



ИНФОРМАТИКА

Учебное пособие

2019 г.



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ДЕПАРТАМЕНТ НАУЧНО-ТЕХНОЛОГИЧЕСКОЙ ПОЛИТИКИ И ОБРАЗОВАНИЯ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО Донской ГАУ)

АЗОВО-ЧЕРНОМОРСКИЙ ИНЖЕНЕРНЫЙ ИНСТИТУТ – ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ» В Г. ЗЕРНОГРАДЕ
(Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ)

Кафедра «Теплоэнергетика и
информационно- управляющие
системы»

ИНФОРМАТИКА

Учебное пособие

Зерноград – 2019

© Литвинов В.Н., Грачева Н.Н.,
Руденко Н.Б., Каун О.Ю., 2019
© Азово-Черноморский инженерный
институт ФГБОУ ВО Донской ГАУ, 2019

Об издании – [1](#), [2](#)
[Содержание](#)
УДК 681.3

*Публикуется по решению методического совета
энергетического факультета
Азово-Черноморского инженерного института – филиала
Федерального государственного бюджетного образовательного учреждения
высшего образования «Донской государственной аграрный университет»
в г. Зернограде*

Рецензенты:

кандидат технических наук, доцент ***Токарева А.Н.***

Информатика [Электронный ресурс]: учебное пособие / В.Н. Литвинов, Н.Н. Грачева, Н.Б. Руденко, О.Ю. Каун. – Электрон. дан. – Зерноград: Азово-Черноморский инженерный институт ФГБОУ ВО ДГАУ, 2019. – 205 с. – Режим доступа: Локальная сеть Библиотеки Азово-Черноморского инженерного института ФГБОУ ВО Донской ГАУ.

Целью учебного пособия является изучение основных разделов дисциплины «Информатика» и освоение компетенции ОПК-1.

Учебное пособие предназначено для студентов очной и заочной форм обучения, изучающих дисциплину «Информатика» и обучающихся по направлению подготовки «Агроинженерия».

© Литвинов В.Н., Грачева Н.Н.,
Руденко Н.Б., Каун О.Ю., 2019

© Азово-Черноморский инженерный
институт ФГБОУ ВО Донской ГАУ, 2019

<u>Введение</u>	4
<u>Лабораторная работа №1.</u> Позиционные и непозиционные системы счисления. Перевод чисел из одной системы счисления в другую.....	5
<u>Лабораторная работа №2.</u> Количество информации	17
<u>Лабораторная работа №3.</u> Логические основы работы компьютера. Логические функции. Таблицы истинности. НКФ. НДФ. Упрощение логических функций	35
<u>Лабораторная работа №4.</u> Логические основы работы компьютера. Основные логические элементы. Логические схемы.	43
<u>Лабораторная работа №5.</u> Основные принципы функционирования операционной системы (ОС) GosLinux.....	55
<u>Лабораторная работа №6.</u> ОС GosLinux. Работа в режиме командной строки	69
<u>Лабораторная работа №7.</u> Python. Основные языковые конструкции. Линейный вычислительный процесс	79
<u>Лабораторная работа №8.</u> Python. Управляющие операторы разветвляющегося вычислительного процесса. Операторы if и if-else.....	101
<u>Лабораторная работа №9.</u> Python. Управляющие операторы разветвляющегося вычислительного процесса. Оператор if-elif-else	108
<u>Лабораторная работа №10.</u> Python. Управляющие операторы циклического вычислительного процесса.....	115
<u>Лабораторная работа №11.</u> Python. Комбинированные алгоритмы	128
<u>Лабораторная работа №12.</u> Python. Алгоритмы накопления суммы и произведения.....	137
<u>Лабораторная работа №13.</u> Python. Обработка одномерных массивов	144
<u>Лабораторная работа №14.</u> Python. Одномерные массивы: накопление суммы и произведения, поиск максимального (минимального) элемента массива	153
<u>Литература</u>	158
<u>Приложения</u>	159

Введение

Учебное пособие «Информатика» соответствует Федеральному государственному образовательному стандарту высшего образования (ФГОС ВО) и является частью учебно-методического комплекса дисциплины базовой части блока Б1. Дисциплины «Информатика».

Пособие предназначено для бакалавров очной и заочной форм обучения, обучающихся по направлению подготовки 35.04.06 «Агроинженерия».

Учебное пособие состоит из лабораторных работ, заданий для самостоятельной работы и вариантов контрольной работы. Пособие составлено таким образом, чтобы студент заочной формы обучения мог самостоятельно изучить необходимые разделы дисциплины и выполнить контрольную работу.

В результате изучения материала, представленного в учебном пособии, студент должен овладеть следующей общепрофессиональной компетенцией:

- способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1).

В результате изучения дисциплины «Информатика» студент должен:

- знать основные понятия, способы и методы осуществления поиска, хранения, обработки и анализа информации из различных источников и баз данных, представления ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий;

- уметь осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий;

- владеть основными средствами компьютерной техники и информационных технологий для поиска, хранения, обработки и анализа информации из различных источников и баз данных, для представления ее в требуемом формате.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: **ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с основными позиционными системами счисления, применяемыми для вычислений в компьютере.

Научиться:

- переводить числа из одной системы счисления в другую;
- выполнять основные арифметические операции в позиционных системах счисления.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Основные понятия

Система счисления – это совокупность правил для обозначения и наименования чисел.

Системы счисления делят на две большие группы: непозиционные и позиционные.

Непозиционными системами счисления называются системы счисления, в которых каждой цифре соответствует величина, не зависящая от её места в записи числа. Примером непозиционной системы счисления может служить римская система счисления, символы алфавита которой и обозначаемое ими количество представлено в [таблице 1.1](#).

Таблица 1.1 – Римская система счисления

Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

Позиционными системами счисления называются системы счисления, в которых вклад каждой цифры в величину числа зависит от её положения (позиции) в последовательности цифр, изображающей число. Примером мо-

жет служить десятичная система счисления, считать в которой учат в школе. Значение каждой цифры в числе зависит от ее местоположения. Сравните два числа: 50 и 500. В первом числе пять стоит в разряде десятков, во втором числе – в разряде сотен.

Основными достоинствами любой позиционной системы счисления являются:

- во-первых, ограниченное количество символов для записи чисел;
- во-вторых, простота выполнения арифметических операций.

Основанием позиционной системы счисления (q) называют количество символов, используемое для записи числа.

Основание системы счисления показывает, во сколько раз изменится количественное значение цифры при перемещении ее на соседнюю позицию.

При записи числа основание системы счисления указывают в виде нижнего индекса: 54_{10} , 1011_2 , $A1E_{16}$.

В компьютере используют двоичную, десятичную, восьмеричную и шестнадцатеричную системы счисления.

От системы счисления будет зависеть:

- скорость вычислений;
- емкость памяти;
- сложность алгоритма выполнения арифметических и логических операций.

Двоичная система счисления используется для организации машинных операций по преобразованию информации:

$q = 2$, алфавит: 0, 1.

Десятичная система счисления – для ввода и вывода информации:

$q = 10$, алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Восьмеричная и шестнадцатеричная системы счисления – для составления программ.

В восьмеричной системе счисления:

$q = 8$, алфавит: 0, 1, 2, 3, 4, 5, 6, 7.

В шестнадцатеричной системе счисления:

$q = 16$, алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

3.2 Правила перевода чисел из одной системы счисления в другую

3.2.1 Перевод чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления

В [таблице 1.2](#) (см. ниже) приведены правила перевода целой и дробной частей чисел десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Поясним правила на примерах.

Для перевода числа из десятичной системы счисления в двоичную (восьмеричную, шестнадцатеричную) системы счисления необходимо:

- целую часть числа разделить на основание системы счисления, в ко-

торую производится перевод (в двоичную – на 2, в восьмеричную – на 8, в шестнадцатеричную – на 16). Деление целочисленное, с остатком. Деление продолжить до тех пор, пока делимое не станет меньше делителя (2, 8, 16). Затем выписать остатки, начиная с последнего. Таким образом, получим целую часть числа в новой системе счисления;

– дробную часть числа умножаем на основание системы счисления, в которую производится перевод (в двоичную – на 2, в восьмеричную – на 8, в шестнадцатеричную – на 16). Если при умножении получается целое число, то в следующем действии оно не учитывается, а умножается дробная часть и т.д. Умножение заканчивается тогда, когда дробная часть равна 0. Если этого не происходит, то умножение производить до третьего знака после запятой. В дробную часть переведенного числа выписываем полученные целые числа.

Таблица 1.2 – Перевод чисел десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления

Направление перевода	Правила перевода
<p>Целая часть числа</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: right; margin-right: 10px;"> N_{10} \nearrow \longrightarrow \searrow </div> <div style="text-align: left; margin-left: 10px;"> N_2 N_8 N_{16} </div> </div>	<p>деление на 2</p> <p>деление на 8</p> <p>деление на 16</p>
<p>Дробная часть числа</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: right; margin-right: 10px;"> N_{10} \nearrow \longrightarrow \searrow </div> <div style="text-align: left; margin-left: 10px;"> N_2 N_8 N_{16} </div> </div>	<p>умножение на 2</p> <p>умножение на 8</p> <p>умножение на 16</p>

Пример 1

Переведите число $67,25_{10}$ из десятичной системы счисления в двоичную.

Перевод целой части	Перевод дробной части
<div style="text-align: right;"> $\begin{array}{r} _67 \quad 2 \\ _66 \quad _33 \quad 2 \\ \hline 1 \quad _32 \quad _16 \quad 2 \\ \hline \quad 1 \quad _16 \quad _8 \quad 2 \\ \hline \quad \quad 1 \quad _8 \quad 2 \\ \hline \quad \quad \quad 0 \quad _4 \quad 2 \\ \hline \quad \quad \quad \quad 0 \quad _2 \quad 2 \\ \hline \quad \quad \quad \quad \quad 2 \quad 1 \\ \hline \quad \quad \quad \quad \quad \quad 0 \end{array}$ </div>	<div style="text-align: center;"> $\begin{array}{r l} 0, & 25 \\ \hline & 2 \\ \hline 0 & 50 \\ \hline & 2 \\ \hline 1 & 00 \end{array}$ </div>

Ответ: $67,25_{10} = 1000011,01_2$

Для перевода числа из двоичной (восьмеричной, шестнадцатеричной) системы счисления в десятичную необходимо:

– над каждой цифрой числа, которое необходимо перевести, поставить номер разряда, на котором стоит цифра. Разряды выставляются в целой части числа справа налево (начиная с 0), в дробной части – слева направо (начиная с -1);

– умножаем каждую цифру переводимого числа на основание системы счисления (из которой происходит перевод), возведенное в степень, равную номеру разряда, на котором стоит соответствующая цифра. Затем складываем эти произведения;

– если число переводится из шестнадцатеричной системы счисления, то буква, при умножении, заменяется соответствующим этой букве значением в десятичной системе счисления ([таблица 1.4](#)).

Таблица 1.4 – Таблица соответствия между системами счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19

Пример 4

Переведите число $57,4_8$ из восьмеричной системы счисления в десятичную.

$$57,4_8 = 5 \cdot 8^1 + 7 \cdot 8^0 + 4 \cdot 8^{-1} = 40 + 7 + \frac{4}{8} = 47 \frac{1}{2} = 47,5_{10}.$$

Ответ: $57,4_8 = 47,5_{10}$

Пример 5

Переведите число $A7,48_{16}$ из шестнадцатеричной системы счисления в десятичную.

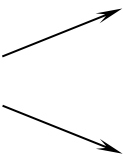
$$A7,48_{16} = 10 \cdot 16^1 + 7 \cdot 16^0 + 4 \cdot 16^{-1} + 8 \cdot 16^{-2} = 160 + 7 + \frac{4}{16} + \frac{8}{256} = 167 \frac{9}{32}_{10}.$$

Ответ: $A7,48_{16} = 167 \frac{9}{32}_{10}$

3.2.3 Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления

В [таблице 1.5](#) приведены правила перевода чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления.

Таблица 1.5 – Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления

Направление перевода		Правила перевода
N_2 	N_8	замена триад цифрами восьмеричной системы счисления
	N_{16}	замена тетрад цифрами шестнадцатеричной системы счисления

Поясним правила на примерах.

Для перевода числа из двоичной системы счисления в восьмеричную (шестнадцатеричную) системы счисления необходимо:

- число разбить на тройки цифр – триады (для восьмеричной системы счисления) или на четверки цифр – тетрады (для шестнадцатеричной системы счисления). Разбиение начинать от запятой, разделяющей целую и дробную части, влево и вправо соответственно;
- если в отделенной последовательности количество цифр меньше трёх (четырёх), то недостающие цифры заменить нулями. Для целой части нули приписать перед числом, для дробной части – после числа;
- выписать соответственно каждой отделенной группе (триаде или тетраде) по таблице соответствия между системами счисления восьмеричную или шестнадцатеричную цифру (см. [таблицу 1.4](#)).

Пример 6

Переведите число $1111101,01011_2$ из двоичной в восьмеричную и шестнадцатеричную системы счисления.

$$1111101,01011_2 = \begin{array}{ccccc} 001 & 111 & 101, & 010 & 110_2 \\ 1 & 7 & 5 & 2 & 6 \end{array} = 175,26_8$$

$$1111101,01011_2 = \begin{array}{cccc} 0111 & 1101, & 0101 & 1000_2 \\ 7 & D & 5 & 8 \end{array} = 7D,58_{16}$$

Ответ: $1111101,01011_2 = 175,26_8$; $1111101,01011_2 = 7D,58_{16}$

3.2.4 Перевод чисел из восьмеричной и шестнадцатеричной систем счисления в двоичную систему счисления

В [таблице 1.6](#) приведены правила перевода чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления.

Таблица 1.6 – Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления

Направление перевода	Правила перевода
$N_8 \xrightarrow{\quad}$ $N_{16} \xleftarrow{\quad}$	замена восьмеричных цифр триадами
N_2	замена шестнадцатеричных цифр тетрадами

Поясним правила на примерах.

Для перевода числа из восьмеричной (шестнадцатеричной) системы счисления в двоичную систему счисления необходимо:

– выписать соответственно каждой цифре восьмеричного числа тройку цифр – триаду по таблице соответствия между системами счисления триаду (см. [таблицу 1.4](#)). Аналогично поступаем и с шестнадцатеричным числом, но выписываем четверку цифр – тетраду;

– в записи полученного двоичного числа нули в начале целой части и в конце дробной части опустить.

Пример 7

Переведите числа $567,561_8$ и $4FE,AD_{16}$ в двоичную систему счисления.

$$567,561_8 = \begin{array}{ccccc} 5 & 6 & 7, & 5 & 6 & 1_8 \\ 101 & 110 & 111 & 101 & 110 & 001 \end{array} = 101110111,101110001_2$$

$$4FE, AD_{16} = \begin{array}{ccccc} & & & 12 & \\ & 4 & F & E, & A & D_{16} & = & 10011111110,10101101_2 \\ 0100 & 1111 & 1110 & 1010 & 1101 & & & \end{array}$$

Ответ: $567,561_8 = 101110111,101110001_2$;
 $4FE, AD_{16} = 10011111110,10101101_2$.

3.2.5 Перевод чисел из восьмеричной системы счисления в шестнадцатеричную систему счисления (или из шестнадцатеричной системы счисления в восьмеричную систему счисления)

Для перевода числа из восьмеричной системы счисления в шестнадцатеричную (из шестнадцатеричной в восьмеричную) необходимо:

- осуществить перевод восьмеричного (или шестнадцатеричного) числа в двоичную систему счисления (см. п.3.2.4);
- осуществить перевод числа из двоичной системы счисления в шестнадцатеричную (или восьмеричную) систему счисления (см. п.3.2.3).

Пример 8

Переведите число $4FE, AD_{16}$ в восьмеричную систему счисления, а число $567,561_8$ в шестнадцатеричную систему счисления.

$$4FE, AD_{16} = 0100 \ 1111 \ 1110, 1010 \ 1101_2 = 10011111110,10101101_2 \\ = 010 \ 011 \ 111 \ 110, 101 \ 011 \ 010_2 = 2376,532_8$$

$$567,561_8 = 101 \ 110 \ 111, 101 \ 110 \ 001_2 = 101110111,101110001_2 \\ = 0001 \ 0111 \ 0111, 1001 \ 1001_2 = 177, B9_{16}$$

3.3 Арифметические операции в позиционных системах счисления

Арифметические операции в восьмеричной и шестнадцатеричной системе счисления сходны с аналогичными операциями в десятичной системе счисления.

Все операции производятся столбиком. Рассмотрим несколько примеров.

Пример 9

Выполните сложение, вычитание и умножение двоичных чисел $1100,11_2$ и $101,10_2$.

сложение	вычитание	умножение
$\begin{array}{r} 1100,11 \\ + 101,10 \\ \hline 10010,01 \end{array}$	$\begin{array}{r} 1100,11 \\ - 101,10 \\ \hline 111,01 \end{array}$	$\begin{array}{r} 1100,11 \\ \times 101,1 \\ \hline 110011 \\ + 110011 \\ \hline 110011 \\ \hline 10000110,001 \end{array}$

Пример 10

Выполните сложение, вычитание и умножение чисел в восьмеричной системе счисления: $215,4_8$ и $73,6_8$.

Необходимо помнить:

– если при сложении или умножении двух цифр в разряде получилось число большее восьми, то полученное число надо разделить на 8 (целочисленное деление). Остаток от деления оставить в текущем разряде, а результат деления прибавить к старшему разряду;

– если при вычитании двух цифр уменьшаемая цифра меньше вычитаемой, то занимаем в старшем разряде 1 (для текущего разряда это будет 8). 8 прибавляем к уменьшаемому и из полученной суммы отнимаем вычитаемое.

сложение	вычитание	умножение
$\begin{array}{r} 111 \\ 215,4 \\ + 73,6 \\ \hline 311,2 \end{array}$ <div style="margin-left: 100px;"> $4+6=10=8+2$ $5+3+1=9=8+1$ $1+7+1=9=8+1$ $2+1=3$ </div>	$\begin{array}{r} 111 \\ 215,4 \\ - 73,6 \\ \hline 121,6 \end{array}$ <div style="margin-left: 100px;"> $8+4-6=6$ $4-3=1$ $1+8-7=2$ </div>	$\begin{array}{r} 215,4 \\ \times 73,6 \\ \hline 15210 \\ + 6504 \\ \hline 17364 \\ \hline 20406,50 \end{array}$

Пример 11

Выполните сложение и вычитание чисел в шестнадцатеричной системе счисления: $8D,8_{16}$ и $3B,C_{16}$.

Необходимо помнить:

– если при сложении или умножении двух цифр в разряде получилось число большее восьми, то полученное число надо разделить на 16 (целочисленное деление). Остаток от деления оставить в текущем разряде, а результат деления прибавить к старшему разряду;

– если при вычитании двух цифр уменьшаемая цифра меньше вычитаемой, то занимаем в старшем разряде 1 (для текущего разряда это будет 16). 16 прибавляем к уменьшаемому и из полученной суммы отнимаем вычитаемое.

сложение	вычитание	умножение
$\begin{array}{r} 11 \\ 8D,8 \\ + 3B,C \\ \hline C9,4 \end{array}$ <div style="margin-left: 40px;"> $8+12=20=16+4$ $13+11+1=25=16+9$ $8+3+1=12=C$ </div>	$\begin{array}{r} 11 \\ 8D,8 \\ - 3B,C \\ \hline 51,C \end{array}$ <div style="margin-left: 40px;"> $8+16=24-12=12=C$ $12-11=1$ $8-3=5$ </div>	$\begin{array}{r} 8D,8 \\ \times 3B,C \\ \hline 6A20 \\ + 6148 \\ \hline 1A88 \\ \hline 2106,A0 \end{array}$

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Переведите числа из десятичной системы счисления в двоичную, восьмеричную, шестнадцатеричную системы счисления.

- | | | | |
|-------------------------|-------------------------|-------------------------|-------------------------|
| 1) 57,9 ₁₀ | 9) 97,5 ₁₀ | 17) 59,5 ₁₀ | 25) 110,1 ₁₀ |
| 2) 123,01 ₁₀ | 10) 55,55 ₁₀ | 18) 29,3 ₁₀ | 26) 25,45 ₁₀ |
| 3) 113,1 ₁₀ | 11) 80,25 ₁₀ | 19) 67,5 ₁₀ | 27) 38,35 ₁₀ |
| 4) 39,2 ₁₀ | 12) 17,17 ₁₀ | 20) 91,25 ₁₀ | 28) 71,1 ₁₀ |
| 5) 91,95 ₁₀ | 13) 11,95 ₁₀ | 21) 77,7 ₁₀ | 29) 44,9 ₁₀ |
| 6) 89,5 ₁₀ | 14) 83,65 ₁₀ | 22) 45,2 ₁₀ | 30) 31,23 ₁₀ |
| 7) 76,45 ₁₀ | 15) 22,22 ₁₀ | 23) 34,25 ₁₀ | 31) 86,2 ₁₀ |
| 8) 190,2 ₁₀ | 16) 78,05 ₁₀ | 24) 66,5 ₁₀ | 32) 29,35 ₁₀ |

2. Переведите числа в десятичную систему счисления.

- | | | | |
|------------------------|--------------------------|--------------------------|--------------------------|
| 1) 101,10 ₂ | 9) 2,3 ₈ | 17) 17,17 ₈ | 25) 100001 ₂ |
| 2) 555 ₁₆ | 10) 11001 ₂ | 18) 77,6 ₈ | 26) 190 ₁₆ |
| 3) 113 ₈ | 11) BA2, A ₁₆ | 19) 1010101 ₂ | 27) 11,75 ₈ |
| 4) 1010 ₂ | 12) 22 ₈ | 20) A38, 1 ₁₆ | 28) 1101,1 ₂ |
| 5) 788 ₁₆ | 13) 111 ₂ | 21) 17,5 ₈ | 29) F6, B ₁₆ |
| 6) 35 ₈ | 14) AB6 ₁₆ | 22) 10001 ₂ | 30) 71,3 ₈ |
| 7) 11011 ₂ | 15) 52 ₈ | 23) F80 ₁₆ | 31) 11,101 ₂ |
| 8) CC3,5 ₁₆ | 16) 111101 ₂ | 24) 11,2 ₈ | 32) BB3, 3 ₁₆ |

3. Переведите числа в восьмеричную и шестнадцатеричную системы счисления.

- | | | | |
|----------------|------------------|-------------------|------------------|
| 1) 1000000_2 | 9) $1010,11_2$ | 17) 1001100_2 | 25) 101000_2 |
| 2) $100,001_2$ | 10) 1110_2 | 18) 1000001_2 | 26) $1001,01_2$ |
| 3) $111,001_2$ | 11) $101,10_2$ | 19) $101,10_2$ | 27) $101,10_2$ |
| 4) $1,0001_2$ | 12) $111,0011_2$ | 20) $11,1011_2$ | 28) $10,0101_2$ |
| 5) 10101_2 | 13) 11000_2 | 21) 10100101_2 | 29) $1110,1_2$ |
| 6) $110,11_2$ | 14) 1000101_2 | 22) $10,001001_2$ | 30) $100,0001_2$ |
| 7) $101,1_2$ | 15) 110011_2 | 23) $110,001_2$ | 31) $1100,11_2$ |
| 8) $100,100_2$ | 16) $11,0101_2$ | 24) $1000,01_2$ | 32) $11001,1_2$ |

4. Переведите числа в двоичную систему счисления.

- | | | | |
|-----------------|------------------|---------------|------------------|
| 1) $65,8_{16}$ | 9) $67,5_8$ | 17) BA_{16} | 25) $51, A_{16}$ |
| 2) $123,8_{16}$ | 10) $55,55_8$ | 18) 29_{16} | 26) $B7_{16}$ |
| 3) 143_{16} | 11) 35_8 | 19) 62_{16} | 27) 47_{16} |
| 4) 39_{16} | 12) $17,17_8$ | 20) 130_8 | 28) $8C_{16}$ |
| 5) $F16_{16}$ | 13) $37, A_{16}$ | 21) $74,6_8$ | 29) $13,6_{16}$ |
| 6) $89,4_{16}$ | 14) $86,5_{16}$ | 22) $43,2_8$ | 30) $41,7_{16}$ |
| 7) $76,4_8$ | 15) $22,67_{16}$ | 23) $41,1_8$ | 31) 38_8 |
| 8) 50_8 | 16) $78,32_{16}$ | 24) 500_8 | 32) $15,5_8$ |

5. Переведите числа из шестнадцатеричной системы счисления в восьмеричную, из восьмеричной системы счисления в шестнадцатеричную.

- | | | | |
|-----------------|------------------|---------------|------------------|
| 1) $65,8_{16}$ | 9) $67,5_8$ | 17) BA_{16} | 25) $51, A_{16}$ |
| 2) $123,8_{16}$ | 10) $55,55_8$ | 18) 29_{16} | 26) $B7_{16}$ |
| 3) 143_{16} | 11) 35_8 | 19) 62_{16} | 27) 47_{16} |
| 4) 39_{16} | 12) $17,17_8$ | 20) 130_8 | 28) $8C_{16}$ |
| 5) $F16_{16}$ | 13) $37, A_{16}$ | 21) $74,6_8$ | 29) $13,6_{16}$ |
| 6) $89,4_{16}$ | 14) $86,5_{16}$ | 22) $43,2_8$ | 30) $41,7_{16}$ |
| 7) $76,4_8$ | 15) $22,67_{16}$ | 23) $41,1_8$ | 31) 38_8 |
| 8) 50_8 | 16) $78,32_{16}$ | 24) 500_8 | 32) $15,5_8$ |

6. Выполните сложение чисел, а затем проверьте результаты, выполнив соответствующие действия в десятичной системе счисления.

- | | | |
|---------------------------|-----------------------------|-----------------------------|
| 1) $101111_2 + 11101_2$ | 11) $35_8 + 17_8$ | 21) $57,25_8 + 71,6_8$ |
| 2) $1011,101_2 + 10,11_2$ | 12) $13,17_8 + 16_8$ | 22) $75_8 + 17_8$ |
| 3) $1011_2 + 110_2$ | 13) $F1_{16} + A_{16}$ | 23) $E, 5_{16} + A, B_{16}$ |
| 4) $1011_2 + 111_2$ | 14) $19_{16} + C_{16}$ | 24) $3D_{16} + E_{16}$ |
| 5) $110,1_2 + 11,1_2$ | 15) $E, 2_{16} + A, B_{16}$ | 25) $10111_2 + 111_2$ |
| 6) $1010_2 + 1111_2$ | 16) $F8_{16} + 9_{16}$ | 26) $19_{16} + F_{16}$ |
| 7) $35_8 + 75_8$ | 17) $BA_{16} + C_{16}$ | 27) $B9_{16} + 12_{16}$ |

- | | | |
|----------------------|--------------------------|-------------------------|
| 8) $165_8 + 37_8$ | 18) $209_{16} + 77_{16}$ | 28) $100111_2 + 111_2$ |
| 9) $7,5_8 + 14,6_8$ | 19) $101,1_2 + 11,1_2$ | 29) $79_{16} + D_{16}$ |
| 10) $55,55_8 + 36_8$ | 20) $1010_2 + 111_2$ | 30) $8A_{16} + 17_{16}$ |

7. Выполните вычитание чисел, а затем проверьте результаты, выполнив соответствующие действия в десятичной системе счисления.

- | | | |
|---------------------------|-----------------------------|-----------------------------|
| 1) $101111_2 - 11101_2$ | 11) $35_8 - 17_8$ | 21) $57,25_8 - 7,6_8$ |
| 2) $1011,101_2 - 10,11_2$ | 12) $13,17_8 - 6_8$ | 22) $75_8 - 17_8$ |
| 3) $1011_2 - 110_2$ | 13) $F1_{16} - A_{16}$ | 23) $E2,5_{16} - A, B_{16}$ |
| 4) $1011_2 - 111_2$ | 14) $19_{16} - C_{16}$ | 24) $3D_{16} - E_{16}$ |
| 5) $110,1_2 - 11,1_2$ | 15) $E5,2_{16} - A, B_{16}$ | 25) $10111_2 - 111_2$ |
| 6) $1010_2 + 11_2$ | 16) $F8_{16} - 9_{16}$ | 26) $C9_{16} - F_{16}$ |
| 7) $35_8 - 7_8$ | 17) $BA_{16} - C_{16}$ | 27) $B9_{16} - 12_{16}$ |
| 8) $165_8 - 37_8$ | 18) $209_{16} - 77_{16}$ | 28) $100111_2 - 111_2$ |
| 9) $7,5_8 - 1,6_8$ | 19) $101,1_2 - 11,1_2$ | 29) $D9_{16} - D_{16}$ |
| 10) $55,55_8 - 36_8$ | 20) $1010_2 - 111_2$ | 30) $8A_{16} - 17_{16}$ |

8. Выполните умножение чисел, а затем проверьте результаты, выполнив соответствующие действия в десятичной системе счисления.

- | | | |
|-------------------------------|---------------------------------|---------------------------------|
| 1) $1011101_2 \cdot 11101_2$ | 11) $35_8 \cdot 17_8$ | 21) $75_8 \cdot 16_8$ |
| 2) $1011,101_2 \cdot 10,11_2$ | 12) $13,17_8 \cdot 6_8$ | 22) $17,17_8 \cdot 6,6_8$ |
| 3) $1011_2 \cdot 110_2$ | 13) $F1_{16} \cdot A_{16}$ | 23) $E, 5_{16} \cdot A, B_{16}$ |
| 4) $1011_2 \cdot 101_2$ | 14) $19_{16} \cdot A_{16}$ | 24) $3D_{16} \cdot E_{16}$ |
| 5) $110,1_2 \cdot 11,1_2$ | 15) $E, 2_{16} \cdot A, B_{16}$ | 25) $10111_2 \cdot 111_2$ |
| 6) $1010_2 \cdot 11_2$ | 16) $F8_{16} \cdot 9_{16}$ | 26) $19_{16} \cdot F_{16}$ |
| 7) $35_8 \cdot 71_8$ | 17) $BA_{16} \cdot C_{16}$ | 27) $B1_{16} \cdot 12_{16}$ |
| 8) $165_8 \cdot 37_8$ | 18) $209_{16} \cdot 77_{16}$ | 28) $100111_2 \cdot 111_2$ |
| 9) $7,5_8 \cdot 1,6_8$ | 19) $101,1_2 \cdot 11,1_2$ | 29) $79_{16} \cdot D_{16}$ |
| 10) $55_8 \cdot 36_8$ | 20) $1010_2 \cdot 111_2$ | 30) $8A_{16} \cdot 17_{16}$ |

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: **КОЛИЧЕСТВО ИНФОРМАЦИИ**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с основными формами представления информации, методами и правилами вычисления объема информации.

Научиться:

- вычислять информационный объем различных видов информации.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Единицы измерения и методы измерения количества информации

Информация – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

В компьютере для представления информации используется двоичное кодирование, так как удалось создать надежно работающие технические устройства, которые могут со стопроцентной надежностью сохранять и распознавать не более двух различных состояний: наличие или отсутствие электрического импульса.

Код – система условных знаков (символов) для представления различной информации.

Информация в компьютере представлена в двоичном коде, состоящем из двух цифр – 0 и 1.

При записи двоичной цифры реализуется выбор одного из двух возможных состояний (одной из двух цифр) и, следовательно, она несет объем информации, равный 1 биту (binary digit – двоичный разряд). Таким образом, за минимальную единицу измерения количества информации принят 1 бит.

Для измерения информации используются и более крупные единицы:

1 байт = 8 бит = 2^3 бит;

1 Кбайт (килобайт) = 2^{10} байт = 1024 байт;

1 Мбайт (мегабайт) = 2^{10} Кбайт = 1024 Кбайт;

1 Гбайт (гигабайт) = 2^{10} Мбайт = 1024 Мбайт;

1 Тбайт (терабайт) = 2^{10} Гбайт = 1024 Гбайт.

3.1.1 Алфавитный подход к измерению количества информации

Алфавитный подход к измерению информации позволяет определить количество информации, заключенной в тексте. Алфавитный подход является объективным, т.е. он не зависит от субъекта (человека), воспринимающего текст.

Множество символов, используемых при записи текста, называется *алфавитом*. Полное количество символов в алфавите называется *мощностью (размером) алфавита*.

Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой (равновероятно), то количество информации, которое несет каждый символ, вычисляется по формуле

$$i = \log_2 N,$$

где N – мощность алфавита.

Следовательно, в 2-символьном алфавите каждый символ «весит» 1 бит ($\log_2 2 = 1$); в 4-символьном алфавите каждый символ несет 2 бита информации ($\log_2 4 = 2$); в 8-символьном – 3 бита ($\log_2 8 = 3$) и т.д.

Один символ из алфавита мощностью 256 несет в тексте 8 бит информации. Такое количество информации называется байт. Алфавит из 256 символов используется для представления текстов на компьютере.

Если весь текст состоит из K символов, то при алфавитном подходе размер содержащейся в нем информации равен:

$$I = K \cdot i,$$

где i – информационный вес одного символа в используемом алфавите.

Пример 1

В зрительном зале две прямоугольные области зрительных кресел: одна 10 на 5, а другая 4 на 8. Какое минимальное количество бит потребуется для кодирования каждого места в автоматизированной системе?

Решение:

$$N = 10 \cdot 5 + 4 \cdot 8 = 82;$$

$$64(2^6) < 82 < 128(2^7);$$

$$N' = 128 = 2^7;$$

$$i = \log_2 N' = \log_2 2^7 = 7 \text{ бит.}$$

Ответ: 7 бит.

Пример 2

В велокроссе участвуют 230 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 20 велосипедистов?

Решение:

$$N = 230;$$

$$128(2^7) < 82 < 256(2^8);$$

$$N' = 256 = 2^8;$$

$i = \log_2 N' = \log_2 2^8 = 8$ бит – это минимальное количество бит для записи номера спортсмена.

Поскольку была записана информация о 20 спортсменах, объем записанного сообщения составил $20 \times 8 = 160$ бит.

Ответ: 160 бит.

Пример 3

Сколько существует различных последовательностей из символов «+» и «-», длиной ровно в шесть символов?

Решение:

В данном случае алфавит состоит из двух элементов, потому информационный объем одного символа – 1 бит. 6 бит позволяют закодировать множество из $2^6 = 64$ элементов.

Ответ: 64.

Пример 4

Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

Решение:

С помощью n лампочек, каждая из которых может находиться в трех состояниях, можно закодировать 3^n сигналов. $3^2 < 18 < 3^3$, поэтому двух лампочек недостаточно, а трех хватит.

Ответ: 3.

Пример 5

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 480 бит. Какова длина сообщения в символах?

Решение:

После перекодировки из 16-битного кода в 8-битный каждый символ сообщения стал занимать на 8 бит меньше, а все сообщение уменьшилось на 480 бит, следовательно, сообщение состояло из $480:8 = 60$ символов.

Ответ: 60.

Пример 6

Книга, набранная с помощью компьютера, содержит 150 страниц. На каждой странице – 40 строк, в каждой строке – 60 символов. Каков объем информации в книге?

Решение:

Мощность компьютерного алфавита равна 256. Значит, один символ несет $256 = 2^8$ – 1 байт информации. Значит, страница содержит $40 \cdot 60 = 2400$ байт информации. Объем всей информации в книге (в разных единицах):

$$2400 \cdot 150 = 360000 \text{ байт};$$

$$\frac{360000}{1024} = 351,5625 \text{ Кбайт};$$

$$\frac{351,5625}{1024} = 0,34332275 \text{ Мбайт.}$$

Ответ: 0,34332275 Мбайт.

3.1.2 Вероятностный подход к измерению количества информации

Рассмотрим способ вычисления информационного объема сообщения с использованием вероятностного подхода на конкретном примере.

Пример 7

В пруду живут 32500 карасей, 5000 щук и 12500 пескарей. Вычислить объем сообщения «рыбак поймал пескаря».

Решение:

Всего в пруду живет 50000 рыб ($32500 + 5000 + 12500 = 50000$). Вероятность попадания на удочку каждого из вида рыб равна его доле в общем количестве. Отсюда:

$$p = \frac{12500}{50000} = 0,25.$$

Качественную связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить так: чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

Количественная зависимость между вероятностью события (p) и количеством информации в сообщении о нем (I) выражается формулой

$$I = \log_2(1/p).$$

В рассматриваемом примере количество информации в сообщении «рыбак поймал пескаря» вычисляется по формуле

$$I = \log_2(1/0,25) = 2 \text{ бита.}$$

Ответ: 2 бита.

Пример 8

В группе 16 студентов – 14 юношей и 2 девушки. Найти объем сообщения «из группы выбрана одна девушка».

Решение:

Вероятность выбора одной девушки из группы студентов равна:

$$\frac{2}{16} = \frac{1}{8}.$$

Информационный объем сообщения равен:

$$\log_2 \left(\frac{1}{1/8} \right) = \log_2 2^3 = 3 \text{ бита.}$$

Ответ: 3 бита.

3.1.3 Представление и измерение звуковой информации

Звук представляет собой распространяющуюся в воздухе, воде или другой среде волну с непрерывно меняющейся интенсивностью и частотой.

Человек воспринимает звуковые волны (колебания воздуха) с помощью слуха в форме звука различных громкости и тона. Чем больше интенсивность звуковой волны, тем громче звук, чем больше частота волны, тем выше тон звука.

Человеческое ухо воспринимает звук с частотой от 20 колебаний в секунду (низкий звук) до 20 000 колебаний в секунду (высокий звук).

Человек может воспринимать звук в огромном диапазоне интенсивностей, в котором максимальная интенсивность больше минимальной в 10^{14} раз (в сто тысяч миллиардов раз). Для измерения громкости звука применяется специальная единица «децибел».

Временная дискретизация звука. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму с помощью временной дискретизации. Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука. Таким образом, непрерывная зависимость громкости звука от времени $A(t)$ заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность «ступенек» ([рисунок 2.1](#)).

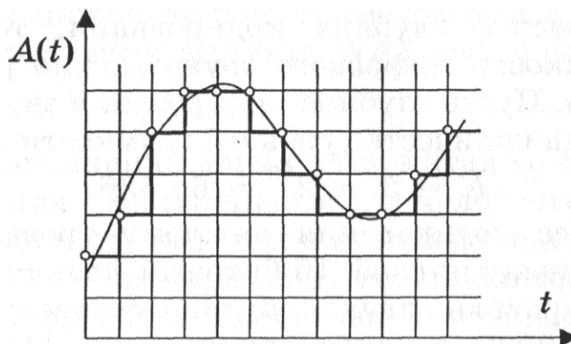


Рисунок 2.1 – Временная дискретизация звука

Частота дискретизации ν . Для записи аналогового звука и его преобразования в цифровую форму используется микрофон, подключенный к звуковой плате. Качество полученного цифрового звука зависит от количества измерений уровня громкости звука в единицу времени, т.е. частоты дискретизации. Чем большее количество измерений производится за 1 секунду, тем больше частота дискретизации.

Частота дискретизации звука – это количество измерений громкости звука за одну секунду.

Частота дискретизации звука может лежать в диапазоне от 8 000 до 48 000 измерений громкости звука за одну секунду.

Одно измерение в секунду соответствует частоте 1 Гц, 1 000 измерений в секунду – 1 кГц.

Глубина кодирования звука R . Каждой «ступеньке» присваивается определенное значение уровня громкости звука. Уровни громкости звука можно рассматривать как набор возможных состояний N , для кодирования которых необходимо определенное количество информации I , которое называется глубиной кодирования звука.

Глубина кодирования звука – это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.

Если известна глубина кодирования, то количество уровней громкости цифрового звука можно рассчитать по формуле $N = 2^I$. Пусть глубина кодирования звука составляет 16 бит, тогда количество уровней громкости звука равно:

$$N = 2^I = 2^{16} = 65\,536.$$

В процессе кодирования каждому уровню громкости звука присваивается свой 16-битовый двоичный код, наименьшему уровню звука будет соответствовать код 0000000000000000, а наибольшему – 1111111111111111.

Различные звуковые карты сегодня могут обеспечить 8- или 16-разрядную глубину кодирования звука.

Качество оцифрованного звука. Чем больше частота и глубина дискретизации звука, тем более качественным будет звучание оцифрованного звука. Самое низкое качество оцифрованного звука, соответствующее качеству телефонной связи, получается при частоте дискретизации 8 000 раз в секунду, глубине дискретизации 8 бит и записи одной звуковой дорожки (режим «моно»). Самое высокое качество оцифрованного звука, соответствующее качеству аудио-CD, достигается при частоте дискретизации 48 000 раз в секунду, глубине дискретизации 16 бит и записи двух звуковых дорожек (режим «стерео»).

