:

```
program stroka1;
var K, N:integer;
    S:string;
begin
    writeln('Введите текст');
    readln(S);
    N:=0;
    K:=Pos('ешь', S);
    while K<>0 do
    begin
        K:=Pos('ешь', S);
        N:=N+1;
        Delete (S, 1, K+1);
    end;
    if N>0 then N:=N-1;
    writeln('В тексте ешь встречается ', N, ' раз');
end.
```

Задания практической работы

1 Даны целые положительные числа n_1 и n_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые n_1 символов S_1 и последние n_2 символов строки S_2 .

2 Даны целое положительное число n и строка S . Преобразовать строку S в строку длины n следующим образом: если длина S больше n , то отбросить первые символы, если меньше, то в начало добавить «.».

3 Даны строки S_1 и S_2 . Проверить, содержится ли S_1 в строке S_2 .

4 Даны строки S_1 и S_2 . Удалить из строки S_1 подстроку S_2 , если S_2 не содержится в S_1 , вывести S_1 без изменений

5 Даны строки S_1 и S_2 . Удалить из строки S_1 первую подстроку S_2 , если S_2 не содержится в S_1 , вывести S_1 без изменений

6 Даны строки S_1 и S_2 . Удалить из строки S_1 последнюю подстроку S_2 , если S_2 не содержится в S_1 , вывести S_1 без изменений

7 Даны строки S_1 и S_2 . Удалить из строки S_1 все подстроки S_2 , если S_2 не содержится в S_1 , вывести S_1 без изменений

8 Даны строки S , S_1 и S_2 . Заменить в строке S последнюю подстроку S_1 на S_2 , если S_1 не содержится в S , вывести S без изменений

9 Даны строки S , S_1 и S_2 . Заменить в строке S первую подстроку S_1 на S_2 , если S_1 не содержится в S , вывести S без изменений

10 Даны строки S , S_1 и S_2 . Заменить в строке S все подстроки S_1 на S_2 , если S_1 не содержится в S , вывести S без изменений

11 Дано полное имя файла, т.е. путь к файлу, имя и расширение. Выделить из строки только имя файла без расширения.

12 Дано полное имя файла, т.е. путь к файлу, имя и расширение. Выделить из строки только расширение файла без точки.

13 Дано полное имя файла, т.е. путь к файлу, имя и расширение. Выделить из строки только имя первого каталога без символа \.

14 Дано полное имя файла, т.е. путь к файлу, имя и расширение. Выделить из строки только имя последнего каталога без символа \.

15 Дана строка, содержащая круглые скобки. Выдать сообщение, правильно ли расставлены скобки (количество открывающихся соответствует количеству закрывающихся)

16 Строка содержит одно слово. Проверить, будет ли оно читаться одинаково слева направо и наоборот.

17 В строке заменить все пробелы на «_», посчитать количество замененных символов

18 В строке заменить все пробелы на «*», посчитать количество замененных символов

19 Дана строка. Определить, сколько в ней «,»

- 20 Дана строка. Определить, сколько в ней «;»
- 21 Удалить часть символьной строки в скобках (вместе со скобками)
- 22 Удалить часть символьной строки в кавычках (вместе с кавычками)
- 23 Дана строка, содержащая кавычки. Выдать сообщение, правильно ли расставлены кавычки (количество открывающихся соответствует количеству закрывающихся)
- 24 Дана строка. Найти количество слов, начинающихся с буквы А
- 25 Дана строка. Найти количество слов, начинающихся с буквы А
- 26 Дана строка. Заменить первую букву слов, начинающихся с буквы А, на малую букву
- 27 Дана строка. Заменить последнюю букву слов, заканчивающихся на е, на большую букву
- 28 Дана строка. Найти количество букв к.
- 29 Дана строка. Определить, сколько раз в нее входит группа букв ель
- 30 Дана строка. Определить, сколько раз в нее входит группа букв ого

Контрольные вопросы:

- 1 Что такое строка в языке Паскаль?
- 2 Чему
равна длина строки в Паскале?
- 3 Операции над строками в Паскале
- 4 Стандартные функции работы со строками
- 5 Стандартные процедуры для работы со строками в Паскале
- 6 Процедуры преобразования типов в Паскале

Практическая работа № 31-32. Составление разветвляющихся программ

Цель: развить навыки составления программ, имеющих разветвляющуюся структуру на языке программирования Паскаль; изучить синтаксис операторов IF...THEN...ELSE и CASE

Теоретический материал

При разработке вычислительных алгоритмов часто возникает необходимость выбора направления дальнейшего решения задачи в зависимости от некоторого заданного условия. Алгоритмы такого типа называются *разветвляющимися*. В языке Паскаль их можно реализовать с помощью условных операторов.

Условные операторы обеспечивают выполнение некоторого оператора или группы операторов в зависимости от заданных условий. Для программирования разветвляющихся алгоритмов используются условные операторы IF или CASE.

Оператор IF

Оператор условного перехода IF в программах представляется в одном из двух форматов:

IF логическое выражение **THEN** 1 блок инструкций
ELSE 2 блок инструкций;

или

IF логическое выражение **THEN** 1 блок инструкций;

Здесь **IF**, **THEN** и **ELSE** – ключевые слова; логическое выражение – оператор сравнения или сложное логическое выражение; блок инструкций (1 и 2) – любые исполняемые операторы языка или блоки операторов, заключенные в операторные скобки (BEGIN ... END).

При выполнении условного оператора **IF** вначале анализируется результат логического выражения и в зависимости от его значения управление передается одному из операторов (блоку операторов), следующему за ключевым словом **THEN** или **ELSE**. Если значение результата **TRUE** «истина», то выполняется оператор (блок операторов), следующий за ключевым словом **THEN**. В противном случае (значение результата **FALSE** «ложь») выполняется оператор (блок операторов), следующий за ключевым словом **ELSE**. Далее в любом случае выполняется оператор, следующий в программе непосредственно за оператором **IF**. Ниже приведен фрагмент программы, поясняющий действие рассматриваемого оператора:

```
IF A <= B THEN D := 2*D ELSE D := ABS(D);
```

В укороченном операторе **IF** отсутствует ключевое слово **ELSE** и блок инструкций, следующий за ним. Действия такой конструкции аналогичны рассмотренной выше. Если значение результата логического выражения **TRUE**, то выполняется оператор (блок операторов), следующий за ключевым словом **THEN**. При значении результата **FALSE** блок инструкций – 1 не выполняется, а сразу же выполняется блок инструкций, непосредственно следующий за оператором **IF**. Пример использования усеченного логического оператора **IF**

```
IF A <= B THEN BEGIN D := 2*D; GOTO 10 END;
```

В последнем фрагменте пришлось после **THEN** вводить не один оператор, а так называемый составной оператор (блок операторов), который заключен в операторные скобки **BEGIN...END**; Формат такого составного оператора имеет вид:

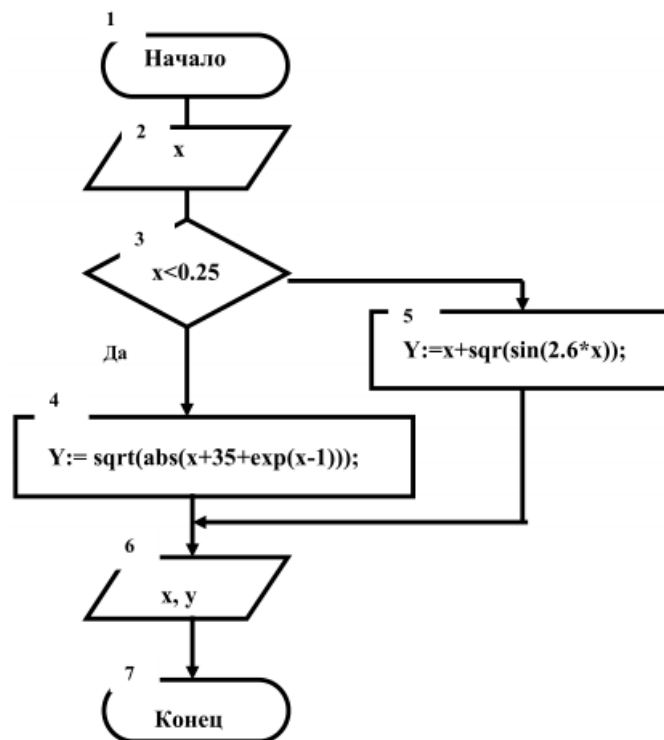
```
BEGIN  
    1 оператор;  
    2 оператор;  
    N оператор  
END;
```


В операторе условного перехода любой из операторов блоков инструкций, в свою очередь, может быть оператором условного перехода, образуя так называемую вложенную конструкцию операторов **IF**. Создавая подобную конструкцию, следует руководствоваться принятым в Паскале соглашением о том, что инструкция **ELSE** всегда относится к ближайшему предшествующему **IF**. С учетом этого правила вложенная конструкция **IF** должна иметь следующий вид:

IF 1 логическое выражение **THEN** 1 блок инструкций
ELSE IF 2 логическое выражение **THEN** 2 блок инструкций
ELSE 3 блок инструкций.

Пример 1. Определить значение функции $Y(x)$.

$$y = \begin{cases} \sqrt{|x + 35 + e^{x-1}|}, & \text{если } x < 0,25 \\ x + \sin^2(2,6x), & \text{если } 0,25 \leq x \end{cases}$$

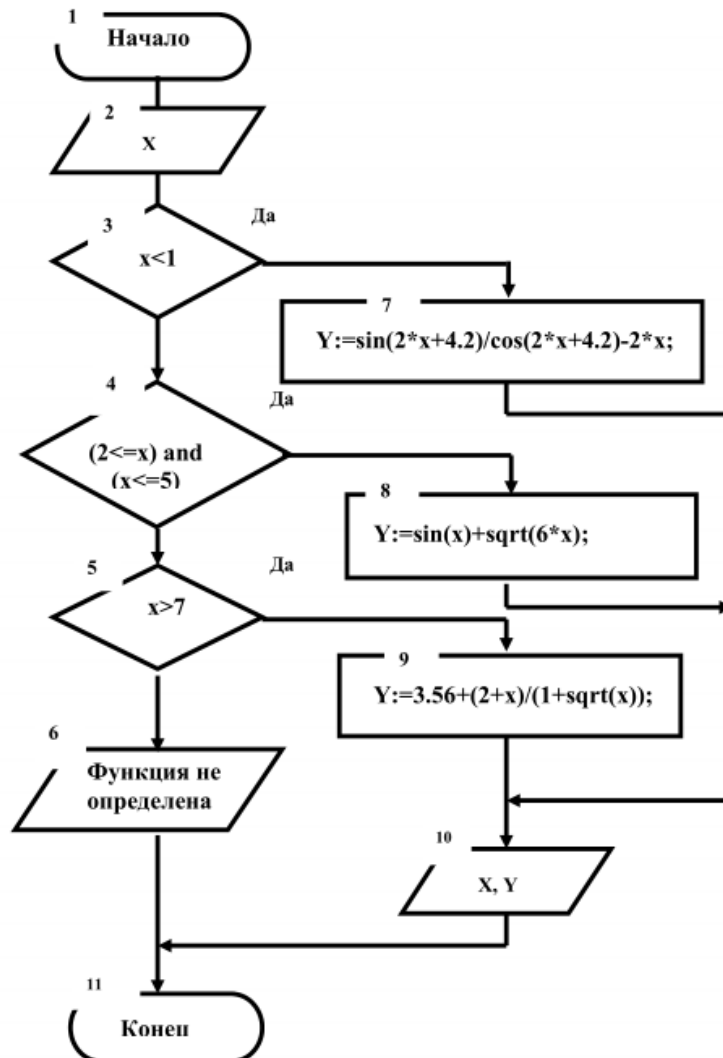


```

Program Project1; //заголовок программы
var x,y: real; //раздел описания переменных
begin //начало раздела операторов
  //вывод текста на экран
  writeln('Введите значение x');
  write('x=');
  readln(x); // ввод значения x
  if x < 0.25 then y := sqrt(abs(x+35+exp(x-1)))
    else y := x + sqrt(sin(2.6*x));
  //вывод результата
  writeln('при x = ',x : 6 : 2 , ' y = ', y : 6 : 2);
  readln; //задержка экрана
end.
  
```

Пример 2. Рассчитать значение функции $Y(x)$, значение x вводится с клавиатуры. При вводе значения x , для которого функция не определена, должно выводиться сообщение «Функция не определена».

$$y = \begin{cases} \operatorname{tg}(2x+4,2) - 2x, & x < 1 \\ \sin x + \sqrt{6x}, & 2 \leq x \leq 5 \\ 3,56x + \frac{2+x}{1+\sqrt{x}}, & x > 7 \end{cases}$$



```

Program Project2;           //заголовок программы
var x,y: real;             //раздел описания переменных
begin                      //начало раздела операторов
    //вывод текста на экран
    writeln('Введите значение x');
    write(_x=' ');
    readln(x);             // ввод значения x
    //1 логическое условие
    if x < 1 then y := sin(2*x+4.2)/cos(2*x+4.2)-2*x
    else
    //2 логическое условие
        if (2<=x) and (x<=5) then y := sin(x)+sqrt(6*x)
        else
    //3 логическое условие
        if x>7 then y:= 3.56+(2+x)/(1+sqrt(x))
        else
            begin

```

```

        writeln('При x=', x, ' функция не определена');
        readln;
        exit;
    end;
    writeln('при x = ', x : 6: 3, ' y = ', y : 6 : 2); //вывод
результата
    readln; //задержка экрана
end.

```

Оператор выбора CASE

Инструкция **Case** используется для выбора одного из нескольких направлений дальнейшего хода программы (последовательности инструкций, которые могут быть выполнены). Эта структура обеспечивает выбор одного из ряда возможных действий, в зависимости от значения, которое принимает переменная-селектор (условие). Выбор варианта действия осуществляется во время выполнения программы в зависимости от равенства значения переменной-селектора константе, указанной перед группой инструкций. В качестве переменной-селектора можно использовать переменную скалярного типа (integer или char). Конструкция имеет следующий формат:

```

CASE переменная-селектор OF
    1 константа: 1 оператор;
    2 константа: 2 оператор;
    -----
    N константа: N оператор
ELSE
    оператор;
END;

```

Эта конструкция имеет и укороченный формат:

```

CASE переменная-селектор OF
    1 константа: 1 оператор;
    2 константа: 2 оператор;
    -----
    K константа: K оператор
END;

```

Пример 3. Вводится число от 0 до 21. Вывести сообщение: число меньше 10, число больше 10, число равно 10. При вводе числа больше 20 или меньше 0 должно выводиться сообщение «Ошибка ввода».

```

Program Project1;           //заголовок программы
var a: integer;           //раздел описания переменных
begin                     //начало раздела операторов
    //вывод текста на экран
    write('Введите число a=');
    readln(a);           //ввод значения a
    case a of
        0..9: writeln( '__число меньше 10`');
        11..20: writeln( '__число больше 10`');
        10: writeln( '__число равно 10`');
    else
        writeln( '__Ошибка ввода`')
    end;
    readln;
end.

```

Задания практической работы

Задание 1. Решить 2 задачи

1.1	$y = \begin{cases} \sin(x+2), & x < 1,35 \\ x + 3,5\text{tg}x, & 1,35 \leq x \end{cases}$	1.2	$y = \begin{cases} \sin(x+2), & x < 1,35 \\ x + 3,5\text{tg}x, & 2 \leq x \leq 4 \\ \sqrt{ 2,56x - 0,35 }, & x > 5 \end{cases}$
2.1	$y = \begin{cases} \text{tg}^2(x+2,5) + e^x, & x < 0,25 \\ 3,5\sin x + \cos^2 x, & 0,25 \leq x \end{cases}$	2.2	$y = \begin{cases} \frac{\text{tg}(x ^{2x}) + x^2}{\cos(x+2)}, & x < -5 \\ 12 + 5x, & 0 \leq x < 5 \\ \sqrt{ x^{2-x} - x^2 } * \text{tg}2x, & x \geq 10 \end{cases}$
3.1	$y = \begin{cases} 3x + 5, & x \leq 1 \\ \sqrt{7x - 6,35}, & 1 \leq x \end{cases}$	3.2	$y = \begin{cases} 5,6(1 + \text{tg}x), & x < 0,5\pi \\ \sin x + 6, & \pi \leq x \leq 3\pi \\ 2,56\sqrt{x^{2x} + 1}, & x > 5\pi \end{cases}$
4.1	$y = \begin{cases} 3^{x+3,5} + \text{tg}(2x), & x < 1,45 \\ \sin^2 x + e^x, & 1,45 \leq x \end{cases}$	4.2	$y = \begin{cases} 3e^{\sin x} * \text{tg}2x, & x \leq 2 \\ 2,47\lg x + x^{2x}, & 3 < x < 4 \\ \sqrt{ \cos^2 x + 6} + 4, & x \geq 6 \end{cases}$
5.1	$y = \begin{cases} \cos x + 3, & x < 2,5 \\ x + 1,35, & 2,5 \leq x \end{cases}$	5.2	$y = \begin{cases} 6(\cos^2 x - \sin^2 x), & x < -4 \\ \sqrt{3,2x^2 + \text{tg}^2 x}, & -1 \leq x < 2 \\ e * \cos(2,58x), & x \geq 5 \end{cases}$
6.1	$y = \begin{cases} \log_3(x+2), & x \geq 1 \\ \frac{x}{\sin(x-6)}, & 1 > x \end{cases}$	6.2	$y = \begin{cases} \text{tg}(2x+4,2) - \lg x , & x < 1 \\ \sin x + \sqrt{6x}, & 2 \leq x \leq 5 \\ 6 + \text{arctg}\left(\frac{2x}{1+\sqrt{x}}\right), & x > 7 \end{cases}$
7.1	$y = \begin{cases} \text{tg}(x+3) - \frac{1}{x}, & x < 2,5 \\ x^2 + 6, & 2,5 < x \end{cases}$	7.2	$y = \begin{cases} 2 + x^{2+x}, & x < 0,1\pi \\ \sin^2(x^2 + 0,5), & \pi \leq x \leq 1,5\pi \\ \sin^2(2,45\sqrt{x-1,2}), & x > 2,5\pi \end{cases}$

8.1	$y = \begin{cases} x^3 + 2, & 0 \leq x \leq 2,5 \\ \sqrt[3]{x+3,5} + \sin x^2, & 2,5 < x \end{cases}$	8.2	$y = \begin{cases} \sqrt{1+2,4x^2}, & x < 1 \\ \lg 5,9x, & 2 \leq x \leq 3 \\ \sqrt{x^2+1} + 2x, & x > 5 \end{cases}$
9.1	$y = \begin{cases} x^2 + 3x \sin(x+6), & x \leq 1,3 \\ \sqrt{ x-10,5 + \operatorname{tg}^2 x}, & 1,3 < x \end{cases}$	9.2	$y = \begin{cases} e^{\sin(x)} \operatorname{tg}(2,3+x), & x < -0,1\pi \\ \operatorname{arctg} x, & -1 \leq x \leq 0,5 \\ -1 + \sqrt{23,5x}, & x > 1,5\pi \end{cases}$
10.1	$y = \begin{cases} x^2 + 5,35, & 4 > x \\ \frac{\sqrt{3x-6,6}}{\operatorname{tg} x}, & 4 \leq x \end{cases}$	10.2	$y = \begin{cases} \frac{e}{\operatorname{tg}(2+x)}, & x < 1 \\ \ln((x+2)^2 - x^2), & 2 \leq x \leq 4 \\ \sin^2(x-6), & x > 5 \end{cases}$
11.1	$y = \begin{cases} 2^{x+3} + \sin x, & x > 2 \\ \sin^2(x-6), & x \leq 2 \end{cases}$	11.2	$y = \begin{cases} \ln(5,3x^{3x} - x^2), & x < 1 \\ \frac{0,025}{\operatorname{tg}(2,6+x)}, & 2 \leq x \leq 4 \\ \sin^2(x-6), & x > 5 \end{cases}$
12.1	$y = \begin{cases} \sin(x+1), & x < 3,5 \\ 2,5x+3, & 3,5 \leq x \end{cases}$	12.2	$y = \begin{cases} \frac{e}{\operatorname{tg}(2,9+3x)}, & x < 1 \\ 0,6 \ln(5-x^2), & 2 \leq x \leq 4 \\ \sin 3x + \lg(x+0,3), & x > 6 \end{cases}$
13.1	$y = \begin{cases} x^2 + 8, & 6 > x \\ \frac{\sqrt{3x-3,6}}{\operatorname{tg} x + \lg(x+2)}, & 6 \leq x \end{cases}$	13.2	$y = \begin{cases} \frac{2,3x+3,56}{\operatorname{tg}(x+1)}, & x < 2 \\ \sin(5,4x^2 - x^{2-x}), & 3 \leq x \leq 4 \\ \operatorname{tg}(x+3) - \frac{1}{x}, & x > 8 \end{cases}$
14.1	$y = \begin{cases} 24x^{2-x}, & 3 > x \\ \frac{\ln x}{\operatorname{tg} x} x^2 + 5,35, & 3 \leq x \end{cases}$	14.2	$y = \begin{cases} 6,25 + 7x, & x < 2 \\ 5 \ln(2-x^2), & 6 \leq x \leq 8 \\ \frac{25x}{\operatorname{tg} x}, & x > 10 \end{cases}$
15.1	$y = \begin{cases} x^2 + 6 \sin(x+6), & x \leq 4 \\ \sqrt{ x-10,5 + \operatorname{tg}^2 x}, & 4 < x \end{cases}$	15.2	$y = \begin{cases} \frac{2,56x+2}{\operatorname{tg}(2+x)}, & x < 2 \\ \ln(x^2-1), & 2 \leq x \leq 4 \\ \operatorname{tg}(2,78x+2), & x > 10 \end{cases}$

Задание 2. При некорректном вводе исходных данных программа должна выводить сообщение «Ошибка ввода».