Необходимо помнить, что чем выше качество цифрового звука, тем больше информационный объем звукового файла. Можно оценить информационный объем цифрового стерео звукового файла длительностью звучания 1 секунда при среднем качестве звука (16 бит, 24 000 измерений в секунду). Для этого глубину кодирования необходимо умножить на количество измерений в 1 секунду и умножить на 2 (стереозвук):

$$16 \text{ бит} \cdot 24000 \cdot 2 \text{ (канала)} = 768000 \text{ бит} = 96000 \text{ байт} = 93,75 \text{ Кбайт.}$$

Качество звука в дискретной форме может быть очень плохим (при 8 битах и 5,5 кГц) и очень высоким (при 16 битах и 44,1 кГц).

Рассмотрим примеры вычисления информационного объема звукового файла.

Пример 9

Оцените объем моно аудиофайла длительностью звучания 10 с при частоте дискретизации 22,05 кГц и разрешении 8 бит. Ответ запишите в Кбайтах.

Решение:

- 1) Для расчета объема моно аудиофайла используем формулу:

$$N = \nu \cdot t \cdot R,$$

где ν – частота дискретизации, Гц;
 R – глубина кодирования звука, бит;
 t – время звучания звукового файла, с.

- 2) Вычислим объем файла:

$$N = 22,05 \cdot 1000 \cdot 10 \cdot 8 = 1\,764\,000 \text{ бит} = 215,33 \text{ Кбайт.}$$

Ответ: 215,33 Кбайт.

Пример 10

Подсчитать время звучания звукового файла объемом 3,5 Мбайт, содержащего стерео запись с частотой дискретизации 44 100 Гц и разрядностью кода 16 бит на 1 измерение.

Решение:

- 1) Для расчета объема стерео аудиофайла используем формулу

$$N = 2 \cdot \nu \cdot t \cdot R,$$

где ν – частота дискретизации, Гц;
 R – глубина кодирования звука, бит;
 t – время звучания звукового файла, с.

- 2) Вычислим время звучания звукового файла:

$$t = \frac{N}{2 \cdot \nu \cdot R} = \frac{3,5 \cdot 1024}{2 \cdot 44100 \cdot 16} = 20,805 \text{ секунд.}$$

Ответ: 20,805 секунд.

3.2 Скорость передачи информации

Основной характеристикой процесса передачи информации является *скорость передачи информации*.

Скорость передачи данных – количество бит информации, передаваемой за единицу времени. Измеряется в битах в секунду (бит/с) и кратных единицах – Кбит/с и Мбит/с.

Объем переданной информации I вычисляется по формуле

$$I = q \cdot t,$$

где q – пропускная способность канала (бит/с),
 t – время передачи (с).

Пример 11

Скорость передачи данных через ADSL-соединение равна 128 000 бит/с. Передача файла через это соединение заняла 2 минуты. Определите размер файла в килобайтах.

Решение:

1) $128000 \cdot 120 = 1536000$ (бит) – количество бит, переданных за 2 минуты.

$$2) \frac{1536000}{8 \cdot 1024} = 1875 \text{ (Кбайт)}.$$

Ответ: 1 875 Килобайт.

Пример 12

Скорость передачи данных через ADSL-соединение равна 128 000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.

Решение:

1) $625 \cdot 1024 \cdot 8 = 512\,000$ (бит) – размер файла в битах;

$$2) \frac{512000}{12800} = 40 \text{ (с)}.$$

Ответ: 40 секунд.

3.3 Обработка графической информации

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов растра.

Графическая информация в компьютере представляется в трех форматах: растровая графика, векторная графика и фрактальная графика. С точки зрения вычисления объема информации, наибольший интерес представляет растровая графика. Рассмотрим этот графический формат более подробно.

Растровое изображение на экране монитора формируется из точек (пикселей).

Растр – прямоугольная сетка пикселей на экране. Объем растрового изображения зависит от глубины цвета.

Глубина цвета – количество бит, отводимое под пиксель.

Число цветов, воспроизводимых на экране монитора K , и глубину N (см. [таблицу 2.1](#)), можно найти по формуле

$$K = 2^N.$$

Таблица 2.1 – Расчет глубины цвета

Глубина цвета	Количество цветов	1 пиксель
1	2 (черный, белый)	1 бит
3	8	3 бита
4	16	4 бита
8	256	8 бит
16	65 536 (high color)	16 бит
24	16 777 216 (true color)	24 бита
32	4 294 967 296 (true color)	32 бита

Код цвета пикселя содержит информацию о доле каждого базового цвета. Чем больше глубина цвета, тем большее количество бит потребуется для кодирования одного пикселя и тем большее количество цветов можно использовать в изображении.

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов раstra.

Пример 13

Для хранения растрового изображения размером 64×64 пикселя отвели 1,5 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

Решение:

Вычислим, сколько бит отводится для хранения одного пикселя, учитывая, что в одном килобайте $1024 = 2^{10}$ байта, а в одном байте $8 = 2^3$ бита. Для этого информационный объем изображения, выраженный в битах, разделим на количество пикселей:

$$1,5 \cdot 1024 \cdot \frac{8}{64 \cdot 64} = 1,5 \cdot 2^{10} \cdot \frac{2^3}{2^6 \cdot 2^6} = 1,5 \cdot 2^{10+3-6-6} = 1,5 \cdot 2 = 3 \text{ (бит)}.$$

Итак, на один пиксель приходится три бита. Три бита позволяют закодировать максимум $2^3 = 8$ различных значений. Поэтому максимально возможное число цветов в палитре изображения равно 8.

Ответ: 8.

Пример 14

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 256×256 пикселей, если известно, что в изображении используется палитра из 216 цветов. Саму палитру хранить не нужно.

Решение:

Найдем минимальный объем памяти, необходимый для хранения одного пикселя. В изображении используется палитра из 2^{16} цветов, следовательно, одному пикселю может быть сопоставлен любой из 2^{16} возможных номеров цвета в палитре. Поэтому минимальный объем памяти для одного пикселя будет равен $\log_2 2^{16} = 16$ бит. Минимальный объем памяти, достаточный для хранения всего изображения будет равен

$$16 \cdot 256 \cdot 256 = 2^4 \cdot 2^8 \cdot 2^8 = 2^{20} \text{бит} = 2^{17} \text{байт} = 2^7 \text{Кбайт} = 128 \text{Кбайт}.$$

Ответ: 128 Кбайт.

Пример 15

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FFFFFF">`?

Решение:

В 24-битной системе кодирования цветов RGB, используемой для мониторов каждая из трех цветовых составляющих (красная, зеленая, синяя) может принимать значение от 0 до 255 и кодируется одним байтом (две шестнадцатеричные цифры). 0 соответствует полному отсутствию данной цветовой составляющей в пикселе, 255 ($FFFFFF_{16}$) – максимальной яркости данного цвета. Максимальная яркость всех трех цветовых сигналов $FFFFFF_{16}$ в RGB обеспечивает белый цвет пикселя. Минимальная (все цвета выключены) 000000_{16} – черный.

Ответ: белый.

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. В следующих задачах вычислите информационный объем текстового файла.

1) Подсчитайте количество информации в тексте «Один символ из алфавита мощностью 256 несет в тексте 8 бит информации. Такое количество информации называется байт» (кодировка Unicode).

2) Подсчитайте количество информации в набранном на компьютере сообщении, имеющем 30 строк по 50 символов.

3) Сколько мегабайт информации содержит сообщение объемом 223 бита?

4) Получено сообщение, информационный объем которого равен 32 бита. Чему равен этот объем в байтах?

5) Сколько дискет объемом 1,44 Мбайт необходимо для хранения энциклопедии из 60 томов по 500 страниц, на каждой из которых по две полосы, содержащей 80 строк из 45 знаков?

6) Сообщение, записанное буквами из 64-х символьного алфавита, содержит 20 символов. Какой объем информации оно несет?

7) Племя восточное имеет 32-х символьный алфавит. Племя западное использует 64-х символьный алфавит. Вожди племен обменивались письмами. Письмо восточного племени содержало 80 символов, а письмо западного племени – 70 символов. Сравните объемы информации, содержащейся в письмах.

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

9) Объем сообщения, содержащего 2048 символов, составил 1/512 Мбайт. Каков размер алфавита, с помощью которого записано сообщение?

10) Сколько символов содержит сообщение, записанное с помощью 16-ти символьного алфавита, если объем его составил 1/16 Мбайт?

11) Сколько килобайт составляет сообщение, содержащее 12 288 бит?

12) Сколько килобайт составит сообщение из 384 символов 16-ти символьного алфавита?

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

14) Сообщение занимает 3 страницы по 25 строк. В каждой строке записано по 60 символов. Сколько символов в использованном алфавите, если все сообщение содержит 1 125 байт?

15) Для записи сообщения использовался 64-х символьный алфавит. Каждая страница содержит 30 строк. Все сообщение содержит 8 775 байт информации и занимает 6 страниц. Сколько символов в строке?

16) Сообщение занимает 2 страницы и содержит 1/16 Кбайт информации. На каждой странице записано 256 символов. Какова мощность использованного алфавита?

17) Два сообщения содержат одинаковое количество символов. Количество информации в первом тексте в 1,5 раза больше, чем во втором. Сколько символов содержат алфавиты, с помощью которых записаны сообщения, если известно, что число символов в каждом алфавите не превышает 10 и на каждый символ приходится целое число бит?

18) Два сообщения содержат одинаковое количество информации. Количество символов в первом тексте в 2,5 раза меньше, чем во втором. Сколько символов содержат алфавиты, с помощью которых записаны сообщения, если известно, что размер каждого алфавита не превышает 32 символов и на каждый символ приходится целое число бит?

19) Выяснить, сколько бит информации несет каждое двузначное число (отвлекаясь от его конкретного числового значения).

20) В студенческой группе 16 студентов, 15 юношей и 1 девушка. Сколько бит информации содержится в сообщении «староста группы – девушка»?

21) В классе 24 ученика, из них 3 мальчика. Сколько бит информации содержится в сообщении «выбран один мальчик»?

22) Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего предложения: «Мой дядя самых честных правил, когда не в шутку занемог, он уважать себя заставил и лучше выдумать не мог».

23) Чему равно максимальное количество книг (каждая объемом 200 страниц, на каждой странице 60 строк, 80 символов в строке), полностью размещенных на лазерном диске емкостью 600 Мбайт?

24) Чему равен объем текстовой информации в сообщении на 40 страницах (на странице 40 строк и 80 символов в строке)?

25) Чему равно количество бит информации в сообщении «выбранный из урны шар является черным» (в урне 8 черных и 56 белых шаров)?

26) Чему равно количество бит информации в сообщении «пойманная в пруду рыба – карп» (всего в пруду 256 карасей, 44 щуки, 100 карпов)?

27) Каков информационный объем сообщения «Миша на олимпиаде по информатике занял одно из 16 мест»?

28) В магазине на 8 полках расставлены предметы, которые можно купить. Какое количество информации несет сообщение о номере полки, на которой находится интересующий нас товар?

29) Вы играете в шашки и сделали первый ход. Сколько информации при этом получил ваш противник?

30) Ученик за четверть получил 10 пятерок, 5 четверок, 3 тройки и 2 двойки. Рассчитайте вероятности получения каждой оценки и сделайте вывод о количестве информации, которую можно получить при получении сообщения о каждой оценке.

31) В корзине 8 чёрных и 24 белых шара. Сколько информации несёт сообщение о том, что достали чёрный шар? Белый?

32) В коробке лежат 64 цветных карандаша. Сообщение о том, что достали белый карандаш, несёт 4 бита информации. Сколько белых карандашей было в корзине?

33) В классе 30 человек. За контрольную работу по математике получено 6 пятёрок, 15 четвёрок, 8 троек и 1 двойка. Какое количество информации в сообщении о том, что Иванов получил четвёрку?

34) Сведения о сотруднике хранятся в виде строки из 2048 символов. Сколько дискет, емкостью 1,2 Мб необходимо, чтобы разместить сведения обо всех 8 192 сотрудниках?

2. В следующих задачах вычислите информационный объем звукового файла.

35) Звуковая плата производит двоичное кодирование аналогового звукового сигнала. Какое количество информации необходимо для кодирования каждого из 65 536 возможных уровней интенсивности сигнала?

36) Оценить информационный объем цифровых звуковых файлов длительностью 10 секунд при глубине кодирования и частоте дискретизации звукового сигнала, обеспечивающих минимальное качество звука: моно, 8 бит, 8000 измерений в секунду.

37) Определить длительность звукового файла, который уместится на диске 3,5" (учтите, что для хранения данных на такой диске выделяется 2 847 секторов объемом 512 байтов каждый) при высоком качестве звука: стерео, 16 битов, 48 000 измерений в секунду.

38) Объем звукового файла 5,25 Мбайт, разрядность звуковой платы – 16. Какова длительность звучания этого файла, записанного с частотой дискретизации 22,05 кГц?

39) Оценить информационный объем цифровых звуковых файлов длительностью 10 секунд при глубине кодирования и частоте дискретизации звукового сигнала, обеспечивающих максимальное качество звука: стерео, 16 бит, 48 000 измерений в секунду.

40) Подсчитать объем файла с 10 минутной речью записанного с частотой дискретизации 11 025 Гц и разрядностью кода 4 бита на 1 измерение.

41) Определить длительность звукового файла, который уместится на диске 3,5" (учтите, что для хранения данных на такой диске выделяется 2 847 секторов объемом 512 байт каждый) при низком качестве звука: моно, 8 битов, 8 000 измерений в секунду.

42) Одна минута записи цифрового аудиофайла занимает на диске 1,3 Мбайт, разрядность звуковой платы – 8. С какой частотой дискретизации записан звук?

43) Рассчитать объем звукового файла продолжительностью звучания в 25 сек., если частота дискретизации составила 44,1 кГц, а для записи значения звукового давления используется 16 бит.

44) Аналоговый звуковой сигнал был дискретизирован сначала с использованием звуковой карты с 16-разрядной глубиной кодирования звука (качество звучания аудио-CD), а затем с использованием звуковой карты с 8-разрядной глубиной кодирования (качество звучания радиотрансляции). Во сколько раз различаются информационные объемы оцифрованного звука?

45) Рассчитать время звучания моно аудиофайла, если при 8-битном кодировании и частоте дискретизации 32 кГц его объем равен 350 Кбайт.

46) Объем свободной памяти на диске – 0,01 Гб, разрядность звуковой платы – 16. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 44 100 Гц?

47) Объем свободной памяти на диске – 5,25 Мб, разрядность звуковой платы – 16. Какова длительность звучания цифрового аудиофайла с частотой дискретизации 22,05.

48) В распоряжении пользователя имеется память объемом 2,6 Мб. Необходимо записать цифровой аудиофайл с длительностью звучания 1 минута. Какой должна быть частота дискретизации и разрядность?

3. В следующих задачах вычислите скорость передачи информации.

49) Скорость передачи данных через ADSL-соединения = 128 000 бит/с. Передача текстового файла через это соединение заняла 1 минуту. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.

- 50) Информационное сообщение объемом 2,5 кбайт передается со скоростью 2 560 бит/мин. За сколько минут будет передано данное сообщение?
- 51) Скорость передачи данных через модемное соединение равна 56 Кбит/с. Передача текстового файла через это соединение заняла 12 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в кодировке Unicode.
- 52) Модем передает данные со скоростью 56 Кбит/с. Передача текстового файла заняла 4,5 мин. Определите, сколько страниц содержал переданный текст, если известно, что он был представлен в кодировке Unicode, а на одной странице – 3 072 символов.
- 53) Скорость передачи данных через модемное соединение равна 51 200 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.
- 54) Средняя скорость передачи данных с помощью модема равна 36 864 бит/с. Сколько секунд понадобится модему, чтобы передать 4 страницы текста в кодировке КОИ8, если считать, что на каждой странице в среднем 2 304 символа?
- 55) Скорость передачи данных через модемное соединение равна 4 096 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.
- 56) Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать цветное растровое изображение размером 800×600 пикселей, при условии, что в палитре 16 миллионов цветов?
- 57) Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать цветное растровое изображение размером 640 × 480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?
- 58) Известно, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 минут. Определите максимальный размер файла (в килобайтах), который может быть передан за время такого подключения, если модем передает информацию в среднем со скоростью 32 Кбит/с.
- 59) Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать 100 страниц текста в 30 строк по 60 символов каждая, при условии, что каждый символ кодируется одним байтом?
- 60) Сколько секунд потребуется модему, передающему сообщения со скоростью 4 400 бит/с, чтобы передать сообщение длиной 225 Кбайт?
- 61) Сколько секунд потребуется модему, передающему информацию со скоростью 32 000 бит/с, чтобы передать 16-цветное растровое изображение размером 800×600 пикселей, при условии, что в одном байте закодировано максимально возможное целое число пикселей?

62) Скорость передачи данных через ADSL-соединение равна 256 000 бит/с. Передача файла через данное соединение заняла 3 мин. Определите размер файла в килобайтах.

63) Скорость передачи данных через ADSL-соединение равна 512 000 бит/с. Передача файла через данное соединение заняла 1 мин. Определите размер файла в килобайтах.

64) Известно, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 мин. Определите максимальный размер файла (в килобайтах), который может быть передан за время такого подключения, если модем передает информацию в среднем со скоростью 32 Кбит/с?

65) Информационное сообщение объемом 2,5 кбайт передается со скоростью 2 560 бит/мин. За сколько минут будет передано данное сообщение?

66) Какой объем информации можно передать за 10 мин по каналу с пропускной способностью 5 кбайт/с

67) Передача данных через ADSL заняла 5 минут. За это время был передан файл, размер которого 3 000 Кбайт. Определите минимальную скорость (бит/с) (пропускную способность канала), при которой такая передача возможна.

68) Какое количество байтов будет передаваться за 1 секунду по каналу с пропускной способностью 100 Мбит/с?

69) Через ADSL-соединение файл размером 2500 Кбайт передавался 40 секунд. Сколько секунд потребуется для передачи файла размером 2 750 Кбайт.

70) Модем передаёт данные со скоростью 56 Кбит/сек. Передача текстового файла заняла 4,5 минут. Определите, сколько страниц содержал переданный текст, если известно, что он был представлен в кодировке Unicode, а на одной странице – 3 072 символа.

71) Скорость передачи данных через ADSL-соединение равна 1 024 000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.

72) Скорость передачи данных через ADSL-соединение равна 256 000 бит/с. Передача файла через это соединение заняла 2 минуты. Определите размер файла в килобайтах.

73) Сколько секунд потребуется обычному модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать цветное растровое изображение размером 640×480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?

74) Скорость передачи данных через модемное соединение равна 51 200 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.

75) Скорость передачи данных через ADSL-соединение равна 128 000 бит/с. Передача текстового файла через это соединение заняла 1 минуту. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.

76) Информационное сообщение объемом 2,5 Кбайт передается со скоростью 2 560 бит/мин. За сколько минут будет передано данное сообщение?

77) Модем передает данные со скоростью 7 680 бит/с. Передача текстового файла заняла 1,5 мин. Определите, сколько страниц содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode, а на одной странице – 400 символов.

78) Средняя скорость передачи данных с помощью модема равна 36 864 бит/с. Сколько секунд понадобится модему, чтобы передать 4 страницы текста в 8-битной кодировке КОИ8, если считать, что на каждой странице в среднем 2 304 символа?

79) Скорость передачи данных через модемное соединение равна 4 096 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.

80) Передача данных через ADSL-соединение заняла 2 минуты. За это время был передан файл, размер которого 3 750 Кбайт. Определите минимальную скорость (бит/с), при которой такая передача возможна.

81) Сколько секунд потребуется модему, передающему сообщения со скоростью 14 400 бит/с, чтобы передать сообщение объемом 225 Кбайт?

82) Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать 100 страниц текста в 30 строк по 16 символов каждая, при условии, что каждый символ кодируется 1 байтом?

83) Предположим, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 минут. Определите максимальный размер файла в Кбайтах, который может быть передан за время такого подключения, если модем передает информация в среднем со скоростью 32 Кбит/с.

4. Решите задачу на вычисление информационного объема растрового изображения.

84) Растровый файл, содержащий черно-белый рисунок, имеет объем 300 байт. Какой размер может иметь рисунок в пикселях?

85) Сколько информации содержится в картинке экрана с разрешающей способностью 800×600 пикселей и 16 цветами?

86) В процессе преобразования растрового графического файла количество цветов уменьшилось с 65 536 до 256. Во сколько раз уменьшился информационный объем файла?

87) Для хранения области экрана монитора размером 256×128 точек выделено 32 Кбайт оперативной памяти. Какое максимальное количество цветов можно использовать для раскраски точек?

88) Для хранения области экрана монитора размером 512×256 точек выделено 64 Кбайт оперативной памяти. Какое максимальное количество цветов можно использовать для раскраски точек?

89) Область экрана имеет размеры 512×128 точек, и каждая точка может иметь один из 65 536 оттенков. Каков минимальный объем памяти (Кбайт), необходимый для хранения этой области?

- 90) Растровый файл, содержащий черно-белый рисунок, имеет объем 400 байт. Какой размер может иметь рисунок в пикселях?
- 91) Область экрана имеет размеры 256×256 точек, и каждая точка может иметь один из 256 оттенков. Каков минимальный объем памяти (Кбайт), необходимый для хранения этой области?
- 92) Цветное растровое графическое изображение, палитра которого включает в себя 65 536 цветов, имеет размер 100×100 точек (пикселей). Какой объем видеопамяти компьютера (Кбайт) занимает это изображение?
- 93) Для хранения области экрана монитора размером точек выделено 128 Кбайт оперативной памяти. Какое максимальное количество цветов можно использовать для раскраски точек?
- 94) В процессе преобразования растрового графического файла количество цветов уменьшилось с 65 536 до 16. Во сколько раз уменьшился информационный объем файла?
- 95) Сколько информации содержится в картинке экрана с разрешающей способностью 800×600 пикселей и 256 цветами?
- 96) Сколько байт памяти необходимо, чтобы закодировать изображение на экране компьютерного монитора, который может отображать 1 280 точек по горизонтали и 1 024 точек по вертикали при 256 цветах?
- 97) Область экрана имеет размеры 512×128 точек, и каждая точка может иметь один из 65 536 оттенков. Каков минимальный объем памяти (Кбайт), необходимый для хранения этой области?
- 98) Растровый файл, содержащий черно-белый рисунок, имеет объем 800 байт. Какой размер может иметь рисунок в пикселях?
- 99) Сколько информации содержится в картинке экрана с разрешающей способностью 800×600 пикселей и 65 536 цветами?
- 100) В процессе преобразования растрового графического файла количество цветов уменьшилось с 65 536 до 8. Во сколько раз уменьшился информационный объем файла?
- 101) Видеопамять имеет объем, в котором может храниться 4-х цветное изображение размером 640×480 . какого размера изображение можно хранить в том же объеме видеопамяти, если оно будет использовать 256-цветную палитру?
- 102) Для хранения области экрана монитора размером 256×128 точек выделено 64 Кбайт оперативной памяти. Какое максимальное количество цветов можно использовать для раскраски точек?
- 103) Для хранения области экрана монитора размером 512×256 точек выделено 128 Кбайт оперативной памяти. Какое максимальное количество цветов можно использовать для раскраски точек?
- 104) Цветное растровое графическое изображение, палитра которого включает в себя 8 цветов, имеет размер 100×100 точек (пикселей). Какой объем видеопамяти компьютера (Кбайт) занимает это изображение?
- 105) Сколько байт памяти необходимо, чтобы закодировать изображение на экране компьютерного монитора, который может отображать 1 280 точек по горизонтали и 1 024 точек по вертикали при 16 цветах?

106) Цветное растровое графическое изображение, палитра которого включает в себя 256 цветов, имеет размер 100×100 точек (пикселей). Какой объем видеопамати компьютера (Кбайт) занимает это изображение?

107) Видеопамать имеет объем, в котором может храниться 4-х цветное изображение размером 300×200 . какого размера изображение можно хранить в том же объеме видеопамати, если оно будет использовать 16-цветную палитру?

108) Сколько цветов можно закодировать при использовании 16-битного способа представления цвета?

109) Растровый графический файл содержит цветное изображение с палитрой из 256 цветов размером 10×10 точек. Каков информационный объём файла?

110) Для хранения области экрана монитора размером 512×256 точек выделено 64 Кбайт оперативной памяти. Можно ли для раскраски точек использовать 256 цветов?

111) На экране с разрешающей способностью 640×200 высвечиваются только двухцветные изображения. Какой минимальный объем видеопамати необходим для хранения изображения?

112) Объем видеопамати 2 Мб, битовая глубина – 24, разрешающая способность дисплея – 640×480 . Какое максимальное количество страниц можно использовать при этих условиях?

113) Объем видеопамати 1 Мб, разрешающая способность дисплея – 800×600 . Какое максимальное количество цветов можно использовать при условии, что видеопамать делится на две страницы?

114) Объем видеопамати 256 Кб, количество используемых цветов – 16. Вычислите варианты разрешающей способности дисплея, при условии, что число страниц может быть равно 1, 2 или 4.

115) Объем видеопамати 2 Мб, битовая глубина – 12, разрешающая способность дисплея – 320×240 . Какое максимальное количество страниц можно использовать при этих условиях?

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: **ЛОГИЧЕСКИЕ ОСНОВЫ РАБОТЫ КОМПЬЮТЕРА.
ЛОГИЧЕСКИЕ ФУНКЦИИ. ТАБЛИЦЫ ИСТИННОСТИ. НКФ. НДФ.
УПРОЩЕНИЕ ЛОГИЧЕСКИХ ФУНКЦИЙ**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с понятием высказывание и основными логическими функциями.

Научиться:

- составлять простые и составные высказывания и определять их истинность;
- считать таблицы истинности для различных логических функций;
- упрощать логические выражения.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Логические основы работы компьютера

Основу компьютера составляют элементарные логические схемы. Работа этих схем основана на законах и правилах алгебры логики, которая оперирует двумя понятиями: истинности и ложности высказывания.

3.1.1 Основные понятия алгебры логики

Алгебра логики (Булева алгебра) – это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Высказывания бывают простыми и сложными.

Простое высказывание (логическая переменная) содержит только одну простую мысль.

Сложное высказывание (логическая функция) содержит несколько простых мыслей, соединенных между собой с помощью логических операций.

Логические переменные обычно обозначаются буквами латинского алфавита: $A, B, C \dots$

Основными логическими операциями являются:

- отрицание (инверсия);
- умножение (конъюнкция);
- сложение (дизъюнкция);

Дополнительными логическими операциями являются:

- следование (импликация);
- равнозначность (эквиваленция).

Значение сложного высказывания (**логической функции**) можно определить с помощью таблицы истинности.

Таблица истинности – это таблица, в которой перечислены все возможные значения входящих логических переменных и соответствующие им значения функции.

3.1.2 Логические операции

Для реализации любых логических операций над двоичными сигналами достаточно элементов трех типов – элементов, реализующих три основные логические операции: **И**, **ИЛИ**, **НЕ**.

Отрицание (инверсия)

Операция, выражаемая словом **НЕ**, называется **инверсией** или **отрицанием** и обозначается чертой над высказыванием \bar{A} . Применяется к одной переменной или логическому выражению.

Обозначается: не A , \bar{A} , $\neg A$, $not A$.

Таблица истинности:

A	\bar{A}
0	1
1	0

Высказывание \bar{A} истинно, когда A ложно, и ложно, когда A истинно.

Пример:

«Луна — спутник Земли» (A)

«Луна — не спутник Земли» (\bar{A})

Логическое сложение (дизъюнкция)

Соответствует союзу **ИЛИ**. Применяется к двум переменным или логическим выражениям.

Обозначается: $+$, или, \vee , *or*.

Таблица истинности:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Дизъюнкция истинна, если хотя бы одно из высказываний истинно.

Логическое умножение (конъюнкция)

Соответствует союзу **И**. Применяется к двум переменным или логическим выражениям.

Обозначается: $*$, и, \wedge , $\&$, *and*.

Таблица истинности:

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Конъюнкция истинна, тогда и только тогда, когда оба высказывания истинны.

Любое сложное высказывание можно записать с помощью основных логических операций.

Логическое следование (импликация)

Соответствует речевому обороту **ЕСЛИ ... ТО**. Применяется к двум переменным или логическим выражениям.

Обозначается: \rightarrow , \Rightarrow .

Таблица истинности:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Импликация истинна всегда, за исключением случая, когда A истинно, а B ложно.

Равнозначность (эквиваленция)

Соответствует речевым оборотам: **ЭКВИВАЛЕНТНО**, **НЕОБХОДИМО И ДОСТАТОЧНО**, **ТОГДА И ТОЛЬКО ТОГДА**. Применяется к двум переменным или логическим выражениям.

Обозначается: $=$, \leftrightarrow , \Leftrightarrow .

Таблица истинности:

A	B	$A \Leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Эквиваленция истинна тогда и только тогда, когда оба высказывания одновременно либо истинны, либо ложны.

Порядок выполнения логических операций:

1. операция в скобках;
2. \bar{A} , $\neg A$ – логическое отрицание;
3. \times , $\&$, \wedge – логическое умножение;
4. $+$, \vee – логическое сложение;
5. \rightarrow , \Rightarrow – импликация;
6. $=$, \Leftrightarrow , \leftrightarrow – эквиваленция.

Пример 1

Постройте таблицу истинности для логической функции $F = \bar{X} \vee X \vee Y$.

Решение:

1) Подсчитаем количество переменных в функции. В нашей функции две переменные – X и Y .

2) Построим таблицу для двух переменных и впишем все возможные сочетания значений двух переменных:

X	Y
0	0
0	1
1	0
1	1

3) Расставим в логической функции порядок выполнения операций.

$$\overset{1}{\bar{X}} \overset{2}{\vee} \overset{3}{X} \vee Y$$

4) Согласно полученному порядку впишем операции в таблицу истинности и подсчитаем значения. Значения логических операций выписываем из таблиц истинности для каждой конкретной логической операции.

X	Y	\bar{X}	$\bar{X} \vee X$	$\bar{X} \vee X \vee Y$
0	0	1	1	1
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

5) Согласно полученному результату делаем вывод.

Вывод: функция тождественно истинна.

Так как в последнем столбце значения функции – все единицы, то функция тождественно истинна. Таким образом, функция принимает значение истина (1) при любых значениях входящих в нее переменных X и Y .

Если в последнем столбце будут все нули, то функция тождественно ложна. Таким образом, функция принимает значение ложь (0) при любых значениях входящих в нее переменных X и Y .

Если в последнем столбце значения функции – набор нулей и единиц, то функция не является ни тождественно ложной, ни тождественно истинной. В таком случае выписываются значения входящих в функцию переменных, при которых функция принимает значения истина (1).

3.1.3 Тождества и законы алгебры логики

Для упрощения сложных логических выражений применяются тождества и законы алгебры логики.

Тождества алгебры логики

Тождества логического сложения	Тождества логического умножения
$A \vee 0 = A$	$A \wedge 0 = 0$
$A \vee 1 = 1$	$A \wedge 1 = A$
$A \vee A = A$	$A \wedge A = A$
$A \vee \bar{A} = 1$	$A \wedge \bar{A} = 0$
$\bar{\bar{A}} = A$ – (двойное отрицание)	

Законы алгебры логики

Переместительный закон	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C)$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$(A \vee B) \wedge C = A \wedge C \vee B \wedge C$ $A \wedge B \vee C = (A \vee C) \wedge (B \vee C)$
Законы де Моргана	$\overline{A \vee B} = \bar{A} \wedge \bar{B}$ $\overline{A \wedge B} = \bar{A} \vee \bar{B}$ $A \rightarrow B = \bar{B} \rightarrow \bar{A} = \bar{A} \vee B$ $A \Leftrightarrow B = A \wedge B \vee \bar{A} \wedge \bar{B} = (\bar{A} \vee B) \wedge (A \vee \bar{B})$

Рассмотрим применение законов и тождеств алгебры логики для упрощения логических функций.

Пример 2

Используя логические законы и тождества, упростите логическое выражение: $A \wedge (A \vee B) \wedge (B \wedge \bar{B})$.

Решение:

1) Первую скобку раскроем по распределительному закону, во второй скобке воспользуемся тождеством логического умножения, получим:

$$A \wedge (A \vee B) \wedge (B \wedge \bar{B}) = (A \wedge A \vee A \wedge B) \wedge 0 = (A \vee A \wedge B) \wedge 0.$$

Можно и далее преобразовывать выражение в скобках. Но согласно тождеству $A \wedge 0 = 0$,

$$A \wedge (A \vee B) \wedge (B \wedge \bar{B}) = (A \wedge A \vee A \wedge B) \wedge 0 = (A \vee A \wedge B) \wedge 0 = 0.$$

3.3 Формы логических функций

Одна и та же логическая функция может быть записана различным образом.

Для исключения неоднозначности записи логические функции представляют в унифицированных формах. Такими формами являются: **дизъюнктивная** и **конъюнктивная**. В них используются элементарные дизъюнкции и конъюнкции.

Элементарной называется **конъюнкция**, в которую входят только переменные и их отрицания.

Например: x_1x_2 ; $x_1\bar{x}_2$; \bar{x}_1x_2 ; $x_1x_2x_3$; $x_1\bar{x}_2\bar{x}_3$.

Элементарной называется **дизъюнкция**, представляющая собой логическую сумму переменных и их отрицаний.

Например: $x_1 + x_2$; $x_1 + \bar{x}_2$; $x_1 + \bar{x}_2 + \bar{x}_3$.

В элементарные конъюнкции (дизъюнкции) не могут входить одинаковые переменные, а также переменные с их отрицаниями. Такие дизъюнкции (конъюнкции) должны преобразовываться. При этом они упрощаются, а также превращаются в 0 или 1.

Например: $x_1x_1x_1 = x_1$; $x_1\bar{x}_1 = 0$; $x_1 + x_1 + x_1 = x_1$; $x_2 + \bar{x}_2 = 1$.

Правильность преобразований может быть проверена подстановкой значений переменных. Элементарная конъюнкция (дизъюнкция) может характеризоваться рангом, равным количеству переменных в конъюнкции (дизъюнкции). Понятия элементарной конъюнкции и дизъюнкции позволяют до-

статочно просто определить дизъюнктивную и конъюнктивную формы записи логических функций.

Дизъюнктивная нормальная форма (ДНФ) – это форма, в которой логическая функция представляется в виде дизъюнкции элементарных конъюнкций, например:

$$F = x_1x_2 + \bar{x}_1x_3 + x_1x_2x_3.$$

Конъюнктивной нормальной формой (КНФ) называется такая форма, в которой функция представляется в виде конъюнкции элементарных дизъюнкций, например:

$$F = (x_1 + \bar{x}_2) \cdot (\bar{x}_1 + x_2 + x_3).$$

Использование нормальных форм не устраняет полностью неоднозначности записи логических функций. Поэтому среди нормальных форм выделяются такие, в которых функции записываются единственным образом. Их называют совершенными. Применяются **совершенная дизъюнктивная и совершенная конъюнктивная нормальные формы (СДНФ и СКНФ)**.

Формы СДНФ и СКНФ имеют две отличительные особенности:

- все элементарные конъюнкции и дизъюнкции имеют одинаковый ранг;
- в элементарные конъюнкции (дизъюнкции) входят все те переменные или их отрицания, от которых зависит функция.

Функция в СДНФ и СКНФ обычно записываются по таблицам истинности по определенным правилам (рассматриваются в лабораторной работе № 4).

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Постройте таблицы истинности для следующих логических функций и сделайте вывод об истинности функций:

- | | | |
|---|--|---|
| 1) $\bar{A} \wedge \bar{B} \rightarrow C$ | 9) $(\bar{A} \rightarrow C) \wedge D$ | 17) $A \Leftrightarrow B \vee C$ |
| 2) $A \wedge \bar{B} \rightarrow C$ | 10) $A \Leftrightarrow B \wedge C$ | 18) $(A \Leftrightarrow \bar{B}) \vee C$ |
| 3) $\bar{A} \wedge (B \rightarrow C)$ | 11) $A \vee (C \Leftrightarrow B)$ | 19) $A \vee (C \Leftrightarrow \bar{B})$ |
| 4) $A \wedge (B \Leftrightarrow C)$ | 12) $A \wedge (\bar{B} \Leftrightarrow C)$ | 20) $\overline{A \Leftrightarrow \bar{B} \wedge C}$ |
| 5) $\bar{A} \wedge \bar{B} \rightarrow C$ | 13) $(A \vee \bar{B}) \wedge C$ | 21) $(A \vee \bar{B}) \wedge \bar{C}$ |
| 6) $\bar{A} \vee (B \Leftrightarrow C)$ | 14) $B \wedge C \wedge \bar{A}$ | 22) $\overline{B \vee C} \vee \bar{A}$ |
| 7) $\overline{A \vee B} \rightarrow C$ | 15) $B \vee C \vee \bar{A}$ | 23) $B \vee \bar{C} \wedge \bar{A}$ |
| 8) $B \wedge C \vee \bar{A}$ | 16) $B \vee C \wedge \bar{A}$ | 24) $(\bar{A} \rightarrow C) \wedge \bar{D}$ |

2. Используя логические законы и тождества, упростите следующие логические выражения:

$$1) A \wedge B \vee A \wedge \bar{B}$$

$$2) \overline{(X \wedge (\bar{X} \wedge \bar{Y}))}$$

$$3) \bar{A} \wedge B \vee A \wedge B$$

$$4) \bar{X} \vee \overline{(\bar{Y} \vee X)}$$

$$5) (A \vee B) \wedge (\bar{A} \vee B)$$

$$6) \bar{B} \wedge (A \vee \overline{A \wedge B})$$

$$7) (A \vee B) \wedge (A \vee \bar{B})$$

$$8) \overline{\bar{X} \wedge \bar{Y}} \vee \bar{X}$$

$$9) \bar{X} \wedge \bar{Y} \vee \bar{X} \wedge Y \vee \bar{X} \wedge Y$$

$$10) \bar{X} \vee \bar{X} \vee \bar{Y} \vee \bar{Y} \wedge \bar{X} \wedge \bar{Y}$$

$$11) \overline{X \vee Y \vee \bar{X} \wedge \bar{Y}} \wedge \bar{Y} \vee \bar{X}$$

$$12) \bar{A} \wedge \bar{B} \vee A \wedge \bar{B}$$

$$13) A \wedge B \wedge \bar{C} \vee \bar{B} \vee \bar{C}$$

$$14) A \wedge A \wedge C \vee A \wedge C \wedge B \vee (A \rightarrow B) \wedge C \vee C$$

$$15) \overline{(A \wedge B \wedge \bar{C} \vee C)} \vee \bar{B} \vee C$$

$$16) A \wedge B \wedge (C \rightarrow D) \vee B \wedge \bar{C} \wedge A \vee B \vee B \wedge A$$

$$17) \bar{B} \wedge \bar{A} \vee A \wedge B \vee (A \rightarrow B) \vee \bar{A} \wedge \bar{B}$$

$$18) \bar{A} \wedge B \vee \bar{B} \wedge A \vee \bar{A} \wedge (B \rightarrow A)$$

$$19) A \wedge B \wedge (\bar{C} \rightarrow \bar{D}) \vee B \vee B \wedge A$$

$$20) \bar{B} \wedge \bar{A} \vee A \wedge B \vee (\bar{A} \rightarrow \bar{B}) \vee \bar{A} \wedge \bar{B}$$

$$21) \bar{X} \wedge (\bar{Y} \rightarrow \bar{X}) \wedge Y \vee \bar{X} \wedge Y$$

$$22) (A \Leftrightarrow B) \vee \bar{A} \wedge B$$

$$23) (A \Leftrightarrow C) \vee \bar{A} \wedge C \vee (\bar{A} \vee \bar{C})$$

$$24) B \wedge \bar{A} \vee \bar{B} \wedge A$$

ЛАБОРАТОРНАЯ РАБОТА № 4

**Тема: ЛОГИЧЕСКИЕ ОСНОВЫ РАБОТЫ КОМПЬЮТЕРА.
ОСНОВНЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ. ЛОГИЧЕСКИЕ СХЕМЫ**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

– с основными логическими элементами.

Научиться:

– составлять логические схемы.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Основные логические элементы

Средством обработки двоичных сигналов в ЭВМ являются логические элементы.

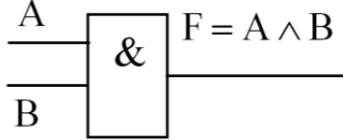
Для реализации любых логических операций над двоичными сигналами достаточно элементов трех типов – элементов, реализующих три основные логические операции: **И**, **ИЛИ**, **НЕ**.

Логические элементы (вентили) – это электронные схемы с одним или несколькими входами и одним выходом, через которые проходят электрические сигналы, представляющие цифры 0 и 1.

1. Элемент **НЕ (инвертор)**. У инвертора один вход и один выход. Сигнал на выходе появится, если на входе его нет, и наоборот. Инвертор реализует логическую операцию **инверсия (логическое отрицание)**.

Функция	Таблица истинности	Логический элемент						
$F = \bar{A}$	<table><tr><td>A</td><td>F</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0	
A	F							
0	1							
1	0							

2. Элемент **И** (**конъюнктор**). Элемент **И** имеет не менее двух входов и один выход. **A** и **B** – входные сигналы, **F** – выходной сигнал. Конъюнктор реализует логическую операцию **конъюнкция** (**логическое умножение**).

Функция	Таблица истинности	Логический элемент															
$F = A \wedge B$	<table border="1"> <tr> <th>A</th><th>B</th><th>F</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	F															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

3. Элемент **ИЛИ** (**дизъюнктор**). Элемент **ИЛИ** имеет не менее двух входов и один выход. **A** и **B** – входные сигналы, **F** – выходной сигнал. Дизъюнктор реализует логическую операцию **дизъюнкция** (**логическое сложение**).

Функция	Таблица истинности	Логический элемент															
$F = A \vee B$	<table border="1"> <tr> <th>A</th><th>B</th><th>F</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	F															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

С помощью логических элементов **НЕ**, **И**, **ИЛИ** можно реализовать (собрать как из конструктора) типовые функциональные узлы (блоки) ЭВМ:

- триггеры;
- сумматоры;
- шифраторы;
- регистры;
- счетчики;
- дешифраторы.

Чтобы понять, как работает интересующее нас устройство, необходимо понять логику его работы, т.е. найти соответствие между входными и выходными сигналами, для этого:

1. Составить таблицу истинности.
2. По таблице записать логическую функцию.
3. Построить логическую схему.

3.2 Построение логических схем по заданной таблице истинности функции

1. Выписывается таблица истинности функции.
2. По данной таблице определяется логическая функция (формула) с помощью следующего метода, называемого дизъюнктивная совершенная нормальная форма (ДСНФ):
 - 2.1. В заданной таблице выбираются наборы переменных, при которых значение функции равно 1.
 - 2.2. Для каждого такого набора записываются конъюнкции (\wedge) всех входных переменных, имеющие значение 1. При этом те переменные, которые имеют значение 0, записываются с отрицанием.
 - 2.3. Все полученные конъюнкции объединяются знаком дизъюнкции (\vee). Это и будет искомая логическая функция, которую можно будет упростить (минимизировать) по законам Булевой алгебры.
3. По упрощенной логической функции строится логическая схема.

Пример 1

По заданной таблице истинности записать логическую функцию, упростить её и построить логическую схему.

X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0

Решение:

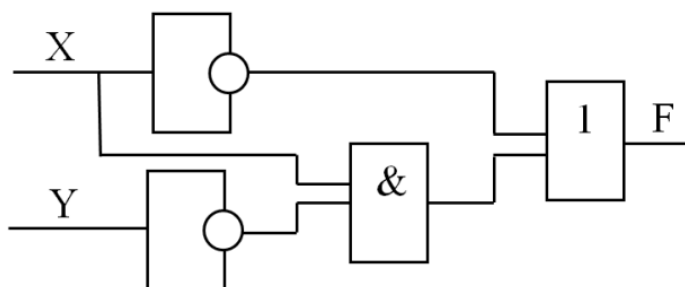
1. Запишем конъюнкцию для каждой строки, где значение функции равно 1 (первая, вторая и третья строки). Переменные, значения которых равны 0, запишем с отрицанием.

2. Объединив полученные конъюнкции дизъюнкцией, получим следующую логическую функцию.

$$F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y}).$$

3. Упростим: $F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y}) = \bar{X} \vee (X \wedge \bar{Y})$.

4. По полученной функции построим логическую схему:



Пример 2

Составить логическую схему, работа которой задана таблицей истинности:

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Решение:

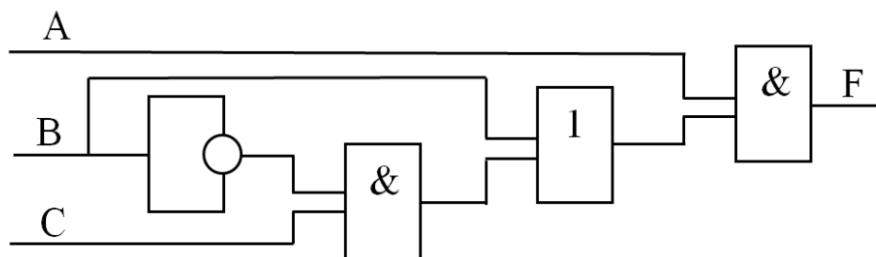
1. Составим логическую функцию по данной таблице:

$$F = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C).$$

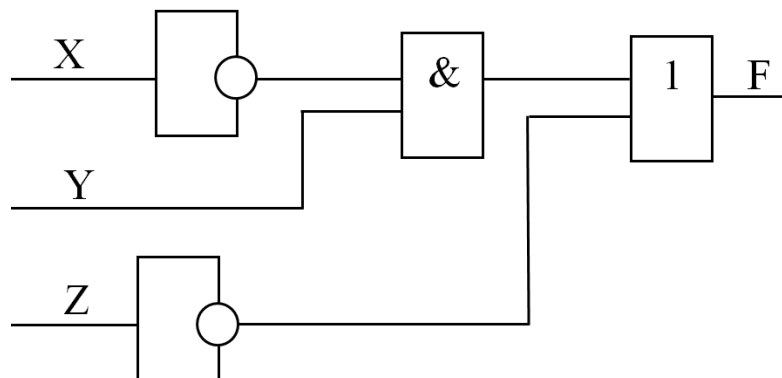
2. Упростим полученную функцию:

$$F = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C) = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge (\bar{C} \vee C)) = A \wedge \bar{B} \wedge C \vee A \wedge B = A \wedge (\bar{B} \wedge C \vee B).$$

3. По полученной функции построим логическую схему:

**Пример 3**

По заданной логической схеме запишите логическую функцию и постройте таблицу истинности:



Решение:

1. Для записи функции необходимо записать значения для каждого элемента схемы:

- 1) \bar{X}
- 2) $\bar{X} \wedge Y$
- 3) \bar{Z}
- 4) $(\bar{X} \wedge Y) \vee \bar{Z}$

2. Следовательно, получится функция:

$$F = (\bar{X} \wedge Y) \vee \bar{Z}.$$

3. По полученной функции построим таблицу истинности:

X	Y	Z	\bar{X}	$\bar{X} \wedge Y$	\bar{Z}	$(\bar{X} \wedge Y) \vee \bar{Z}$
0	0	0	1	0	1	0
0	0	1	1	0	0	0
0	1	0	1	1	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	1	0	0	0	1
1	1	0	0	0	1	1
1	1	1	0	0	0	1

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Составьте логические схемы для следующих логических выражений:

- 1) $A \wedge \bar{B} \wedge (B \rightarrow A)$
- 2) $\overline{(X \wedge (\bar{X} \rightarrow \bar{Y}))} \vee Y$
- 3) $\bar{A} \wedge B \vee A \wedge B \vee \bar{A}$
- 4) $X \vee (\bar{Y} \vee X) \rightarrow Y$
- 5) $(A \vee B) \wedge (\bar{A} \vee B)$
- 6) $\bar{B} \wedge (A \vee \bar{A} \wedge \bar{B}) \vee \bar{A} \wedge B$
- 7) $(A \vee B) \wedge (A \rightarrow \bar{B})$
- 8) $\bar{X} \wedge \bar{Y} \vee \bar{X} \wedge Z$
- 9) $\bar{X} \wedge \bar{Y} \vee \bar{X} \wedge Y \vee \bar{X} \wedge Z$
- 10) $\bar{X} \vee \bar{X} \vee \bar{Y} \vee \bar{Y} \wedge \bar{X} \wedge \bar{Z}$
- 11) $\bar{X} \vee Z \vee \bar{X} \wedge \bar{Y} \wedge \bar{Y} \vee \bar{X}$
- 12) $\bar{A} \wedge \bar{B} \vee C \wedge \bar{B}$
- 13) $A \wedge B \wedge \bar{C} \vee \bar{B} \vee C$
- 14) $\overline{(A \wedge C \wedge B \vee (A \rightarrow B) \wedge C)} \rightarrow \bar{C}$
- 15) $\overline{(A \wedge B \wedge \bar{C} \vee C)} \vee \bar{B} \vee C$
- 16) $A \wedge B \wedge (C \rightarrow D) \vee B \wedge \bar{C} \vee B \wedge A$
- 17) $\bar{B} \wedge \bar{A} \vee A \wedge B \vee (A \rightarrow B) \vee \bar{A} \wedge \bar{B}$
- 18) $\bar{A} \wedge B \vee \bar{B} \wedge A \vee \bar{A} \wedge (B \rightarrow A)$
- 19) $A \wedge B \wedge (\bar{C} \rightarrow \bar{D}) \vee B \vee B \wedge A$
- 20) $\bar{B} \wedge \bar{A} \vee (\bar{A} \rightarrow \bar{B}) \vee \bar{A} \wedge \bar{B}$
- 21) $\bar{X} \wedge (\bar{Y} \rightarrow \bar{X}) \wedge Y \vee \bar{X} \wedge Y$
- 22) $(A \Leftrightarrow B) \vee \bar{A} \wedge B$
- 23) $(A \Leftrightarrow C) \vee \bar{A} \wedge C \vee (\bar{A} \vee \bar{C})$
- 24) $B \wedge \bar{A} \vee \bar{B} \wedge A$

2. По заданным логическим схемам составить соответствующие им логические выражения и упростите их:

№	Схема
1	2
1)	
2)	
3)	
4)	
5)	

1	2
6)	
7)	
8)	
9)	
10)	

1	2
11)	<p>Logic circuit for problem 11): Inputs X, Y, and Z. X is inverted and then ANDed with Y. This result is ORed with Z. The output of this OR gate is ANDed with the output of an inverter that takes X as input. The final output is F.</p>
12)	<p>Logic circuit for problem 12): Inputs A, B, and C. A and B are ANDed together. C is inverted. The output of the first AND gate is ORed with the inverted C. The final output is F.</p>
13)	<p>Logic circuit for problem 13): Inputs A and B. A is inverted and then ANDed with B. The output of this AND gate is ORed with the output of an inverter that takes A as input. The result is ANDed with the output of an inverter that takes B as input. The final output is F.</p>
14)	<p>Logic circuit for problem 14): Inputs A and B. A is ANDed with B. The output of this AND gate is ORed with the output of an inverter that takes A as input. The result is ANDed with the output of an inverter that takes B as input. The final output is F.</p>

3. По заданным таблицам истинности запишите логические функции, составьте логические схемы:

№	Таблица	№	Таблица																																																			
1	2	3	4																																																			
1)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	0	0	1	1	1	0	1	1	1	0	13)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	1	0	1	0	1	0	1	1	1	1																					
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	0																																																				
0	1	1																																																				
1	0	1																																																				
1	1	0																																																				
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	1																																																				
0	1	0																																																				
1	0	1																																																				
1	1	1																																																				
2)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	1	0	1	1	1	0	0	1	1	1	14)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	1	0	1	0	1	0	1	1	1	0																					
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	1																																																				
0	1	1																																																				
1	0	0																																																				
1	1	1																																																				
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	1																																																				
0	1	0																																																				
1	0	1																																																				
1	1	0																																																				
3)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	1	0	1	1	1	0	0	1	1	0	15)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	0	0	1	1	1	0	0	1	1	1																					
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	1																																																				
0	1	1																																																				
1	0	0																																																				
1	1	0																																																				
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	0																																																				
0	1	1																																																				
1	0	0																																																				
1	1	1																																																				
4)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	<i>A</i>	<i>B</i>	<i>F</i>	0	0	0	0	1	1	1	0	1	1	1	1	16)	<table><tr><td><i>A</i></td><td><i>B</i></td><td><i>C</i></td><td><i>F</i></td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
<i>A</i>	<i>B</i>	<i>F</i>																																																				
0	0	0																																																				
0	1	1																																																				
1	0	1																																																				
1	1	1																																																				
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																			
0	0	0	0																																																			
0	0	1	1																																																			
0	1	0	0																																																			
0	1	1	0																																																			
1	0	0	1																																																			
1	0	1	1																																																			
1	1	0	1																																																			
1	1	1	1																																																			

1	2	3	4																																																			
5)	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	1	1	1	1	17)	<table><tr><th>A</th><th>B</th><th>C</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	C	F	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
A	B	F																																																				
0	0	0																																																				
0	1	0																																																				
1	0	1																																																				
1	1	1																																																				
A	B	C	F																																																			
0	0	0	0																																																			
0	0	1	0																																																			
0	1	0	0																																																			
0	1	1	0																																																			
1	0	0	1																																																			
1	0	1	1																																																			
1	1	0	1																																																			
1	1	1	1																																																			
6)	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	1	18)	<table><tr><th>A</th><th>B</th><th>C</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	C	F	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1
A	B	F																																																				
0	0	1																																																				
0	1	1																																																				
1	0	1																																																				
1	1	1																																																				
A	B	C	F																																																			
0	0	0	1																																																			
0	0	1	1																																																			
0	1	0	1																																																			
0	1	1	0																																																			
1	0	0	0																																																			
1	0	1	0																																																			
1	1	0	0																																																			
1	1	1	1																																																			
7)	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0	19)	<table><tr><th>A</th><th>B</th><th>C</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	C	F	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1
A	B	F																																																				
0	0	1																																																				
0	1	1																																																				
1	0	1																																																				
1	1	0																																																				
A	B	C	F																																																			
0	0	0	1																																																			
0	0	1	0																																																			
0	1	0	1																																																			
0	1	1	0																																																			
1	0	0	0																																																			
1	0	1	0																																																			
1	1	0	1																																																			
1	1	1	1																																																			

1	2	3	4																																																																								
8)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	0	20)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	1	0
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	1																																																																								
0	1	0	1																																																																								
0	1	1	1																																																																								
1	0	0	0																																																																								
1	0	1	0																																																																								
1	1	0	0																																																																								
1	1	1	0																																																																								
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	1																																																																								
0	1	0	0																																																																								
0	1	1	0																																																																								
1	0	0	1																																																																								
1	0	1	1																																																																								
1	1	0	0																																																																								
1	1	1	0																																																																								
9)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	21)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	0
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	1																																																																								
0	1	0	0																																																																								
0	1	1	0																																																																								
1	0	0	0																																																																								
1	0	1	0																																																																								
1	1	0	1																																																																								
1	1	1	1																																																																								
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	0																																																																								
0	1	0	1																																																																								
0	1	1	0																																																																								
1	0	0	1																																																																								
1	0	1	0																																																																								
1	1	0	1																																																																								
1	1	1	0																																																																								
10)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	22)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	0																																																																								
0	1	0	0																																																																								
0	1	1	1																																																																								
1	0	0	0																																																																								
1	0	1	0																																																																								
1	1	0	1																																																																								
1	1	1	1																																																																								
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	0																																																																								
0	0	1	0																																																																								
0	1	0	1																																																																								
0	1	1	1																																																																								
1	0	0	0																																																																								
1	0	1	0																																																																								
1	1	0	1																																																																								
1	1	1	1																																																																								

1	2	3	4																																																																								
11)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td></td></tr> <tr><td>0</td><td>0</td><td>1</td><td></td></tr> <tr><td>0</td><td>1</td><td>0</td><td></td></tr> <tr><td>0</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td>0</td><td></td></tr> <tr><td>1</td><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td></td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0		0	0	1		0	1	0		0	1	1		1	0	0		1	0	1		1	1	0		1	1	1		23)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	1	1	0
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0																																																																									
0	0	1																																																																									
0	1	0																																																																									
0	1	1																																																																									
1	0	0																																																																									
1	0	1																																																																									
1	1	0																																																																									
1	1	1																																																																									
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	1																																																																								
0	1	0	0																																																																								
0	1	1	0																																																																								
1	0	0	1																																																																								
1	0	1	0																																																																								
1	1	0	1																																																																								
1	1	1	0																																																																								
12)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	24)	<table> <tr><th><i>A</i></th><th><i>B</i></th><th><i>C</i></th><th><i>F</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	0	1	1	1	1
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	0																																																																								
0	1	0	1																																																																								
0	1	1	0																																																																								
1	0	0	1																																																																								
1	0	1	1																																																																								
1	1	0	1																																																																								
1	1	1	1																																																																								
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>																																																																								
0	0	0	1																																																																								
0	0	1	0																																																																								
0	1	0	1																																																																								
0	1	1	0																																																																								
1	0	0	1																																																																								
1	0	1	0																																																																								
1	1	0	0																																																																								
1	1	1	1																																																																								

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема: **ОСНОВНЫЕ ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ
ОПЕРАЦИОННОЙ СИСТЕМЫ (ОС) GOSLINUX**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с интерфейсом рабочего стола ОС GosLinux;
- с главным меню;
- с файловым менеджером Nautilus;
- с навигацией с помощью Обзорателя файлов;
- с файловой системой ОС GosLinux.

Научиться:

- запускать ОС GosLinux;
- работать с Главным меню;
- работать в Обзорателе файлов.

**2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК
ЕЁ ВЫПОЛНЕНИЯ**

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ****3.1 Основные понятия**

Операционная система (ОС) GosLinux – это многопользовательская, многозадачная, многотерминальная операционная система из семейства UNIX. Она предназначена для работы на серверах и рабочих станциях, обеспечивает подключение дополнительных терминалов и допускает в этом режиме использование графических оболочек.

ОС GosLinux является сетевой операционной системой для 32-х или 64-х разрядных платформ. Она не связана с конкретной моделью компьютеров. Её ядро реализовано на языке СИ, что позволяет достаточно легко переносить эту систему с одной платформы на другую. Система распространяется по лицензии GNU либо подобным свободным лицензиям, обеспечивается как коммерческое, так и свободное сопровождение через Internet. Поставка исходных модулей системы обеспечивает возможность адаптации прикладных программ в случае перехода на другую платформу и дает возможность контроля кодов, реализующих несанкционированный доступ.

Диалог с ОС осуществляется посредством командных интерпретаторов с использованием системных библиотек.

Каждая системная библиотека представляет собой набор программ, динамически вызываемых операционной системой.

Для удобства пользователей при работе с командными интерпретаторами используются интерактивные рабочие среды, представляющие пользователю удобный интерфейс для работы с ОС.

Если на вашем компьютере установлена ОС Windows, то ОС GosLinux можно установить на виртуальную машину. Где скачать и как установить виртуальную машину и ОС GosLinux показано в [Приложении А](#) и [Приложении Б](#) соответственно.

2.2 Загрузка ОС GosLinux

Для того чтобы загрузить GosLinux, достаточно просто включить компьютер. После загрузки на экране появится окно для выбора пользователя ([рисунок 5.1](#)).

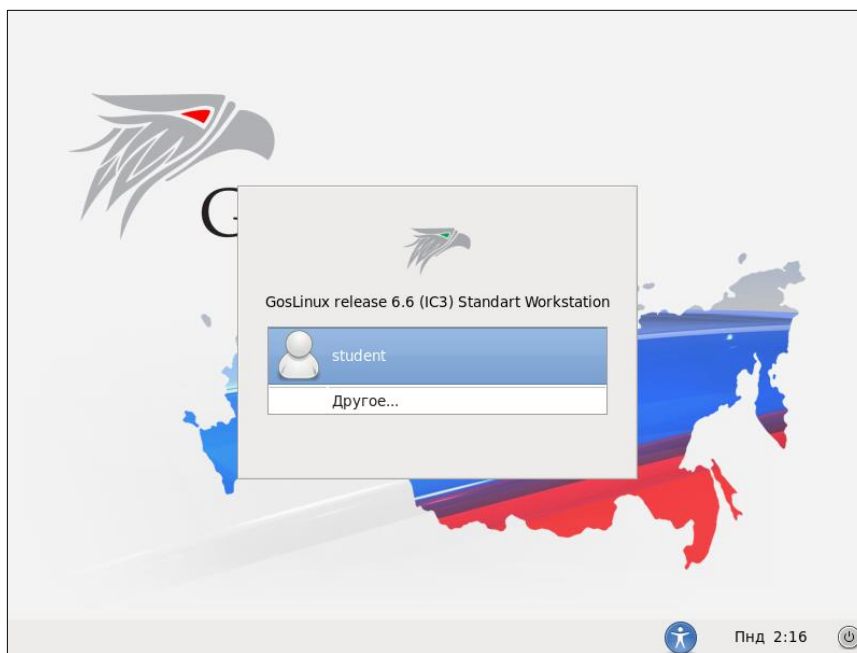


Рисунок 5.1 – Окно выбора пользователя

Выберите пользователя student. На экране появится диалоговое окно с полем ввода пароля пользователя student ([рисунок 5.2](#)). В этом окне в поле ввода введите пароль: student. В нижней части экрана, на панели расположен раскрывающийся список для выбора языка интерфейса. Выберите язык русский (Российская Федерация), если установлен другой язык. Щелкните по кнопке **Войти в систему**. Операционная система загрузится и на экране появится среда рабочего стола GNOME 2.82.2 ([рисунок 5.3](#)).

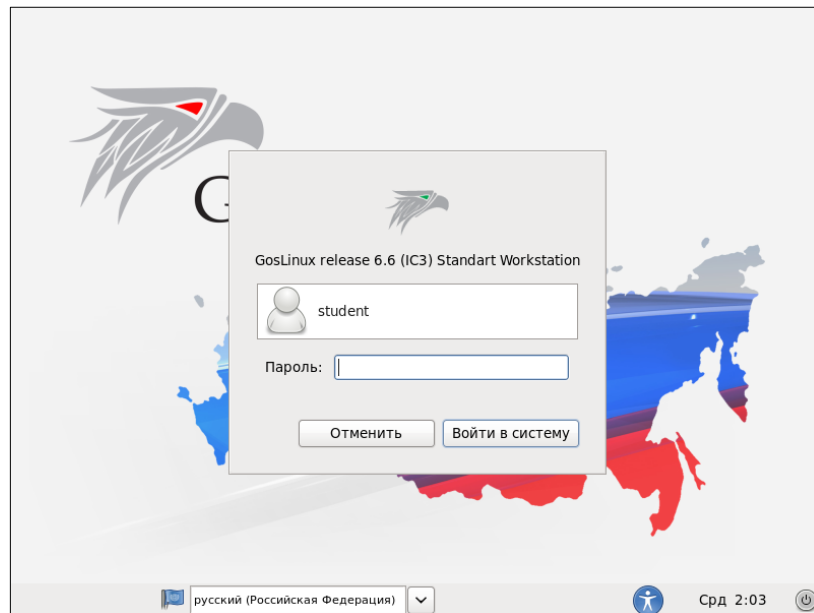


Рисунок 5.2 – Окно ввода пароля и список выбора языка интерфейса ОС

Среда рабочего стола состоит из следующих компонентов ([рисунок 5.3](#)):

1. Верхняя панель, на которой расположены: главное меню, панель быстрого запуска, область уведомлений, информация о текущей учетной записи.
2. Нижняя панель, на которой расположены: кнопки переключения между рабочими столами, корзина и кнопки открытых окон (приложений).
3. Рабочий стол.

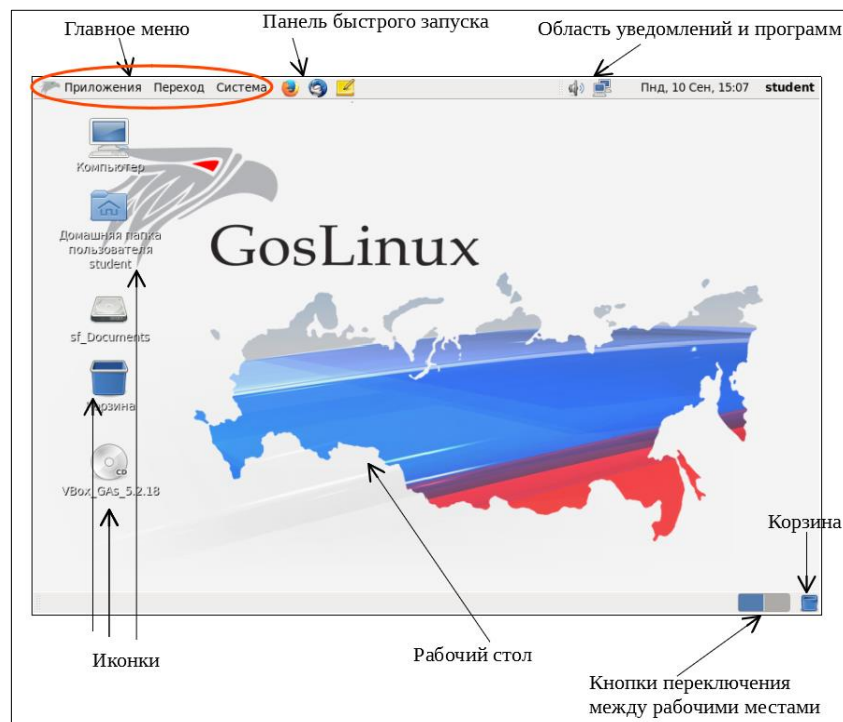


Рисунок 5.3 – Среда рабочего стола

2.3 Главное меню

На верхней панели слева находится **Главное меню**. Оно состоит из трех пунктов: **Приложения**, **Переход**, **Система**. Первый пункт – **Приложения** – это аналог пункта меню **Пуск – Программы** в Windows. В пункте **Приложения** программы отсортированы по категориям, что значительно облегчает их поиск ([рисунок 5.4](#)). Каждый раз, когда вы будете ставить новую программу, она появится в Главном меню.

Следующий пункт главного меню – **Переход** ([рисунок 5.5](#)). В этом меню содержатся ярлыки на быстрый переход к определенным каталогам в вашем домашнем каталоге, на примонтированные диски, сеть, последние документы и т.д.

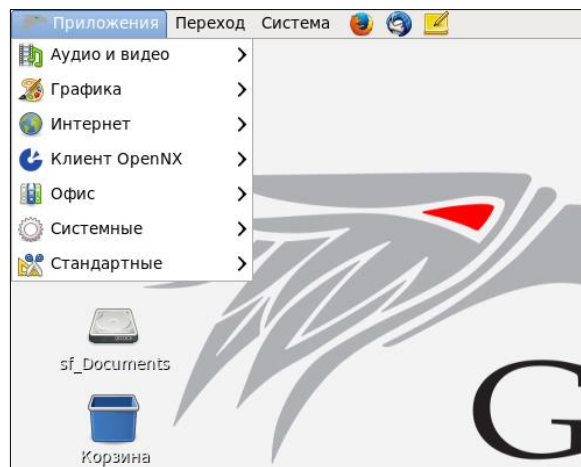


Рисунок 5.4 – Пункт главного меню Приложения

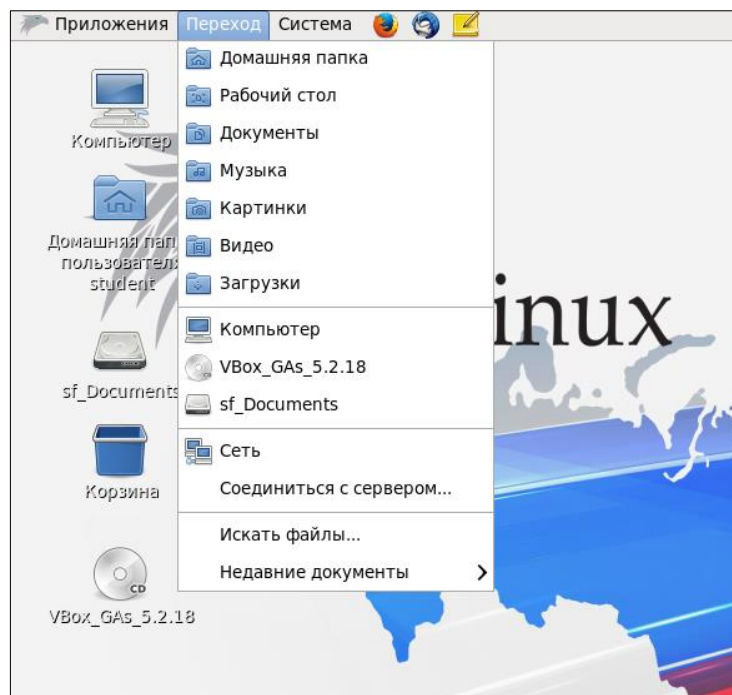


Рисунок 5.5 – Пункт главного меню Переход

Третий пункт меню – **Система** (аналог – панель управления Windows). Только здесь все управление разделено на две категории: параметры и администрирование. Разница между ними в том, что настройки из первой категории носят персональный характер, в то время как настройки из категории «администрирование» влияют на всю систему в целом.

Упражнение 1. В этом упражнении вы изучите содержимое главного меню.

1. Откройте программу **Калькулятор** и выполните вычисления: $\sqrt{3} * 45 - 34$. Для этого выполните команду **Приложения – Стандартные – Калькулятор**. Обратите внимание, что для вычисления корня квадратного калькулятор нужно переключить в научный вид, выполнив команду **Вид – Научный**.

2. Откройте домашнюю папку с помощью **Главного меню**. Для этого выполните команду **Переход – Домашняя папка**. На экране откроется домашняя папка текущего пользователя, в данном случае **Student**. Откройте папку **Документы**. Обратите внимание, что папка **Документы** открывается в отдельном окне ([рисунок 5.6](#)). Такой режим работы с окнами называется **пространственный**. Закройте все открытые папки.

3. Выполните команду **Система – Параметры – Управление файлами**. В открывшемся диалоговом окне на вкладке **Поведение** установите флажок **Всегда открывать папки в обозревателе**. Откройте снова папку **Документы**. Теперь домашняя папка открыта в окне **Обозревателя файлов** ([рисунок 5.7](#)).

4. Просмотрите информацию об операционной системе, выполнив команду **Система – О системе**. В появившемся диалоговом окне изучите содержимое всех вкладок. Закройте окно.

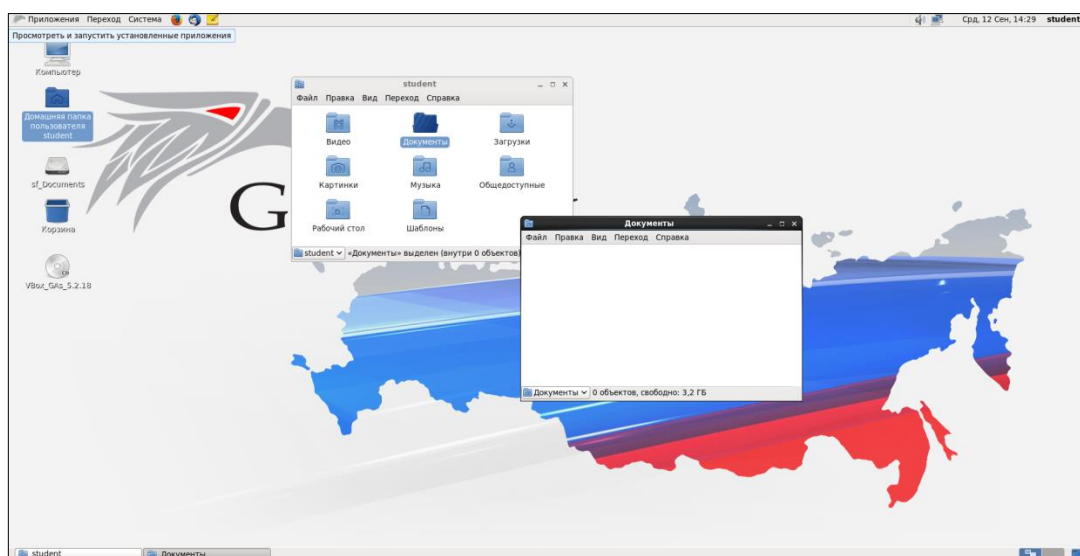


Рисунок 5.6 – Домашняя папка, открытая в отдельном окне

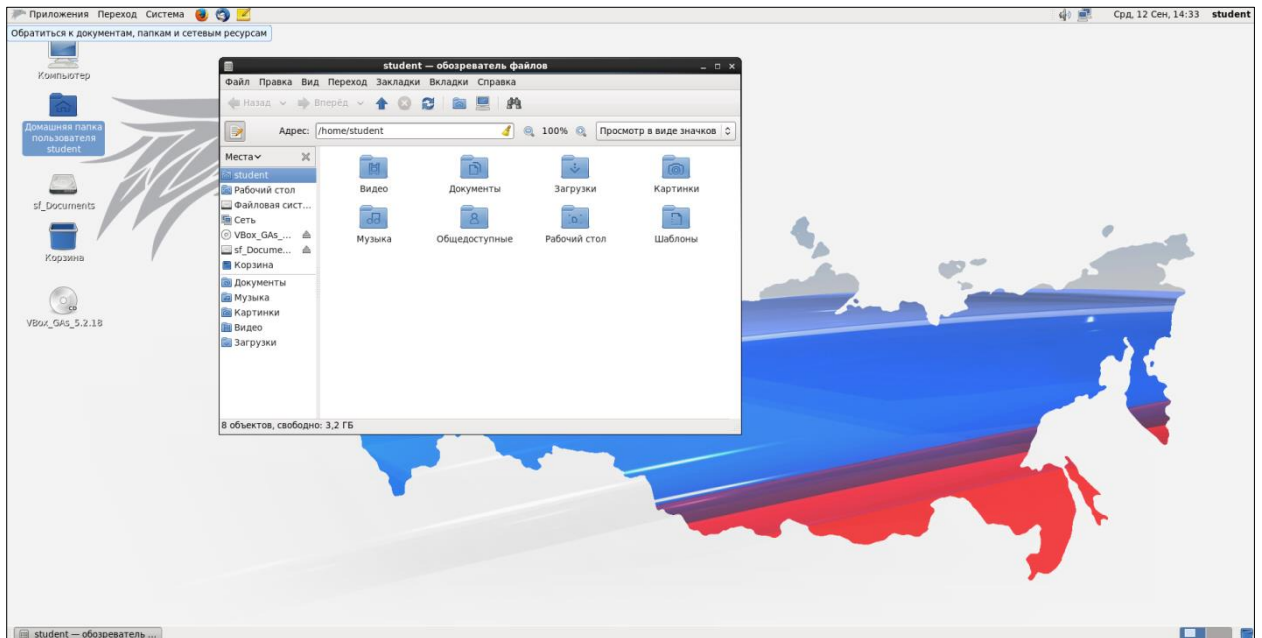


Рисунок 5.7 – Домашняя папка открытая в Обозревателе файлов

2.4 Рабочие места

Рабочие места позволяют выбирать окна, которые показываются на экране. Все рабочие места содержат один и тот же Рабочий стол, те же панели и то же меню, но на каждом Рабочем месте можно запускать свои приложения и открывать свои окна. Приложения останутся на своём рабочем месте, когда вы переключитесь на другое место. Переключаться между рабочими местами можно с помощью кнопок переключения (переключателя между рабочими местами) (см. [рисунок 5.3](#)).

Упражнение 2. В этом упражнении вы научитесь переключаться между рабочими местами.

1. Откройте домашнюю папку пользователя.
2. Переключитесь на второе рабочее место. Откройте приложение **Калькулятор**.
3. Вернитесь на первое рабочее место. Обратите внимание, что при переключении между рабочими местами, приложения и папки остаются открытыми.

2.5 Файловый менеджер Nautilus

Управление папками и файлами в ОС GosLinux выполняется с использованием файлового менеджера Nautilus. В ОС GosLinux он установлен как основной файловый менеджер, а также как менеджер рабочего стола.

С помощью файлового менеджера вы можете:

- подключать и отключать устройства для хранения данных (жесткие диски, сетевые диски, флеш носители, оптические диски и т.д.);

- работать с удаленными серверами (FTP, SSH, WebDAV, SMB);
- просматривать файлы и каталоги;
- просматривать миниатюры файлов (видео, изображения, PDF, DJVU, текстовые файлы);
- просматривать свойства файлов и каталогов (в том числе дополнительные свойства на отдельных вкладках с помощью сторонних приложений);
- создавать, изменять, удалять файлы и каталоги (в т.ч. с помощью шаблонов файлов размещенных в каталоге ~/Templates или ~/Шаблоны);
- запускать сценарии и приложения;
- искать файлы и каталоги по названиям;
- размещать файлы и каталоги на рабочем столе;
- записывать CD/DVD диски (при помощи Brasero);
- выполнять другие действия через дополнения и сценарии для Nautilus.

Файловый менеджер может работать в двух режимах: пространственном и режиме **Обозревателя файлов**.

2.6 Обозреватель файлов

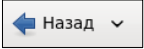
Обратиться к обозревателю файлов можно одним из следующих способов:

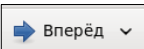
1. С помощью команды **Главного меню Приложения – Системные – Обозреватель файлов**.
2. С помощью команды контекстного меню **Открыть папку в обозревателе**.
3. Если в параметрах Nautilus (см. Упражнение 1, п. 3) указано всегда использовать режим обозревателя, то при двойном щелчке на любой папке будет открыто окно обозревателя.

Обозреватель файлов состоит из следующих частей (рисунок 5.8):

1. **Строка меню** – содержит команды, через которые можно выполнять различные задачи в файловом менеджере.




2. **Панель инструментов** – содержит кнопки, с помощью которых можно выполнять различные задачи в файловом менеджере.

–  – возвращает к предыдущему посещённому местоположению, рядом с кнопкой находится выпадающий список недавно посещённых адресов;

–  – выполняет действие, обратное действию кнопки Назад;

–  – поднимает на один уровень вверх, к родительской папке;

–  – обновляет содержимое текущей папки;

-  – открывает домашнюю папку;
-  – открывает папку Компьютер;
-  – открывает панель поиска.

3. **Панель адреса** – средство для перемещения по компьютеру. Строка адреса в файловом менеджере может содержать поле адреса, строку кнопок или поле поиска. Для переключения **Панели адреса** в вид «кнопки» или «текст» необходимо использовать **Кнопку переключения вида Панели адреса** ([рисунок 5.8](#)).

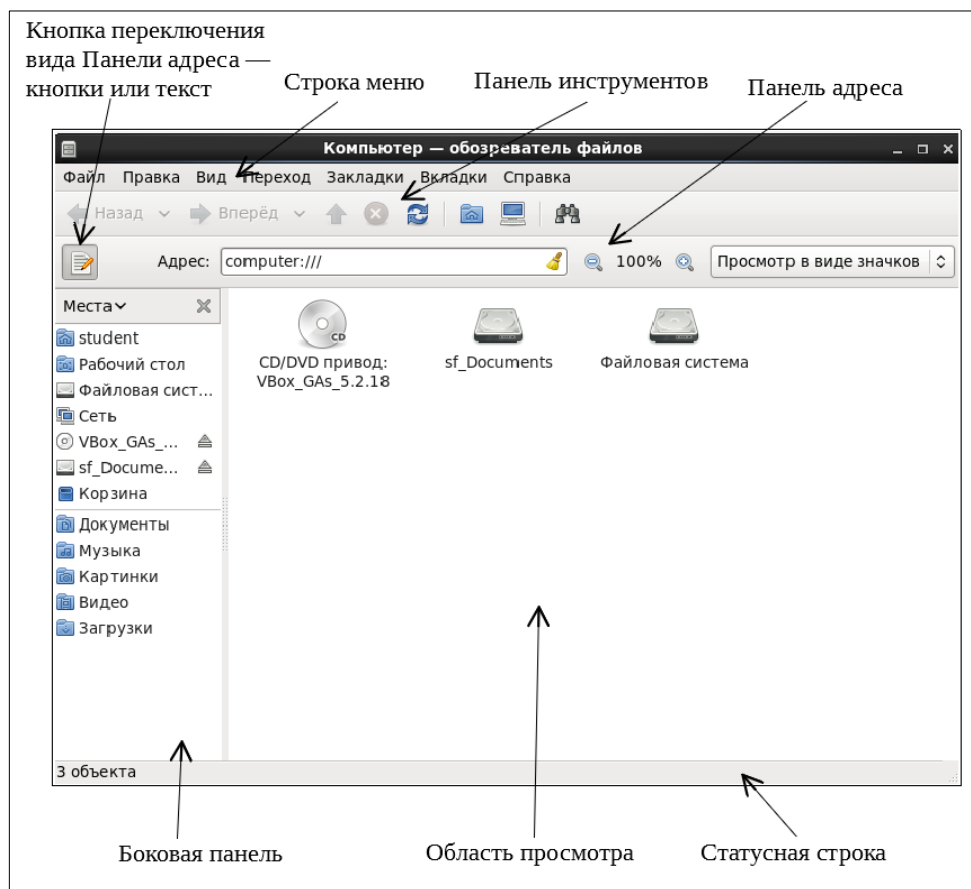


Рисунок 5.8 – Обозреватель файлов

4. **Боковая панель** – показывает сведения о текущем файле или папке и позволяет перемещаться по файловой системе. Боковая панель имеет выпадающий список, который позволяет выбирать содержимое панели ([рисунок 5.9](#)).