1. Вводится число от 1 до 4, определяющее пору года. Дать название этой поры года (1 — зима, 2 — весна, 3 — лето, 4 — осень).
2. Вводится число от 1 до 7, определяющее день недели. Дать название этого дня (1 — понедельник, 2 — вторник, ..., 7 — воскресенье).
3. Вводятся числа 12, 1, 2, определяющие зимний месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, 12 — декабрь).
4. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два, ..., 10 — десять).
5. Дано натуральное число A ($A < 20$), определяющее сумму денег в рублях. Дать для этого числа наименование: "рубль", "рубля", "рублей".
6. Дано натуральное число N ($N < 100$), определяющее возраст человека в годах. Дать для этого числа наименование: "год", "года", "лет".
7. Вводится число от 1 до 12, определяющее месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, ..., 12 — декабрь).
8. Вводится число от 1 до 20. Дать название этого числа (1 — один, 2 — два, ..., 20 — двадцать).
9. Вводятся числа 3, 4, 5, определяющие весенний месяц года. Дать название этого месяца года (3 — март, 4 — апрель, 5 — май).
10. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два, ..., 10 — десять).
11. Вводится число от 2 до 10. Вывести сообщение: четное или нечетное введенное число.
12. Вводится число от -10 до 10. Вывести сообщение: введенное число больше 0, меньше 0 или равно 0.
13. Написать программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота», «Воскресенье».
14. Вводится номер месяца (1, 2, ..., 12). Вывести количество дней в указанном месяце.
15. Вводится значение года в укороченной форме (от 0 по 10). Вывести значение года текущего столетия в полном формате (0 — 2000, 1 — 2001 и т.д.).

Контрольные вопросы

1. Что такое алгоритм с ветвлением?
2. Как записывается условный оператор (оператор ветвления) в Паскале?
3. Что такое полная и сокращенная записи условного оператора?
4. Что используется в качестве условий в операторе ветвления?
5. Какие знаки отношений можно использовать при составлении условий?
6. Что такое составное условие?
7. Каковы правила записи составных условий?
8. Какие вы знаете логические операции?
9. Какие действуют правила старшинства логических операций?
10. Что располагается после служебных слов THEN и ELSE?
11. Что такое составной оператор? Какую структуру он имеет?
12. В каких случаях используется составной оператор?
13. Как в Паскале записывается оператор выбора?
14. Для чего предназначен оператор выбора?
15. Что такое полная и сокращенная записи оператора выбора?
16. Какого типа должна быть переменная, значения которой выбирают с помощью оператора CASE?

Практическая работа № 33-34. Составление циклических программ

Цель: развить навыки составления структурных схем циклических алгоритмов; развить навыки составления программного кода по структурной схеме; развить навыки тестирования программных кодов.

Методические указания

Пример 1. Составить таблицу значений функции $y = e^{\sin(x)} \cos(x)$ на отрезке $[0; \pi]$ с шагом 0,1.

Решение.

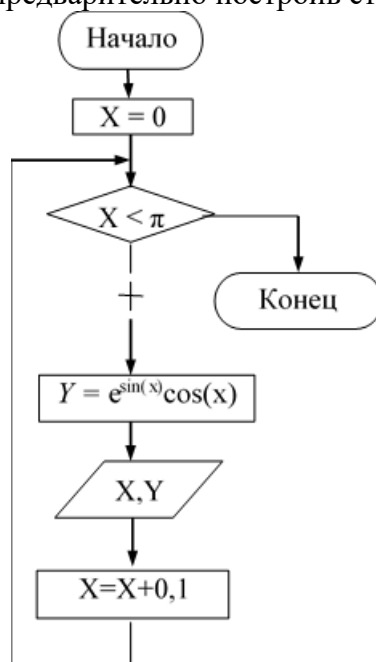
1. Построение математической модели.

Входные данные: начальное значение аргумента - 0, конечное значение аргумента - π , шаг изменения аргумента - 0,1.

Выходные данные: множество значений аргумента X и соответствующее им множество значений функции Y .

2. Составление структурной схемы.

В условии задачи количество повторений цикла явно не задано, но известно, как изменяется параметр цикла X и каковы его начальное и конечное значения. Поэтому решим эту задачу, используя цикл с предусловием, предварительно построив структурную схему:



3. Программный код.

```
program Example_11;
const pi=3.14;
var x,y:real;
begin
    { Присвоение начального значения параметру цикла. }
    x:=0;
    {Перед каждым выполнением тела цикла проверяется условие  $x \leq \pi$ ,
    если оно истинно, то тело цикла выполняется, иначе управление
    передается следующему за циклом оператору}
    while x<=pi do
        begin
            {Расчет значения функции для соответствующего значения
            аргумента X}
            y:=exp(sin(x))*cos(x);
            { Печать значений функции и аргумента. }
            writeln (x:4:2, ' ' \y:8:5);
        end
    end
```

```

                {Вычисление нового значения аргумента.}
                x:=x+0.1;
    end;           {Конец цикла while}
    readln;
end.

```

2 способ.

Теперь решим задачу другим способом, используя циклический оператор *for ... do*, предварительно определив количество повторений тела цикла *n*.

1. Построение математической модели.

Если параметр цикла *x* принимает значения в диапазоне от *x_n* до *x_k*, изменяясь с шагом *dx*, то количество повторений тела цикла можно определить по формуле:

$$n = \frac{x_k - x_n}{dx} + 1$$

округлив результат деления до целого числа.

2. Программный код.

```

program Example_12;
var x,y:real,-n,i:integer;
begin
    { Присвоение начального значения аргументу. }
    x:=0;
    {Расчет количества повторений цикла. Функция round округляет результат деления
до целого}
    n:=round((pi-0)/0.1)+1;
    for i:-1 to n do      { Тело цикла выполнится ровно n раз }
    begin
        {Расчет значения функции для соответствующего значения аргумента X . }
        Y:=exp (sin(x))*cos(x) ;
        {Печать значений функции и аргумента. }
        Writeln (x:4:2,' ',y:8:4);
        {Вычисление нового значения аргумента.}
        x:=x+0.1;
    { Конец цикла for ... do }
    end;
end.

```

Пример 2. Дано натуральное число *n*. Подсчитайте количество цифр данного числа.

Решение.

1. Построение математической модели.

Подсчет количества цифр начнем с последней цифры числа. Увеличим счетчик цифр на единицу. Число уменьшим в 10 раз (тем самым мы избавляемся от последней цифры числа). Далее с получившимся числом сделаем ту же последовательность действий и т.д., пока число не станет равным нулю.

2. Программный код.

```

Program Example 2;
var m,    n:    Longint;
    {счетчик цифр}
    k:    Integer;
begin
    {вводим целое число n}
    writeln('Введите целое число');
    readln(n);
    m:=n;
    k:=0;
    {пока число m<>0 делать }
    while m<>0 do

```



```

begin
  Inc(k); {или k:=k+1;}
  {«уменьшаем» число на последнюю цифру}
  m:=m Div 10;
end;
{вывод количества цифр}
writeln ('В числе ', n, ' - ', k, ' цифр. ');
readln;
end.

```

3. Тестирование программы.

Рассмотрим выполнение этой программы в пошаговом режиме для числа 65387.

В результате работы программы на экране появится предложение:

В числе 65387 – 5 цифр.

n	m	k
65387	65387	0
65387	6538	1
65387	653	2
65387	65	3
65387	6	4
65387	0	5

Пример 3. Вычислить факториал числа N ($N! = 1 \cdot 2 \cdot \dots \cdot N$).

1. Построение математической модели.

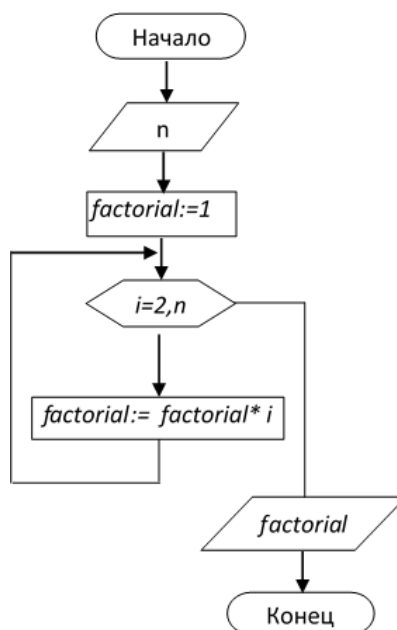
Входные данные: N - целое число, факториал которого необходимо вычислить.

Выходные данные: factorial - значение факториала числа N .

Промежуточные данные: i - целочисленная переменная, принимающая значения от 2 до N с шагом 1, параметр цикла.

Итак, вводится число N . Переменной factorial, предназначенной для хранения значения произведения последовательности чисел, присваивается начальное значение, равное единице. Затем организуется цикл, параметром которого выступает переменная i . Если значение параметра цикла меньше или равно N , то выполняется оператор тела цикла, в котором из участка памяти с именем factorial считывается предыдущее значение произведения, умножается на текущее значение параметра цикла, а результат снова помещается в участок памяти с именем factorial. Когда параметр i становится больше N , цикл заканчивается и на печать выводится значение переменной factorial, которая была вычислена в теле цикла.

2. Структурная схема:



3. Программный код.

```
Program Example_3;  
var  
    factorial, n, i: integer;  
begin  
    write('n='); readln (n);  
    {Стартовое значение.}  
    factorial:=1;  
    for i:=2 to n do  
        factorial:=factorial*i;  
    writeln (factorial);  
end.
```

Пример 4. Вычислить a^n ($n > 0$).

1. Построение математической модели.

Входные данные: a - вещественное число, которое необходимо возвести в целую положительную степень n .

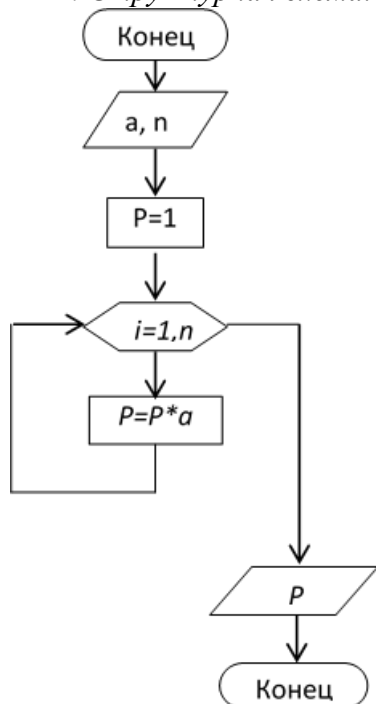
Выходные данные: p (вещественное число) - результат возведения вещественного числа a в целую положительную степень n .

Промежуточные данные: i - целочисленная переменная, принимающая значения от 1 до n с шагом 1, параметр цикла.

Известно, что для того чтобы получить целую степень n числа a , нужно умножить его само на себя n раз. Результат будет храниться в участке памяти с именем p . При выполнении очередного цикла из этого участка предыдущее значение будет считываться, умножаться на основание степени a и снова записываться в участок памяти p . Цикл выполняется n раз.

В таблице отображен протокол выполнения алгоритма при возведении числа 2 в пятую степень: $a = 2, n = 5$.

2. Структурная схема.



3. Программный код.

```
Program Example_4;  
var  
    a, p: real;
```

```

i, n: word;
begin
  write('Введите основание степени a=');
  readln(a);
  write('Введите показатель степени n=');
  readln(n);
  p:=1;
  for i:=1 to n do
    P:=P*a;
  writeln ('P=', P:1:3);
  readln;
end.

```

4. Тестирование программы.

i	1	2	3	4	5
P	2	4	8	16	32

Пример 5. Вычислить сумму натуральных четных чисел, не превышающих N.

1. Построение математической модели.

Входные данные: N - целое число.

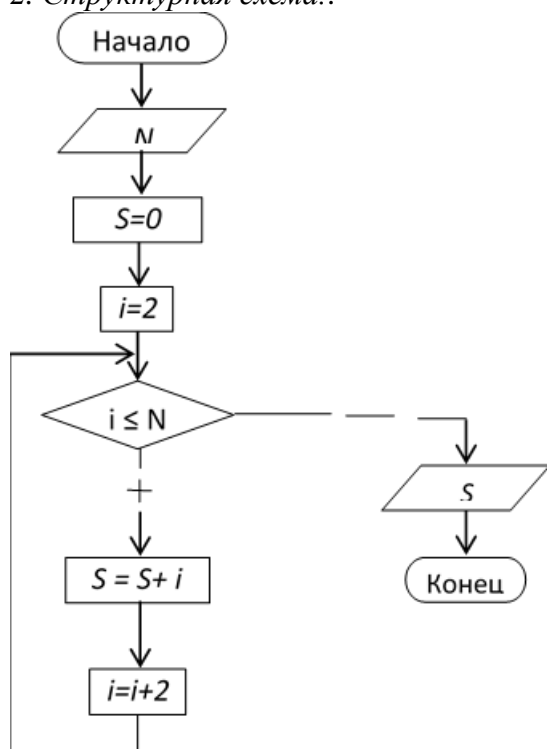
Выходные данные: S - сумма четных чисел.

Промежуточные данные: i - переменная, принимающая значения от 2 до N с шагом 2, следовательно, также имеет целочисленное значение.

При сложении нескольких чисел необходимо накапливать результат в определенном участке памяти, каждый раз считывая оттуда предыдущее значение суммы и прибавляя к нему следующее слагаемое. Для выполнения первого оператора накапливания суммы из участка памяти необходимо взять такое число, которое не влияло бы на результат сложения. Иными словами, перед началом цикла переменной, предназначенной для накапливания суммы, необходимо присвоить значение нуль.

Так как параметр цикла i изменяется с шагом 2, в блок-схеме использован цикл с предусловием, который реализуется при составлении программы с помощью оператора while ... do.

2. Структурная схема.:



3. Программный код.

```
Program Example 5;  
var n, i, S:word;  
begin  
  write( 'n=' ) ; readln (n);  
S:=0; i:=2;  
while i<=n do  
  begin  
    S:=S+i;  
    i:=i+2;  
  end;  
writeln ('S=', S) ;  
end.
```

Эту же задачу можно решить иначе, используя цикл for ... do:

```
Program Example_51;  
var n, i, S: word;  
begin  
write ('n=' ) ;  
  {Если остаток от деления параметра цикла на 2 равен 0, то это  
  число четное. Следовательно, происходит накопление суммы}  
  if i mod 2 = 0 then  
    S: =S+i ;  
writeln ( ` S=' , S);  
end.
```

4. Тестирование программы.

В таблице приведены результаты тестирования программы для $n = 7$. Несложно заметить, что при нечетных значениях параметра цикла значение переменной, предназначенной для накопления суммы, не изменяется.

i	1	2	3	4	5	6	7
s	0	2	2	6	6	12	12

Пример 6. Даны натуральные числа n и k . Составить программу вычисления выражения $1^k+2^k+\dots+n^k$.

1. Построение математической модели.

Для вычисления указанной суммы целесообразно организовать цикл с параметрами i , в котором, во-первых, вычислялось бы очередное значение $y=i^k$ и, во-вторых, осуществлялось бы накопление суммы прибавлением полученного слагаемого к сумме всех предшествующих ($s=s+y$).

2. Программный код.

```
program Example 6;  
var n, k, y, i, s, m: integer;  
begin  
  writeln ('Введите исходные данные n и k');  
  readln (n, k);  
  s:=0  
  for i:=1 to n do  
    begin  
      y:=1  
      {нахождение степени k числа i}  
      for m:=1 to k do y:= y*I;  
      s:=s+y;  
    end;  
  writeln ( ` Ответ: ` , s);  
end.
```

Таким образом, для решения задачи потребовалось организовать два цикла, один из которых пришлось поместить внутрь другого. Такие конструкции называют вложенными циклами.

Пример 7. Модифицировать предыдущую программу так, чтобы она вычисляла сумму $1^1+2^2+\dots+n^n$.

1. Построение математической модели.

Данная задача отличается от предыдущей тем, что показатель степени очередного слагаемого совпадает со значением ее основания, следовательно, параметры внутреннего цикла (цикла, в котором вычисляется очередное слагаемое) совпадают с параметрами внешнего цикла.

2. Программный код.

```

program Example 7;
var n, y, i, s, m: integer;
begin
    writeln ('Введите начальное значение n` ');
    readln (n) ;
    s:=0;
    for i:=1 to n do
        begin
            y:=1;
            {нахождение степени k числа i }
            for m:=1 to i do y:=y*i;
            s:=s+y;
        end;
    writeln ('Ответ: ',s);
end.

```

Внутренний и внешний циклы могут быть любыми из трех рассмотренных ранее видов: циклами с параметром, циклами с предусловием или циклами с постусловием. Правила организации как внешнего, так и внутреннего циклов такие же, как и для простого цикла каждого из этих видов. Но при использовании вложенных циклов необходимо соблюдать следующее условие: внутренний цикл должен полностью укладываться в циклическую часть внешнего цикла.

Пример 8. Старинная задача. Сколько можно купить быков, коров и телят, если плата за быка 10руб., за корову – 5руб., за теленка – полтинник (0,5руб.), если на 100 руб. надо купить 100 голов скота.

1. Построение математической модели.

Обозначим b – число быков; k — число коров; t – число телят.

После этого можно записать два уравнения:

$$10b + 5k + 0,5t = 100 \text{ и } b + k + t = 100.$$

Преобразуем их: $20b + 10k + t = 200$ и $b + k + t = 100$.

На 100 рублей можно купить:

- не более 10 быков, т. е. $0 < b < 10$;
- не более 20 коров, т.е. $0 < k < 20$;
- не более 200 телят, т.е. $0 < t < 200$.

2. Программный код.

```

program Example 8;
var b, k, t: integer;
begin
    for b:=0 to 10 do
        for k:=0 to 20 do
            for t:=0 to 200 do
                if (20*b+10*k+t=200) and (b+k+t=100) then
                    writeln (`быков `, b, ` коров`, k, ` телят `, t);
end.

```

Сколько раз будет проверяться условие в данной программе? Значение переменной b изменяется 11 раз (от 0 до 10), для каждого её значения переменная k изменяется 21 раз, а для каждого значения переменной k переменная t изменяется 201 раз. Таким образом, условие будет проверяться 11 21 201 раз. Но если известно число быков и коров, то число телят можно вычислить по формуле $t = 100 - (b + k)$ и цикл по переменной t исключается.

```

program Example 81;
var b, k, t: integer;
begin
    for b:=0 to 10 do
        for k:=0 to 20 do
            begin
                t:=100-(b+k);
                if (20*b+10*k+t=200) then
                    writeln (`быков `, b, ` коров`, k, ` телят `, t);
            end;
        end;
    end.

```

Пример 9. Дана непустая последовательность натуральных чисел, за которой следует 0. Составить программу поиска в данной непустой последовательности порядкового номера наименьшего элемента.

1. Построение математической модели.

Обозначим через x, i – очередной член последовательности и его номер; min, k - минимальный член последовательности и его номер. Считывание членов последовательности производится до тех пор, пока не будет наеден 0, то есть пока $x <> 0$. Начальное значение минимума определяется значением первого члена последовательности. Очередное вводимое число требуется сравнивать с текущим значением минимума, и если текущее значение min окажется больше очередного члена последовательности, то его надо изменить.

2. Программный код.

```

program Example_9;
var x, i, min, k: Longint;
begin
    writeln (`Введите первый член последовательности`);
    read (x); k:=1;
    min:=x; i:=2;
    while x<>0 do
        begin
            if x<min then
                begin
                    min:=x;
                    k:=i
                end;
            writeln (`Введите`, i, ` элемент последовательности`);
            read (x);
            inc (i);
        end;
    writeln (` Номер минимального элемента - `, k);
end.

```

Пример 10. Составить программу планирования закупки товара в магазине на сумму, не превышающую заданной величины.

1. Построение математической модели.

Обозначим через x, k соответственно цену и количество товара, через p – заданную предельную сумму, через s – общую стоимость покупки. Начальное значение общей стоимости покупки (s) равно нулю. Значение предельной суммы донесся с клавиатуры. Необходимо повторять запрос цены и количества выбранного товара, вычислять его стоимость, суммировать ее с общей стоимостью и выводить результат на экран до тех пор, пока она не превысит предельную сумму p . В этом случае на экран надо вывести сообщение о превышении.

2. Программный код.

```

program Example_10;
var x, k, p, s: integer;

```

```

begin
  writeln ( `Предельная сумма -`); readln (p);
  s:=0;
  repeat
    writeln ('Введите цену товара и его количество`);
    readln (x ,k);
    s:=s+x*k;
    writeln ('Стоимость покупки равна ', s);
  until s>p;
  writeln ( 'Суммарная стоимость покупки превысила предельную
сумму`);
end.

```

Пример 11. Дано натуральное число N . Определить K - количество делителей этого числа, не превышающих его. (Например, $N= 12$, его делители 1, 2, 3, 4, 6, $K = 5$).

1. Построение математической модели.

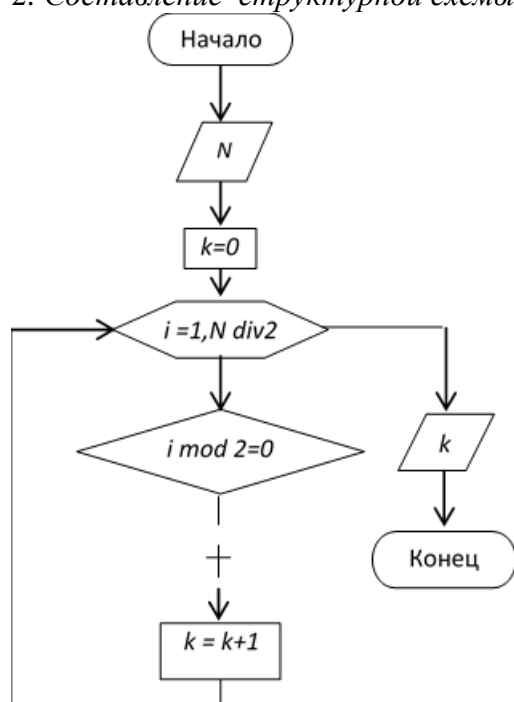
Входные данные: N - целое число.

Выходные данные: целое число K - количество делителей N .

Промежуточные данные: i - параметр цикла, возможные делители числа N .

В блок-схеме реализован следующий алгоритм: в переменную K , предназначенную для подсчета количества делителей заданного числа, помещается значение, которое не влияло бы на результат, то есть нуль. Далее организовывается цикл, в котором изменяющийся параметр i выполняет роль возможных делителей числа N . Если заданное число делится нацело на параметр цикла, это означает, что i является делителем N , и значение переменной K следует увеличить на единицу. Цикл необходимо повторить $N/2$ раз.

2. Составление структурной схемы.



В таблице отображены результаты тестирования алгоритма при определении делителей числа $N = 12$.

i		1	2	3	4	5	6
K	0	1	2	3	4	4	5

2. Программный код.

```

var N, i, K:word;
begin

```

```

write ('N=');
readln (N);
{ Присвоение начального значения. }
K:=0;
for i:=1 to N div 2 do
  { Если N делится нацело на i, то }
  if N mod i= 0 then
    { увеличить счетчик на единицу. }
    k:=K+1;
writeln (' K= ', K);
end.