Можно выбрать следующее:

- **Места** – показывает часто используемые места;
- **Сведения** – показывает значок текущей папки и сведения о ней;

- **Дерево** – показывает файловую систему в иерархическом виде;
- **История** – содержит список файлов, папок, FTP-сайтов и URL (адресов), которые вы недавно посещали;
- **Заметки** – позволяет оставлять заметки о файлах и каталогах;
- **Эмблемы** – содержит эмблемы, которые можно добавлять к файлам и каталогам.

5. **Область просмотра** – отображает содержимое папок, FTP-сайтов, общих папок Windows, серверы WebDAV.

6. **Статусная строка** – показывает сведения в текущем состоянии.

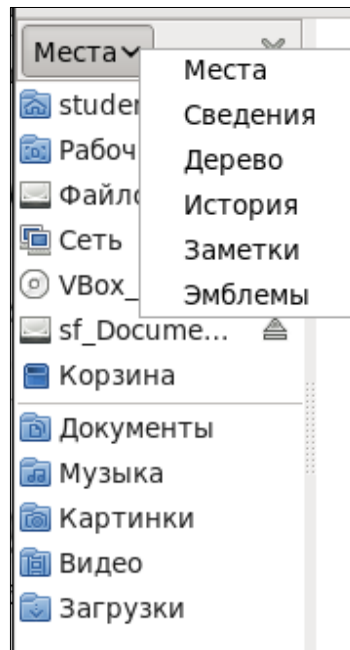


Рисунок 5.9 – Выпадающий список Боковой панели

Упражнение 3. В этом упражнении вы отредактируете файл `yum.repos.d`.

1. Откройте домашнюю папку в **Обозревателе файлов**.
2. Переключите **Панель адреса** в вид «текст», используя **Кнопку переключения вида Панели адреса**.

3. Измените раскладку клавиатуры на английскую с помощью комбинации клавиш **[Ctrl + Shift]**.

4. В адресной строке наберите `/etc/yum.repos.d/`. Обратите внимание, что при вводе происходит автоматическая подстановка названий каталогов и файлов. Нажмите **Enter**.

5. В области просмотра появится файл `goslinux-Base.repo`. Откройте его. Все строки вида

`baseurl=http://fap.fssprus/goslinux_fssp/$releasever/os/$basearch/ gpgcheck=1`

измените на следующую:

`baseurl=http://fap.fssprus.ru/goslinux_fssp/$releasever/os/$basearch/ gpgcheck=1.`

6. Сохраните файл и закройте.

При установке операционной системы GosLinux обязательно необходимо выполнить исправления в указанном файле. Если этого не сделать, то пакетный менеджер **yum** работать не будет.

Упражнение 4. В этом упражнении вы изучите навигацию с помощью **Обозревателя файлов**.

1. Откройте домашнюю папку в **Обозревателе файлов**. Домашней папкой в GosLinux является папка с именем, соответствующим имени учетной записи. Т. к. все студенты заходят в ОС GosLinux под учетной записью с именем **student**, то и домашней папкой является папка **/home/student**.

2. Создайте в текущей папке папку **Work**. Для этого выполните команду **Файл – Создать папку**, введите **Work** и нажмите **Enter**. Создать папку можно также с помощью команды **Создать папку** контекстного меню, щелкнув правой кнопкой мыши по свободному месту **Области просмотра**.

3. Откройте папку **Work** и создайте в ней две новые папки – **Факультет** и **Группа**. Таким образом, получим следующую структуру папок ([рисунок 5.10](#)).

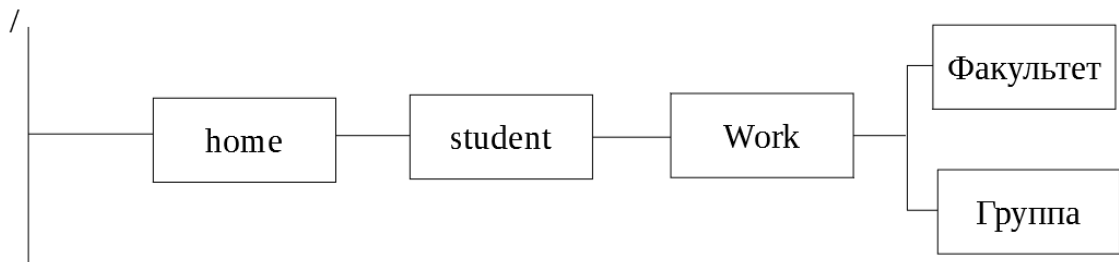


Рисунок 5.10 – Структура папок

4. Выведите файловую систему в иерархическом виде (Дерево). Изучите содержимое вашей домашней папки ([рисунок 5.11](#)).

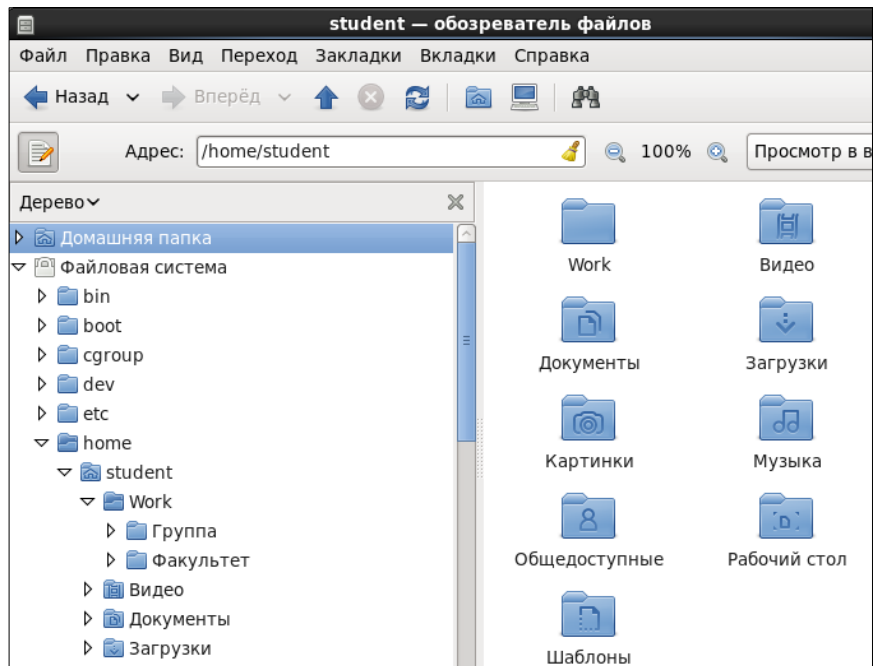


Рисунок 5.11 – Отображение структуры папок в Обозревателе файлов

5. Перейдите в папку **Факультет** и создайте текстовый документ с именем **Пример**, с помощью команды **Файл – Создать документ – Пустой файл**. Содержимое файла – ваша фамилия, имя, отчество и номер группы. Сохраните файл и закройте его.
6. Скопируйте файл **Пример** в папку **Группа**. Откройте файл, добавьте в содержимое название факультета и сохраните файл с новым именем **Студент**.
7. Закройте **Обозреватель файлов**.
8. Откройте папку **Компьютер**. Найдите папку **Факультет** с помощью панели поиска. Для этого перейдите в **Файловую систему**, нажмите на кнопку поиска и в строке поиска введите название папки со своей фамилией.
9. Удалите папку **Work**.
10. Закройте **Обозреватель файлов**.

2.7 Файловая система

Файловая система GosLinux представляет собой иерархическую, древовидную структуру. Самый верхний уровень файловой системы – это корневой каталог, который обозначается символом **/**.

В ОС GosLinux всё, включая жесткие диски, разделы и сменные носители, считается файлами. Это значит, что все файлы и каталоги (включая диски и разделы) существуют под корневым каталогом ([рисунок 5.12](#)).

Под корневым каталогом находится ряд важных системных каталогов, которые используются для работы ОС. Список системных каталогов находится в [Приложении В](#).

ОС Linux поддерживает множество файловых систем, в настоящее время наиболее широко используются: ext2, ext3, ext4, reiserfs. Так же, современные ОС Linux совместимы с файловыми системами, используемыми ОС Windows, такими как NTFS и FAT32.

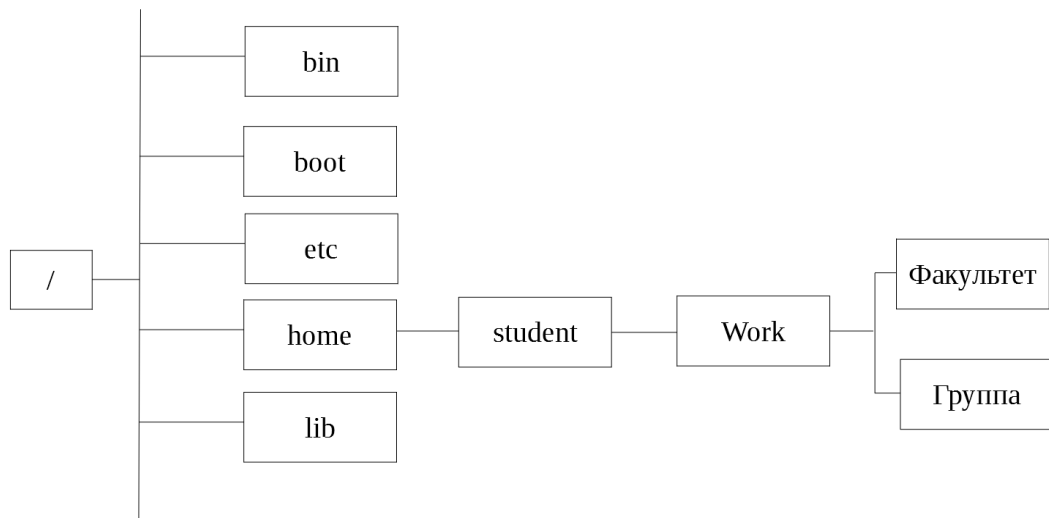


Рисунок 5.12 – Файловая система ОС GosLinux

Цепочка имен каталогов, через которые необходимо пройти для доступа к заданному файлу, вместе с именем этого файла образуют **путь к файлу**. Путь может быть абсолютным (например, /temp/foo – от корневого каталога) или относительным (например, book4/filesystem – от текущего каталога).

Существует ограничение на длину имени файла – не более 255 символов. В имя нельзя включать символ косой черты и нулевые символы. Также есть ограничение на длину пути, который передается ядру в качестве аргумента системного вызова – 4095 байт.

2.7.1 Монтирование и размонтирование файловых систем

Разделы дисков в GosLinux подключаются к системе совсем не так, как в Windows. Здесь есть корневая файловая система, куда подключаются все другие разделы и устройства, которые вы будете использовать. Системные разделы монтируются (присоединяются) автоматически при старте системы.

Монтирование – это процесс, благодаря которому файловая система устройства становится доступной для использования. В результате монтирования происходит подключение в один из каталогов целой файловой системы, находящейся где-то на другом устройстве.

Для монтирования необходим пустой каталог – он называется **точкой монтирования**. Точкой монтирования может служить любой каталог, никаких ограничений на этот счёт в GosLinux нет. После операции монтирования в каталоге (точке монтирования) появятся все те файлы и каталоги, которые находятся на соответствующем устройстве. В результате пользователь может даже и не знать, на каком устройстве какие файлы располагаются.

Подключённую («смонтированную») файловую систему можно в любой момент отключить – размонтировать, после чего тот каталог, куда она была смонтирована, снова окажется пустым.

Для GosLinux самой важной является корневая файловая система (root filesystem). Именно к ней затем будут подключаться (монтироваться) все остальные файловые системы на других устройствах. Обратите внимание, что корневая файловая система тоже монтируется, но только не к другой файловой системе, а к «самой Linux», причём точкой монтирования служит “/” (корневой каталог). Поэтому при загрузке системы прежде всего монтируется корневая файловая система, а при останове она размонтируется (в последнюю очередь).

Пользователю обычно не требуется выполнять монтирование и размонтирование вручную: при загрузке системы будут смонтированы все устройства, на которых хранятся части файловой системы, а при останове (перед выключением) системы все они будут размонтированы. Файловые системы на съёмных носителях (лазерных дисках, дискетах и пр.) также монтируются и размонтируются автоматически – либо при подключении носителя, либо при обращении к соответствующему каталогу.

Упражнение 5. В этом упражнении вы научитесь правильно выключать систему.

1. Закройте все открытые программы и окна.
2. Выполните команду **Система – Выключить...** В появившемся диалоговом окне щелкните по кнопке **Выключить**.

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Создать в домашней папке следующую структуру папок ([рисунок 5.13](#)).

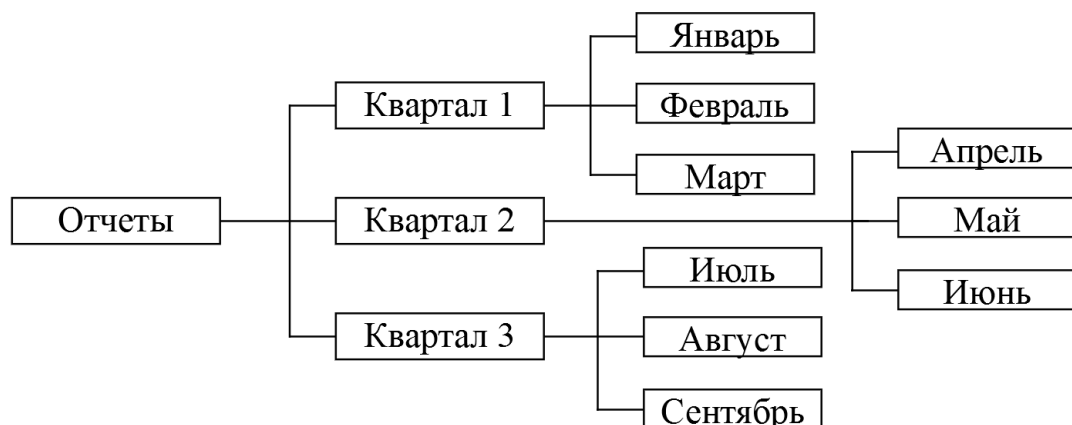


Рисунок 5.13 – Структура папок

2. В папке **Январь** создайте текстовый файл **Отчет1**, в папке **Февраль** создайте текстовый файл **Отчет2**, и т.д. для всех месяцев.
3. Скопируйте файлы **Отчет1**, **Отчет2**, **Отчет3** в папку **Квартал1**.

4. Перенесите файлы **Отчет4**, **Отчет5**, **Отчет6** в папку **Квартал2**.
5. Удалите файлы **Отчет7**, **Отчет8**, **Отчет9**.
6. Переименуйте файлы **Отчет1**, **Отчет2**, **Отчет3** папки **Квартал1** в файлы **Первый_Отчет**, **Второй_Отчет**, **Третий_Отчет**.
7. Найдите в домашней папке все файлы и папки с именем **Отчет**.
8. Покажите результаты работы преподавателю.
9. Удалите папку **Отчеты**.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Из каких компонентов состоит среда рабочего стола ОС GosLinux?
2. Где находится **Главное меню**?
3. Для чего предназначен файловый менеджер Nautilus?
4. В каких режимах может работать Nautilus?
5. Как открыть папку в **Обозревателе файлов**?
6. Что представляет собой файловая система ОС GosLinux?
7. Что такое путь к файлу?
8. Какая папка в ОС GosLinux является домашней?
9. Какой процесс называется монтированием файловой системы?
10. Что является точкой монтирования?

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема: ОС GOSLINUX. РАБОТА В РЕЖИМЕ КОМАНДНОЙ СТРОКИ

1 ЦЕЛЬ РАБОТЫ

Научиться:

- запускать Терминал;
- создавать в Терминале файлы и папки;
- выполнять в Терминале основные команды работы с файлами и папками.

2 ОБЪЕМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

3.1 Терминал и консоль: основные понятия

В GosLinux существует два вида интерфейса: графический интерфейс пользователя и интерфейс командной строки.

Графический интерфейс пользователя (англ. *Graphical user interface, GUI*) – управление программами с помощью графических кнопок, всплывающих меню, окон и других элементов. Множество действий можно выполнять с помощью мыши. Преимущества использования GUI: визуальное отображение программ и их содержимого, возможности программ можно изучать без чтения документации.

Интерфейс командной строки (англ. *Command Line Interface, CLI*) – управление программами с помощью команд. Команды состоят из букв, цифр, символов, набираются построчно, выполняются после нажатия клавиши Enter. Основным инструментом здесь является клавиатура. Данный интерфейс встроен в ядро системы, он будет доступен, даже если графический интерфейс не запустится.

Преимущества использования командной строки:

1. Командная строка обеспечивает более быстрый доступ к некоторым возможностям операционной системы, а иногда – это единственный способ запустить тот или иной процесс.
2. Текстовый интерфейс менее требовательный к ресурсам компьютера, чем графический.
3. Бывает, что графический интерфейс просто не нужен (например, на части серверов).

4. Программируемость работы операционной системы и программ. Командная оболочка позволяет загружать файл с последовательностью команд и, затем, выполняет их. Т.е. от человека не требуется постоянного ввода команд.

Добраться до командной строки можно двумя способами: через консоль или терминал.

Терминал – графическая программа, эмулирующая консоль. Такие программы позволяют не выходя из графического режима выполнять команды. Терминал по сравнению с консолью имеет дополнительный функционал (управление мышью, контекстное меню, полоса прокрутки, вкладки, запуск нескольких окон, главное меню, графические настройки) ([рисунок 6.1](#)).

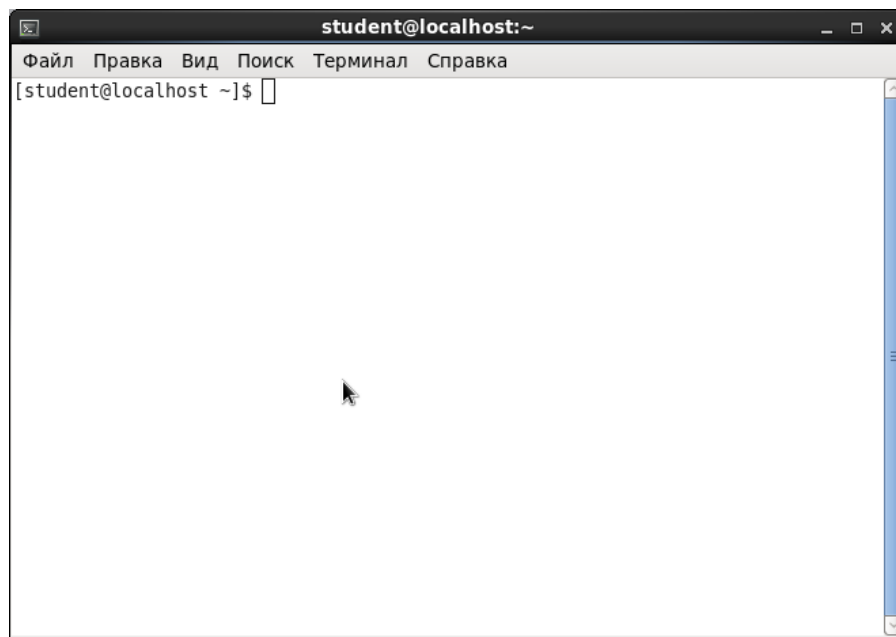


Рисунок 6.1 – Окно Терминала

Консолью называют программное обеспечение, реализующее текстовый интерфейс.

Эмулятор терминала – это графическая программа, которая предоставляет вам доступ к **tty** или **pts** терминалу. Например, Gnome Terminal, Konsole, Terminix, Xterm и многие другие.

Командная оболочка – интерпретатор команд ОС, обеспечивающий интерфейс для взаимодействия пользователя с функциями системы. Командных оболочек достаточно много, это **bash**, **sh**, **zsh**, **ksh** и другие, но чаще всего применяется **bash**.

Командная строка – это то место куда вы будете вводить свои команды, приглашение терминала для ввода.

Запустить эмулятор терминала можно через главное меню, выполнив команду **Приложения – Системные – Терминал** или в любой открытой папке с помощью команды контекстного меню **Open in Terminal** (графический режим терминала). Для работы в текстовом режиме следует выполнить

комбинацию клавиш **Ctrl+Alt+F2** (выход из текстового режима **Ctrl+Alt+F1**).

После запуска терминала, в окне появится приглашение к работе вида **[student@localhost ~]\$** (рисунок 6.1). Оно несет в себе информацию об имени пользователя (**student**), имени компьютера (**localhost**) и текущей папке (каталоге) (**~**). Тильда (**~**) обозначает домашнюю папку. Если на том же компьютере зайдет другой пользователь, то изменится первая строка (до знака **@**), при работе на другом компьютере изменится также строка после **@**. А если поменять текущий каталог, то знак тильды будет заменен на имя нового текущего каталога.

3.2 Команды работы с папками и файлами

Команд много, они позволяют выполнить прикладные задачи (получать информацию, работать с текстом, программировать и др.), управлять операционной системой и компьютером (железом).

Чтобы выполнить команду ее надо записать после приглашения и нажать **Enter**.

Синтаксис команды:

имя_команды [короткие_ключи] [длинные_ключи] [аргументы]

- **имя_команды** – командой является любой исполняемый файл (например, **ls** – команда показывает список файлов в текущем каталоге);
- **[короткие_ключи]** – короткие ключи состоят из одного символа и начинаются с символа **-** и указываются через пробел (например, **ls -l -a**). Для компактности короткие ключи могут объединяться в один ключ (например, **ls -la**).
- **[длинные_ключи]** – длинные ключи состоят из нескольких символов, образующих осмысленные слова и начинаются с двух символов – указываются через пробел (например, **ls --all**). Длинные ключи НЕ МОГУТ объединяться в один ключ.

Краткую справку о команде можно получить вызовом команды с ключом **-h** или **--help**. Перемещение по тексту вниз и вверх с помощью стрелок и клавиш **<PgUp>** и **<PgDn>**. Выход нажатием клавиши **<q>**.

Команда **pwd**

Назначение: выводит имя текущей папки, т. е. папки, с которой пользователь работает в данный момент времени. Текущую папку нужно знать при указании относительных путей к файлам и папкам, также некоторые программы сохраняют результаты своего выполнения в текущую папку.

Формат: **pwd**

Упражнение 1

1. Запустите Терминал из домашней папки.
2. Выполните команду `pwd`. В терминале появится строка `/home/student`, где `/` – корневой каталог, `home` – папка, находящаяся в корневом каталоге, `student` – текущая папка, вложенная в папку `home`.

Команда ls

Назначение: выводит список файлов и папок, поддерживает несколько различных форматов вывода.

Формат: `ls [ключ1] [ключ2] ... [ключN] [путь к папке]`

Ключи:

- a – выводит список всех файлов и папок (каталогов), в том числе и скрытые файлы (краткий формат);
- l – выводит список файлов в расширенном формате, показывая тип каждого элемента, полномочия, владельца, размер и дату последней модификации (полный формат);
- r – выводит список в порядке, обратном заданному;
- S – выводит размеры каждого файла;
- t – перечисляет файлы и каталоги в соответствии с датой их последней модификации;
- u – перечисляет файлы и каталоги в порядке, обратном в соответствии с датой их последней модификации.

Если команда `ls` применяется без указания пути к папке, то по умолчанию на экран выводится содержимое текущей папки.

Упражнение 2

1. Выполните команду `ls`. На экран выведутся папки, содержащиеся в текущей ([рисунок 6.2](#)).
2. Выполните команду `ls -l`. На экран выведется список папок, содержащихся в текущей, с атрибутами ([рисунок 6.3](#)).
3. Выполните команду `ls -la`. На экран выведется список папок и файлов, содержащихся в текущей папке, с атрибутами ([рисунок 6.4](#)).

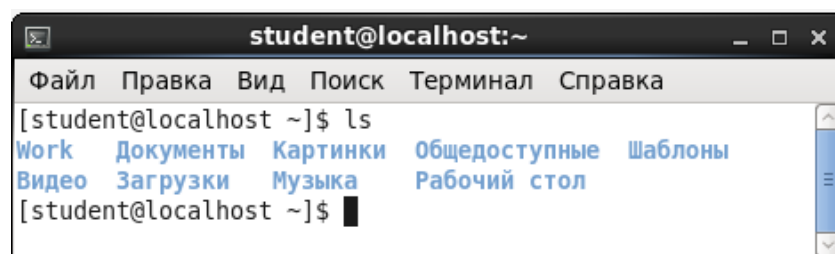


Рисунок 6.2 – Результат работы команды `ls`

```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ ls -l
итого 36
drwxrwxr-x. 4 student student 4096 Сен 14 15:11 Work
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Видео
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Документ
ы
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Загрузки
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Картинки
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Музыка
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Общедост
упные
drwxr-xr-x. 2 student student 4096 Сен 14 13:25 Рабочий
стол
drwxr-xr-x. 2 student student 4096 Сен 10 14:19 Шаблоны
[student@localhost ~]$

```

Рисунок 6.3 – Результат работы команды ls -l

```

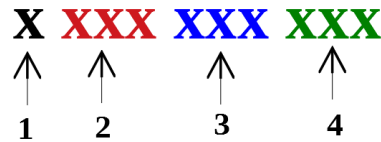
student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ ls -la
итого 200
drwx-----, 35 student student 4096 Сен 18 12:57 .
drwxr-xr-x. 3 root root 4096 Сен 6 11:58 ..
drwxrwxr-x. 2 student student 4096 Сен 10 14:19 .abrt
-rw-----, 1 student student 160 Сен 18 15:08 .bash_histo
ry
-rw-r--r--. 1 student student 18 Map 4 2015 .bash_logou
t
-rw-r--r--. 1 student student 176 Map 4 2015 .bash_profi
le
-rw-r--r--. 1 student student 124 Map 4 2015 .bashrc
drwxr-xr-x. 7 student student 4096 Сен 12 14:10 .cache
drwxr-xr-x. 8 student student 4096 Сен 12 13:33 .config
drwx-----, 3 student student 4096 Сен 10 14:19 .dbus
-rw-----, 1 student student 16 Сен 10 14:19 .esd_auth
drwxr-xr-x. 2 student student 4096 Сен 10 15:10 .fontconfig
drwx-----, 4 student student 4096 Сен 18 12:56 .gconf
drwx-----, 2 student student 4096 Сен 18 15:08 .gconfd
drwxr-xr-x. 22 student student 4096 Сен 12 12:56 .gimp-2.6
drwxr-xr-x. 7 student student 4096 Сен 18 14:18 .gnome2
drwx-----, 2 student student 4096 Сен 10 14:19 .gnome2_pri
vate
drwxrwxr-x. 3 student student 4096 Сен 10 14:19 .gnote
drwx-----, 2 student student 4096 Сен 18 12:56 .gnupg
drwxrwxr-x. 2 student student 4096 Сен 10 14:19 .gstalker-
0.10
-rw-rw-r--. 1 student student 326 Сен 18 12:56 .gtk-bookma
rks
dr-x-----, 2 student student 0 Сен 18 12:56 .gvfs
-rw-----, 1 student student 2480 Сен 18 12:56 .ICEauthori
ty
drwxrwxr-x. 2 student student 4096 Сен 12 13:33 .icons
-rw-r--r--. 1 student student 661 Сен 18 12:56 .imsettings
.log
drwxr-xr-x. 3 student student 4096 Сен 10 14:19 .local

```

Рисунок 6.4 – Результат работы команды ls -la

Обратите внимание, что перед названием файла или каталога выведены атрибуты, указывающие на принадлежность файла (папки), возможность чтения, записи и выполнения различными группами пользователей.

Атрибуты представляют собой цепочку из десяти символов x xxx xxx xxx.



1 – первый символ байта атрибутов определяет тип файла и может интерпретироваться со следующими значениями:

- – обычный файл;
- d – папка (каталог);
- l – символическая связь.

2 – вторая группа символов атрибутов определяет доступ к папкам и файлам для владельца.

3 – третья группа символов атрибутов определяет доступ к папкам и файлам для членов группы.

4 – четвертая группа символов атрибутов определяет доступ к папкам и файлам для остальных пользователей.

Для каждого уровня доступа существуют свои байты атрибутов, значение которых расшифровывается следующим образом:

- r – разрешение на чтение;
- w – разрешение на запись;
- x – разрешение на выполнение;
- – отсутствие разрешения.

В домашней папке пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).

После байтов атрибутов на экран выводится следующая информация о файле:

- число связей файла;
- имя владельца файла;
- размер файла в байтах;
- дата создания файла (или модификации);
- время;
- имя файла.

Например, с использованием команды `ls -la` выведем содержимое папки student и рассмотрим атрибуты папки Документы:

d rwx r-x r-x. 2 student 4096 Сен 10 14:19 Документы,

где d – папка;

rwx – разрешение на чтение, запись и выполнение для владельца (student);

r-x – разрешение на чтение и выполнение для членов группы;

r-x – разрешение на чтение и выполнение для остальных пользователей.

Выполните команду `clear` для очистки содержимого терминала.

Команда mkdir

Назначение: создание новой папки.

Формат: mkdir имя_папки

Упражнение 3

1. Выполните команду `mkdir Work`.
2. Откройте домашнюю папку в **Обозревателе файлов** и убедитесь, что в домашней папке появилась папка **Work**.
3. Создайте в папке **Work** две папки **Факультет** и **Группа**. Для этого выполните команды `mkdir Work/Факультет` и `mkdir Work/Группа`.

Команда `cd`

Назначение: изменение текущей папки.

Формат: `cd [имя_папки]`

Упражнение 4

1. Проверьте, какая папка является текущей с помощью команды `pwd`.
2. Измените текущую папку на **Факультет**. Для этого выполните команду `cd Work/Факультет`.
3. Убедитесь, что текущей стала папка **Факультет**.
4. Перейдите в папку **группа** с помощью команды `cd ~/Work/Группа`.
5. Выполните команду `cd -`. Таким образом, мы вернулись в предыдущую папку **Факультет**.
6. Перейдите в папку **Work** – родительскую папку для папки **Факультет**. Выполните команду `cd ..`. Таким образом, мы вернулись на один уровень выше.
7. Перейдите в домашнюю папку, выполнив команду `cd`. Команда `cd` без параметров возвращает оболочку в домашнюю папку.

Команда `cat`

Назначение: просмотр содержимого файлов, создание текстовых файлов, объединение содержимого нескольких файлов.

Формат:

`cat [ключи][путь/]имя файла1 [[путь/]имя файла2]... [[путь/]имя файлаN]`

Рассмотрим основные ключи:

- b – нумеровать только непустые строки;
- E – показывать символ \$ в конце каждой строки;
- n – нумеровать все строки;
- S – удалять пустые повторяющиеся строки;
- T – отображать табуляции в виде ^I.

Упражнение 5

1. Проверьте, какая папка является текущей с помощью команды `pwd`.
2. Измените текущую папку на **Work**.
3. Создайте в текущей папке текстовый файл **text1**. Для этого выполните команду `cat > text1`. На экране курсор перейдет на новую строку и оболочка будет ожидать ввода текста. Переключите раскладку клавиатуры на

русский язык ([**Ctrl+Shift**]) и введите текст по предложенному образцу, указывая свои личные данные, в конце каждой строки нажимайте **Enter**:

Иванов Иван Иванович
дата рождения 11 октября 2000 г.
студент группы АЭ-11 Энергетического факультета Азово-Черноморского инженерного института
год поступления - 2018

4. После окончания ввода текста нажмите комбинацию клавиш [**Ctrl+d**]. Оболочка выведет в терминал приглашение к работе. Файл **text1** создан.

5. Проверьте наличие файла **text1** в папке **Work**.

6. Выведите в терминал содержимое созданного файла, выполнив команду **cat text1**.

7. Создайте в текущей папке файл **text2**. Содержимым файла является список фамилий студентов вашей группы.

8. Выполните объединение содержимого файлов **text1** и **text2** и перенаправьте полученный текст в новый файл **text3**. Для этого выполните команду **cat text1 text2 > text3**.

9. Просмотрите содержимое файла **text3**.

10. Просмотрите содержимое папки **Work**.

11. Сделайте текущей домашнюю папку.

Команда `cp`

Назначение: копирует файлы и папки.

Формат: `cp [ключ1] ... [ключN] [путь/]источник [путь/]приёмник`

источник – папка, которую мы хотим скопировать;

приёмник – папка, в которую мы хотим скопировать.

Ключи:

-f, -force – перезаписывать существующие файлы;

-i, -interactive – спрашивать, нужно ли перезаписывать существующие файлы;

-n – не перезаписывать существующие файлы;

-r – копировать папку рекурсивно, т. е. с учетом вложенных папок;

-u – скопировать файл, только если он был изменён;

-p – сохранять владельца, временные метки и флаги доступа при копировании;

-t – считать файл-приемник директорией и копировать файл-источник в эту директорию.

Упражнение 6

1. Проверьте, какая папка является текущей.

2. Сделайте текущей папку **Work**.

3. Скопируйте папку **Группа** в папку **Факультет**. Для этого выполните команду `cp -a Группа Факультет`. Копирование происходит внутри папки **Work**, которая является текущей. Поэтому мы не указываем путь к папкам.

4. В папке **Факультет** создайте папку **Группа2**.

5. Убедитесь, что текущей является папка **Work**. Если это не так, сделайте папку **Work** текущей.

6. Скопируйте файл **text2** в папку **Группа2**. Для этого выполните команду `cp text2 Факультет/Группа2`.

Команда mv

Назначение: перемещение, переименование файла или папки (каталога).

Формат: `mv [ключ1] ... [ключN] [путь/]источник [путь/]приёмник`

источник – папка или файл, которые мы хотим переименовать или переместить;

приёмник – новое имя папки или файла при переименовании, имя папки, в которую перемещаем папку или файл.

Упражнение 7

1. Проверьте, какая папка является текущей.
2. Сделайте текущей папку **Work**.
3. Переименуйте папку **Факультет** в **Энергетический_Факультет**, выполнив команду `mv Факультет Энергетический_Факультет`.
4. В папке **Work** переименуйте файл **text3** на **АЭ-11**, выполнив команду `mv text3 АЭ-11`.
5. Переместите файл **АЭ-11** в папку **Энергетический_Факультет**, выполнив команду `mv АЭ-11 Энергетический_Факультет`.

Команда rm

Назначение: удаление файла или папки (каталога).

Формат: `rm [ключ1] ... [ключN] папка (или файл)`

Ключи:

- r – происходит рекурсивное удаление папки вместе с её содержимым;
- i – перед удалением каждого файла запрашивается подтверждение.

Упражнение 8

1. Проверьте, какая папка является текущей.
2. Сделайте текущей папку **Work**.
3. Удалите файл **text1**, выполнив команду `rm text1`.
4. Сделайте текущей домашнюю папку.
5. Удалите папку **Work** вместе со всем содержимым.
6. Для выхода из Терминала выполните команду `exit`.

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Создать в режиме командной строки в домашней папке следующую структуру папок ([рисунок 6.5](#)).

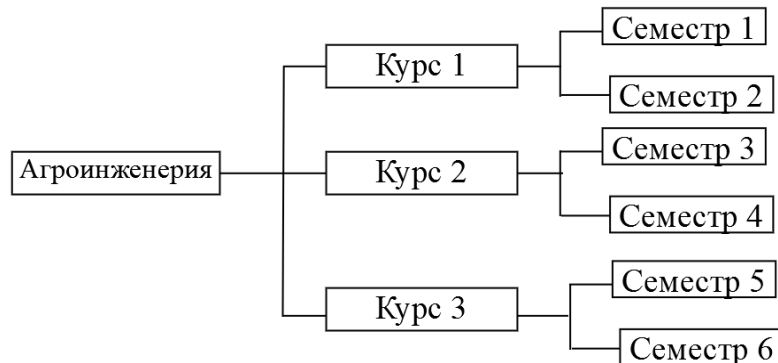


Рисунок 6.5 – Структура папок

2. В папке **Семестр1** создайте текстовый файл **Сессия1**, в папке **Семестр2** создайте текстовый файл **Сессия2**, и т.д. для всех семестров. Содержимым файлов будет являться перечень дисциплин с оценками.
3. Скопируйте файлы **Сессия1**, **Сессия1** в папку **Курс1**.
4. Перенесите файлы **Сессия3**, **Сессия4** в папку **Курс2**.
5. Удалите файлы **Сессия5**, **Сессия6**.
6. Переименуйте файлы **Сессия1**, **Сессия1** папки **Курс1** в файлы **Первая_сессия**, **Вторая_сессия**.
7. Покажите результаты работы преподавателю.
8. Удалите папку **Агроинженерия**.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое **Терминал**?
2. Для чего предназначен Терминал в ОС GosLinux?
3. Что такое **Консоль**?
4. Что такое **Командная оболочка**?
5. Для чего предназначена **Командная строка**?
6. Какая папка называется текущей?
7. Какая папка называется домашней? Как она обозначается?
8. Назначение и формат команды **pwd**?
9. Назначение и формат команды **ls**?
10. Назначение и формат команды **mkdir**?
11. Назначение и формат команды **cd**?
12. Назначение и формат команды **cat**?
13. Назначение и формат команды **cp**?
14. Назначение и формат команды **mv**?
15. Назначение и формат команды **rm**?
16. Назначение и формат команды **exit**?
17. Назначение и формат команды **clear**?

ЛАБОРАТОРНАЯ РАБОТА № 7

Тема: **PYTHON. ОСНОВНЫЕ ЯЗЫКОВЫЕ КОНСТРУКЦИИ.
ЛИНЕЙНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с основными языковыми конструкциями;
- с базовыми синтаксическими конструкциями языка Python.

Научиться:

- создавать программы, реализующие линейный вычислительный процесс.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

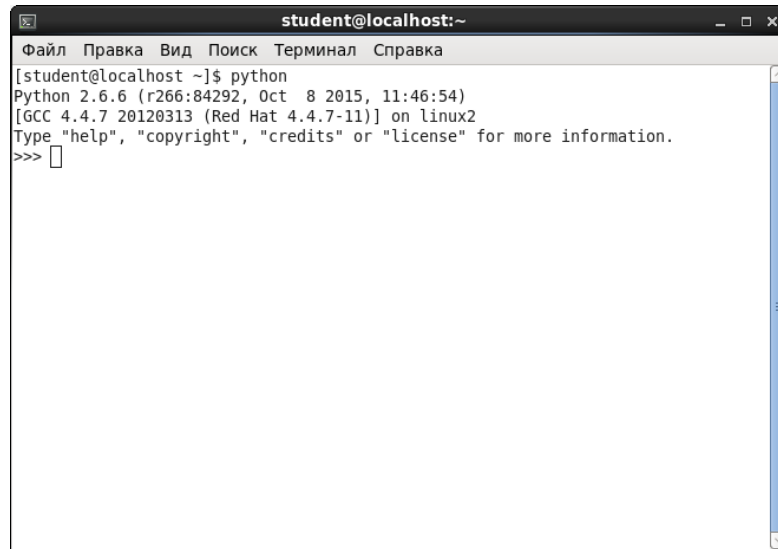
3.1 Основные понятия

Python – современный язык программирования, работающий на всех распространенных операционных системах для настольных компьютеров. Язык программирования Python разрабатывается чуть более 20 лет. В настоящее время активно используется две версии языка – более старая версия 2 и современная версия 3.

Python – современный универсальный интерпретируемый язык программирования. Его достоинства:

1. Кроссплатформенность и бесплатность.
2. Простой синтаксис и богатые возможности позволяют записывать программы очень кратко, но в то же время понятно.
3. По простоте освоения язык сравним с бейсиком.
4. Богатая стандартная библиотека, возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т. д.).