```

Пример 12. Дано натуральное число N . Определить, является ли оно простым. Натуральное число N называется простым, если оно делится нацело без остатка только на единицу и N . Число 13 - простое, так как делится только на 1 и 13, $N=12$ не является простым, так как делится на 1, 2, 3, 4, 6 и 12.

1. Построение математической модели.

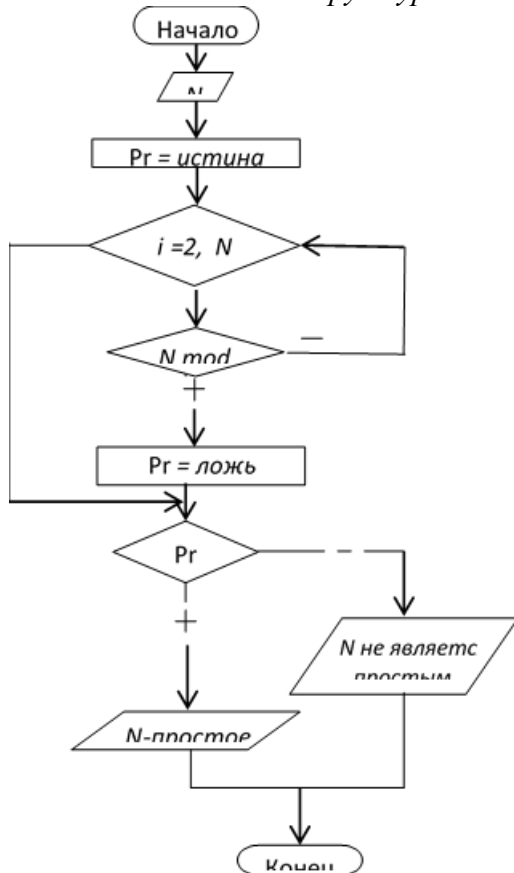
Входные данные: N - целое число.

Выходные данные: сообщение.

Промежуточные данные: i - параметр цикла, возможные делители числа N .

Алгоритм решения этой задачи заключается в том, что число N делится на параметр цикла i , изменяющийся в диапазоне от 2 до $N/2$. Если среди значений параметра не найдется ни одного числа, делящего заданное число нацело, то N - простое число, иначе оно таковым не является. Обратите внимание на то, что в алгоритме предусмотрено два выхода из цикла. Первый - естественный, при исчерпании всех значений параметра, а второй - досрочный. Нет смысла продолжать цикл, если будет найден хотя бы один делитель из указанной области изменения параметра.

2. Составление структурной схемы.



3. Программный код.

При составлении программы на языке Турбо Паскаль досрочный выход из цикла удобно выполнять при помощи оператора *break*:

```
var N, i: integer;
Pr :boolean;
begin
  write ('N='),
  readln (N);
  Pr:=true;           { Предположим, что число простое. }
  for i:=2 to N div 2 do
    {Если найдется хотя бы один делитель, то}
    if N mod i = 0 then
      begin
        Pr:=false;    { число простым не является, и }
        break;       { досрочный выход из цикла.   }
      end;
    {Проверка значения логического параметра и вывод на печать
    соответствующего сообщения.}
    if Pr then
      writeln ('Число ',N,' - простое')
    else
      writeln ( 'Число ',N,' простым не является') ;
end.
```

Пример 3. Дано натуральное число N . Определить самую большую цифру и ее позицию в числе ($N= 573863$, наибольшей является цифра 8, ее позиция - четвертая слева).

1. Построение математической модели.

Входные данные: N - целое число.

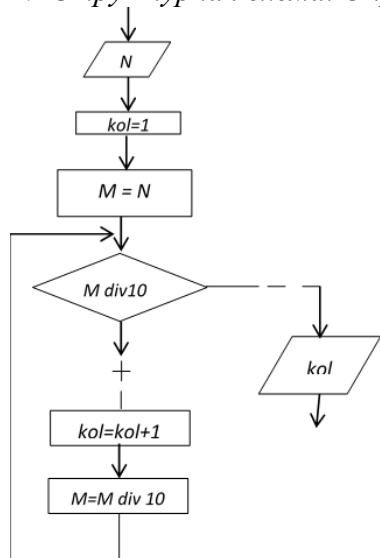
Выходные данные: max - значение наибольшей цифры в числе, pos - позиция этой цифры в числе.

Промежуточные Данные: i - параметр цикла, kol - количество цифр в числе, M - переменная для временного хранения значения N .

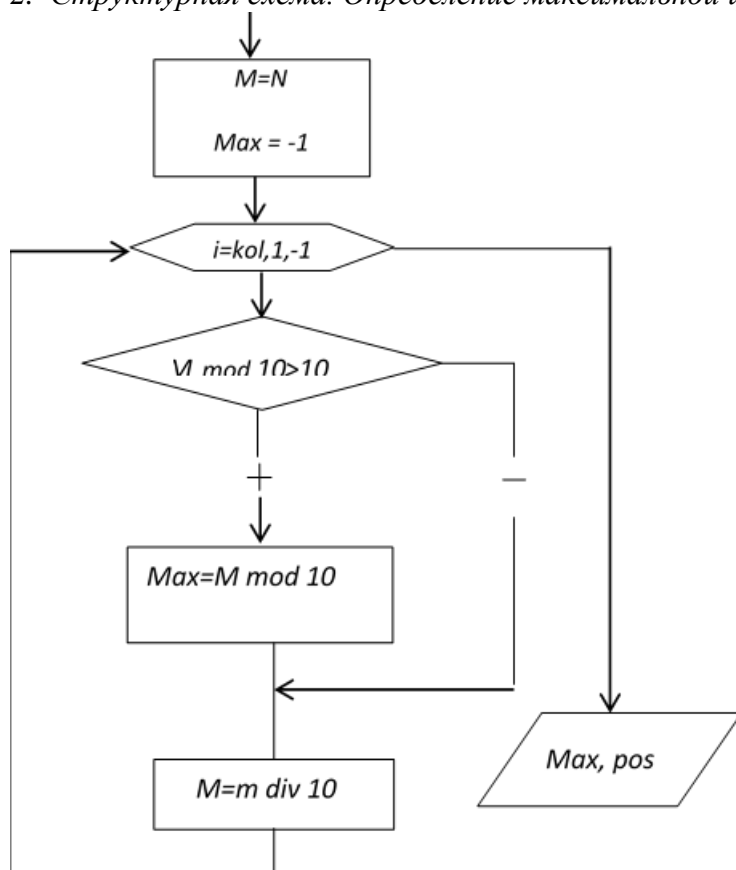
Разобьем решение этой задачи на два этапа. Вначале найдем количество цифр в заданном числе (рис. 1), а затем определим наибольшую цифру и ее позицию (рис. 2).

Для того чтобы подсчитать количество цифр в числе, необходимо определить, сколько раз заданное число можно разделить на десять нацело. Например, пусть $N = 12345$, тогда количество цифр $kol = 5$.

1. Структурная схема. Определение количества цифр в числе



2. Структурная схема. Определение максимальной цифры и её позиции в цифре.



Результаты вычислений приведены в таблицах:

Определение количества цифр числа

Kol	N
1	12345
2	12345 div 10 = 1234
3	1234 div 10 = 123
4	123 div 10 = 12
5	12 div 10 = 1
	1 div 10 = 0

Определение текущей цифры числа

i	Число M
1	12345
2	12345 div 10 = 1234
3	1234 div 10 = 123
4	123 div 10 = 12
5	12 div 10 = 1

Если цифры числа известны, определить наибольшую из них не составит труда. Алгоритм поиска максимального значения в некоторой последовательности цифр заключается в следующем. В ячейку, в которой будет храниться максимальный элемент (max), записывают значение, меньшее любого из элементов последовательности (в нашем случае max = -1, так как цифры числа находятся в диапазоне от 0 до 9). Затем сравнивают элементы последовательности с значением ячейки max, если найдется элемент, превышающий значение предполагаемого максимума, то ячейке max необходимо присвоить значение этого элемента и, соответственно, запомнить его номер в последовательности (в нашем случае переменной pos присваивается значение параметра цикла). В алгоритме поиска минимума вначале в переменную min записываем значение, заранее большее любого элемента последовательности. Затем все

элементы последовательности сравниваем с `min`. Если встретится значение меньшее, чем `min`, переписываем его в переменную `min`.

3. Программный код.

```

var M,N,max:longint;
      i,kol,pos:word;
begin
  { Так как речь идет о натуральных числах, }
  { при вводе предусмотрена проверка. }
  { Закончить цикл, если введено положительное число, }
  { иначе повторить ввод. }
  repeat
    write ('N=' ) ; readln (N);
  until N>0;
  M:=N; { Сохранить значение переменной N. }
  { Предположим, что число состоит из одной цифры. }
  kol:=1;
  {Выполнять тело цикла, пока число делится нацело на 10. }
  while M div 10 > 0 do
  begin
    kol:=kol+1; { Счетчик количества цифр. }
    M:=M div 10; { Изменение числа. }
  end;
  max:=-1; { Присвоение максимуму начального значения. }
  pos:=1; { Предполагаемая позиция максимума в числе. }
}
M:=N;
For i:=kol downto 1 do { Нумерация цифр в числе слева}
begin
  { Если цифра больше предполагаемого максимума, то }
  if M mod 10 > max then
  begin
    { заменить максимум этой цифрой }
    max:=M mod 10;
    {запомнить 'ее позицию в числе}
    pos:=i;
  end;
  {изменение числа. }
  M:=M div 10;
end;
writeln ( 'max=',max,' pos=',pos);
end.

```

Задание для практической работы

Выполнение практических заданий.

Вариант	№ задач	Вариант	№ задач	Вариант	№ задач
1	1, 7, 37, 47, 71, 81	11	5, 17, 47, 57, 71, 91	21	3, 27, 46, 67, 71, 111
2	2, 8, 38, 48, 72, 82	12	6, 18, 37, 58, 72, 92	22	4, 28, 37, 68, 72, 112
3	3, 9, 39, 49, 73, 83	13	1, 19, 38, 59, 73, 93	23	5, 29, 38, 69, 73, 113
4	4, 10, 40, 50, 74, 84	14	2, 20, 39, 60, 74, 94	24	6, 30, 39, 70, 74, 85
5	5, 11, 41, 51, 75, 85	15	3, 21, 40, 61, 75, 95	25	1, 31, 40, 47, 75, 86

6	6, 12, 42, 52, 76, 86	16	4, 22, 41, 62, 76, 96	26	2, 32, 41, 48, 76, 87
7	1, 13, 43, 53, 77, 87	17	5, 23, 42, 63, 77, 97	27	3, 33, 42, 49, 77, 88
8	2, 14, 44, 54, 78, 88	18	6, 24, 43, 64, 78, 98	28	4, 34, 43, 50, 78, 100
9	3, 15, 45, 55, 79, 89	19	1, 25, 44, 65, 79, 99	29	5, 35, 44, 51, 79, 105
10	4, 16, 46, 56, 80, 90	20	2, 26, 45, 66, 80, 100	30	6, 36, 45, 52, 80, 113

1. Используя операторы цикла, составить таблицу значений функции (протабулировать функцию):

$$y = \sqrt{\operatorname{tg}^2 \frac{x}{5} + 2x^2 + 3}, \text{ если } 2 \leq x \leq 7, \Delta x = 0,56$$

где Δx – шаг, с которым изменяется аргумент x .

$$2. y = \sqrt{\sin^2 \frac{2\pi x}{5} + 6x^2 + |x|}, \text{ если } 3 \leq x \leq 9, \Delta x = 0,74$$

$$3. y = \sqrt{\cos^2 \frac{28^\circ}{5x} + 8x^2}, \text{ если } 1 \leq x \leq 5, \Delta x = 0,46.$$

$$4. y = \sqrt{\frac{\sin 12^\circ + 3x^2}{\operatorname{tg} \frac{\pi}{7}}}, \text{ если } 1 \leq x \leq 4, \Delta x = 0,35.$$

$$5. y = \sqrt{\frac{\cos 22^\circ + 3x^2}{\operatorname{tg} \frac{\pi}{7} + 7x}}, \text{ если } 2 \leq x \leq 6, \Delta x = 0,34.$$

$$6. y = \sqrt{\frac{3x^2 + \sin 20^\circ}{5x}}, \text{ если } 1 \leq x \leq 6, \Delta x = 0,42.$$

7. Вычислить квадратный корень $y = \sqrt{x}$ по рекуррентной формуле Герона

$$y = \frac{1}{2} \left(y + \frac{x}{y} \right) \text{ с точностью } \xi = 10^{-6}.$$

8. Составить схему алгоритма вычисления значения полинома

$$P = 11x^5 + 9x^4 + 7x^3 + 5x^2 + 3x + 1 \text{ при } x = 1.25$$

9. Подсчитать количество цифр в целом числе X .