Программы на языке Python выполняются интерпретатором. В ОС GosLinux интерпретатор можно вызвать, набрав команду **python** в терминале. После запуска интерпретатора появляется подсказка **>>>**, в которой можно начать отладку операторов программы в простом цикле чтения/выполнения ([рисунок 7.1](#)). Такой режим работы интерпретатора называют интерактивным.



```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python
Python 2.6.6 (r266:84292, Oct  8 2015, 11:46:54)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 

```

Рисунок 7.1 – Результат запуска Python в интерактивном режиме

Несмотря на удобства интерактивного режима работы с Python, часто требуется сохранить исходный программный код для последующего использования. В таком случае подготавливаются файлы, которые передаются затем интерпретатору на исполнение. По отношению к интерпретируемым языкам программирования часто исходный код называют скриптом. Файлы с кодом на Python обычно имеют расширение **.py**.

Скрипты можно писать в любом текстовом редакторе (например, стандартная программа gedit с подсветкой синтаксиса Python). Также существуют специальные программы для разработки, предоставляющие дополнительные возможности и удобства.

Запускать подготовленные файлы можно в Терминале с помощью команды **python адрес/имя_файла**.

3.2 Переменные в Python

Совокупность величин, с которыми работает компьютер, принято называть **данными**. По отношению к программе данные делятся на **исходные, результаты** (окончательные данные) и **промежуточные**, которые получаются в процессе вычислений.

Для успешного освоения программирования необходимо усвоить следующее правило: всякая величина занимает свое определенное место в памяти ЭВМ (иногда говорят – ячейку памяти). Хотя термин «ячейка», если учесть архитектуру современных ЭВМ, несколько устарел, однако в учебных целях его удобно использовать.

У всякой величины имеются три основных свойства: **имя, значение и тип**. В алгоритмах и языках программирования величины делятся на **константы и переменные**.

Константа – величина, которая не изменяет свое значение в процессе выполнения программы. В программе она представляется собственным значением, например: 15, 34.7, 'k', true и пр.

Переменная – это простейшая именованная структура данных, в которой может быть сохранён промежуточный или конечный результат работы программы. Переменная может изменять свое значение в процессе выполнения программы и представляется символическим именем – **идентификатором**, например: X, S2, cod15 и пр.

Правила написания имен переменных (констант):

1. Переменная должна иметь уникальное имя, состоящее из букв латинского алфавита, цифр и знака подчеркивания.
2. Имя переменной должно начинаться с буквы.
3. В имени переменной нельзя использовать ключевые слова и встроенные команды (например, import, help...).
4. При написании имени переменной необходимо учитывать регистр букв – X и x разные переменные.

Переменную в Python создать очень просто – нужно присвоить некоторому идентификатору значение при помощи оператора присваивания «=».

Упражнение 1. В этом упражнении вы научитесь работать с Python в режиме интерпретатора.

1. Откройте домашнюю папку. Откройте **Терминал**, выполнив команду контекстного меню **Open in Terminal** в домашней папке.
2. Запустите интерпретатор Python. Для этого наберите в командной строке команду **python** и нажмите клавишу **Enter**.
3. После приглашения к вводу команд интерпретатора **>>>** введите следующие команды, нажимая клавишу **Enter** после каждой строки.

```
a = 10
b = 3.1415926
c = "Hello"
d = [1, 2, 3]
```

4. Просмотрите значения переменных **a**, **b**, **c**, **d**. Для этого в интерпретаторе введите имя переменной и нажмите **Enter**.

В этом упражнении используются четыре переменные:

- переменная **a** хранит значение типа **int** (целое число),
- переменная **b** хранит значение типа **float** (действительное число),
- переменная **c** хранит значение типа **str** (строка),
- переменная **d** хранит значение типа **list** (список, в данном случае из трех целых чисел).

Никакого специального объявления переменных не требуется, первое присваивание переменной значения и является ее объявлением.

Python – язык с динамической типизацией: каждая переменная в каждый момент времени имеет определенный тип, но этот тип может меняться по ходу выполнения программы, достаточно просто присвоить ей новое значение другого типа.

На самом деле переменная в Python является лишь ссылкой на объект в памяти. При создании любой переменной (число, строка или массив) в неё записывается ссылка на объект, а сам объект находится где-то в оперативной памяти далеко от самой переменной со ссылкой. Таким образом, несколько переменных могут указывать на один объект, и при изменении объекта (например, списка) изменится результат обращения к нему с использованием каждой переменной.

3.3 Структура простейшей программы в Python

Как правило, программа на Python должна состоять из следующих частей:

1. Считывание данных.
2. Решение задачи.
3. Вывод результата.

Например, программа, считающая сумму двух чисел может иметь следующий вид:

```
a = int(input())
b = int(input())
sum = a + b
print "sum= ", sum
```

В данной программе с клавиатуры вводятся значения переменных **a** и **b**, вычисляется их сумма **sum**, и выводится значение переменной **sum** на экран.

3.4 Синтаксис языка Python

Программа на Python – обычный текстовый файл с инструкциями. Каждая инструкция располагается на отдельной строке. Если инструкция не является вложенной, то она начинается с начала строки (отступ вложенной инструкции может быть, например, 4 пробела). В конце инструкции обычно ничего не ставится (; ставится, если несколько инструкций в одной строке). Если инструкция длинная, то ее можно разбить на несколько частей с помощью \ или заключив инструкцию в круглые скобки.

Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков.

Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

3.5 Комментарии

Комментарии предназначены для пояснений и игнорируются интерпретатором. В Python можно использовать только однострочный комментарий.

Упражнение 2. В этом упражнении вы научитесь писать комментарии.

1. Для корректного отображения символов русского алфавита необходимо после запуска интерпретатора ввести команду

```
#coding=utf-8
```

2. В режиме интерпретатора введите следующую команду:

```
print "Первая программа на Python" # Вывод текста на экран
```

Обратите внимание, что в результате выполнения команды на экране появится текст, заключенный в кавычки в операторе print, а комментарий выводиться не будет.

3.6 Типы данных

В языке Python выделяют несколько типов данных: целые числа, числа с плавающей точкой (вещественные), строки, логический тип и т.д.

Тип каждой переменной может динамически изменяться по ходу выполнения программы. Определить, какой тип имеет переменная, можно с помощью команды `type(имя переменной)`.

3.6.1 Тип Numbers (числа)

Python поддерживает четыре различных числовых типа: целые числа (`int`), длинные целые числа (`long`), числа с плавающей запятой или вещественные (`float`) и комплексные числа (`complex`). Целые числа являются положительными или отрицательными целыми числами без десятичной точки. Длинные целые числа имеют неограниченную точность, действительные числа записываются с десятичной точкой, отделяющие целые и дробные части. Комплексные числа имеют вещественную и мнимую часть, `a + bj`, где `a` – вещественная часть, а `b` – мнимая часть.

Целые числа (`int`) и длинные целые числа (`long`)

Целое число в Python имеет тип `int`, длинное целое – тип `long`. Тип `int` представляет целое число, для хранения которого используется 4 байта в памяти компьютера. Числа данного типа имеют 32-битную точность (или допустимый диапазон от -2147483648 до +2147483647). Длинные целые числа ограничений не имеют.

Вещественные числа (`float`)

Вещественные числа или числа с плавающей запятой имеют тип `float`. Представляют числа с плавающей точкой, для хранения которого используется 8 байт, например, `1.2` или `34.76`. Точность чисел с плавающей точкой равна 15 десятичным знакам в дробной части. Обратите внимание, что в Python разделителем целой и дробной частей числа является точка.

Комплексные числа (complex)

Тип данных `complex` относится к категории неизменяемых и хранит пару значений типа `float`, одно из которых представляет действительную часть комплексного числа, а другое – мнимую. Принцип описания комплексных чисел в Python похож на описание комплексных чисел в математике. Комплексные числа записываются в форме `a+bj`, где `a` – действительная часть числа, `b` – мнимая часть числа.

Упражнение 3. В этом упражнении вы научитесь выполнять арифметические операции с комплексными числами.

1. В режиме интерпретатора введите следующие команды:

```
c = 3.5 + 2j
print "c=", c
d=4j
print "d=", d
```

Если действительная часть равна 0, то эту часть при записи комплексного числа можно опустить.

Отдельные части комплексного числа доступны в виде атрибутов `real` и `imag`.

2. В режиме интерпретатора введите следующие команды:

```
m = -4.5-5j
print m.real, '\n', m.imag #Вывод действительной и мнимой части числа m
'\n' – перевод курсора на новую строку.
```

В Python с комплексными числами можно выполнять следующие арифметические операции:

```
a + b – сумма;
a - b – разность;
a * b – произведение;
a / b – частное;
a * * b – возведение в степень.
```

Указанные арифметические операции можно выполнять как с самим комплексным числом, так и с его частями.

3. В режиме интерпретатора введите следующие команды:

```
e = c + d
print "e=", e
```

Метод `conjugate()` изменяет знак мнимой части на противоположный, т. е. Возвращает комплексно-сопряженное число.

4. В режиме интерпретатора введите следующие команды:

```
m = -4.5-5j
print m.conjugate()
```

Основные операции с целыми и вещественными числами

$a + b$ – сумма;

$a - b$ – разность;

$a * b$ – произведение;

a / b – частное (результатом этого действия является вещественное или целое число);

$a \% b$ – возвращает остаток от деления a на b ;

$a // b$ – возвращает целую часть от деления a на b (целочисленное деление);

$a ** b$ – возведение a в степень b ;

`pow(a, b)` – возведение a в степень b ;

$-a$ – отрицание a ;

`divmod(a, b)` – возвращает пару $(a // b, a \% b)$, т. е. целую часть от деления и остаток от деления.

Упражнение 4. В этом упражнении вы научитесь выполнять арифметические операции с целыми и вещественными числами.

1. В режиме интерпретатора введите следующие команды:

`a = 25`

`b = 4.3`

`c = a * b`

`print "c=", c # Вывод результата произведения двух чисел`

`d = a % b`

`print "d=", d # Вывод остатка от деления`

Приоритеты выполнения операций

Приоритеты операций в Python совпадают с приоритетом операций в математике:

1. Возведение в степень справа налево, т.е. $3 * 3 * 3$ это $3 * (3 * 3)$.

2. Унарный минус (отрицание).

3. Умножение и деление слева направо. Операции умножения и деления имеют одинаковый приоритет.

4. Сложение и вычитание слева направо. Операции сложения и вычитания имеют одинаковый приоритет.

5. Для изменения порядка действий нужно использовать скобки.

Арифметические операции с присвоением (унарные операции)

Ряд специальных операций позволяют присвоить результат операции первому операнду:

$+=$ – присвоение результата сложения;

$-=$ – присвоение результата вычитания;

$*=$ – присвоение результата умножения;

$/=$ – присвоение результата от деления;

$//=$ – присвоение результата целочисленного деления;

$**=$ – присвоение результата возведения в степень;

$\%=$ – присвоение остатка от деления.

1. В режиме интерпретатора введите следующие команды:

```
a = 5
a += 2
print "a=", a
d = 23
d //= 3
print "d=", d
```

3.6.2 Тип String (строки)

Строки в Python имеют тип `str`. Строкой называется последовательность символов: букв, цифр, знаков препинания и т.д., представленных в кавычках. Python позволяет использовать пару одиночных или двойных кавычек. Подмножества строк могут быть взяты с использованием оператора «извлечения среза» или «доступа по индексу» (`[]` и `[:]`) с индексами, начинающимися с 0 в начале строки, с -1, если отсчет идет с конца строки.

Основные операции со строками

`a + b` – конкатенация (строка `b` приписывается к строке `a`);
`a * n` – повторение строки `a` `n` раз, значение `n` должно быть целого типа.

Упражнение 5. В этом упражнении вы научитесь выполнять операции со строками.

1. В режиме интерпретатора введите следующие команды:

```
stroka1 = "primer 1"
print stroka1 # Вывод всей строки
print stroka1[0] # Вывод первого символа строки
print stroka1[2:5] # Вывод символов строки с третьего по пятый
print stroka1[2:] # Вывод строки с третьего символа
print stroka1*2 # Вывод строки дважды
r1 = "Python"
r2 = " - язык программирования высокого уровня"
r3 = r1 + r2 # Операция конкатенация
print r3
```

3.6.3 Тип List (списки)

Список представляет собой упорядоченную последовательность элементов. Он очень гибкий и является одним из самых используемых типов в Python. Элементы списка не обязательно должны быть одного типа.

Объявить список довольно просто. Внутри квадратных скобок помещаются элементы списка, разделённые запятой:

```
a = [1, 2.4, "Python"]
```

Для извлечения элемента из списка используется оператор `[]`. В Python индексация начинается с нуля.

```
a = [5, 20, 13, 15, 17, 18]
print "a[2]=", a[2] # Вывод элемента списка с индексом 2
```

3.7 Преобразование типов в Python

Иногда бывает полезно целое число записать как строку. И, наоборот, если строка состоит из цифр, то полезно эту строку представить в виде числа, чтобы дальше можно было выполнять арифметические операции с ней. Для этого используются функции, название которых совпадает с именем типа, то есть `int`, `float`, `str`.

Существует несколько встроенных функций для преобразования из одного типа данных в другой. Эти функции возвращают новый объект, представляющий преобразованное значение ([таблица 7.1](#)).

Таблица 7.1 – Преобразование типов в Python

№	Функция	Описание
1	<code>int(x [,base])</code>	Преобразует x в целое число
2	<code>float(x)</code>	Преобразует x в число с плавающей запятой
3	<code>complex(real [,imag])</code>	Создает комплексное число
4	<code>str(x)</code>	Преобразует объект x в строковое представление
5	<code>repr(x)</code>	Преобразует объект x в строковое выражение
6	<code>eval(str)</code>	Обрабатывает строку и возвращает объект. Используется для выполнения программного кода представленного строкой
7	<code>list(s)</code>	Преобразует s в список
8	<code>chr(x)</code>	Преобразует целое число в символ
9	<code>unichr(x)</code>	Преобразует целое число в символ Unicode
10	<code>ord(x)</code>	Преобразует один символ в его целочисленное значение

3.8 Оператор присваивания

Оператором присваивания является символ `=`. Выполняется оператор стандартным образом: сначала вычисляется выражения справа от знака равенства, а затем полученное значение записывается в переменную, указанную слева от знака равенства.

```
a = 3.14
```

```
b = a + 10
```

3.9 Встроенные функции

В среде Python без дополнительных операций импорта доступно более сотни встроенных объектов, в основном, функций и исключений.

Для удобства функции условно разделены по категориям:

1. Функции преобразования типов и классы: `coerce`, `str`, `repr`, `int`, `list`,

tuple, long, float, complex, dict, super, file, bool, object.

2. Числовые и строковые функции: **abs, divmod, ord, pow, len, chr, uni-
chr, hex, oct, cmp, round, unicode.**

3. Функции обработки данных: **apply, map, filter, reduce, zip, range,
xrange, max, min, iter, enumerate, sum.**

4. Функции определения свойств: **hash, id, callable, issubclass, isin-
stance, type.**

5. Функции для доступа к внутренним структурам: **locals, globals, vars,
intern, dir.**

6. Функции компиляции и исполнения: **eval, execfile, reload,
__import__, compile.**

7. Функции ввода-вывода: **input, raw_input, open.**

8. Функции для работы с атрибутами: **getattr, setattr, delattr, hasattr.**

9. Функции-"украшатели" методов классов: **staticmethod, classmethod,
property.**

10. Прочие функции: **buffer, slice.**

3.10 Функция для ввода данных

Язык Python включает много уже определенных, т. е. встроенных в него, функций. Программист не видит их определений. Достаточно знать, что эти функции принимают и что возвращают, то есть их синтаксис.

Ввод данных осуществляется при помощи функции **input()**.

Синтаксис:

input([строка])

Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст. После этого, когда он нажмет Enter, функция **input()** заберет введенный текст и передаст его программе, которая уже будет обрабатывать его согласно своим алгоритмам.

Если в интерактивном режиме ввести команду **input()**, то компьютер будет ждать, когда вы что-нибудь введете и нажмете Enter или просто нажмете Enter. Введенное значение сразу же отобразится на экране.

Упражнение 6. В этом упражнении вы научитесь использовать функцию **input()**.

Функцию **input()** можно вызвать двумя способами.

Способ 1. Функция **input()** передает введенные данные в программу, которые можно присвоить переменной. В этом случае интерпретатор не выводит строку сразу же.

Способ 2. Функция **input()** вводит входные значения как строку символов. Поэтому, чтобы присвоить переменной целочисленное значение, следует воспользоваться функцией преобразования типов, например **int()**.

1. В режиме интерпретатора введите следующие команды. Ввод каждой команды заканчивается нажатием клавиши **Enter**.

```
input() #Ввод в интерактивном режиме
a = input("Введите значение переменной a= ") #Ввод значения
переменной a
b = int(input()) #Ввод значения целочисленной переменной b
```

3.11 Оператор вывода данных

Для вывода данных используется оператор **print**, который может выводить не только значения переменных, но и значения любых выражений

Синтаксис:

```
print [object, ...][, sep=' '][, end='\n'][, file=sys.stdout]
```

где

object – объекты, которые требуется отправить в файл;

sep=' ' – строка, которой следует разделить объекты, **None** – использовать значение по умолчанию;

end='\n' – строка, которой следует поставить после всех объектов, **None** – использовать значение по умолчанию;

file=sys.stdout – ожидается объект, реализующий метод `write(string)`.

При помощи оператора **print** можно выводить значение не одного, а нескольких выражений, для этого нужно перечислить их через запятую.

Упражнение 7. В этом упражнении вы научитесь использовать оператор **print**.

1. В режиме интерпретатора введите следующие команды:

```
a = 1
b = 2
print "a=", a, '\n', "b=", b, '\n', "a+b=", a + b
```

В данном случае будет напечатан текст **a=1, b=2, a+b=3**: сначала выводится значение переменной **a**, затем значение переменной **b**, затем значение суммы **a + b**.

3.12 Основные стандартные модули Python

Одним из важных преимуществ языка Python является наличие большой библиотеки модулей и пакетов, входящих в стандартную комплектацию.

Перед тем как приступить к изучению модулей стандартной библиотеки, необходимо определить то, что в Python называется **модулем**.

В соответствии с модульным подходом к программированию большая задача разбивается на несколько более мелких, каждую из которых (в идеале) решает отдельный модуль. В разных методологиях даются различные ограничения на размер модулей, однако при построении модульной структуры программы важнее составить такую композицию модулей, которая позволила бы свести к минимуму связи между ними. Набор классов и функций, имею-

щий множество связей между своими элементами, было бы логично расположить в одном модуле.

Есть и еще одно полезное замечание: модули должно быть легче использовать, чем написать заново. Это значит, что модуль должен иметь удобный интерфейс: набор функций, классов и констант, который он предлагает своим пользователям.

Модули выполняют как минимум три важных функции:

1. Повторное использование кода: такой код может быть загружен много раз во многих местах.

2. Управление адресным пространством: модуль – это высокоуровневая организация программ, это пакет имен, который избавляет вас от конфликтов. Каждый объект «проживает» свой цикл внутри своего модуля, поэтому модуль – это средство для группировки системных компонентов.

3. Глобализация сервисов и данных: для реализации объекта, который используется во многих местах, достаточно написать один модуль, который будет импортирован.

В программе на Python модуль представлен объектом-модулем, атрибутами которого являются имена, определенные в модуле:

```
from math import *
c = cos(0)
print "c=", c
```

В данном примере импортируется модуль `math`. В результате работы оператора `import` в текущем пространстве имен появляется объект с именем `math`.

Модули для использования в программах на языке Python по своему происхождению делятся на обычные (написанные на Python) и модули расширения, написанные на другом языке программирования (как правило, на C). С точки зрения пользователя они могут отличаться разве что быстродействием.

Python распространяется с библиотекой стандартных модулей. Библиотека включает в себя более 200 модулей, которые выполняют платформенно-зависимую поддержку таких задач, как: интерфейс к операционной системе, управление объектами, поиск, сеть + интернет, GUI и т.д. Полный список стандартных модулей можно посмотреть на <https://docs.python.org/3/library/>.

3.12.1 Модули `math` и `cmath`

В этих модулях собраны математические функции для действительных и комплексных аргументов. В [таблице 7.2](#) даны функции модуля `math`.

Там, где аргумент обозначен буквой `Z`, аналогичная функция определена и в модуле `cmath`.

Таблица 7.2 – Основные функции модуля **math** для работы с математическими функциями

Функция или константа	Описание функции
<code>cos(z)</code>	косинус z
<code>sin(z)</code>	синус z
<code>tan(z)</code>	тангенс z
<code>acos(z)</code>	арккосинус z
<code>asin(z)</code>	арксинус z
<code>atan(z)</code>	арктангенс z
<code>atan2(y,x)</code>	$\text{atan}(y/x)$
<code>e</code>	константа e
<code>pi</code>	константа π
<code>exp(z)</code>	экспонента
<code>pow(x,y)</code>	возведение числа x в степень y
<code>sqrt(z)</code>	корень квадратный от z
<code>abs(x)</code>	модуль x
<code>floor(x)</code>	наибольшее целое, меньшее или равное x
<code>fmod(x,y)</code>	остаток от деления x на y
<code>hypot(x,y)</code>	корень квадратный из суммы квадратов x и y
<code>log(z)</code>	натуральный логарифм z
<code>log10(z)</code>	десятичный логарифм z
<code>modf(x)</code>	возвращает пару (y,q) — дробную и целую части числа x ; обе части имеют знак исходного числа
<code>ceil(x)</code>	наименьшее целое, большее или равное x

Рассмотрим пример записи математической функции $y = \ln(x) - \frac{x^3}{|x+1|}$ с использованием модуля **math**. На языке Python эта функция будет выглядеть так:

`y = log(x) — x**3 / abs(x + 1)`

3.13 Линейный вычислительный процесс

Линейным называется процесс обработки данных, если составляющие его действия выполняются последовательно одно за другим.

Упражнение 8. В этом упражнении вы составите блок-схему алгоритма и разработаете программу на языке Python для вычисления значения функции $y = a + \frac{\sin(10*b)}{e^{c+2}}$, где a, b, c – произвольные числа, введенные с клавиатуры.

1. Составьте блок-схему алгоритма ([рисунок 7.2](#)).

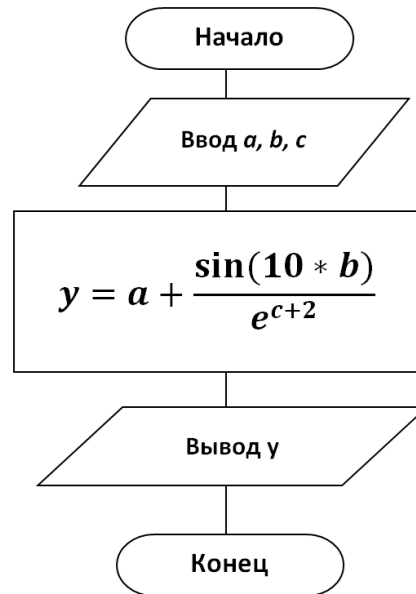


Рисунок 7.2 – Блок-схема алгоритма

2. Выйдите из интерактивного режима Python, нажав комбинацию клавиш **[Ctrl+d]**.
3. В домашней папке создайте файл **Lab7_1.py**.
4. Откройте созданный файл в программе **gedit** и наберите текст программы ([рисунок 7.3](#)).

```

*Lab1_1.py X
#coding=utf-8
from math import *          # Подключение модуля math
a=float(input('Введите a= ')) # Ввод значения вещественной переменной a
b=float(input('Введите b= ')) # Ввод значения вещественной переменной b
c=float(input('Введите c= ')) # Ввод значения вещественной переменной c
y=a+sin(10*b)/exp(c+2)      # Вычисление y
print 'y=', y              # Вывод y
  
```

Рисунок 7.3 – Текст программы

5. Сохраните файл.
6. Запустите программу на выполнение. Для этого в Терминале введите команду `python Lab7_1.py`.
7. После выполнения программы интерпретатор Python будет ожидать ввода значений указанных переменных. После ввода каждого значения нажимайте **Enter**. На экране появится результат работы вашей программы.

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

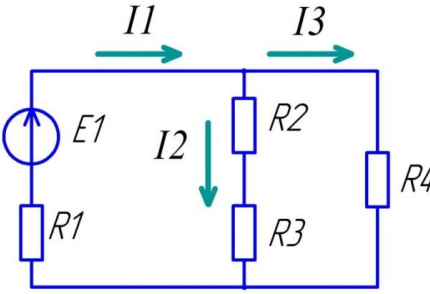
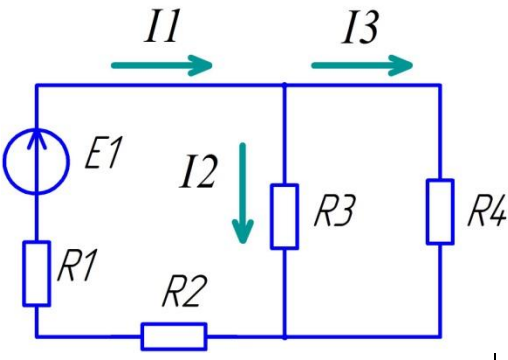
1. Составьте блок-схему алгоритма и напишите программу, согласно варианту, указанному преподавателем. Имя файла – **Lab7_2.py**.

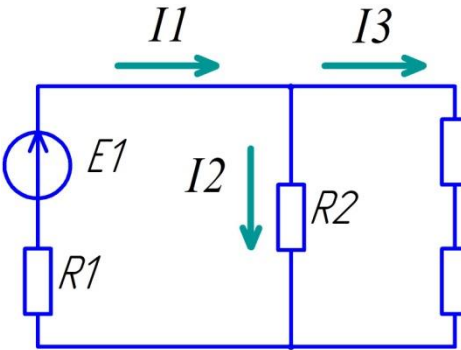
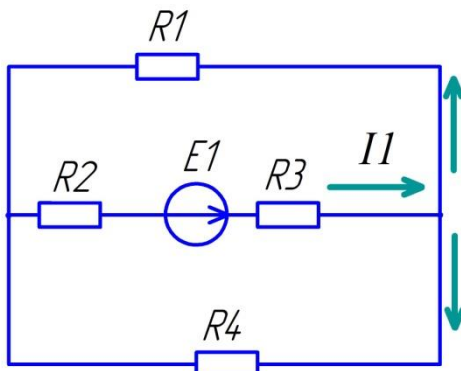
Задача. Разработать программу для расчета параметров электрической схемы N, приведенной в таблице исходных данных (N – номер варианта, заданный преподавателем). Значения ЭДС (E_1) и сопротивлений ($R_1 - R_4$) должны вводиться пользователем. В результате расчета должны быть выведены следующие значения:

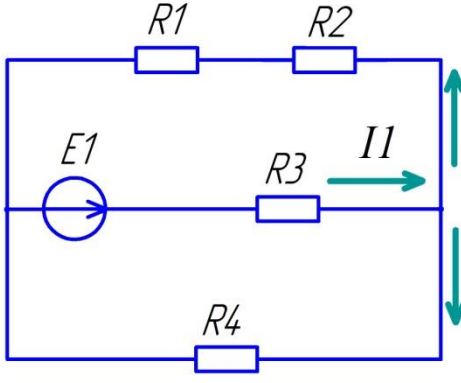
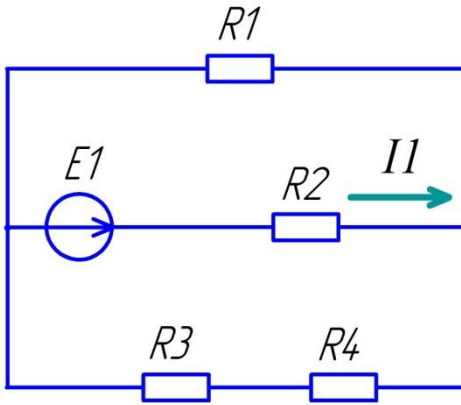
- а) эквивалентное сопротивление электрической цепи, Ом;
- б) токи в ветвях электрической цепи, А;
- в) мощности, потребляемые активными сопротивлениями, Вт;
- г) суммарная мощность, потребляемая нагрузкой, Вт;
- д) мощность, генерируемая источником питания, Вт.

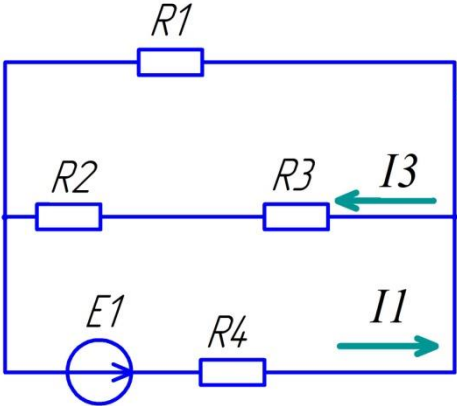
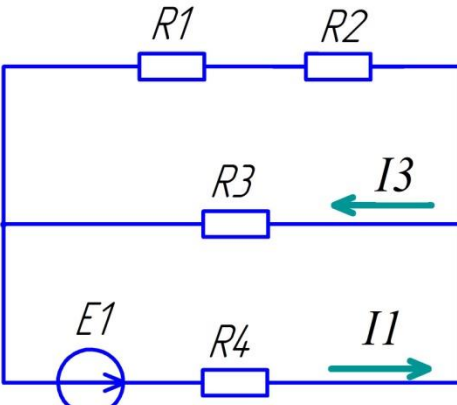
Все числовые данные, выводимые на консоль, округлять до сотых и приводить поясняющие надписи.

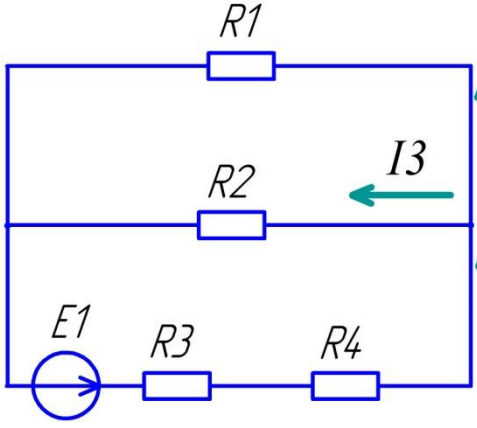
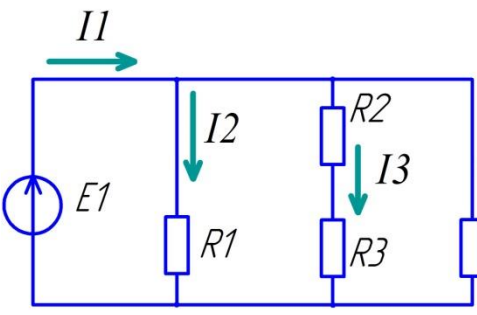
Таблица исходных данных

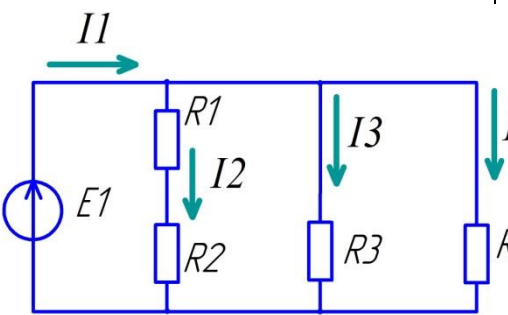
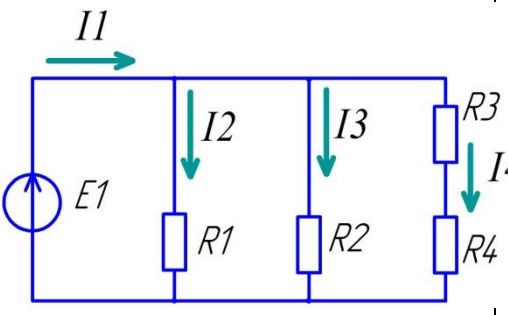
№ варианта	Электрическая схема	Порядок расчета
1	2	3
1		1) Эквивалентное сопротивление цепи, Ом $R_{ekv} = R1 + \frac{(R2 + R3)R4}{R2 + R3 + R4}$ 2) Ток в первой ветви, А $I1 = \frac{E1}{R_{ekv}}$ 3) Ток во второй ветви, А $I2 = \frac{E1 - I1 \cdot R1}{R2 + R3}$ 4) Ток в третьей ветви, А $I3 = \frac{E1 - I1 \cdot R1}{R4}$ 5) Мощность, потребляемая сопротивлением $R1$, Вт $P1 = I1^2 R1$ 6) Мощность, потребляемая сопротивлением $R2$, Вт $P2 = I2^2 R2$ 7) Мощность, потребляемая сопротивлением $R3$, Вт $P3 = I2^2 R3$ 8) Мощность, потребляемая сопротивлением $R4$, Вт $P4 = I3^2 R4$ 9) Суммарная мощность, потребляемая нагрузкой, Вт $P_{nagr} = P1 + P2 + P3 + P4$ 10) Мощность, генерируемая источником питания, Вт $P_{ist} = E1 \cdot I1$
2		1) Эквивалентное сопротивление цепи, Ом $R_{ekv} = R1 + R2 + \frac{R3 \cdot R4}{R3 + R4}$ 2) Ток в первой ветви, А $I1 = \frac{E1}{R_{ekv}}$ 3) Ток во второй ветви, А $I2 = \frac{E1 - I1 \cdot (R1 + R2)}{R3}$ 4) Ток в третьей ветви, А $I3 = \frac{E1 - I1 \cdot (R1 + R2)}{R4}$ 5) Мощность, потребляемая сопротивлением $R1$, Вт $P1 = I1^2 R1$ 6) Мощность, потребляемая сопротивлением $R2$, Вт $P2 = I1^2 R2$ 7) Мощность, потребляемая сопротивлением $R3$, Вт $P3 = I2^2 R3$ 8) Мощность, потребляемая сопротивлением $R4$, Вт $P4 = I3^2 R4$ 9) Суммарная мощность, потребляемая нагрузкой, Вт $P_{nagr} = P1 + P2 + P3 + P4$ 10) Мощность, генерируемая источником питания, Вт $P_{ist} = E1 \cdot I1$

1	2	3
3		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R1 + \frac{R2(R3 + R4)}{R2 + R3 + R4}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot R1}{R2}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot R1}{(R3 + R4)}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I1^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I2^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I3^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I3^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ист} = E1 \cdot I1$
4		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R2 + R3 + \frac{R1 R4}{R1 + R4}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot (R2 + R3)}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot (R2 + R3)}{R4}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I1^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I1^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I3^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ист} = E1 \cdot I1$

1	2	3
5		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R3 + \frac{(R1 + R2)R4}{R1 + R2 + R4}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot R3}{R1 + R2}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot R3}{R4}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I2^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I1^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I3^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$
6		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R2 + \frac{R1(R3 + R4)}{R1 + R3 + R4}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot R2}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot R2}{R3 + R4}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I1^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I3^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I3^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$

1	2	3
7		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R4 + \frac{R1(R2 + R3)}{R1 + R2 + R3}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot R4}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot R4}{R2 + R3}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I3^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I3^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I1^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$
8		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R4 + \frac{(R1 + R2)R3}{R1 + R2 + R3}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1 \cdot R4}{R1 + R2}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1 \cdot R4}{R3}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I2^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I3^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I1^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$

1	2	3
9		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = R3 + R4 + \frac{R1 \cdot R2}{R1 + R2}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1 - I1(R3 + R4)}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1 - I1(R3 + R4)}{R2}$ <p>5) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>6) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I3^2 R2$ <p>7) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I1^2 R3$ <p>8) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I1^2 R4$ <p>9) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>10) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$
10		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{ekv} = \frac{1}{\frac{1}{R1} + \frac{1}{R2 + R3} + \frac{1}{R4}}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{ekv}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1}{R2 + R3}$ <p>5) Ток в четвертой ветви, А</p> $I4 = \frac{E1}{R4}$ <p>6) Мощность, потребляемая сопротивлением $R1$, Вт</p> $P1 = I2^2 R1$ <p>7) Мощность, потребляемая сопротивлением $R2$, Вт</p> $P2 = I3^2 R2$ <p>8) Мощность, потребляемая сопротивлением $R3$, Вт</p> $P3 = I3^2 R3$ <p>9) Мощность, потребляемая сопротивлением $R4$, Вт</p> $P4 = I4^2 R4$ <p>10) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{nagr} = P1 + P2 + P3 + P4$ <p>11) Мощность, генерируемая источником питания, Вт</p> $P_{ist} = E1 \cdot I1$

1	2	3
11		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{\text{equiv}} = \frac{1}{\frac{1}{R1 + R2} + \frac{1}{R3} + \frac{1}{R4}}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{\text{equiv}}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1}{R1 + R2}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1}{R3}$ <p>5) Ток в четвертой ветви, А</p> $I4 = \frac{E1}{R4}$ <p>6) Мощность, потребляемая сопротивлением R1, Вт</p> $P1 = I2^2 R1$ <p>7) Мощность, потребляемая сопротивлением R2, Вт</p> $P2 = I2^2 R2$ <p>8) Мощность, потребляемая сопротивлением R3, Вт</p> $P3 = I3^2 R3$ <p>9) Мощность, потребляемая сопротивлением R4, Вт</p> $P4 = I4^2 R4$ <p>10) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{\text{nagr}} = P1 + P2 + P3 + P4$ <p>11) Мощность, генерируемая источником питания, Вт</p> $P_{\text{ист}} = E1 \cdot I1$
12		<p>1) Эквивалентное сопротивление цепи, Ом</p> $R_{\text{equiv}} = \frac{1}{\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3 + R4}}$ <p>2) Ток в первой ветви, А</p> $I1 = \frac{E1}{R_{\text{equiv}}}$ <p>3) Ток во второй ветви, А</p> $I2 = \frac{E1}{R1}$ <p>4) Ток в третьей ветви, А</p> $I3 = \frac{E1}{R2}$ <p>5) Ток в четвертой ветви, А</p> $I4 = \frac{E1}{R3 + R4}$ <p>6) Мощность, потребляемая сопротивлением R1, Вт</p> $P1 = I2^2 R1$ <p>7) Мощность, потребляемая сопротивлением R2, Вт</p> $P2 = I3^2 R2$ <p>8) Мощность, потребляемая сопротивлением R3, Вт</p> $P3 = I4^2 R3$ <p>9) Мощность, потребляемая сопротивлением R4, Вт</p> $P4 = I4^2 R4$ <p>10) Суммарная мощность, потребляемая нагрузкой, Вт</p> $P_{\text{nagr}} = P1 + P2 + P3 + P4$ <p>11) Мощность, генерируемая источником питания, Вт</p> $P_{\text{ист}} = E1 \cdot I1$

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схему алгоритма, листинг программы **Lab_7_2**, исходные данные, результат выполнения программы и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая величина называется константой?
2. Какая величина называется переменной?
3. Что представляет собой программа на языке Python?
4. Для чего нужны комментарии в программе?
5. Перечислите основные типы данных в Python.
6. Перечислите основные операции с вещественными и целыми числами?
7. Приоритет выполнения операций при записи арифметических выражений.
8. Как записываются унарные операции?
9. Перечислите основные операции со строками.
10. Что представляет собой тип список?
11. Синтаксис функции для ввода данных.
12. Синтаксис оператора вывода данных.
13. Какой вычислительный процесс называется линейным?

ЛАБОРАТОРНАЯ РАБОТА № 8

Тема: PYTHON. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ РАЗВЕТВЛЯЮЩЕГОСЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА. ОПЕРАТОРЫ IF И IF-ELSE

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с управляющими операторами разветвляющегося вычислительного процесса.

Научиться:

- реализовывать разветвляющиеся алгоритмы на языке программирования Python.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

Вычисления, в процессе которых необходимо совершить те или иные действия в зависимости от результата проверки какого-либо условия, получили название **разветвляющегося вычислительного процесса**.

Разветвляющийся вычислительный процесс реализуется с помощью условных операторов.

3.1 Оператор условного перехода if

Оператор **if** содержит в себе логическое условие, в котором производится сравнение данных и по результату выбирается дальнейшее действие.

Синтаксис оператора:

```
if логическое_выражение:  
    инструкция1  
    инструкция2  
...  
    инструкцияN
```

Если логическое выражение принимает значение истина (True), то выполняются инструкции, определяемые данным оператором. Блок кода, который необходимо выполнить в случае истинности логического выражения от-

деляется четырьмя пробелами слева либо количеством пробелов, кратным четырем.

Структура «Разветвление» для оператора `if` выглядит, как показано на [рисунке 8.1](#).



Рисунок 8.1 – Структура «Разветвление» для оператора `if`

Упражнение 1. В этом упражнении вы составите блок-схему и разработаете программу для следующей задачи.