10. Найти минимальное значение функции $Y = x \sin x$, на отрезке $[c, d]$ с шагом $\nabla x = 0.001$.

11. Дано натуральное число n . Вычислить $a(a+1)(a+2)\dots(a+n-1)$.

12. Дано натуральное число n . Вычислить $(1 + \frac{1}{a})(1 + \frac{1}{2a})\dots(1 + \frac{1}{na})$

13. Дано натуральное число n . Вычислить $a(a-n)(a-2n)\dots(a-nn)$

14. Дано натуральное число x . Подсчитать, сколько цифр в этом числе и чему равна их сумма.

15. Дано натуральное число x . Найти первую цифру этого числа.

16. Даны натуральные числа m, n . Получить сумму m последних цифр числа n .

17. Дано 50 вещественных чисел. Найти величину наибольшего из них.

18. Дана непустая последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.

19. Даны целое $n > 0$ и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти величину наибольшего среди отрицательных чисел этой последовательности.

20. Дано n вещественных чисел. Вычислить разность между максимальным и минимальным из них.

21. Составить программу, определяющую, является ли данное число простым.

22. Найти сумму первых чисел Фибоначчи, не превышающих числа a .

23. Дано n вещественных чисел. Определить, образуют ли они возрастающую последовательность.

Из истории. Грекам были известны первые четыре совершенных числа: 6, 28, 496, 8128. Эти числа высоко ценились. Даже в 12 веке церковь утверждала, что для спасения души необходимо найти пятое совершенное число. Это число было найдено только 15 веке. До сих пор совершенные числа полностью не исследованы – не известно, имеется ли конечное число совершенных чисел или их число бесконечно, кроме того, не известно ни одного нечетного числа, но и не доказано, что таких чисел нет.

Контрольные вопросы

1. Какие циклы называются условными?
2. Дайте определение понятию «итерация».
3. Какой вид цикла можно использовать, если заранее не известно число итераций?
4. При использовании какого вида цикла, тело цикла может не вычисляться не разу?
5. При использовании какого вида цикла, тело цикла обязательно вычисляется хотя бы один раз?
6. Напишите программу вычисления $N!$
7. Напишите программу вычисления a^n .

Практическая работа № 35-36. Составление программ с массивами

Цель: развить навыки обработки одномерных массивов; развить навыки обработки двумерных массивов; развить навыки тестирования программных кодов.

Теоретический материал

Пример 1.

Рассмотрим алгоритм пузырьковой сортировки.

1. Построение математической модели.

Сравним первый элемент массива со вторым. Если первый окажется больше второго, то поменяем их местами. Те же действия выполним для второго и третьего, третьего и четвертого, I-го и (I + 1)-го, (n - 1)-го и n-го элементов. В результате этих действий самый большой элемент станет на последнее (n-е) место. Теперь повторим данный алгоритм сначала, но последний (n-й) элемент, рассматривать не будем, так как он уже занял свое место. После проведения данной операции самый большой элемент оставшегося массива встанет на (n - 1)-е место. Так повторяем до тех пор, пока не упорядочим весь массив.

Обратите внимание на то, что для перестановки элементов (блок 4) используется буферная переменная b, в которой временно хранится значение элемента, подлежащего замене.

Процесс упорядочивания элементов в массиве по возрастанию представим в таблице:

	Номер элемента				
	1	2	3	4	5
Исходный массив	7	3	5	4	2
Первый просмотр	3	5	4	2	7
Второй просмотр	3	4	2	5	7
Третий просмотр	3	2	4	5	7
Четвертый просмотр	2	3	4	5	7

2. Программный код.

Листинг программы упорядочивания элементов в массиве по возрастанию их значений.

```
var
  i, n, j: integer;
  b: word;
  y: array [1 . . 100] of word;
begin
  writeln ( 'введите размер массива' );
  readln (n);
  for i :=1 to n do
  begin
    write ('y[' , i , ']=');
    readln (y[i]);
  end;
  writeln ('массив y');
  for i:=1 to n do write (y [ i ], ' ');
  writeln;
  repeat
    f:=0;
    { Если текущий элемент больше следующего, то }
    if y[ i ] > y [ i + 1 ] then
      { поменять их местами. }
  begin
```

```

        { Сохранит значение текущего элемента. }
        b:=y[ i ] ;
        { Заменить текущий элемент следующим. }
        y[ i ]:=y[i + 1] ;
        { Заменить следующий элемент текущим. }
        y[i + 1]:=b ;
        f:=1;
    end;
    writeln ( 'упорядоченный массив' ) ;
    for i :=1 to n do
        write (y[ i ], ' ' ) ;
    writeln;
end.

```

Пример 2. Разработать программу генерации следующих комбинаций: 1111, 1112, 1113, 1121, 1122, 1123, 1131, 1132, 1133, 1211,..., 3333.

Вариант 1. Для генерации комбинаций используем вложенные циклы. Количество разрядов - 4. Следовательно, для генерации всех комбинаций понадобится 4 цикла, вложенных один в другой (рис. а). Если ввести параметр m - максимальное значение в каждом разряде, то эта же программа будет генерировать комбинации от 1111 до mmmm.

Если количество разрядов генерируемой комбинации изменить, то придется переписывать программу, увеличивая или уменьшая количество вложенных циклов.

Вариант 2. С точки зрения увеличения универсальности программы для генерации всех комбинаций лучше использовать программно реализуемый счетчик в общем случае на n разрядов, который в каждом разряде считает от 1 до n. Каждое состояние такого счетчика соответствует комбинации. Для генерации следующего варианта в младший разряд счетчика добавляют 1 и осуществляют межразрядные переносы в соответствии с характеристиками каждого разряда.

Ниже представлена программа, реализующая данный алгоритм для $n < 10$.

```

Program ex2;
const
    a:array[1..10] of byte=(1, 1,1,1,1,1,1,1,1,1);
var i,m,n:integer;
begin
    readLn (n,m);
    while a[1]<m+1 do {условие «сброса» счетчика}
        begin
            for i:=1 to n do write (a[i]); {вывод комбинации}
            write(' ');
            a[n]:=a[n]+1; {добавление 1 в последний разряд}
            for i:=n downto 2 do {анализ и осуществление переносов}
                if a[i]>m then
                    begin
                        a[i]:=1;
                        a[i-1]:=a[i-1]+1;
                    end;
            end;
        end;
end.

```

Ассоциируя комбинации счетчика с вариантами данных и проверяя полученные комбинации по критериям конкретной задачи, можно осуществить полный перебор вариантов.

Как уже упоминалось выше, для реализации ограниченного перебора используется вторая стратегия, при которой генерацию и анализ комбинаций исходных данных выполняют поэтапно. С этой целью удобно использовать рекурсию, при которой процесс генерации комбинации

продолжается, пока полученная часть комбинации перспективна, или не исчерпаны все варианты.

Для генерации всех комбинаций исходных данных можно использовать вложенные циклы или программно реализуемый счетчик (по типу, например, электросчетчика). Второй способ позволяет получить более общее решение.

Ограниченный перебор, реализованный с использованием рекурсии. Для генерации комбинаций будем использовать древовидную рекурсию, на каждом уровне которой ставится один ферзь m различными способами. При этом появляется возможность проверки перспективности уже имеющейся комбинации. Поясним сказанное на примере доски 4×4 .

На первом шаге первого ферзя можно поставить на одно из четырех полей первого вертикального ряда, что соответствует комбинациям

1..., 2..., 3..., 4....

На втором шаге второго ферзя можно поставить на одно из четырех полей второго вертикального ряда. При этом мы получим 4 комбинации для каждого варианта установки первого ферзя:

11., 12., 13., 14.. 21., 22., 23., 24.. 31., 32., 33., 34.. 41., 42., 43., 44..

На этом этапе уже можно исключить все варианты, для которых первые два ферзя бьют друг друга: 11., 12., 21., 22., 23., 32., 33., 34., 43., 44.. Генерацию остальных комбинаций необходимо продолжить, установив третьего ферзя одним из четырех способов и исключив неперспективные решения. На последнем этапе необходимо установить четвертого ферзя и опять исключить варианты, в которых ферзи бьют друг друга. Оставшиеся варианты являются решениями задачи.

Program ex21;

Type p=array[1 ..100] of integer;

Var

Pole: p;

k,m: integer;

{функция проверки перспективности комбинации}

function new_r(n:integer;pole:p):boolean;

var j: integer;

begin

new_r:=false;

for j:=1 to n-1 do

if (pole[j]=pole[n]) or (abs(pole[j]-pole[n])=n-j) then

exit;

new_r:=true;

End;

{рекурсивная функция генерации комбинации}

procedure ferz (n,m:integer; var pole.p);

var i:integer;

begin

if n=m+1 then {если установлено m ферзей, то вывести решение}

begin

for i:=1 to m do write(pole[i]:2);

writeln;

end

else {иначе - пытаемся установить следующего ферзя}

for i:=1 to m do { m способами}

begin

pole[n]:=i; {установка n -го ферзя}


```

                if new_r(n,pole)                {проверка
перспективности комбинации}
                thenferz (n+1,m,pole); {рекурсивный вызов установки
следующего ферзя}
                end;
end;
{основная программа}
begin
writeln ('Введите размер доски:');
readln (m);
к:=0;
ferz (1,m,pole); end.

```

Размер доски	Количество вариантов, которое проверяется при полном переборе	Количество вариантов, рассмотренных при ограниченном переборе	Количество полученных решений
4x4	$4^4 = 256$	17	2
5x5	$5^5 = 3125$	54	10
6x6	$6^6 = 46656$	153	4
7x7	$7^7 = 823543$	552	40
8x8	$8^8 = 16777216$	2057	92

Пример 3. Переписать элементы массива x в обратном порядке. Блок-схема Представлена на рис.4.7. Алгоритм состоит в следующем: меняем местами 1-й и n-й элемент, затем 2-й и (n-1)-й и т.д. середины массива (элемент с номером I следует обменять с элементом (n+1-i)).

```

program mas_5;

type
massiv=array[1..100] of real;
var
x:massiv;
i, n : integer;
b : real;
begin
writeln ('введите размер массива');
readln(n);
for i:=1 to n do
begin
write ('x[',i,']=');
readln (x [ i ] );
end;
{Элементы массива меняются местами : 1- й - с n-м,2-й- с (n-1) и т.д.}
for i :=1 to n div 2 do
begin
b:=x[n+1-i];
x[n+1-i ]:=x[i];
x[i]:=b;
end;
writeln ( ' преобразованный массив');
for i :=1 to n do
write ( x [i]:1:2,');
end.

```

Пример 4. Найти сумму элементов матрицы, лежащих выше главной диагонали.

Алгоритм решения главной задачи построен следующим образом: обнуляется ячейка для накопления суммы (переменная s). Затем с помощью двух циклов (первый по строкам, второй по столбцам) просматривается каждый элемент матрицы, но суммирование происходит только в том случае, если этот элемент находится выше главной диагонали, то есть выполняется свойство $i < j$.

```
program ex6;
var
a:array [1..15,1..10] of real;
i,j,n,m: integer;
s: real;
begin
writeln (' введите размеры матрицы');
writeln ('n-количество строк, m-количество столбцов');
readln (n,m);
for i:=1 to n do
for j:=1 to n do
begin
write ('a[',i, ', ',j, ']=');
readln (a[i,j]);
end;
s:=0;
for i:=1 to n do
for j:=1 to n do
    if j>i then {Если элемент лежит выше главной диагонали, то }
        s:=s+a [i,j]; {наращиваем сумму. }
writeln ('матрица A');
for i:=1 to m do
    {Здесь важен формат, особенно общая ширина поля! }
write (a[i, j] :8:3, ' ');
writeln
end;
writeln ('сумма элементов матрицы' , s:8:3);
end.
```

Пример 5. Преобразовать матрицу A (m,n) так, чтобы строки с нечетными номерами были упорядочены по убыванию, с четными – по возрастанию.

```
program ex7;
var
    a: array [1...15,1..15] of real;
    j, i. k. m, n: byte;
    b: real;
begin
    writeln ('введите m и n');
    readln (m,n);
    for i:=1 to m do
        for j:=1 to n do
            begin
                write ('a[',i,',',j,'] =');
            readln (a[i,j]);
            end;
    writeln ('матрица a');
    for i:=1 to m do
```

```

begin
  for j:=1 to n do
    write (a[I,j]:7:3,' ');
    writeln;
  end;
  for i:=1 to m do
    if (I mod 2)=0 then {если номер строки четный, то упорядочить
ее элементы по
возрастанию.}
    begin
      for k:1 to n-1 do
        for j:1 to n-k do
          if a[i,j] >a[i,j+1] then
            begin
              b:=a[i,j];
              a [i,j]:=a [i,j+1];
              a[I,j+1]:=b;
            end;
          end;
        else {если номер строки нечетный, то упорядочить ее
элементы по убыванию.}
        for k:=1 to n-1 do
          for j:=1 to n-k do
            if a[I;j]< a[i,j+1] then
              begin
                b:=a[I,j];
                a[i,j]:=a [i,j+1];
                a[I,j+1]:=b;
              end;
            writeln ('преобразованная матрица a);
            for i:=1 to m do
              begin
                for j:=1 to m do
                  write (a [I,j]:7:3,' ');
                writeln;
              end;
            end.

```

Пример 6. Написать программу умножения двух матриц $A(n, m)$ и $B(m, l)$. Например, необходимо перемножить две матрицы:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}.$$

Вспользуемся правилом «строка на столбец», получим матрицу

$$\begin{pmatrix} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} & a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} & a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} \\ a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31} & a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix}.$$

В общем виде формула для нахождения элемента C_{ij} матрицы выглядит следующим образом:

$$C_{ij} = \sum_{k=1}^m A_{ik} B_{kj},$$

где $i = 1, N$ и $j = 1, L$. Обратите внимание, что проводить операцию умножения можно только в том случае, если количество строк левой матрицы совпадает с количеством столбцов правой. Кроме того, $A \times B \neq B \times A$.

```

var
    a,b,c : matrica;
    i,j,m,n,l,k : bute;
    s : real;
begin
    writeln ('Введите n,m,l');
    readln (n,m,l);
    for i:=1 to n do
        begin
            write ('a[', i, ', ', j, ']=');
            readln (a[i, j]);
            end;
        for i:=1 to m do
            for j:=1 to l do
                begin
                    write ('b[', i, ', ', j, ']=');
                    readln (b[i,j]);
                    end;
            for i:=1 to n do
                for j:=1 to l do
                    begin
                        {В переменной S будет храниться результат скалярного произведения
i-й строки на j-й столбец.}
                        S:=0;
                        for k:=1 to m do s:=s+a[i,k]*b[k,j];
                        c[i,j]:=s;
                        end;
                        writeln ('Матрица a');
                    for i:=1 to n do
begin
                        for j:=1 to m do writeln (a[i, j]:7:3, ' ');
                        writeln;
                    end;
                    writeln ('Матрица b');
                    for i:=1 to m do
begin
                        for j:=1 to l do write (b[i, j]:7:3, ' ');
                        writeln;
                    end;
                    writeln ('Матрица c=a*b');
                    for i:=1 to n do
begin
                        for j:=1 to l do write (c[i,j]:7:3, ' ');
                        writeln;
                    end;
                end;
            end.

```

Пример 7. Поменять местами n -й и 1-й столбцы матрицы $A(k,m)$.

```
program ex9;  
type  
matrica = array [1..15,1..15] of real;  
var  
a : matrica;  
i, j, k, m, n, l :byte;  
b : real;  
begin  
  write ('k=');  
  readln (k);  
  write ('m=');  
  readln (m);  
  for i:=1 to k do  
  for j:=1 to m do  
  begin  
    write ('a['i','j']=');  
    readln (a[i,j])  
  end;  
  repeat  
    write ('n=');  
    readln (n);  
    write ('l=');  
    readln (l);  
  until (n<=m) and (l<=m) and (n<=1); {Ввод считается верным, если n и  
  l меньше m и не верны друг другу.}  
  for I :=1 to k do{Элементы столбца с номером l заменить элементами  
  столбца с номером n.}  
    begin  
      b :=a[i,n];  
      a[i,n]:= a[i,l];  
      a[i,l]:=b  
    end;  
  for I :=1 to k do  
  begin  
    for j:=1 to m do write ('a['i','j']:7:3,' ');  
  writeln;  
    end;  
  end.
```

Пример 8. Преобразовать матрицу $A(m,n)$ таким образом, чтобы каждый столбец был упорядочен по убыванию. Алгоритм решения этой задачи сводится к тому, что уже известный нам по предыдущей главе алгоритм упорядочивания элементов в массиве выполняется для каждого столбца матрицы.

```
program ex10;  
type  
  matrica = array [1..15,1..15] of real;  
var  
  a : matrica ;  
  I,j,k,m,n : byte ;  
  b: real;  
begin
```

```

write ( 'm=' ) ;
readln (m) ;
write ( 'n=' ) ;
readln (n);
for i:=1 to m do
  for j :=1 to n do
    begin
      write ( 'a[',i,',',j,'] =' ) ;
      readln (a[I,j]) ;
    end;
writeln ( 'матрица a' );
for i:=1 to m do
  begin
    for j:=1 to n do write (a[I, j]:7:3, ' ');
    writeln;
  end;
for j:=1 to n do      {j- номер столбца.}
  for k:=1 to m-1 do {k- номер просмотра.}
    for i:=1 to m-k do {I - номер строки .}
      if a [I,j] <a [i+1,j] then
        begin
          b:= [a , j];
          a [I,j] :=a[i+1,j];
          a [i+1,j] :=b;
        end;
      writeln ( ' преобразованная матрица a' );
for i:= to n do
  begin
    for j:= to n do write (a[I,j] :7:3, ' ');
    writeln;
  end;
end.

```

Задание практической работы

Вариант	№ задач	Вариант	№ задач	Вариант	№ задач
1	1, 19, 37, 62, 82,100, 120	11	11, 29, 47, 72, 92, 110, 130	21	3, 21, 57, 62, 99, 105, 121
2	2, 20, 38, 63, 83, 101, 121	12	12, 30, 48, 73, 93, 111, 131	22	4, 22, 58, 63, 82, 106, 122
3	3, 21, 39, 64, 84, 102, 122	13	13, 31, 49, 74, 94, 112, 132	23	5, 23, 59, 64, 83, 107, 123
4	4, 22, 40, 65, 85, 103, 123	14	14, 32, 50, 75, 95, 113, 133	24	6, 24, 60, 65, 84, 108, 124
5	5, 23, 41, 66, 86, 104, 124	15	15, 33, 51, 76, 96, 114, 134	25	14, 25, 61, 66, 85, 109, 125
6	6, 24, 42, 67, 87, 105,125	16	16, 34, 52, 77, 97, 115, 135	26	15, 26, 62, 67, 86, 110, 126
7	7, 25, 43, 68, 88, 106, 126	17	17, 35, 53, 78, 98, 116, 136	27	16, 27, 61, 68, 87, 111, 127
8	8, 26, 44, 69, 89, 107, 127	18	18, 36, 54, 79, 99, 117, 137	28	17, 28, 40, 69, 88, 112, 128
9	9, 27, 45, 70, 90, 108, 128	19	1, 19, 55, 80, 86, 118, 138	29	18, 29, 39, 70, 89, 113, 129
10	10, 28, 46, 71, 91, 109, 129	20	2, 20, 56, 81, 93, 119, 120	30	10, 30, 46, 71, 90, 114, 130

1. Подсчитать и вывести на экран количество элементов массива, равных заданному значению. Заданное значение вводится с клавиатуры.
2. Составить и вывести на экран новый массив с номерами элементов исходного массива, которые равны заданному значению. Заданное значение вводится с клавиатуры.
3. Составить и вывести на экран новый массив со значениями элементов исходного массива, которые не равны заданному значению. Заданное значение вводится с клавиатуры.
4. Поменять местами максимальный и x минимальный элементы массива. Вывести измененный массив на экран.
5. Подсчитать и вывести на экран количество отрицательных, положительных и нулевых элементов массива.
6. Все элементы массива, меньшие заданного значения, и их номера записать в новые массивы. Вывести новые массивы на экран. Заданное значение вводится с клавиатуры.
7. Все элементы массива, большие заданного значения, и их номера записать в новые массивы. Вывести новые массивы на экран. Заданное значение вводится с клавиатуры.
8. Подсчитать и вывести на экран процент нулевых элементов в исходном массиве.
9. Определить и выдать сообщение на экран, какая сумма по абсолютной величине больше, отрицательных или положительных элементов вектора.
10. Вводится дополнительный массив разрешенных значений. Определить и вывести на экран, сколько элементов исходного массива имеют неразрешенные значения.
11. Вводится дополнительный массив разрешенных значений. Определить и вывести на экран, сколько элементов исходного массива имеют разрешенные значения.
12. Вводится дополнительный массив разрешенных значений. Составить массив из элементов исходного массива, имеющих неразрешенные значения. Вывести результативный массив на экран.
13. Составить и вывести на экран массив с M минимальными значениями исходного массива. M вводится дополнительно.
14. Составить и вывести на экран массив с M максимальными значениями исходного массива. M вводится дополнительно.
15. Переписать элементы массива в обратном порядке на том же месте. Вывести измененный массив на экран.
16. Составить и вывести на экран массив из различных элементов исходного массива.
17. Составить и вывести на экран массив номеров элементов исходного вектора, встречающихся один раз.
18. Составить и вывести на экран два массива: массив повторяющихся элементов исходного массива и массив их частот.
19. Ввести размер прямоугольной матрицы и значения ее элементов. Построить вектор V , где $V(i)$ - число неотрицательных элементов в i -ой строке.
 19. число нулевых элементов в i -ом столбце.
 20. среднее арифметическое отрицательных элементов в i -ой строке.
 21. среднее арифметическое положительных элементов в i -ом столбце.
 22. минимальное значение в i -ой строке.
 23. максимальное значение в i -ом столбце.