Задача. Даны два числа a и b . Возвести в квадрат эти числа, если a больше b . Вывести результат на экран.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 8.2](#)).

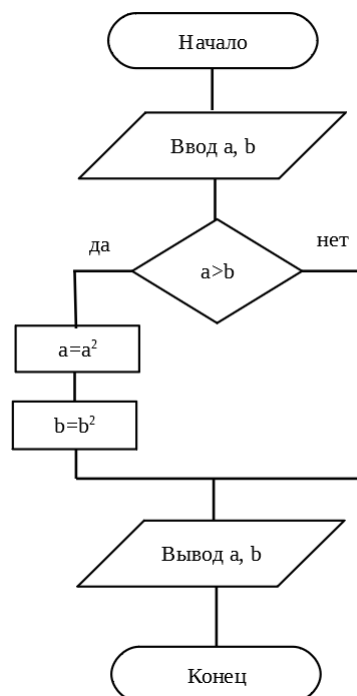
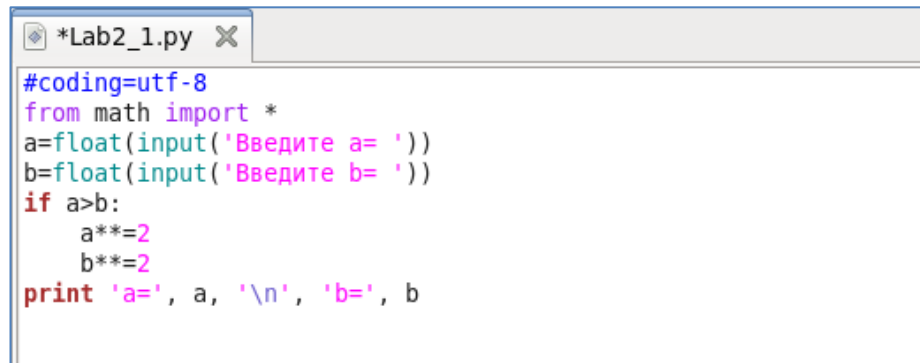


Рисунок 8.2 – Блок-схема алгоритма

2. Создайте файл с именем **Lab8_1.py**.

3. Откройте созданный файл и наберите текст программы ([рисунок 8.3](#)).



```
#coding=utf-8
from math import *
a=float(input('Введите a= '))
b=float(input('Введите b= '))
if a>b:
    a**=2
    b**=2
print 'a=', a, '\n', 'b=', b
```

Рисунок 8.3 – Текст программы

4. Сохраните файл и закройте его.
5. Выполните отладку программы для значений a=3, b=7 и a=8, b=2.

3.2 Логические выражения. Логический тип данных

Для записи условий используются шесть знаков **отношений**: **>** (больше), **<** (меньше), **>=** (больше или равно), **<=** (меньше или равно), **==** (равно), **!=** (не равно).

Операция отношения может иметь только одно из двух результирующих значений – True (Истина) и False (Ложь), которые могут быть присвоены переменным типа Boolean или определенному свойству объекта.

С переменными типа Boolean (логическими переменными) можно выполнять ряд специальных операций, которые называются логическими или булевыми. К ним относят:

- and** – конъюнкция (логическое умножение);
- or** – дизъюнкция (логическое сложение);
- not** – инверсия (логическое отрицание).

Так же, как и для операций отношения, результатом выполнения логической операции может быть True или False.

Логические выражения типа **a >= 1023** являются простыми, так как в них выполняется только одна логическая операция. Однако на практике нередко возникает необходимость в более сложных выражениях.

В таких случаях используются специальные операторы, объединяющие два и более простых логических выражения. Широко используются два оператора – так называемые логические **и (and)** и **или (or)**.

Допустим, переменной x было присвоено значение 10 (**x = 10**), переменной y присвоили 23 (**y = 23**). Логическое выражение **y<23 and x>10** будет выполняться следующим образом. Сначала выполнится выражение **y<23**. Его результатом будет True. Затем выполнится выражение **x>8**. Его результатом будет False. Далее выражение сведется к True and False, что вернет False.

Если бы мы записали выражение так: **x>10 and y<23**, то оно также вернуло бы False.

В случае с оператором **or** второе простое выражение проверяется, если первое вернуло ложь, и не проверяется, если уже первое вернуло истину. Так как для истинности всего выражения достаточно единственного True, неважно по какую сторону от **or** оно стоит.

В языке Python есть еще унарный логический оператор **not**, т. е. отрицание. Он превращает правду в ложь, а ложь в правду. Унарный он потому, что применяется к одному выражению, стоящему после него, а не справа и слева от него как в случае бинарных **and** и **or**.

3.3 Оператор условного перехода if–else

Оператор **if–else** содержит в себе логическое условие, в котором производится сравнение данных и по результату выбирается дальнейшее действие.

Синтаксис оператора:

if логическое_выражение:

инструкция1

инструкция2

...

инструкцияN

else:

инструкция1

инструкция2

...

инструкцияN

Если логическое выражение принимает значение истина (True), то выполняются инструкции, определяемые оператором **if**. Если логическое выражение принимает значение истина (False), то выполняются инструкции, определяемые оператором **else** ([рисунок 8.4](#)).

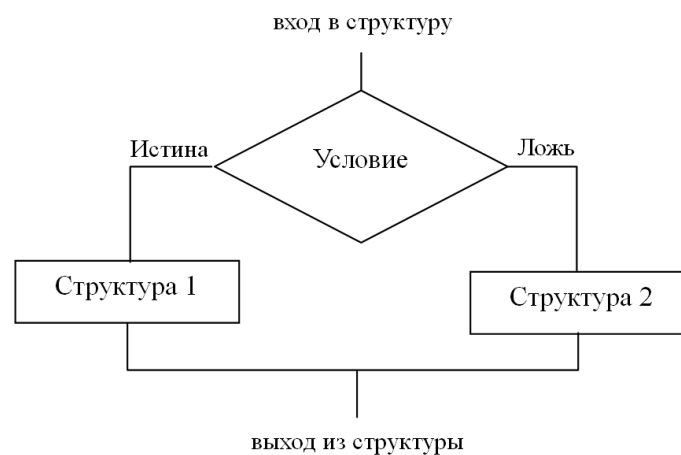


Рисунок 8.4 – Структура «Разветвление» для оператора if-else

Упражнение 2

В этом упражнении вы составите блок-схему и напишете программу, вычисляющую значение функции

$$y = \begin{cases} \frac{1}{\sqrt{|x-a|}} + \ln^2(x+a), & \text{если } x > a \\ \cos^3(x+a)^2, & \text{если } x \leq a \end{cases},$$

где x, a – произвольные числа, введённые с клавиатуры.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 8.5](#)).

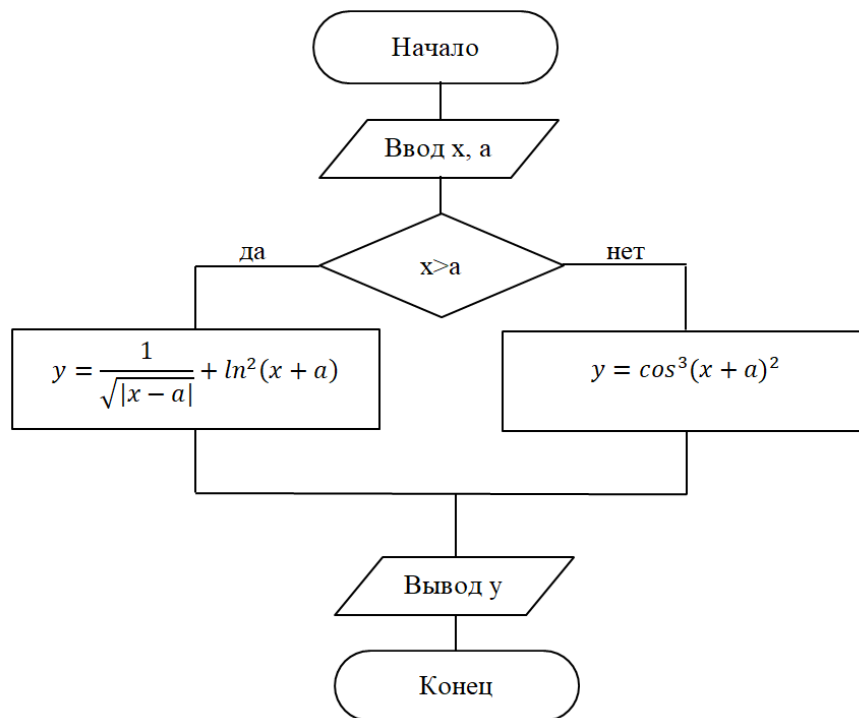


Рисунок 8.5 – Блок-схема алгоритма

2. Создайте файл с именем **Lab8_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 8.6](#)).

```

#coding=utf-8
from math import *
a=float(input('Введите a= '))
x=float(input('Введите x= '))
if x>a:
    y=1/sqrt(abs(x-a))+log(x+a)**2
else:
    y=cos((x+a)**2)**3
print 'y=', y
  
```

Рисунок 8.6 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение.

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Составьте блок-схему алгоритма и напишите программу, согласно варианту, указанному преподавателем. Имя файла – **Lab8_3.py**.

Варианты заданий.

1. Даны два целых числа. Составьте схему алгоритма и напишите программу деления первого числа на второе. Причем, если ввести второе число, равное нулю, то должно выводиться соответствующее сообщение.
2. Составьте схему алгоритма и напишите программу. Даны вещественные числа x , y ($x \neq y$). Заменить полусуммой чисел меньшее из них, удвоенным произведением чисел – большее из них.
3. Составьте схему алгоритма и напишите программу. Даны вещественные числа x , y ($x \neq y$). Каждое из чисел заменить модулем, если оба числа отрицательные, в противном случае – увеличить числа в три раза.
4. Составьте схему алгоритма и напишите программу. Определить, имеется ли среди введенных с клавиатуры положительных целых чисел a , b хотя бы одно четное.
5. Составьте схему алгоритма и напишите программу. Определить, имеется ли среди введенных с клавиатуры положительных целых чисел a , b хотя бы одно нечетное.
6. Составьте схему алгоритма и напишите программу. Даны две переменные целого типа: c и d . Если их значения равны, то присвоить каждой переменной среднее этих значений, а если не равны, то присвоить переменным значение 1.
7. Составьте схему алгоритма и напишите программу. Даны две переменные целого типа: c и d . Если их значения не равны, то присвоить каждой переменной минимальное из этих значений, а если равны, то присвоить переменным значение 3.
8. Составьте схему алгоритма и напишите программу для проверки условия: имеется ли среди двух целых чисел хотя бы одно положительное значение.
9. Составьте схему алгоритма и напишите программу для проверки условия: является ли треугольник со сторонами a , b , c прямоугольным.
10. Составьте схему алгоритма и напишите программу. Даны два числа a и b . Выведите на экран эти числа в порядке возрастания.
11. Составьте схему алгоритма и напишите программу. Даны два числа a и b . Выведите на экран эти числа в порядке убывания.

12. Составьте схему алгоритма и напишите программу. Даны два целых числа a и b . Проверьте, является ли сумма двух этих чисел кратной 3.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схему алгоритма, листинг программы **Lab8_3.py**, исходные данные, результат выполнения программы и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой вычислительный процесс называется разветвляющимся?
2. С помощью каких операторов реализуется разветвляющийся вычислительный процесс?
3. Синтаксис оператора `if`.
4. Синтаксис оператора `if-else`.

ЛАБОРАТОРНАЯ РАБОТА № 9**Тема: PYTHON. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ
РАЗВЕТВЛЯЮЩЕГОСЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА.
ОПЕРАТОР IF-ELIF-ELSE****1 ЦЕЛЬ РАБОТЫ**

Познакомиться:

- с управляющими операторами разветвляющегося вычислительного процесса.

Научиться:

- реализовывать разветвляющиеся алгоритмы на языке программирования Python.

**2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК
ЕЁ ВЫПОЛНЕНИЯ**

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ****3.1 Оператор условного перехода if-elif-else**

Синтаксис оператора:

```
if логическое_выражение_1:
    инструкция1
...
    инструкцияN
elif логическое_выражение_2:
    инструкция1
...
    инструкцияN
...
[elif логическое_выражение_N:
    инструкция1
...
    инструкцияN]
else:
    инструкция1
...
    инструкцияN]
```

Оператор `elif` позволяет выполнять множественную проверку выражения и выполнять блок кода, как только результат будет считаться истиной.

Упражнение 1

В этом упражнении вы составите блок-схему и напишете программу, вычисляющую значение функции

$$S = \begin{cases} \sqrt[3]{ab^2}, & \text{если } a \leq 3 \\ a - b, & \text{если } 3 < a \leq 8, \\ a + b, & \text{если } a > 8 \end{cases}$$

где a, b – произвольные положительные числа, введённые с клавиатуры.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 9.1](#)).

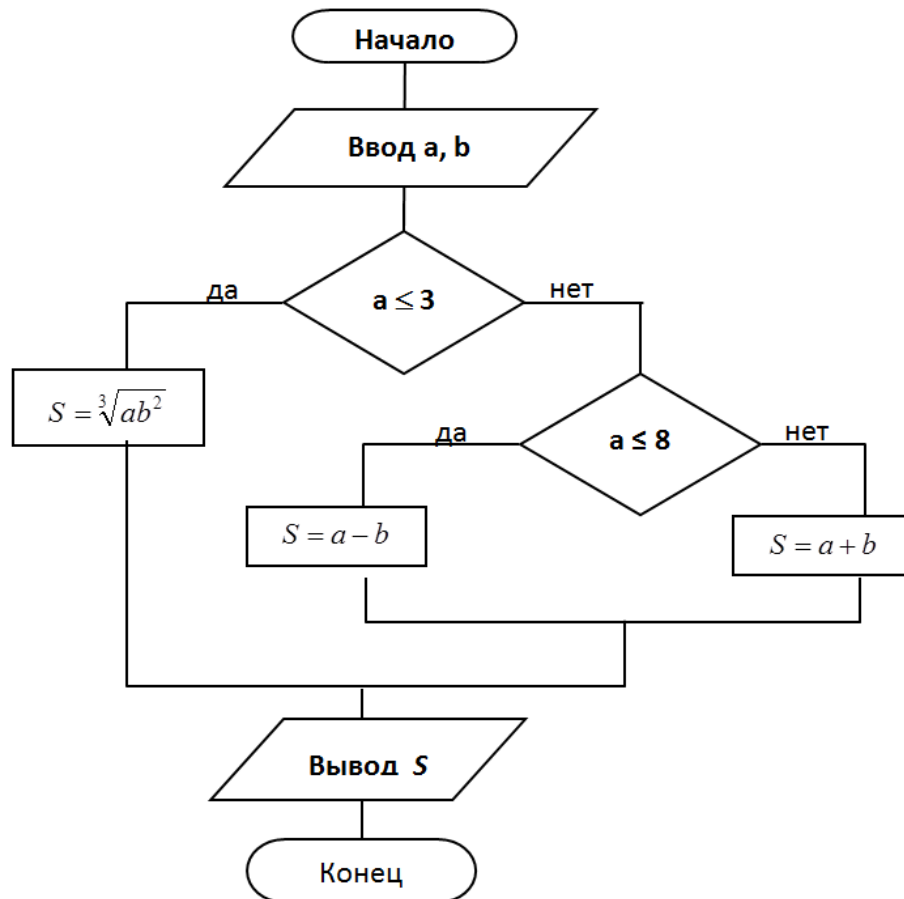


Рисунок 9.1 – Блок-схема алгоритма

2. Создайте файл с именем **Lab9_1.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 9.2](#)).
4. Сохраните файл.
5. Запустите программу на выполнение.
6. Выполните отладку программы для значений $b=3, a=1, a=7, a=10$.

```

Lab3_1.py
#coding=utf-8
from math import *
a=float(input('Введите a= '))
b=float(input('Введите b= '))
if a<=3:
    S=(a*b**2)**(1.0/3)
elif a<=8:
    S=a-b
else:
    S=a+b
print 'S=', S

```

Рисунок 9.2 – Текст программы

Упражнение 2

В этом упражнении вы составите блок-схему и напишите программу для следующей задачи.

Задача. Дана точка А с координатами (х, у), х и у отличны от нуля. Укажите, какой четверти координатной плоскости принадлежит данная точка.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 9.3](#)).

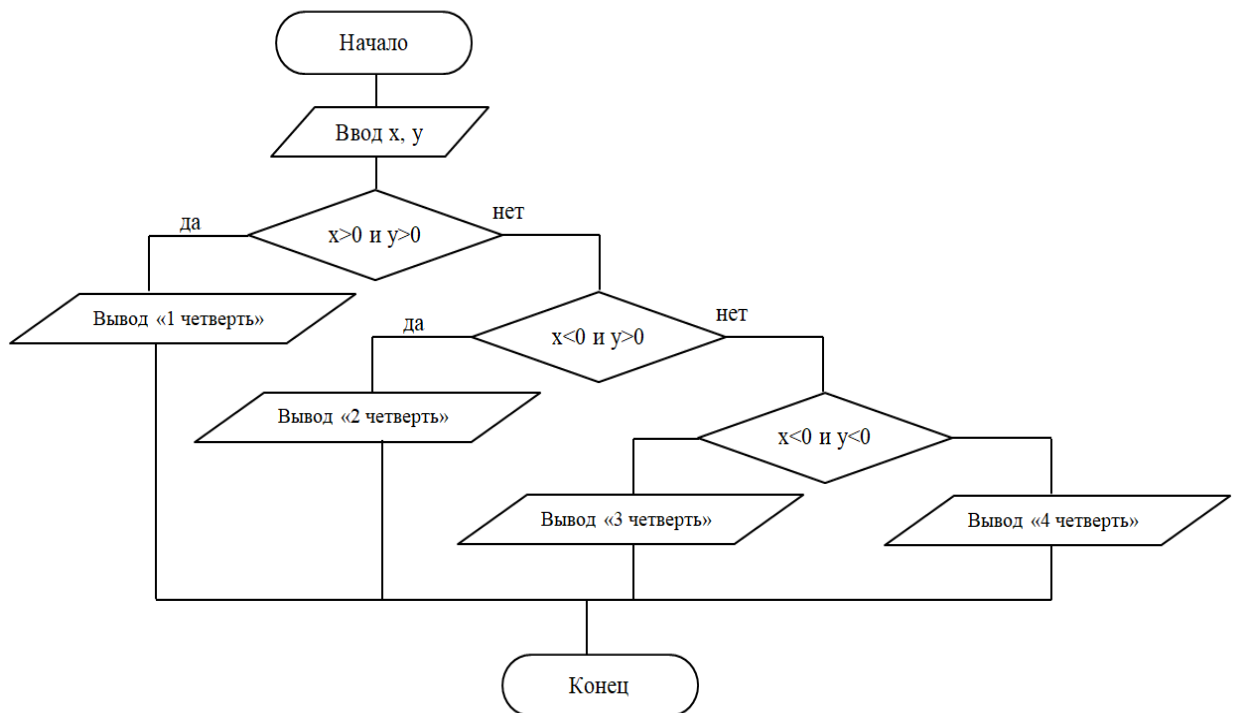
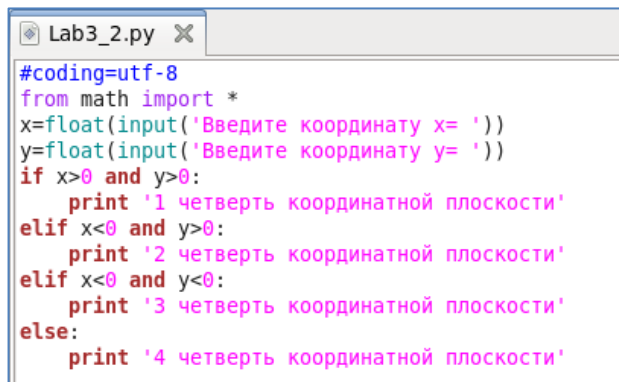


Рисунок 9.3 – Блок-схема алгоритма

2. Создайте файл с именем **Lab9_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 9.4](#)).



```
#coding=utf-8
from math import *
x=float(input('Введите координату x= '))
y=float(input('Введите координату y= '))
if x>0 and y>0:
    print '1 четверть координатной плоскости'
elif x<0 and y>0:
    print '2 четверть координатной плоскости'
elif x<0 and y<0:
    print '3 четверть координатной плоскости'
else:
    print '4 четверть координатной плоскости'
```

Рисунок 9.4 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение.
6. Выполните отладку программы для различных значений x и y .

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab9_3.py**.

1.1. Составьте схему алгоритма и напишите программу для вычисления значения функции Z :

$$Z = \begin{cases} k^3 + \sqrt{k}, & \text{при } k > 7 \\ e^{k+1} - \sin^2 k, & \text{при } -2,4 < k \leq 7, \\ \sin(|k - 7,6|), & \text{при } k \leq -2,4 \end{cases}$$

где k – произвольное число из области допустимых значений.

1.2. Составьте схему алгоритма и напишите программу для вычисления значения функции Y :

$$Y = \begin{cases} \frac{\sqrt[4]{|x+2|}}{c - \lg^3(x+7)}, & \text{если } -2 < x < 5 \\ e^{(c-x)^2}, & \text{если } x \geq 5 \\ \operatorname{arctg}(x-1), & \text{в остальных случаях} \end{cases},$$

где $c = \cos x + \sin^2(1+x) - e^x$, x – произвольное.

1.3. Составьте схему алгоритма и напишите программу для вычисления значения функции Z :

$$Z = \begin{cases} y, & \text{при } x = 0 \\ e^{x+1} - \sin^2 y, & \text{при } x > 0, \\ xy - \cos 20^\circ, & \text{при } x < 0 \end{cases}$$

где x, y – произвольные.

1.4. Составьте схему алгоритма и напишите программу для вычисления значения функции F :

$$F = \begin{cases} |x - \operatorname{tg} b|, & \text{при } x < 5 \\ e^{x+1,2}, & \text{при } 5 \leq x < 12 \\ b + \log_3(x + 1), & \text{при } x \geq 12 \end{cases},$$

где x, b – произвольные.

1.5. Составьте схему алгоритма и напишите программу для вычисления значения функции Y :

$$Y = \begin{cases} ax + \ln(x + b), & \text{если } x < b \\ \operatorname{tg} x + \frac{\operatorname{ctg}(a+x)}{b-1}, & \text{если } b \leq x \leq a, \\ \sqrt[3]{ax + b \cdot \lg x}, & \text{если } x > a \end{cases}$$

где $a = 7,8, b = 1,12, x$ – произвольное.

1.6. Составьте схему алгоритма и напишите программу для вычисления значения функции z :

$$z = \begin{cases} \frac{\ln(x+b)^2}{b \cdot e^{ax}}, & \text{если } x < 0 \\ \operatorname{ctg} x + b \cdot \sin^2(ax), & \text{если } 0 \leq x \leq 5, \\ \operatorname{tg} x + |x - a|, & \text{если } x > 5 \end{cases}$$

где $a = 3,2, b = 7, x$ – произвольное.

1.7. Составьте схему алгоритма и напишите программу для вычисления значения функции f :

$$f = \begin{cases} \sqrt{|t+8|} - \frac{1}{at-c}, & \text{если } t < 0 \\ \cos^3(a+t) + ct \cdot \operatorname{arctg}(at), & \text{если } 0 \leq t \leq 3, \\ \frac{\sqrt{|t-c|}}{\ln^3 t}, & \text{если } t > 3 \end{cases}$$

где $a = -4,3, c = 6, t$ – произвольное.

1.8. Составьте схему алгоритма и напишите программу для вычисления значения функции z :

$$z = \begin{cases} \cos(x+1), & \text{если } y < 0 \\ \operatorname{tg}|y-1| - e^{x+2,5}, & \text{если } 0 \leq y < 1, \\ y + \log_5 x, & \text{если } y \geq 1 \end{cases}$$

где $y = \cos^3 x + \sin^3 x, x$ – произвольное.

1.9. Составьте схему алгоритма и напишите программу для вычисления значения функции w :

$$w = \begin{cases} \sqrt{t}, & \text{если } t = 20 \\ e^{k+1} - \sin^4 k, & \text{если } t > 20, \\ \cos 20^\circ, & \text{если } t < 20 \end{cases}$$

где $t = \frac{\lg k + \sqrt[3]{\lg(k+7)}}{7}$, k – произвольное.

1.10. Составьте схему алгоритма и напишите программу для вычисления значения функции r :

$$r = \begin{cases} \frac{\sqrt[3]{x}}{\lg(x+0,1)-c}, & \text{если } 0,2 < x < 3 \\ \lg x + |x - c|, & \text{если } x \geq 3 \\ \lg(x + c), & \text{в остальных случаях} \end{cases},$$

где $c = \cos^3(1 - x) - e^{x+1}$, x – произвольное.

1.11. Составьте схему алгоритма и напишите программу для вычисления значения функции f :

$$f = \begin{cases} |x + \operatorname{ctg}(b)|, & \text{если } x < 5 \\ e^{x+1,2}, & \text{если } 5 \leq x \leq 12 \\ b + \ln(x + 1)^2, & \text{в остальных случаях} \end{cases},$$

где $b = \cos x^2$, x – произвольное.

1.12. Составьте схему алгоритма и напишите программу для вычисления значения функции Y :

$$Y = \begin{cases} ax + \ln(x + b), & \text{если } x < b \\ \frac{\sqrt{|x-ab|}}{\ln^3(x+10)}, & \text{если } b \leq x \leq a \\ e^{x+1,2}, & \text{если } x > a \end{cases},$$

где $a = -1,8$, $b = 2$, x – произвольное.

2. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab9_4.py**.

2.1. Составьте схему алгоритма и напишите программу. Даны три числа x , y , z . Возвести эти числа в квадрат, если $x < y < z$, в противном случае – увеличить каждое число на 2.

2.2. Составьте схему алгоритма и напишите программу. Даны три числа. Вывести на экран те из них, которые принадлежат интервалу $[3, 9)$.

2.3. Составьте схему алгоритма и напишите программу. Даны три числа x , y , z . Возвести в квадрат неотрицательные числа.

- 2.4. Составьте схему алгоритма и напишите программу. Даны три вещественных числа. Возвести в квадрат отрицательные числа и в пятую степень – положительные (число 0 не изменять).
- 2.5. Составьте схему алгоритма и напишите программу для проверки условия: является ли треугольник со сторонами a , b , c равносторонним.
- 2.6. Составьте схему алгоритма и напишите программу для проверки условия: является ли треугольник со сторонами a , b , c равнобедренным.
- 2.7. Составьте схему алгоритма и напишите программу. Даны вещественные числа x , y ($x \neq y$). Каждое из чисел заменить модулем, если оба числа отрицательные, в противном случае – увеличить числа в три раза.
- 2.8. Составьте схему алгоритма и напишите программу. Даны вещественные числа x , y ($x \neq y$). Заменить суммой чисел меньшее из них, удвоенным произведением чисел – большее из них.
- 2.9. Составьте схему алгоритма и напишите программу. Даны три точки x , y , z на числовой оси. Определить, какая из двух последних точек (y или z) расположена ближе к x . Вывести координату этой точки и расстояние до точки x .
- 2.10. Составьте схему алгоритма и напишите программу. Даны координаты точки на координатной плоскости. Вывести на экран: 0 – если точка не лежит на координатных осях; 1 – лежит на оси OX или 2 – лежит на оси OY .
- 2.11. Составьте схему алгоритма и напишите программу для проверки условия: лежит ли точка с координатами (x, y) в первой четверти координатной плоскости.
- 2.12. Даны три числа x , y , z . Вывести на экран те числа, которые кратны 5.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схемы алгоритмов, листинги программ, исходные данные, результаты выполнения программ и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой вычислительный процесс называется разветвляющимся?
2. С помощью каких операторов реализуется разветвляющийся вычислительный процесс?
3. Синтаксис оператора `if`.
4. Синтаксис оператора `if-else`.
5. Синтаксис оператора `if-elif-else`.

ЛАБОРАТОРНАЯ РАБОТА № 10

Тема: **PYTHON. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ ЦИКЛИЧЕСКОГО ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с управляющими операторами циклического вычислительного процесса.

Научиться:

- реализовывать циклические алгоритмы на языке программирования Python.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ**

Процесс, в котором оператор или группа операторов повторяются многократно при различных значениях входящих величин, называется *циклическим*.

Циклический вычислительный процесс реализуется с помощью следующих операторов цикла:

- `while`;
- `for`.

3.1 Оператор цикла while

С помощью оператора цикла `while` реализуется циклический вычислительный процесс с предусловием – условие выхода из цикла проверяется до выполнения тела цикла (в этом случае цикл может не выполниться ни разу).

Синтаксис оператора цикла `while`:

```
while логическое_выражение:  
    инструкция1  
...  
    инструкцияN
```


Семантика оператора цикла while. В этом цикле инструкции выполняются до тех пор, пока значение выражения равно true (истина). Как только условие становится ложным, управление программой передаётся строке кода, следующей непосредственно после цикла.

Упражнение 1

В этом упражнении вы составите блок-схему и разработаете программу, вычисляющую элементы последовательности a_n с заданной точностью $\varepsilon = 0,1$.

$$a_n = \frac{1}{n+2}$$

Решение:

1. Составьте блок-схему алгоритма ([рисунок 10.1](#)).

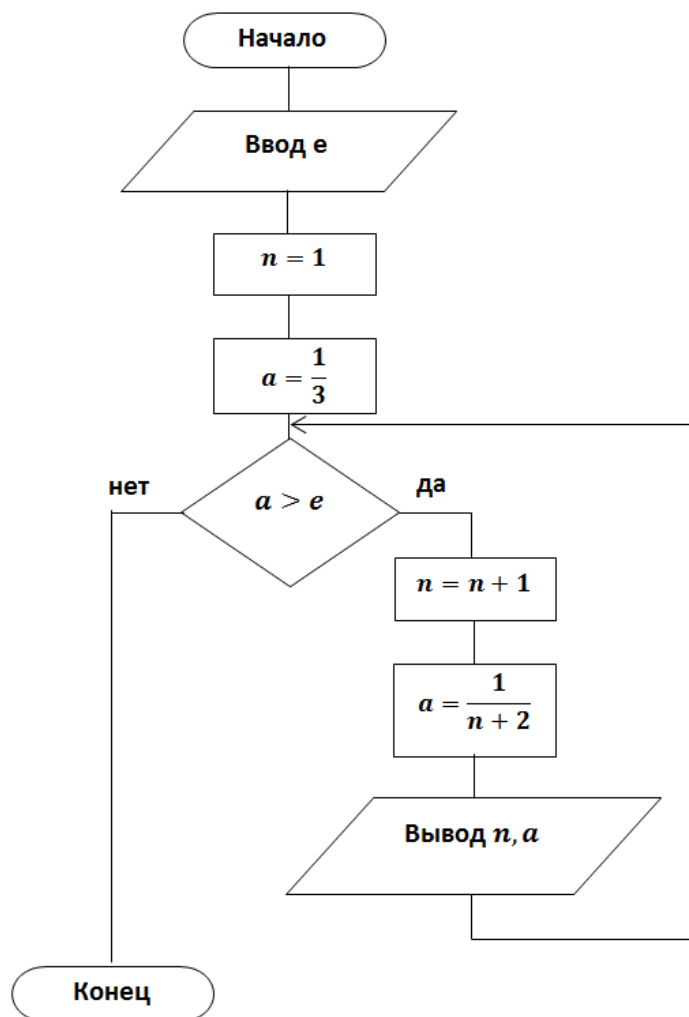
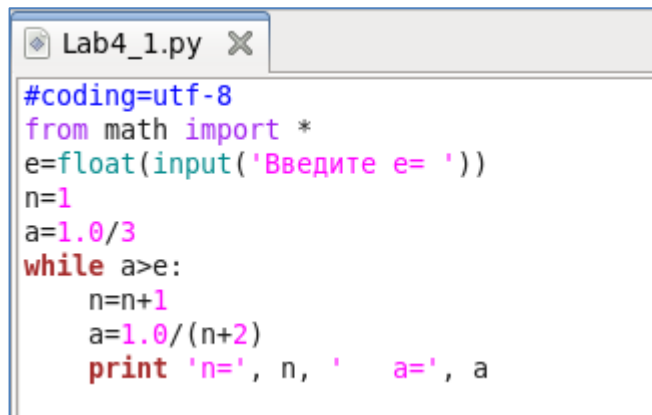


Рисунок 10.1 – Блок-схема алгоритма

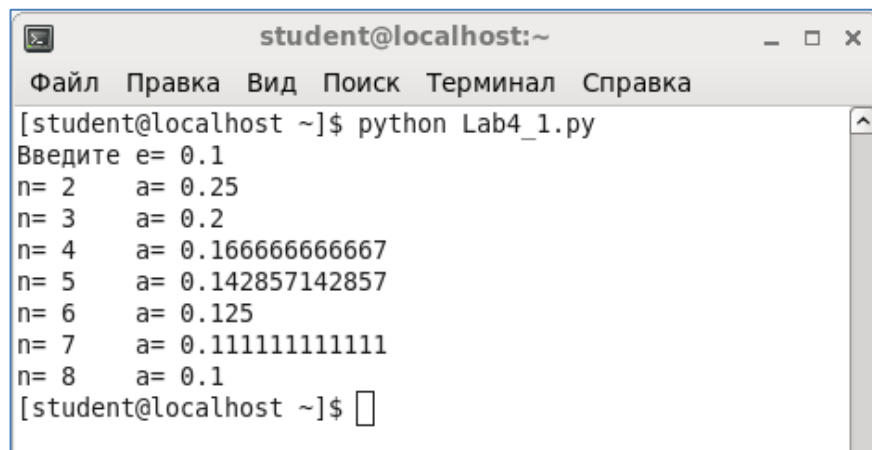
2. Создайте файл с именем **Lab10_1.py**.
3. Откройте созданный файл и наберите текст программы (см. [рисунок 10.2](#)).
4. Сохраните файл.

5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунком 10.3](#).



```
#coding=utf-8
from math import *
e=float(input('Введите e= '))
n=1
a=1.0/3
while a>e:
    n=n+1
    a=1.0/(n+2)
    print 'n=', n, '    a=', a
```

Рисунок 10.2 – Текст программы



```
student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python Lab4_1.py
Введите e= 0.1
n= 2    a= 0.25
n= 3    a= 0.2
n= 4    a= 0.1666666666667
n= 5    a= 0.142857142857
n= 6    a= 0.125
n= 7    a= 0.1111111111111
n= 8    a= 0.1
[student@localhost ~]$
```

Рисунок 10.3 – Результат выполнения программы

Упражнение 2

В этом упражнении вы составите блок-схему и напишите программу, вычисляющую значения функции z :

$$z = \frac{\sqrt[3]{bx^2} + \cos(x - y)}{e^{x+1,2}},$$

где $x \in [3; 6]$, $\Delta x = 0,2$, $y = 4,1$, $b = 7,3$.

Решение:

1. Составьте блок-схему алгоритма ([рисунком 10.4](#)).
2. Создайте файл с именем **Lab10_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунком 10.5](#)).

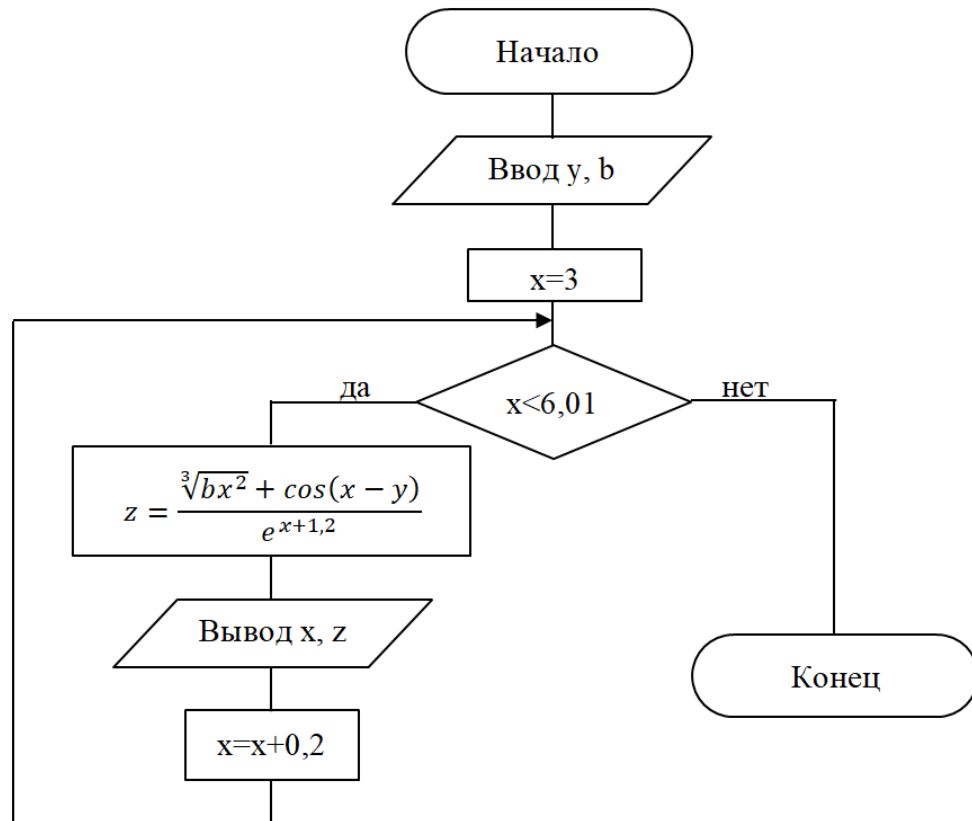


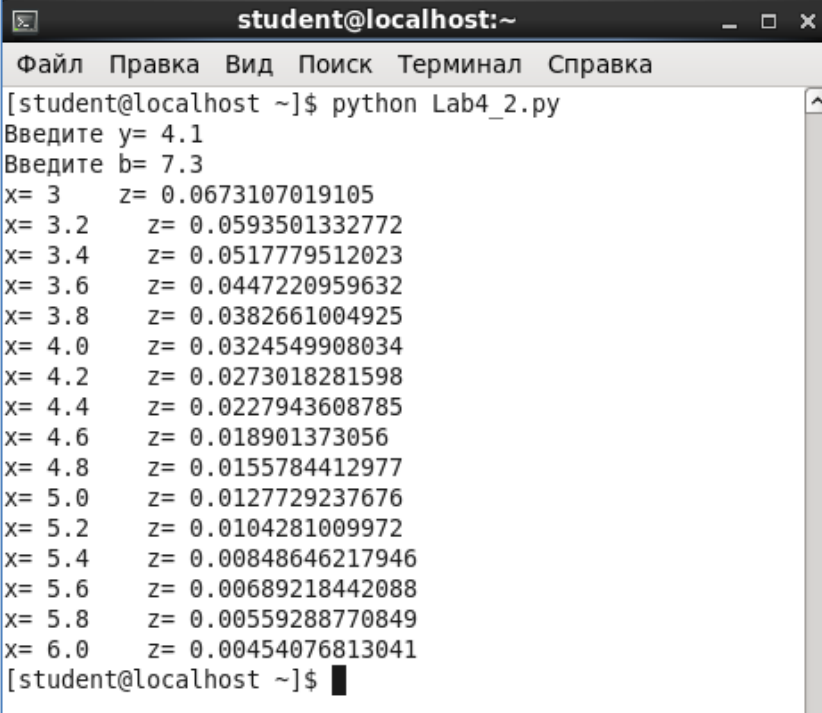
Рисунок 10.4 – Блок-схема алгоритма

```

#coding=utf-8
from math import *
y=float(input('Введите y= '))
b=float(input('Введите b= '))
x=3
while x<6.01:
    z=((b*x**2)**(1.0/3)+cos(x-y))/exp(x+1.2)
    print 'x=', x, ' z=', z
    x+=0.2
  
```

Рисунок 10.5 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунком 10.6](#).



```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python Lab4_2.py
Введите y= 4.1
Введите b= 7.3
x= 3      z= 0.0673107019105
x= 3.2    z= 0.0593501332772
x= 3.4    z= 0.0517779512023
x= 3.6    z= 0.0447220959632
x= 3.8    z= 0.0382661004925
x= 4.0    z= 0.0324549908034
x= 4.2    z= 0.0273018281598
x= 4.4    z= 0.0227943608785
x= 4.6    z= 0.018901373056
x= 4.8    z= 0.0155784412977
x= 5.0    z= 0.0127729237676
x= 5.2    z= 0.0104281009972
x= 5.4    z= 0.00848646217946
x= 5.6    z= 0.00689218442088
x= 5.8    z= 0.00559288770849
x= 6.0    z= 0.00454076813041
[student@localhost ~]$

```

Рисунок 10.6 – Результат выполнения программы

3.2 Операторы break и continue

При работе с циклами используются операторы `break` и `continue`. Оператор `break` предназначен для досрочного прерывания работы цикла `while` и выход из него.