Контрольные вопросы

1. Какими параметрами характеризуется массив?
2. Чем отличается динамический массив от статического?
3. Опишите двумерный массив – константу размерностью 2×4 .
4. Какие методы сортировки элементов массива Вам известны?
5. Опишите алгоритм сортировки массива методом «пузырька».
6. Опишите алгоритм сортировки массива методом выбора.
7. Составьте программу нахождения минимального элемента одномерного массива.
8. Составьте программу нахождения суммы положительных элементов двумерного массива.

9. С переменными, относящимися к перечисленному типу данных, можно выполнить операции:

- 1) присваивания;
- 2) сложения;
- 3) вывода на экран;
- 4) отношения;
- 5) умножения на константу.

10. Выберите все правильные варианты ответа. При описании массива тип индекса может быть:

- 1) вещественным;
- 2) byte;
- 3) интервальным;
- 4) integer;
- 5) перечисляемым.

9. Выберите все правильные утверждения. С массивами в целом можно выполнять операции:

- 1) сравнения;
- 2) присваивания;
- 3) сложения;
- 4) вычитания;
- 5) умножения.

11. Какое число получится в результате работы программы?

```
const a : array[1..8] of integer = (3, 8, 0, -6, 0, -1, -9, 3);
```

```
var i, k : integer;
```

```
begin
```

```
  k := abs(a[8]);
```

```
  for i := 1 to 8 do
```

```
    if abs(a[i]) > k then k := abs(a[i]);
```

```
  write(k)
```

```
end.
```

12. Какую величину вычисляет программа?

```
const a : array[1..8] of integer = (3, 8, 0, -6, 0, -1, -9, 3);
```

```
var i, k : integer;
```

```
begin
```

```
  k := 0; j := 4;
```

```
  for i := 1 to 8 do
```

```
    if abs(a[i]) > j then k := a[i];
```

```
  write(k)
```

```
end.
```

13. Составить алгоритмы для решения следующих задач:

- 1) Первый из элементов массива, больший, чем 4.
- 2) Первый из элементов массива, больший по модулю, чем 4.
- 3) Модуль последнего из элементов массива, больших, чем 4.
- 4) Модуль первого из элементов массива, больших, чем 4.
- 5) Последний из элементов массива, больший по модулю, чем 4.

Практическая работа № 37. Составление программ, использующих подпрограммы

Цель: приобретение навыков написания программ с подпрограммами пользователя на языке Паскаль

Теоретический материал

В языке Паскаль имеется два вида подпрограмм – процедуры и функции.

Имея один и тот же смысл и аналогичную структуру, процедуры и функции различаются назначением и способом их использования.

Все процедуры в языке Паскаль делятся на 2 вида: встроенные и пользовательские.

Встроенные определены заранее и могут вызываться без предварительного описания.

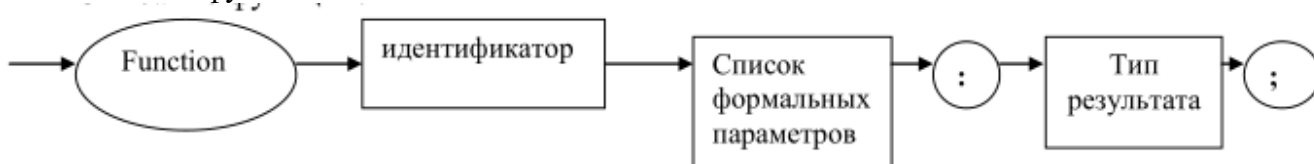
Пользовательские – именованная группа операторов, реализующая определенную последовательность действий.

Вызов пользовательских процедур и функций осуществляется по имени.

Описание процедуры:



Описание функции:



Пример 1. Рассчитать значение $y(x)$ по формуле

$$y = tg^2(x+1) + tg(x+0.5) + 7 * tg(x)$$

Для расчета значения $tg(x)$ используется подпрограмма-функция $tg(a)$ с одним параметром.

```
Program Project1;
var x,y:real;
function tg(a:real):real;
begin
    tg:=cos(a)/sin(a);
end;
begin
    write('x='); readln(x);
    y:=sqr(tg(x+1))+tg(x+0.5)+7*tg(x);
    writeln('y=',y:6:2);
    readln;
end.
```

Пример 2. Условие из предыдущего примера. В программе для расчета значения $tg(x)$ используется подпрограмма-процедура $tg(a,b)$ с двумя параметрами.

```
Program Project2;
var x,y:real; x1,x2,x3:real;
procedure tg(a:real; var b:real);
begin
    b:=cos(a)/sin(a);
end;
begin
    write('x='); readln(x);
    tg(x+1,x1); tg(x+0.5,x2); tg(x,x3);
    y:=sqr(x1)+x2+7*x3;
end.
```

```

        writeln('y=',y:6:2);
        readln;
end.

```

Пример 3. Рассчитать периметр (в м) и площадь прямоугольника (в м²). Длины сторон (x, y) вводятся с клавиатуры в см. В программе используется подпрограмма-процедура parametr(a, b, p, s).

```

Program Project3;
var x,y:integer; s1,p1:real;
procedure parametr(a,b:integer; var p,s:real);
begin
    p:=2*(a+b)/100;
    s:=a*b/10000;
end;
begin
    x:=30; y:=40;
    parametr(x,y,p1,s1);
    writeln('p1=',p1:6:2, ' s1=',s1:6:2);
    readln;
end.

```

Пример 4. Условие из предыдущего примера. В программе используются две подпрограммы-функции.

```

Program Project4;
var x,y:integer; s1,p1:real;
function p(a,b:integer):real;
begin
    p:=2*(a+b)/100;
end;
function s(a,b:integer):real;
begin
    s:=a*b/10000;
end;
begin
    x:=30; y:=40;
    p1:=p(x,y);
    s1:=s(x,y);
    writeln('p1=',p1:6:2, ' s1=',s1:6:2);
    readln;
end.

```

Пример 5. Условие из предыдущего примера. В программе используется процедура parametr без параметров.

```

Program Project5;
var a,b:integer; p,s:real;
procedure parametr;
begin
    p:=2*(a+b)/100;
    s:=(a*b)/10000;
end;
begin
    write('a='); readln(a); write('b='); readln(b);
    parametr;
    writeln('p=',p:6:2, ' s=',s:6:2);
    a:=50; b:=80;
end.

```

```

    parametr;
    writeln('p=',p:6:2, ' s=',s:6:2);
    readln;
end.

```

Пример 6. Условие из предыдущего примера. В программе используются две функции (s и p) без параметров.

```

Program Project6;
var a,b:integer; p1,s1:real;
function p:real;
begin
    p:=2*(a+b)/100;
end;
function s:real;
begin
    s:=a*b/10000;
end;
begin
    a:=30; b:=40;
    p1:=p;
    s1:=s;
    writeln('p1=',p1:6:2, ' s1=',s1:6:2);
    readln;
end.

```

Задания практической работы

Задание 1. Написать программу с подпрограммой-функцией.

Задание 2. Написать программу с подпрограммой-процедурой (с параметром или параметрами).

Варианты

1. Даны действительные s и t. Рассчитать $f(t,-2s,1.17)+f(2.2,t,s-t)$, где

$$f(a,b,c) = \frac{2a - b - \sin(c)}{5 + |c|}$$

подпрограмма.

2. Даны действительные a, b, c. Получить

$$\frac{\max(a-b, a, a+b) + \max(a, b+c, a-c)}{1 + \max(a+bc, 1.15, a/c)}$$

, где $\max(x,y)$ подпрограмма.

3. Даны действительные числа S и t. Рассчитать

$f(t,-3*s,2.5-t)+f(5.2,3*t,s-t)$, где

$$f(a,b,c) = \frac{2a - b - \cos(c)}{3.5 - |c|}$$

подпрограмма.

4. Даны действительные числа S и t. Рассчитать

$f(1.5*t,2*s)+f(t,3-s)-f(2*s-3,t+s)$, где

$$f(a,b) = \frac{a^3 - b^2 + \sin(ab)}{2ab + 5b^2}$$

подпрограмма.

5. Даны действительные числа S и t. Рассчитать

$h(s,t)+h(h^2(s-t,s*t),h^4(s-t,s+t))+h(1,1)$, где

$$h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2} - (a-b)^2$$

подпрограммы.

6. Даны действительные числа a , b . Рассчитать $Y = \min(a, b)$, $Y1 = \min(\min(a*b, a+b), \min(a^2 - b*a, b^3 - 3*a))$, $Y2 = \min(Y1^2 + Y, 5.25)$, где $\min(x, x1)$ подпрограмма.

7. Даны действительные числа S и t . Рассчитать $f(t, 4*s, 2.5*t) + f(5, 2*t, s+t)$,

$$f(a, b, c) = \frac{a^2 + b^3 + \sqrt{|c|}}{3 * |c|}$$

где подпрограмма.

8. Даны действительные числа S и t . Рассчитать $h(s, t) + h^3(s-t, s*t) + h^2(s-t, s+t) + h(1, 1)$, где

$$h(a, b) = \frac{a^2}{1+b} + \frac{b}{1+a^3} + (a-b)^3$$

подпрограмма.

9. Даны действительные числа a , b , c . Рассчитать

$$y = \frac{\max(a, a+b) + \max(a, b+c)}{1 + \max(a+b*c, 2*a, c*b)}$$

, где $\max(x, x1)$ описать как подпрограмму.

10. Вычислить $K = (x + y + z) / 3$, где x – наибольшее значение из параметров $x1, x2$, y – из $y1, y2$, z – из $z1, z2$. Использовать подпрограмму для нахождения наибольшего значения из двух параметров.

11. В порт в среднем приходят 3 корабля в день. Какова вероятность того, что в день придет 2 корабля, 4 корабля? Вероятность вычислять по формуле $P = 3 * e^{-3} / k!$. Использовать подпрограмму для расчета $k!$.

12. Вычислить $Z = (v1 + v2 + v3) / 3$, где $v1, v2, v3$ – объемы шаров с радиусами $r1, r2, r3$ соответственно. Использовать подпрограмму для расчета объемов шаров. Объем шара вычислять по формуле $V = 4 / 3 * \Pi * R^3$.

13. Определить число сочетаний из n по m ($n > m$), по формуле $C = n! / (m!(n-m))$. Использовать одну подпрограмму для расчета $n!$ и $m!$.

14. Вычислить $Z = (n + m) / 2$, где n – наименьшее значение из параметров $n1, n2$, m – из $m1, m2$. Использовать подпрограмму для нахождения наименьшего значения из двух параметров.

15. Составить программу вычисления значения функции

$$S = x^2 + y^2 + \sin(2 * x^2 * y^2) + x + z + \sin(2 * x * z) + y^2 + z^2 + \sin(2 * y^2 * z^2),$$

используя подпрограмму для расчета $a + b + \sin(2 * a * b)$.