Оператор `continue` вызывает немедленный переход в начало цикла. Этот оператор иногда позволяет избежать использования вложенных инструкций.

3.3 Оператор цикла for

Оператор цикла **for** используется в том случае, если известно количество повторений каких-либо операций. Оператор **for** выполняет указанный набор инструкций заданное количество раз, которое определяется количеством элементов в наборе.

Синтаксис оператора:

for переменная in диапазон:
 тело цикла

Тело цикла – инструкции, выполняемые многократно.

Блок кода после заголовка выполняется, пока **переменная** принадлежит указанному **диапазону** (причём этот диапазон может быть списком, числовой последовательностью, массивом каких-либо значений).

3.4 Функция range

Range() является универсальной функцией Python для создания списков (**list**), содержащих арифметическую прогрессию. Чаще всего она используется в циклах **for**.

Функция **range()** может принимать от одного до трех аргументов, при этом аргументами должны быть целые числа (**int**).

Синтаксис функции:

range(старт, стоп, шаг)

где **старт** – начальное значение списка, если старт не задан, то равняется нулю;

шаг – разность между соседними элементами списка, если шаг не задан, то он равен единице, шаг может принимать целые положительные и отрицательные значения, не должен равняться нулю;

стоп – конечное значение списка.

Рассмотрим применение функции **range()** на следующих примерах.

```
>>>range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Эта функция возвращает все целые числа в диапазоне от 0 до числа 10, не включая само число 10.

```
>>>range(5, 10)
[5, 6, 7, 8, 9]
```

Эта функция возвращает все целые числа в диапазоне от 5 до числа 10, не включая само число 10, но включая начальное число 5.

```
>>>range(0, 12, 3)
[0, 3, 6, 9]
```

Эта функция возвращает все целые числа в диапазоне от 0 до числа 10 с шагом 3, включая начальное число 0, но не включая число 12.

```
>>>range(-10, -100, -30)
[-10, -40, -70]
```

Эта функция возвращает все целые отрицательные числа в диапазоне от -10 до числа -100 с шагом -30, включая начальное число -10, но не включая число -100.

Упражнение 3

В этом упражнении вы составите блок-схему и напишите программу, вычисляющую значения функции z :

$$z = \cos\left(\frac{a^2 - b}{27}\right)$$

для $b \in [-2; 14]$, $\Delta b = 1$, a – любое число.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 10.7](#)).
2. Создайте файл с именем **Lab10_3.py**.

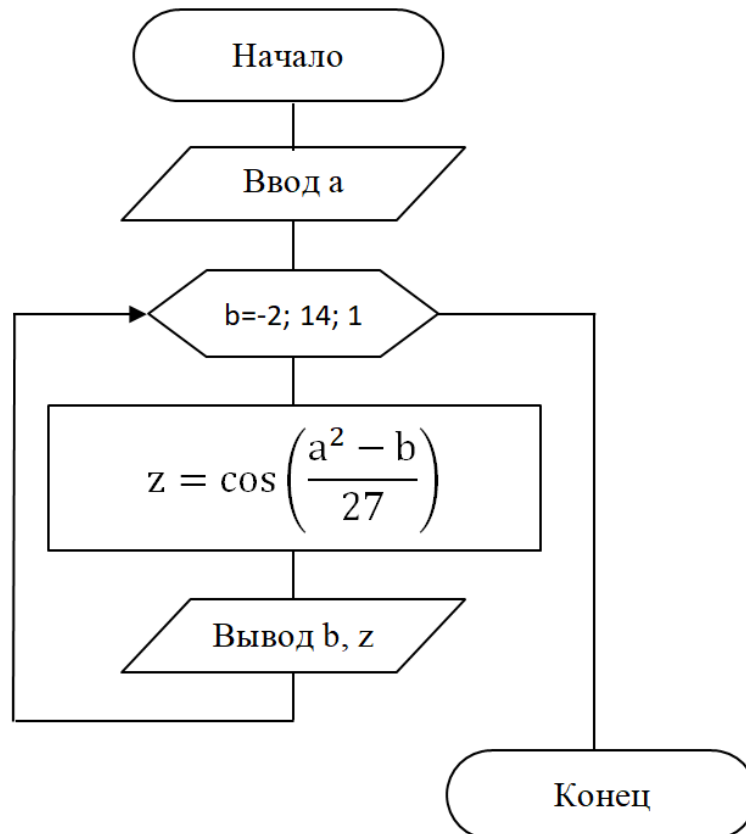
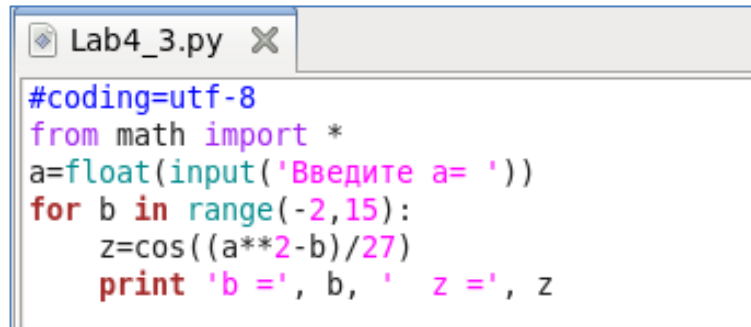


Рисунок 10.7 – Блок-схема алгоритма

3. Откройте созданный файл и наберите текст программы ([рисунок 10.8](#)). Обратите внимание на запись функции `range(-2, 15)`. Параметры функции указывают на начальное значение (старт) переменной b – это -2 , и на конечное значение (стоп) переменной b – это 15 . Конечное значение отличается от значения в задаче, т.к. для достижения значения 14 необходимо указать значение больше, согласно семантике функции `range()`. Третьего значения у функции `range()` нет, т.к. шаг равен 1 .

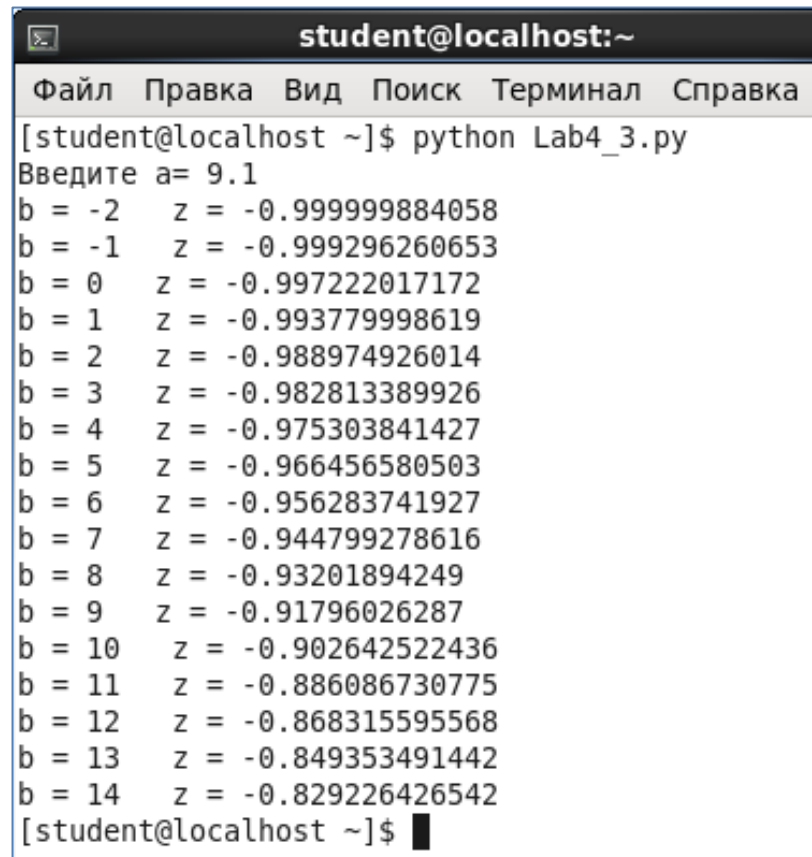
4. Сохраните файл.

5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунок 10.9](#).



```
#coding=utf-8
from math import *
a=float(input('Введите a= '))
for b in range(-2,15):
    z=cos((a**2-b)/27)
    print 'b =', b, ' z =', z
```

Рисунок 10.8 – Текст программы



```
student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab4_3.py
Введите a= 9.1
b = -2 z = -0.999999884058
b = -1 z = -0.999296260653
b = 0 z = -0.997222017172
b = 1 z = -0.993779998619
b = 2 z = -0.988974926014
b = 3 z = -0.982813389926
b = 4 z = -0.975303841427
b = 5 z = -0.966456580503
b = 6 z = -0.956283741927
b = 7 z = -0.944799278616
b = 8 z = -0.93201894249
b = 9 z = -0.91796026287
b = 10 z = -0.902642522436
b = 11 z = -0.886086730775
b = 12 z = -0.868315595568
b = 13 z = -0.849353491442
b = 14 z = -0.829226426542
[student@localhost ~]$
```

Рисунок 10.9 – Результат выполнения программы

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab10_4.py**.

1.1. Переходный процесс описывается уравнением $i = i_{np} + Ae^{pt}$, A . Разработать программу, позволяющую пользователю ввести параметры: i_{np} , A , p ($p < 0$). Параметр t изменяется от 0 до $6/p$ с шагом $0,5/p$. На основе введенных данных программа должна вычислить значения токов в указанном временном диапазоне с заданным шагом и вывести результаты расчетов.

1.2. Переходный процесс описывается уравнением $u = Ae^{pt}$, В. Разработать программу, позволяющую пользователю ввести параметры: A, p ($p < 0$). Параметр t изменяется от 0 до $5/p$ с шагом $0,25/p$. На основе введенных данных программа должна вычислить значения напряжений в указанном временном диапазоне с заданным шагом и вывести результаты расчетов.

1.3. Активное сопротивление (R_1 , Ом) и катушка индуктивности (R_k , Ом – активное сопротивление катушки; L_k , Гн – индуктивность катушки) соединены последовательно и подключены к идеальному источнику переменного напряжения. Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля полного сопротивления от частоты питающего напряжения в заданном пользователем диапазоне (f_1, f_2) с заданным пользователем шагом (df). Использовать следующие зависимости:

$$\omega = 2\pi f - \text{угловая частота, рад/с;}$$

$$X_L = \omega L_k - \text{индуктивное сопротивление катушки;}$$

$$R_{ekv} = R_1 + R_k, \text{ Ом} - \text{полное активное сопротивление цепи, Ом;}$$

$$X_{ekv} = X_L, \text{ Ом} - \text{полное реактивное сопротивление цепи, Ом;}$$

$$Z = \sqrt{R_{ekv}^2 + X_{ekv}^2} - \text{полное сопротивление цепи, Ом.}$$

1.4. Активное сопротивление (R_1 , Ом) и катушка индуктивности (R_k , Ом – активное сопротивление катушки; L_k , Гн – индуктивность катушки) соединены последовательно и подключены к идеальному источнику переменного напряжения (U). Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля тока от частоты питающего напряжения в заданном пользователем диапазоне (f_1, f_2) с заданным пользователем шагом (df). Использовать следующие зависимости:

$$\omega = 2\pi f - \text{угловая частота, рад/с;}$$

$$X_L = \omega L_k - \text{индуктивное сопротивление катушки;}$$

$$R_{ekv} = R_1 + R_k, \text{ Ом} - \text{полное активное сопротивление цепи, Ом;}$$

$$X_{ekv} = X_L, \text{ Ом} - \text{полное реактивное сопротивление цепи, Ом;}$$

$$Z = \sqrt{R_{ekv}^2 + X_{ekv}^2} - \text{полное сопротивление цепи, Ом.}$$

$$I = \frac{U}{Z}, \text{ А.}$$

1.5. Активное сопротивление (R_1 , Ом) и катушка индуктивности (R_k , Ом – активное сопротивление катушки; L_k , Гн – индуктивность катушки) соединены последовательно и подключены к идеальному источнику переменного напряжения с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля полного сопротивления от величины индуктивности в заданном

пользователем диапазоне (L_{k1}, L_{k2}) с заданным пользователем шагом (dL_k) . Использовать следующие зависимости:

$$\omega = 2\pi f - \text{угловая частота, рад/с;}$$

$$X_L = \omega L_k - \text{индуктивное сопротивление катушки;}$$

$$R_{ekv} = R_1 + R_k, \text{ Ом} - \text{полное активное сопротивление цепи, Ом;}$$

$$X_{ekv} = X_L, \text{ Ом} - \text{полное реактивное сопротивление цепи, Ом;}$$

$$Z = \sqrt{R_{ekv}^2 + X_{ekv}^2} - \text{полное сопротивление цепи, Ом.}$$

1.6. Активное сопротивление $(R_1, \text{ Ом})$ и катушка индуктивности $(R_k, \text{ Ом} - \text{активное сопротивление катушки; } L_k, \text{ Гн} - \text{индуктивность катушки})$ соединены последовательно и подключены к идеальному источнику переменного напряжения $(U, \text{ В})$ с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля тока от величины индуктивности в заданном пользователем диапазоне (L_{k1}, L_{k2}) с заданным пользователем шагом (dL_k) . Использовать следующие зависимости:

$$\omega = 2\pi f - \text{угловая частота, рад/с;}$$

$$X_L = \omega L_k - \text{индуктивное сопротивление катушки;}$$

$$R_{ekv} = R_1 + R_k, \text{ Ом} - \text{полное активное сопротивление цепи, Ом;}$$

$$X_{ekv} = X_L, \text{ Ом} - \text{полное реактивное сопротивление цепи, Ом;}$$

$$Z = \sqrt{R_{ekv}^2 + X_{ekv}^2} - \text{полное сопротивление цепи, Ом.}$$

$$I = \frac{U}{Z}, \text{ А.}$$

1.7. Активное сопротивление $(R_1, \text{ Ом})$ и конденсатор C ($-$ ёмкость конденсатора, Φ) соединены последовательно и подключены к идеальному источнику переменного напряжения f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля полного сопротивления от частоты питающего напряжения в заданном пользователем диапазоне (f_1, f_2) с заданным пользователем шагом (df) . Использовать следующие зависимости:

$$\omega = 2\pi f - \text{угловая частота, рад/с;}$$

$$X_C = \frac{1}{\omega C} - \text{емкостное сопротивление конденсатора, Ом;}$$

$$Z = \sqrt{R_1^2 + X_C^2} - \text{полное сопротивление цепи, Ом.}$$

1.8. Активное сопротивление $(R_1, \text{ Ом})$ и конденсатор C ($-$ ёмкость конденсатора, Φ) соединены последовательно и подключены к идеальному источнику переменного напряжения $(U, \text{ В})$ с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля тока от частоты питающего напряжения в заданном пользователем диапазоне (f_1, f_2) с заданным пользователем шагом (df) . Использовать следующие зависимости:

$\omega = 2\pi f$ – угловая частота, рад/с;

$X_C = \frac{1}{\omega C}$ – емкостное сопротивление конденсатора, Ом;

$Z = \sqrt{R_1^2 + X_C^2}$ – полное сопротивление цепи, Ом.

$I = \frac{U}{Z}$, А.

1.9. Активное сопротивление (R_1 , Ом) и конденсатор (C – ёмкость конденсатора, Ф) соединены последовательно и подключены к идеальному источнику переменного напряжения с частотой f . Для заданной электрической цепи разработать консольное приложение, позволяющее исследовать зависимость изменения модуля полного сопротивления Z от величины ёмкости конденсатора в заданном пользователем диапазоне (C_1, C_2) с заданным пользователем шагом (dC). Использовать следующие зависимости:

$\omega = 2\pi f$ – угловая частота, рад/с;

$X_C = \frac{1}{\omega C}$ – емкостное сопротивление конденсатора, Ом;

$Z = \sqrt{R_1^2 + X_C^2}$ – полное сопротивление цепи, Ом.

1.10. Активное сопротивление (R_1 , Ом) и конденсатор (C – ёмкость конденсатора, Ф) соединены последовательно и подключены к идеальному источнику переменного напряжения (U, V) с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля тока I от величины ёмкости конденсатора в заданном пользователем диапазоне (C_1, C_2) с заданным пользователем шагом (dC). Использовать следующие зависимости:

$\omega = 2\pi f$ – угловая частота, рад/с;

$X_C = \frac{1}{\omega C}$ – емкостное сопротивление конденсатора, Ом;

$Z = \sqrt{R_1^2 + X_C^2}$ – полное сопротивление цепи, Ом.

$I = \frac{U}{Z}$, А.

1.11. Активное сопротивление (R , Ом) и конденсатор C (C – ёмкость конденсатора, Ф) соединены последовательно и подключены к идеальному источнику переменного напряжения с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля полного сопротивления от величины активного сопротивления в заданном пользователем диапазоне (R_1, R_2) с заданным пользователем шагом (dR). Использовать следующие зависимости:

$\omega = 2\pi f$ – угловая частота, рад/с;

$X_C = \frac{1}{\omega C}$ – емкостное сопротивление конденсатора, Ом;

$Z = \sqrt{R^2 + X_C^2}$ – полное сопротивление цепи, Ом.

1.12. Активное сопротивление (R , Ом) и конденсатор (C – ёмкость конденсатора, Ф) соединены последовательно и подключены к идеальному источнику переменного напряжения (U , В) с частотой f . Для заданной электрической цепи разработать программу, позволяющую исследовать зависимость изменения модуля тока от величины активного сопротивления в заданном пользователем диапазоне (R_1, R_2) с заданным пользователем шагом (dR). Использовать следующие зависимости:

$\omega = 2\pi f$ – угловая частота, рад/с;

$X_C = \frac{1}{\omega C}$ – емкостное сопротивление конденсатора, Ом;

$Z = \sqrt{R^2 + X_C^2}$ – полное сопротивление цепи, Ом.

$I = \frac{U}{Z}$, А.

2. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab10_5.py**.

2.1. Составьте схему алгоритма и напишите программу. Вычислите значение функции $z = \lg(m)$ для $m \in [2; 7]$, $\Delta m = 1$.

2.2. Составьте схему алгоритма и напишите программу. Вычислите значение функции $y = 3x + \sqrt{|ax - e^{3x}|}$ для $x \in [2; 16]$, $\Delta x = 2$, $a = \lg(x - 1)$.

2.3. Составьте схему алгоритма и напишите программу. Вычислите значение функции $z = \sin^2(t + b) - \sqrt[4]{|x - b|}$ для $t \in [-1; 9]$, $\Delta t = 1$, $b = e^{2t-t^2}$.

2.4. Составьте схему алгоритма и напишите программу. Вычислите значение функции $f = \operatorname{tg}(x + ax)^2$ для $x \in [0; 8]$, $\Delta x = 1$, $a = \frac{\ln|3b-2|}{7}$, $b = 2,1$.

2.5. Составьте схему алгоритма и напишите программу. Вычислите значение функции $R = \ln|x + ax| - \sqrt[3]{|x - ba^2|}$ для $x \in [-1; 11]$, $\Delta x = 2$, $a = 2b + \cos^2(b)$, $b = 3,1$.

2.6. Составьте схему алгоритма и напишите программу. Вычислите значение функции $f = \sin(2x + y) - a$ для $x \in [1; 20]$, $\Delta x = 1$, $a = 4,2$, $y = 1,3$.

2.7. Составьте схему алгоритма и напишите программу. Вычислите значение функции $M = x + 2,4y^3 - b$ для $y \in [2; 13]$, $\Delta y = 2$, $b = 1,3$, $x = 5,3$.

2.8. Составьте схему алгоритма и напишите программу. Вычислите значение функции $z = \cos(x - y) + bx^2$ для $y \in [3; 8]$, $\Delta y = 1$, $x = 4,1$, $b = 7,3$.

2.9. Составьте схему алгоритма и напишите программу. Вычислите значение функции $f = 2\sin(3t - y) + bt$ для $t \in [1; 8]$, $\Delta t = 1$, $y = 5,1$, $b = 1,2$.

2.10. Составьте схему алгоритма и напишите программу. Вычислите значение функции $f = 2cd - 3x^2$ для $x \in [1; 9]$, $\Delta x = 1$, $c = 2,3$, $d = 4,2$.

2.11. Составьте схему алгоритма и напишите программу. Вычислите значение функции $z = \ln(x^3 - ax^2) - y$ для $x \in [2; 9]$, $\Delta x = 2$, $a = 0,21$, $y = 3,4$.

2.12. Плотность воздуха убывает с высотой h по закону $\rho = \rho_0 \cdot e^{-hz}$, где $\rho_0 = 1,3 \text{ кг/м}^3$, $z = 1,4 \cdot 10^{-4} \text{ 1/м}$. Составьте схему алгоритма и напишите программу, которая выводит на экран таблицу зависимости плотности воздуха от высоты для значений h , изменяющихся от 0 до 1000 м с шагом 20 м.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схемы алгоритмов, листинги программ, исходные данные, результаты выполнения программ и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой процесс называется циклическим?
2. Какие вы знаете операторы цикла?
3. Оператор цикла `while`. Его синтаксис и семантика.
4. Назначение операторов `break` и `continue`.
5. Оператор цикла `for`. Его синтаксис.
6. Назначение и синтаксис функции `range()`.

ЛАБОРАТОРНАЯ РАБОТА № 11**Тема: PYTHON. КОМБИНИРОВАННЫЕ АЛГОРИТМЫ****1 ЦЕЛЬ РАБОТЫ**

Научиться:

- реализовывать комбинированные алгоритмы на языке программирования Python.

**2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК
ЕЁ ВЫПОЛНЕНИЯ**

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ**

К *комбинированным алгоритмам* относятся алгоритмы, в которых для решения задачи используется несколько алгоритмических структур. Например, комбинирование разветвляющейся структуры и циклической или двух циклических структур.

Упражнение 1

В этом упражнении вы составите блок-схему и напишете программу, вычисляющую значения функции Y :

$$Y = \begin{cases} a + \cos(x + b), & \text{если } x < b \\ \operatorname{tg} x + \frac{\operatorname{ctg}(a+x)}{b-1}, & \text{если } b \leq x \leq a, \\ \sqrt[3]{|a + b \cdot \lg x|}, & \text{если } x > a \end{cases}$$

где $a = 5,6$, $b = 1,2$, $x \in [1; 10]$, $\Delta x = 1$.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 11.1](#)).

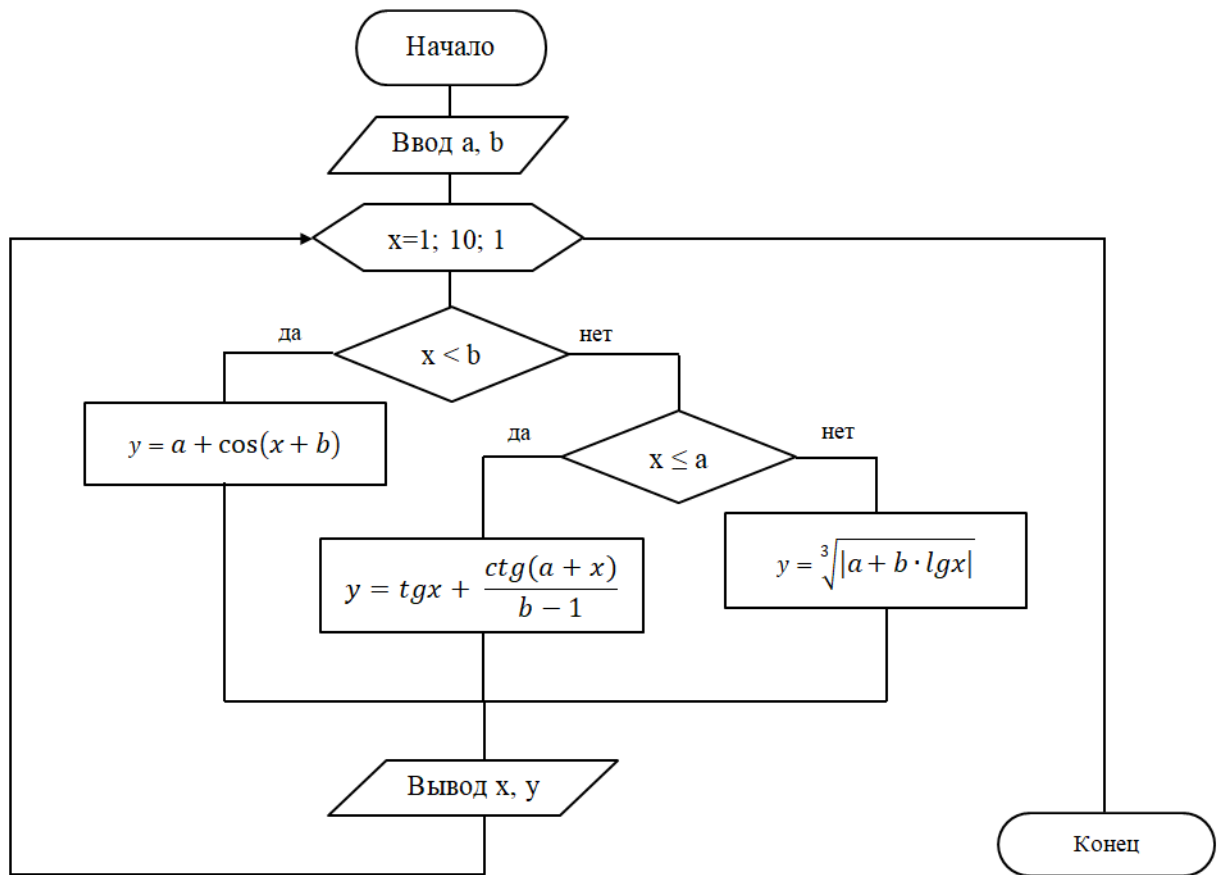


Рисунок 11.1 – Блок-схема алгоритма

2. Создайте файл с именем **Lab11_1.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 11.2](#)).

```

Lab5_1.py
#coding=utf-8
from math import *
a=float(input('Введите a= '))
b=float(input('Введите b= '))
for x in range(1,11):
    if x<b:
        y=a+cos(x+b)
    elif x<=a:
        y=tan(x)+1.0/(tan(a+x)*(b-1))
    else:
        y=abs(a+b*log(x))**(1.0/3)
    print 'x =', x, ' y =', y
  
```

Рисунок 11.2 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунок 11.3](#).

```

student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab5_1.py
Введите a= 6.8
Введите b= 3.4
x = 1   y = 6.49266713002
x = 2   y = 7.43469287594
x = 3   y = 7.79318491876
x = 4   y = 1.24036568419
x = 5   y = -3.81334751104
x = 6   y = 1.45987804083
x = 7   y = 2.37615829782
x = 8   y = 2.40266493572
x = 9   y = 2.42556941674
x = 10  y = 2.44569781236
[student@localhost ~]$ █

```

Рисунок 11.3 – Результат выполнения программы

Упражнение 2

В этом упражнении вы составите блок-схему и напишете программу, вычисляющую значения функции z :

$$z = \frac{\lg(x + y)}{\sqrt{(x + y)^2 + 1}},$$

где $x \in [1; 10]$, $\Delta x = 1$, $y \in [0,1; 1]$, $\Delta y = 0,1$.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 11.4](#)).

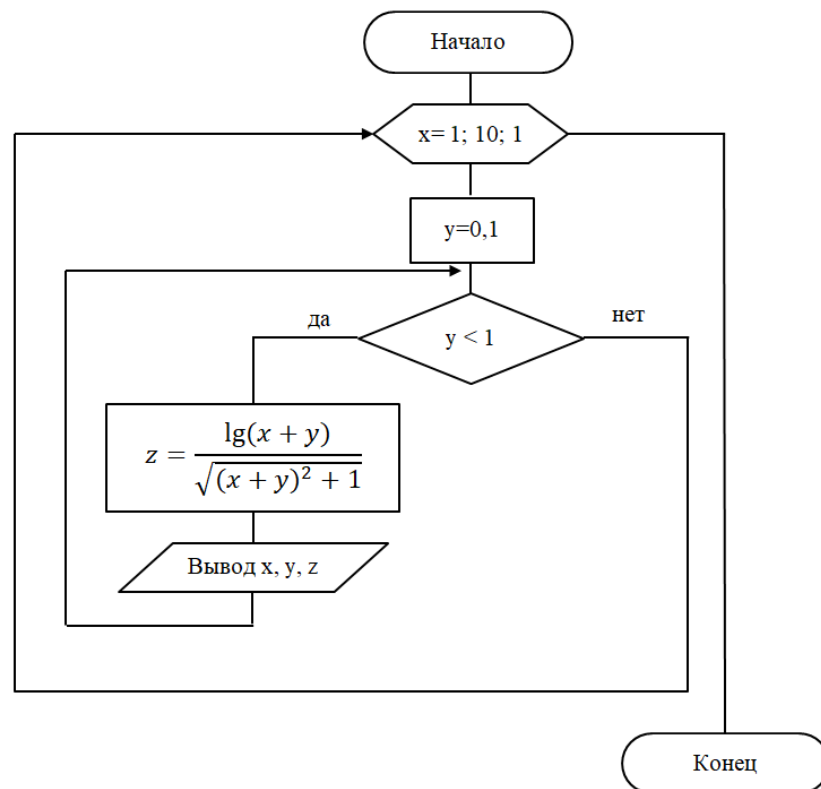
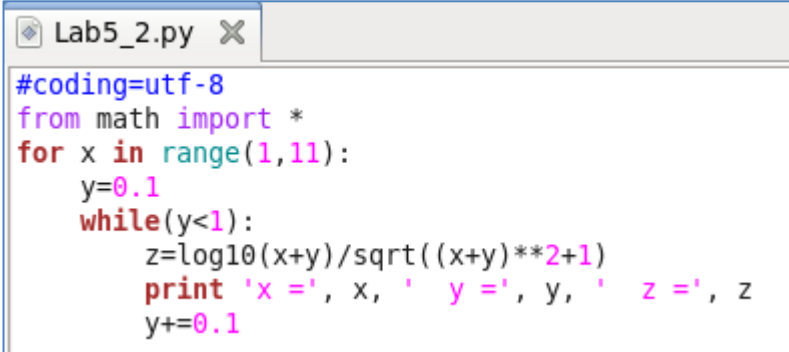


Рисунок 11.4 – Блок-схема алгоритма

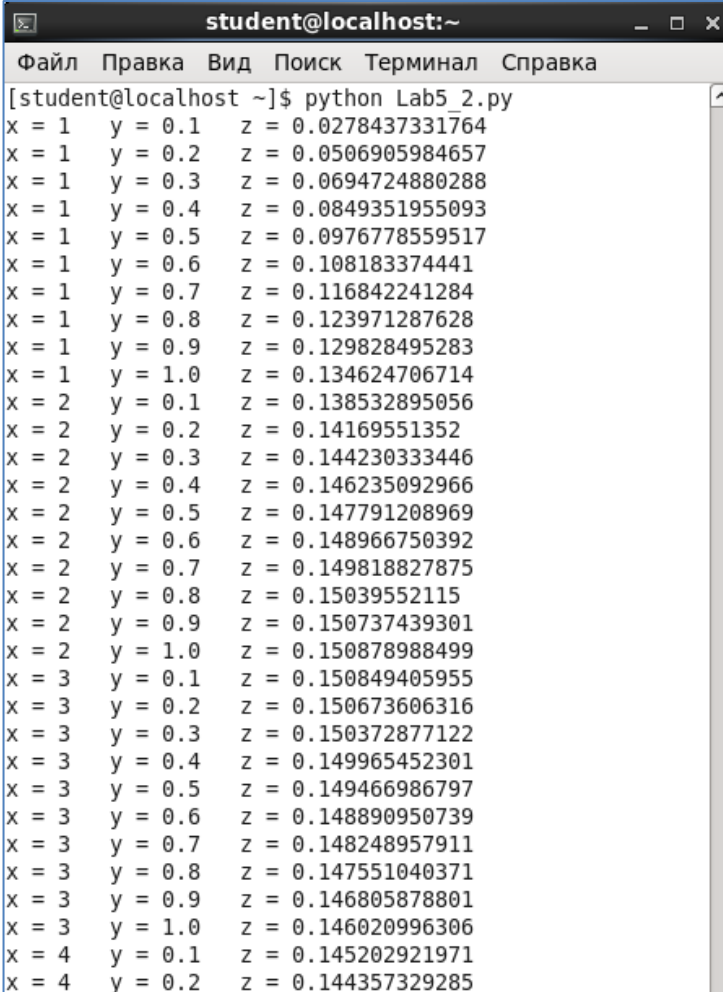
2. Создайте файл с именем **Lab11_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 11.5](#)).



```
#coding=utf-8
from math import *
for x in range(1,11):
    y=0.1
    while(y<1):
        z=log10(x+y)/sqrt((x+y)**2+1)
        print 'x =', x, ' y =', y, ' z =', z
        y+=0.1
```

Рисунок 11.5 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунком 11.6](#).



```
student@localhost:~$ python Lab5_2.py
x = 1  y = 0.1  z = 0.0278437331764
x = 1  y = 0.2  z = 0.0506905984657
x = 1  y = 0.3  z = 0.0694724880288
x = 1  y = 0.4  z = 0.0849351955093
x = 1  y = 0.5  z = 0.0976778559517
x = 1  y = 0.6  z = 0.108183374441
x = 1  y = 0.7  z = 0.116842241284
x = 1  y = 0.8  z = 0.123971287628
x = 1  y = 0.9  z = 0.129828495283
x = 1  y = 1.0  z = 0.134624706714
x = 2  y = 0.1  z = 0.138532895056
x = 2  y = 0.2  z = 0.14169551352
x = 2  y = 0.3  z = 0.144230333446
x = 2  y = 0.4  z = 0.146235092966
x = 2  y = 0.5  z = 0.147791208969
x = 2  y = 0.6  z = 0.148966750392
x = 2  y = 0.7  z = 0.149818827875
x = 2  y = 0.8  z = 0.15039552115
x = 2  y = 0.9  z = 0.150737439301
x = 2  y = 1.0  z = 0.150878988499
x = 3  y = 0.1  z = 0.150849405955
x = 3  y = 0.2  z = 0.150673606316
x = 3  y = 0.3  z = 0.150372877122
x = 3  y = 0.4  z = 0.149965452301
x = 3  y = 0.5  z = 0.149466986797
x = 3  y = 0.6  z = 0.148890950739
x = 3  y = 0.7  z = 0.148248957911
x = 3  y = 0.8  z = 0.147551040371
x = 3  y = 0.9  z = 0.146805878801
x = 3  y = 1.0  z = 0.146020996306
x = 4  y = 0.1  z = 0.145202921971
x = 4  y = 0.2  z = 0.144357329285
```

Рисунок 11.6 – Результат выполнения программы

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем.
Имя файла – **Lab11_5.py**.

1.1. Составьте схему алгоритма и напишите программу. Вычислить значения функции z :

$$z = \begin{cases} \frac{\ln^2(x+b)^2}{b+e^{ax}}, & \text{если } x < 0 \\ \arctg x + b \cdot \sin^2(ax), & \text{если } 0 \leq x \leq 2, \\ \lg x + |x - a|, & \text{если } x > 2 \end{cases}$$

где $a = 2$, $b = 8$, $x \in [-2; 4]$, $\Delta x = 0,5$.

1.2. Составьте схему алгоритма и напишите программу. Вычислить значения функции f :

$$f = \begin{cases} \sqrt{|t+c|} - \frac{1}{at+c}, & \text{если } t < 0 \\ \sin^3(c+t) + at \cdot \arctg(at), & \text{если } 0 \leq t \leq 3, \\ \frac{\sqrt{|t-c|}}{\ln^3(t+10)}, & \text{если } t > 3 \end{cases}$$

где $a = -3,7$, $c = 10$, $t \in [-4; 7]$, $\Delta t = 1$.

1.3. Составьте схему алгоритма и напишите программу. Вычислить значения функции z :

$$z = \begin{cases} \frac{1}{\sqrt{x+10}}, & \text{если } y < 0 \\ \lg|y-1| - e^{x+2,5}, & \text{если } 0 \leq y < 1, \\ y + \log_5(x+7,7), & \text{если } y \geq 1 \end{cases}$$

где $y = \sqrt{|x+3|}$, $x \in [-6; 3]$, $\Delta x = 0,5$.

1.4. Составьте схему алгоритма и напишите программу. Вычислить значения функции w :

$$w = \begin{cases} \sqrt{|t+8|}, & \text{если } t = 20 \\ e^{2k+1} - \sin^4 k, & \text{если } t > 20 \\ \cos 20^\circ, & \text{если } t < 20 \end{cases},$$

где $t = \frac{\lg k + \sqrt[3]{\lg(k+7)}}{7}$, $k \in [5; 9]$, $\Delta k = 0,5$.

1.5. Составьте схему алгоритма и напишите программу. Вычислить значение функции r :

$$r = \begin{cases} \frac{\sqrt[7]{x}}{\lg(|x+0,1|)-c}, & \text{если } 0,2 < x < 3 \\ \operatorname{tg}(x+c), & \text{в остальных случаях} \end{cases},$$

где $c = \sin^3(1-x) - e^{7x-1}$, $x \in [-3; 6]$, $\Delta x = 1$.

1.6. Составьте схему алгоритма и напишите программу. Вычислить значение функции f :

$$f = \begin{cases} \sqrt{|x + \operatorname{ctg}(b)|}, & \text{если } x < 1 \\ e^{x+1,2}, & \text{если } 1 \leq x \leq 4,7 \\ b + \ln(x+1)^2, & \text{в остальных случаях} \end{cases},$$

где $b = \cos x^2$, $x \in [0; 6]$, $\Delta x = 0,2$.

1.7. Составьте схему алгоритма и напишите программу. Вычислить значение функции Z :

$$Z = \begin{cases} k^3 + \sqrt{k}, & \text{при } k > 0 \\ e^{k+1} - \sin^2 k, & \text{при } k \leq 0 \end{cases}$$

где $k \in [-2; 3]$, $\Delta k = 0,5$.

1.8. Составьте схему алгоритма и напишите программу. Вычислить значение функции Y :

$$Y = \begin{cases} \frac{\sqrt[4]{|x+2|}}{c - \lg^3|x|}, & \text{если } -2 < x < 5 \\ \operatorname{arctg}(x-1), & \text{в остальных случаях} \end{cases},$$

где $c = \cos x + \sin^2(1+x) - e^x$, $x \in [-3,2; 6,4]$, $\Delta x = 0,6$.

1.9. Составьте схему алгоритма и напишите программу. Вычислить значение функции Z :

$$Z = \begin{cases} y, & \text{при } x = 0 \\ e^{x+1} - \sin^2 y, & \text{при } x > 0, \\ xy - \cos 20^\circ, & \text{при } x < 0 \end{cases}$$

где x — произвольное, $y \in [-2; 12]$, $\Delta y = 2$.

1.10. Составьте схему алгоритма и напишите программу. Вычислить значение функции F :

$$F = \begin{cases} |x - tgb|, & \text{при } x < 5 \\ e^{x+1,2}, & \text{при } 5 \leq x < 12, \\ b + \log_3(x + 1), & \text{при } x \geq 12 \end{cases}$$

где $x \in [2; 18]$, $\Delta x = 3$, b – произвольное.

1.11. Составьте схему алгоритма и напишите программу. Вычислить значение функции Y :

$$Y = \begin{cases} ax + \ln(x + b), & \text{если } x < b \\ tgx + \frac{ctg(a+x)}{b-1}, & \text{если } b \leq x \leq a, \\ \sqrt[3]{ax + b \cdot lgx}, & \text{если } x > a \end{cases}$$

где $a = 7,8$, $b = 1,12$, $x \in [0; 8]$, $\Delta x = 1,5$.

1.12. Составьте схему алгоритма и напишите программу. Вычислить значение функции m :

$$m = \begin{cases} \sqrt{|n + \cos(a)|}, & \text{если } n < 3 \\ e^{n-1,7}, & \text{если } 3 \leq n \leq 6, \\ a + \ln(n + 1)^2, & \text{если } n > 6 \end{cases}$$

где $a = tgn$, $n \in [2; 7]$, $\Delta n = 1$.

2. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab11_6.py**.

1.1. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$f = \sqrt[3]{1 + xy},$$

где $x \in [3; 5]$, $\Delta x = 0,25$, $y \in [-1; 1]$, $\Delta y = 0,4$.

1.2. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$f = \ln(ax) + 1,3e^{8y-x},$$

где $x \in [2; 8]$, $\Delta x = 0,3$, $y \in [1; 7]$, $\Delta y = 1$, $a = 3,1$.

1.3. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$z = \cos 2x + \operatorname{tg}^2(ay),$$

где $y \in [2; 3]$, $\Delta y = 0,2$, $a \in [1; 8]$, $\Delta a = 1,2$, $a = 1,2$.

1.4. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$y = \sqrt{x} + 1,1a^2b,$$

где $x \in [1,5; 2,5]$, $\Delta x = 0,1$, $a \in [2; 12]$, $\Delta a = 2$, b – произвольное.

1.5. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$y = 1 + e^{ax} - \sqrt{bx},$$

где $x \in [1; 2]$, $\Delta x = 0,1$, $b \in [2; 11]$, $\Delta b = 1$, $a = 5 + \frac{x}{10}$.

1.6. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$f = e^{x+1} - \sin^2 y,$$

где $x \in [4; 5,5]$, $\Delta x = 0,3$, $y \in [2; 7]$, $\Delta y = 1$.

1.7. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$f = \frac{\sqrt[4]{|x+b|}}{y - \lg^3 x},$$

где $x \in [2; 7]$, $\Delta x = 0,5$, $y \in [-3; 1]$, $\Delta y = 0,2$, b – произвольное.

1.8. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$z = \sqrt{|x + \operatorname{ctg}(b)|},$$

где $x \in [2; 3]$, $\Delta x = 0,2$, $b \in [1; 8]$, $\Delta b = 1,2$.

1.9. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$y = \sin^3(c+t) + at \cdot \operatorname{arctg}(at),$$

где $c \in [1; 2]$, $\Delta c = 0,1$, $t \in [2; 11]$, $\Delta t = 1$, $a = 5 - c$.

1.10. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$M = \sqrt[3]{|\sin^4(d^2 - c \cdot \operatorname{tg}(d + c))|}$$

где $d \in [0; 7]$, $\Delta d = 0,5$, $c \in [1; 6]$, $\Delta c = 1$.

1.11. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$L = \cos^2 \frac{a - f^3 + e^{(a-f)}}{a + f},$$

где $a \in [1,5; 2,5]$, $\Delta a = 0,1$, $f \in [6; 22]$, $\Delta f = 2$.

1.12. Составьте схему алгоритма и напишите программу. Вычислить значение функции

$$Y = \frac{z^2 + \cos^3(n + 14,5)}{|n - 3,9|}$$

где $n \in [1; 4]$, $\Delta n = 0,25$, $z \in [2; 11]$, $\Delta z = 2$.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схемы алгоритмов, листинги программ, исходные данные, результаты выполнения программ и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой процесс называется циклическим?
2. Какие вы знаете операторы цикла?
3. Назначение и синтаксис функции `range()`.
4. Функция `range()` записана в виде: `range(2,7)`. Перечислите параметры данной функции.

ЛАБОРАТОРНАЯ РАБОТА № 12**Тема: PYTHON. АЛГОРИТМЫ НАКОПЛЕНИЯ СУММЫ И ПРОИЗВЕДЕНИЯ****1 ЦЕЛЬ РАБОТЫ**

Научиться:

- реализовывать алгоритмы накопления суммы и произведения на языке Python.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ**

Типовыми алгоритмами для циклического процесса являются задачи с вычислением суммы нескольких слагаемых или произведения нескольких сомножителей. Вычисление суммы или произведения сводится к их накоплению в какой-либо ячейке памяти.

Например, необходимо найти сумму значений числа x или произведение значений числа x . Накопление суммы часто обозначают символом \sum , а накопление произведения символом \prod . Если параметры цикла известны, то они указываются, например

$$\sum_{x=1}^{10} x^2.$$

Накапливать сумму будем при помощи переменной S , а произведение – при помощи переменной P . Принцип накопления реализуется с использованием рекуррентных выражений:

$$\begin{aligned} S &= S + x; \\ P &= P \cdot x. \end{aligned}$$

Причем до начала процесса накопления (до цикла) необходимо ячейку для накопления суммы обнулить ($S = 0$), а в ячейку для накопления произведения положить единицу ($P = 1$).

Накопленные сумма и произведение выводятся за циклом.

Упражнение 1

В этом упражнении вы составите блок-схему и напишете программу, вычисляющую L по формуле

$$L = \sum_{a=3}^7 \frac{15}{(a+1)}.$$

Решение:

1. Составьте блок-схему алгоритма ([рисунок 12.1](#)).

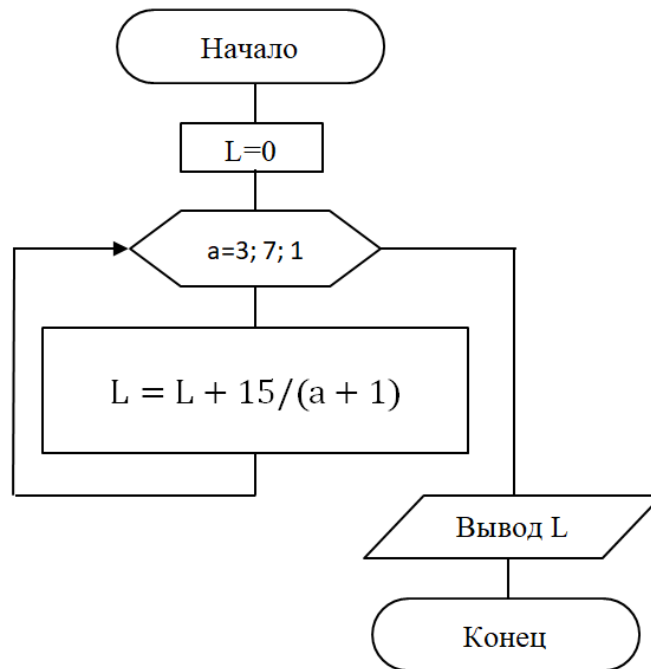


Рисунок 12.1 – Блок-схема алгоритма

2. Создайте файл с именем **Lab12_1.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 12.2](#)).

```

Lab6_1.py
#coding=utf-8
from math import *
L=0
for a in range(3,8):
    L+=15.0/(a+1)
print 'L =', L
  
```

Рисунок 12.2 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение. Результат выполнения программы сравните с [рисунком 12.3](#).

```

student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab6_1.py
L = 13.2678571429
[student@localhost ~]$

```

Рисунок 12.3 – Результат выполнения программы

Упражнение 2

В этом упражнении вы составите блок-схему и напишите программу, вычисляющую сумму ряда S с точностью ε по формуле:

$$S = \frac{1}{x} - \frac{2}{x^2} + \frac{3}{x^3} - \frac{4}{x^4} + \dots + (-1)^{i-1} \frac{i}{x^i} + \dots$$

Решение:

1. Составьте блок-схему алгоритма ([рисунок 12.4](#)).

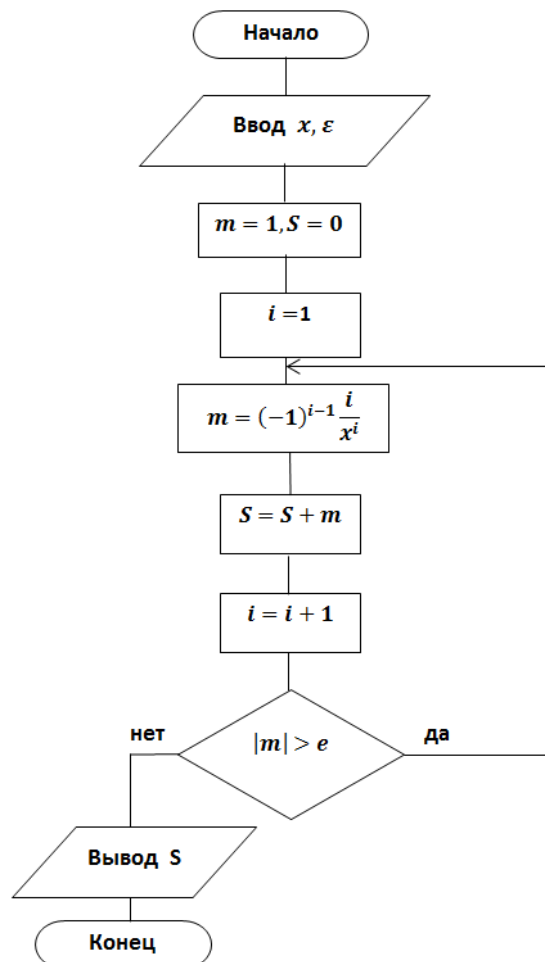
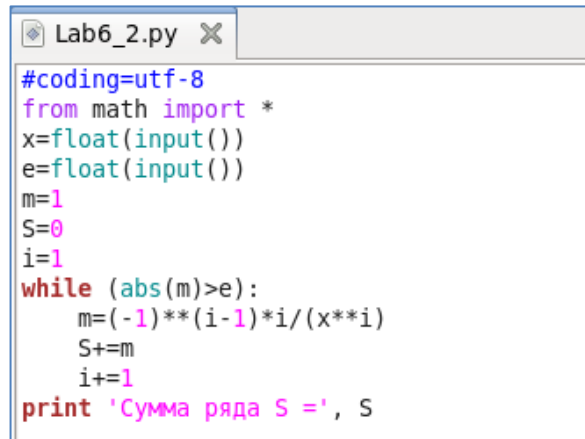


Рисунок 12.4 – Блок-схема алгоритма

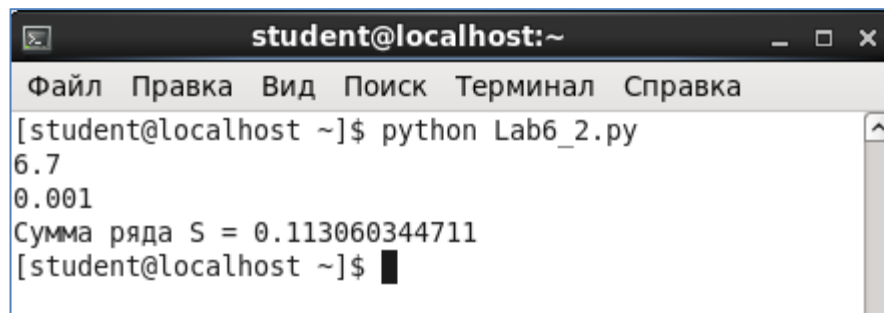
2. Создайте файл с именем **Lab12_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 12.5](#)).



```
#coding=utf-8
from math import *
x=float(input())
e=float(input())
m=1
S=0
i=1
while (abs(m)>e):
    m=(-1)**(i-1)*i/(x**i)
    S+=m
    i+=1
print 'Сумма ряда S =', S
```

Рисунок 12.5 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение. Выполните отладку программы для значений $\varepsilon = 0,01$ и $x = 8$.
6. Сравните результат выполнения программы с [рисунком 12.6](#).



```
student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab6_2.py
8.7
0.001
Сумма ряда S = 0.113060344711
[student@localhost ~]$
```

Рисунок 12.6 – Результат выполнения программы

Упражнение 3

В этом упражнении вы составите блок-схему алгоритма и напишите программу для вычисления z по формуле

$$z = \prod_{y=2}^6 (2ab + y), \quad \Delta y = 0,5, \quad a = -1,58, \quad b = 2.$$

Решение:

1. Составьте блок-схему алгоритма ([рисунком 12.7](#)).

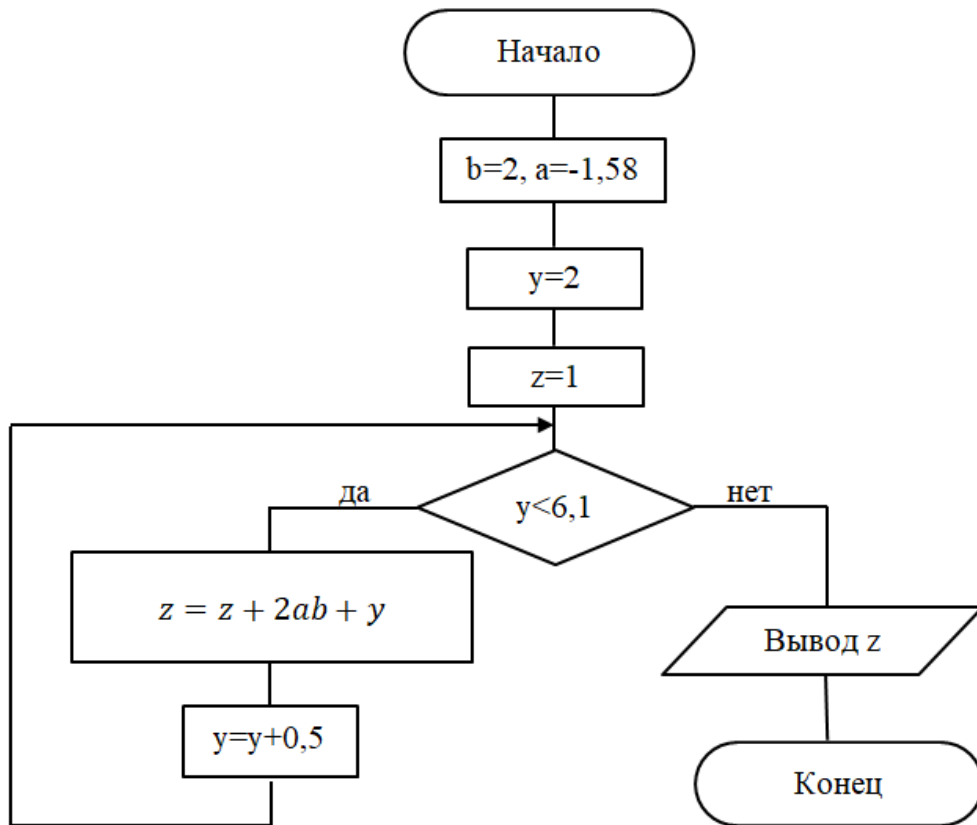


Рисунок 12.7 – Блок-схема алгоритма

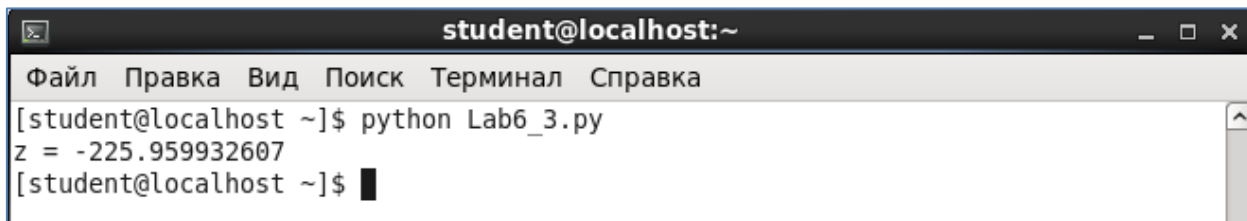
2. Создайте файл с именем **Lab6_3.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 12.8](#)).

```

Lab6_3.py
#coding=utf-8
from math import *
b=2
a=-1.58
y=2
z=1
while (y<6.1):
    z*=2*a*b+y
    y+=0.5
print 'z =', z
  
```

Рисунок 12.8 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение.
6. Сравните результат выполнения программы с [рисунок 12.9](#).



```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python Lab6_3.py
z = -225.959932607
[student@localhost ~]$

```

Рисунок 12.9 – Результат выполнения программы

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab12_4.py**.

1.1. Составьте схему алгоритма и напишите программу для вычисления f по формуле

$$f = \sum_{x=1}^5 2 \sin(3x - y), \quad \Delta x = 0,4, \quad y = 5,1.$$

1.2. Составьте схему алгоритма и напишите программу для вычисления p по формуле

$$p = \prod_{x=2}^6 \frac{3x}{\sqrt{|\sin x^3|}}, \quad \Delta x = 0,2.$$

1.3. Составьте схему алгоритма и напишите программу для вычисления k по формуле

$$k = \sum_{x=1}^2 bx^2, \quad \Delta x = 0,2, \quad b = 4,5.$$

1.4. Составьте схему алгоритма и напишите программу для вычисления h по формуле

$$h = \prod_{n=5}^{15} \frac{1}{(3n - 2)(3n + 1)}.$$

1.5. Составьте схему алгоритма и напишите программу для вычисления p по формуле

$$p = \sum_{y=-2}^2 (2x^2 + 3y) - a, \quad \Delta y = 0,1, \quad a = 3,7, \quad x = 8,3.$$

1.6. Составьте схему алгоритма и напишите программу для вычисления y по формуле

$$y = \prod_{x=1}^3 \frac{1}{2} \sin \frac{x}{4} + 10, \quad \Delta x = 0,1.$$

1.7. Составьте схему алгоритма и напишите программу для вычисления z по формуле

$$z = \sum_{x=2}^5 \ln(x^3 - ax^2) - y, \quad \Delta x = 0,2, \quad a = 0,2, \quad y = 4,3.$$

1.8. Составьте схему алгоритма и напишите программу для вычисления f по формуле

$$f = \prod_{n=3}^9 \frac{n^{\ln n}}{(\ln n)^n}.$$

1.9. Составьте схему алгоритма и напишите программу для вычисления M по формуле

$$M = \sum_{x=-\frac{\pi}{2}}^{\frac{\pi}{2}} tg(-ax)^2, \quad \Delta x = \frac{\pi}{6}, \quad a = \ln|3b - 2|, \quad b = 2,1.$$

1.10. Составьте схему алгоритма и напишите программу для вычисления y по формуле

$$y = k - \prod_{x=0}^1 \frac{\sin x + tg^3 x}{\sqrt[3]{x+3}}, \quad \Delta x = 0,1, \quad k = -4,58.$$

1.11. Составьте схему алгоритма и напишите программу для вычисления z по формуле

$$z = \sum_{x=2}^8 3x + \sqrt{|ax - e^{3x}|}, \quad \Delta x = 1,2, \quad a = \ln(x - 1).$$

1.12. Составьте схему алгоритма и напишите программу для вычисления g по формуле

$$g = \prod_{n=1}^8 \frac{2n - 1}{2^n}, \quad \Delta n = 0,5.$$

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схему алгоритма, листинг программы, исходные данные, результат выполнения программы.

ЛАБОРАТОРНАЯ РАБОТА № 13

Тема: **PYTHON. ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ**

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с основными характеристиками массивов, с типами массивов;
- с основными способами обработки массивов на языке Python.

Научиться:

- реализовывать алгоритмы обработки массивов на языке программирования Python.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Массивы: основные понятия и определения

Массив – это упорядоченное множество однотипных переменных (элементов массива), объединенных общим именем и отличающихся номерами (*индексами*).

Массивы применяются там, где возникает необходимость обработки (поиска, перестановки, сортировки) большого количества данных, как числовых, так и символьных.

Идея дать многим переменным одно имя и разрешить использовать в качестве индекса переменную позволяет создавать очень компактные программы по обработке больших наборов данных.

Для характеристики массивов используют следующие величины:

- *Размерность* – это число индексов. Чаще всего используются одномерные (один индекс) и двумерные (два индекса) массивы.
- *Размер (глубина)* – это количество элементов в массиве.

Работа с массивами сводится к работе с его элементами. *Чтобы обратиться к элементу массива*, необходимо назвать имя массива и индекс (порядковый номер) элемента массива.

В одномерном массиве положение элемента (его порядковый номер) определяется одним индексом, который обозначается буквой *i*. Аналогом одномерного массива является линейная таблица (вектор).

Нумерация элементов массива начинается с нуля. Например, пусть X – одномерный массив, содержащий 5 вещественных чисел:

i	0	1	2	3	4
x_i	10,1	5	4,8	8,2	10

Тогда, $X_0 = 10,1$; $X_1 = 5$; $X_2 = 4,8$; $X_3 = 8,2$; $X_4 = 10$.

3.2 Объявление и инициализация одномерных массивов

Одномерный массив в Python представляет собой список элементов.

Для работы с массивами в Python необходимо импортировать модуль `array`, который определяет массивы. Для этого необходимо написать команду

```
from array import *
```

Для инициализации (определения) массива в программе необходимо указать его имя, тип и значения элементов массива, например,

```
x = array('i', [2, 4, 6, 8])
```

Данная команда создает новый массив x и инициализирует его значения. Значения элементов массива перечисляются через запятую в квадратных скобках `[]`. Любой элемент массива вызывается по индексу (номеру). Номера элементов массива начинаются с нуля, т.е. $x[0]=2$, $x[1]=4$, $x[2]=6$, $x[3]=8$.

В объявлении массива `"i"` – это `typecode`, который сообщает Python, что массив будет хранить целые числа. Этот параметр в Python является необязательным.

Функция `len()` возвращает количество элементов массива.

Для перечисления элементов массива используется цикл `for`. С помощью оператора `for` можно перебирать элементы массива или их индексы:

```
for x in [2, 4, 6, 8]
```

или

```
for i in range(0,5).
```

3.3 Обработка одномерных массивов

Рассмотрим на примерах инициализацию и обработку одномерных массивов на языке Python.

Упражнение 1

В этом упражнении вы составите блок-схему и напишите программу для следующей задачи.

Даны массивы X и Y, состоящие из 10 вещественных чисел каждый. Образовать массив Z, каждый элемент которого вычисляется по правилу:

$$Z_i = \sqrt[3]{X_i^2 + Y_i^2}.$$

Полученный массив вывести на экран.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 13.1](#)).

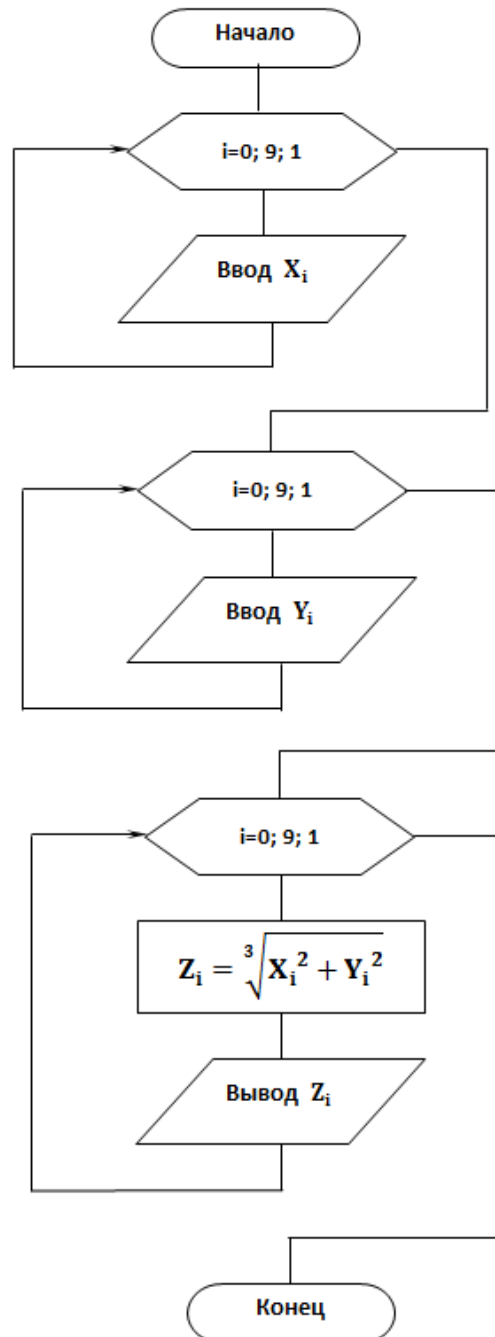
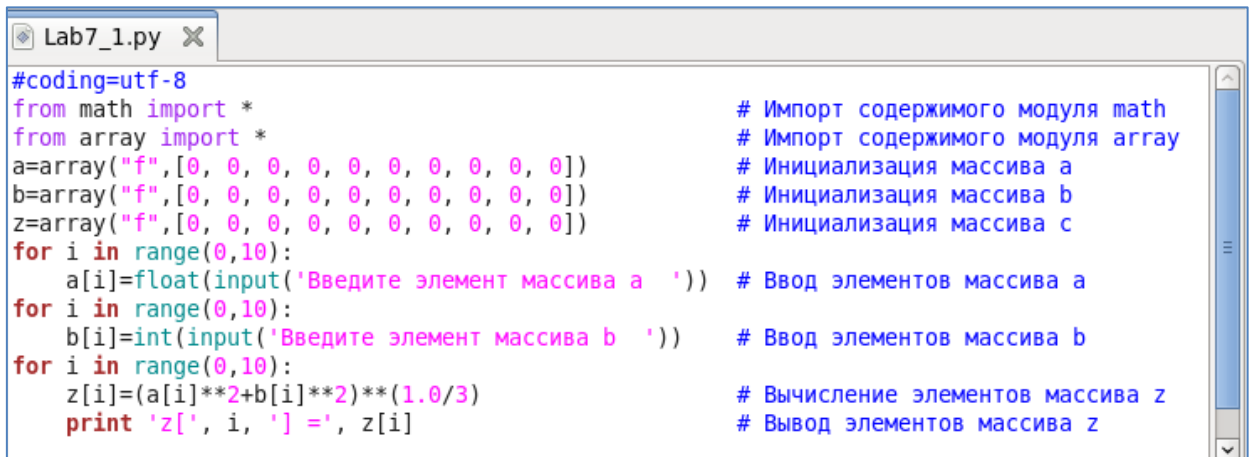


Рисунок 13.1 – Блок-схема алгоритма

2. Создайте файл с именем **Lab13_1.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 13.2](#)).



```
#coding=utf-8
from math import *
from array import *
a=array("f",[0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
b=array("f",[0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
z=array("f",[0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
for i in range(0,10):
    a[i]=float(input('Введите элемент массива a '))
for i in range(0,10):
    b[i]=int(input('Введите элемент массива b '))
for i in range(0,10):
    z[i]=(a[i]**2+b[i]**2)**(1.0/3)
print 'z[', i, ']' =', z[i]
```

Рисунок 13.2 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение.
6. Сравните результат выполнения программы с [рисунком 13.3](#).

Рассмотрим код программы подробнее. В первой и второй строках программы импортируется содержимое двух модулей `math` и `array` соответственно.

Третья, четвертая и пятая строка – инициализация массивов `a`, `b`, `z` с указанием типа элементов и их количества.

Шестая и седьмая строки – оператор `for` для ввода элементов массива `a`.

Восьмая и девятая строки – оператор `for` для ввода элементов массива `b`.

Десятая, одиннадцатая и двенадцатая строки кода – оператор `for` для вычисления и вывода элементов массива `z`.


```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python Lab7_1.py
Введите элемент массива a  5.4
Введите элемент массива a  3.1
Введите элемент массива a  0.45
Введите элемент массива a  -0.96
Введите элемент массива a  -12
Введите элемент массива a  2
Введите элемент массива a  23
Введите элемент массива a  45
Введите элемент массива a  1
Введите элемент массива a  1
Введите элемент массива b  0.87
Введите элемент массива b  65
Введите элемент массива b  -43
Введите элемент массива b  2.34
Введите элемент массива b  1
Введите элемент массива b  2
Введите элемент массива b  5
Введите элемент массива b  7
Введите элемент массива b  9.01
Введите элемент массива b  56
z[ 0 ] = 3.07795667648
z[ 1 ] = 16.178483963
z[ 2 ] = 12.2742462158
z[ 3 ] = 1.70099139214
z[ 4 ] = 5.25358772278
z[ 5 ] = 2.0
z[ 6 ] = 8.21302700043
z[ 7 ] = 12.7527227402
z[ 8 ] = 4.3444814682
z[ 9 ] = 14.6387786865
[student@localhost ~]$

```

Рисунок 13.3 – Результат выполнения программы

Упражнение 2

В этом упражнении вы составите блок-схему и напишите программу для следующей задачи.

В одномерном массиве D, состоящем из 12 целых чисел, элементы с четными номерами, значения которых находятся в диапазоне от 3 до 6, заменить на ноль. Полученный массив вывести на экран.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 13.4](#)).
2. Создайте файл с именем **Lab13_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 13.5](#)).
4. Сохраните файл.
5. Запустите программу на выполнение.
6. Сравните результат выполнения программы с [рисунком 13.6](#).

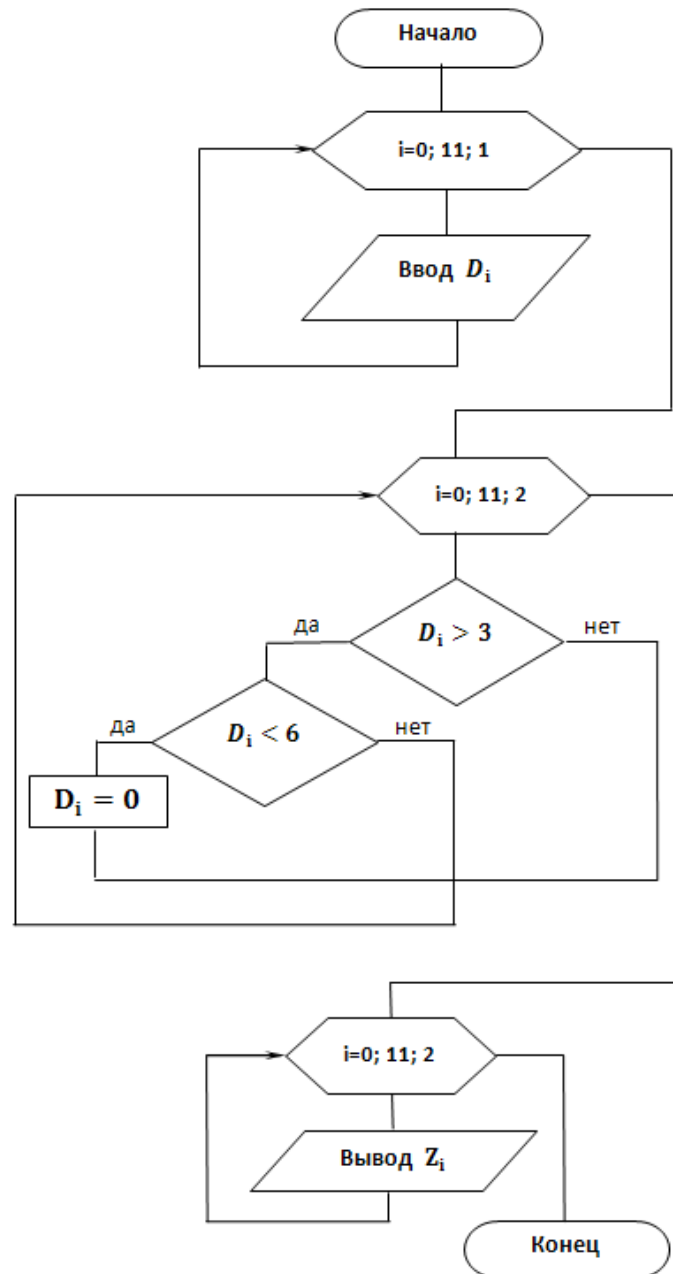


Рисунок 13.4 – Блок-схема алгоритма

```

Lab7_2.py (~) - gedit
Файл  Правка  Вид  Поиск  Сервис  Документы  Справка
Открыть  Сохранить  Отменить
Lab7_2.py
#coding=utf-8
from math import *
from array import *
d=array("i",[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
for i in range(0,12):
    d[i]=int(input('Введите элемент целочисленного массива d= '))
for i in range(0,12,2):
    if d[i]>3 and d[i]<6:
        d[i]=0
for i in range(0,12):
    print 'd[', i, ']' '=', d[i]
Python  Ширина табуляции: 8  Стр 6, Стлб 20  ВСТ

```

Рисунок 13.5 – Текст программы

```

student@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[student@localhost ~]$ python Lab7_2.py
Введите элемент целочисленного массива d= 8
Введите элемент целочисленного массива d= 5
Введите элемент целочисленного массива d= 6
Введите элемент целочисленного массива d= 4
Введите элемент целочисленного массива d= 3
Введите элемент целочисленного массива d= -9
Введите элемент целочисленного массива d= 10
Введите элемент целочисленного массива d= 3
Введите элемент целочисленного массива d= 4
Введите элемент целочисленного массива d= 5
Введите элемент целочисленного массива d= 6
Введите элемент целочисленного массива d= 4
d[ 0 ] = 8
d[ 1 ] = 5
d[ 2 ] = 6
d[ 3 ] = 4
d[ 4 ] = 3
d[ 5 ] = -9
d[ 6 ] = 10
d[ 7 ] = 3
d[ 8 ] = 0
d[ 9 ] = 5
d[ 10 ] = 6
d[ 11 ] = 4
[student@localhost ~]$

```

Рисунок 13.6 – Результат выполнения программы

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем.
Имя файла – **Lab13_3.py**.

1.1. Составьте схему алгоритма и напишите программу, в результате выполнения которой были бы напечатаны номера всех нулевых элементов массива C(11).

1.2. Составьте схему алгоритма и напишите программу, в результате выполнения которой были бы напечатаны номера всех положительных элементов массива M(9).

1.3. Составьте схему алгоритма и напишите программу. В одномерном массиве F(10) отрицательные элементы заменить нулями, а все положительные увеличить на 2.

1.4. Составьте схему алгоритма и напишите программу. В одномерном массиве S(11) замените на единицы все элементы, значения которых больше 5.

1.5. Составьте схему алгоритма и напишите программу. В одномерном массиве K(10) элементы с нечетными номерами необходимо уменьшить в четыре раза. Полученный массив вывести на экран.

1.6. Составьте схему алгоритма и напишите программу. В одномерном массиве S(11) элементы с четными номерами разделите на их порядковый номер. Полученный массив вывести на экран.

1.7. Составьте схему алгоритма и напишите программу. В одномерном массиве A(9) элементы с четными номерами необходимо уменьшить в два раза, а элементы с нечетными номерами увеличить в три раза. Полученный массив вывести на экран.

1.8. Составьте схему алгоритма и напишите программу. Дан массив R, состоящий из 8 вещественных чисел. Упорядочить по возрастанию элементы массива R с нечетными индексами.

1.9. Составьте схему алгоритма и напишите программу замены всех нулевых элементов массива F(12) единицами.

1.10. Составьте схему алгоритма и напишите программу замены всех отрицательных элементов массива K(12) их квадратами.

1.11. Составьте схему алгоритма и напишите программу. В одномерном массиве M(13) отрицательные элементы с нечетными номерами необходимо возвести в квадрат. Полученный массив вывести на экран.

1.12. Составьте схему алгоритма и напишите программу для вычисления значений функции Z:

$$Z = \begin{cases} \operatorname{ctg}^2\left(x_i - \frac{a}{b}\right), & \text{если } x_i < a \\ 1 - (x_i + a^3)^2, & \text{если } a \leq x_i < b, \\ x_i^2 + \frac{1}{ab}, & \text{если } x_i \geq b \end{cases}$$

где $a = 4$, $b = 6,4$, $x(10)$ – массив целых чисел.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схему алгоритма, листинг программы, исходные данные, результат выполнения программы и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется массивом?
2. Что такое размерность массива?
3. Что называется размером массива?
4. Как обратиться к элементу массива?
5. С какого значения начинается нумерация элементов массива в Python?
6. Какой модуль необходимо использовать для работы с массивами?
7. Какой командой импортируется содержимое модуля для работы с массивами?
8. Какая функция в Python возвращает количество элементов массива?

ЛАБОРАТОРНАЯ РАБОТА № 14

Тема: PYTHON. ОДНОМЕРНЫЕ МАССИВЫ: НАКОПЛЕНИЕ СУММЫ И ПРОИЗВЕДЕНИЯ, ПОИСК МАКСИМАЛЬНОГО (МИНИМАЛЬНОГО) ЭЛЕМЕНТА МАССИВА

1 ЦЕЛЬ РАБОТЫ

Познакомиться:

- с основными способами обработки массивов на языке Python.

Научиться:

- реализовывать алгоритмы обработки массивов на языке программирования Python.

2 ОБЪЁМ ЛАБОРАТОРНОЙ РАБОТЫ И ПОРЯДОК ЕЁ ВЫПОЛНЕНИЯ

Данная лабораторная работа выполняется в течение двух часов. Выполнение работы идет одновременно с изучением теоретической части. Для закрепления материала необходимо выполнить упражнения для самостоятельной работы и оформить отчет.

**3 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.
ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ****3.3 Обработка одномерных массивов**

Вычисление суммы и произведения элементов одномерного массива производится по тем же принципам, что и накопление суммы и произведения. В одномерных массивах вычисление происходит в цикле. Начальное значение суммы и количества равно 0, а произведения – 1. Количество элементов в массиве накапливается подобно сумме, только вместо значения элемента массива в рекуррентной формуле прибавляется 1. Накопленные величины выводятся за циклом. Рассмотрим несколько примеров.

Упражнение 1

В этом упражнении вы составите блок-схему и напишите программу для следующей задачи.

Дан массив T , состоящий из 8 элементов. Найти сумму положительных элементов массива $T(8)$.

Решение:

1. Составьте блок-схему алгоритма ([рисунок 14.1](#)).

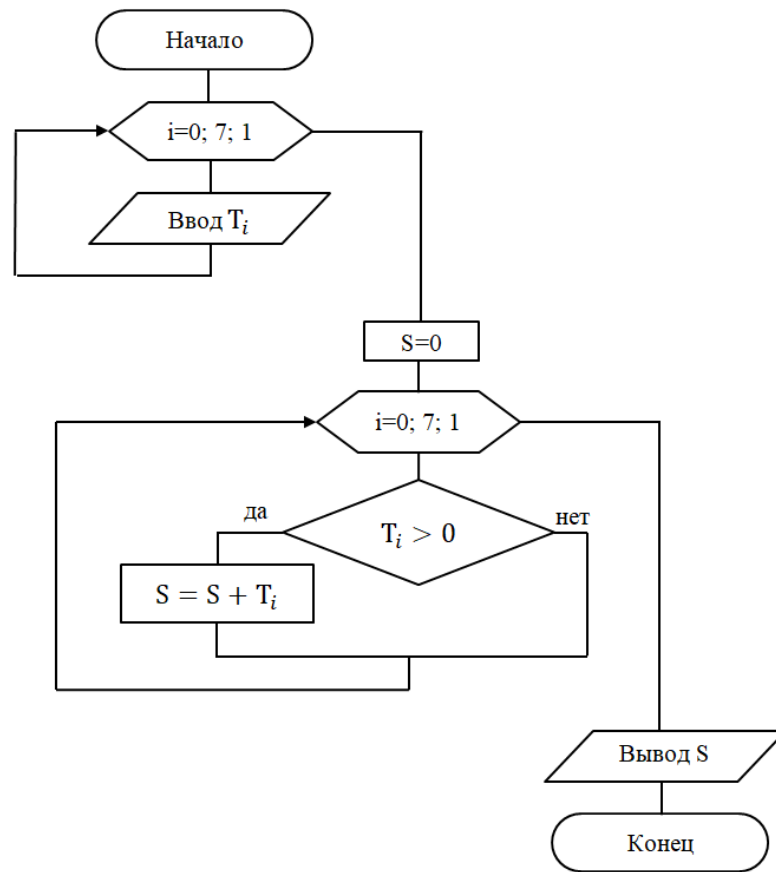


Рисунок 14.1 – Блок-схема алгоритма

2. Создайте файл с именем **Lab14_1.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 14.2](#)).
4. Сохраните файл.
5. Запустите программу на выполнение.
6. Сравните результат выполнения программы с [рисунок 14.3](#).

```

Lab8_1.py X
#coding=utf-8
from math import *
from array import *
T=array("f",[0, 0, 0, 0, 0, 0, 0, 0])
for i in range(0,8):
    T[i]=float(input('Введите элемент массива T = '))
S=0
for i in range(0,8):
    if T[i]>0:
        S+=T[i]
print "Сумма S положительных элементов массива T равна ", S
  
```

Рисунок 14.2 – Текст программы

```

student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab8_1.py
Введите элемент массива T = -6.7
Введите элемент массива T = 4.5
Введите элемент массива T = 3.21
Введите элемент массива T = 6.78
Введите элемент массива T = -0.987
Введите элемент массива T = -0.45
Введите элемент массива T = -4.32
Введите элемент массива T = 4.5
Сумма S положительных элементов массива T равна 18.990000248
[student@localhost ~]$

```

Рисунок 14.3 – Результат выполнения программы

Упражнение 2

В этом упражнении вы составите блок-схему и разработаете консольное приложение для следующей задачи.

Дан массив K , состоящий из 12 вещественных чисел. Найдите значение и номер максимального элемента массива K .

Решение:

1. Составьте блок-схему алгоритма ([рисунок 14.4](#)).

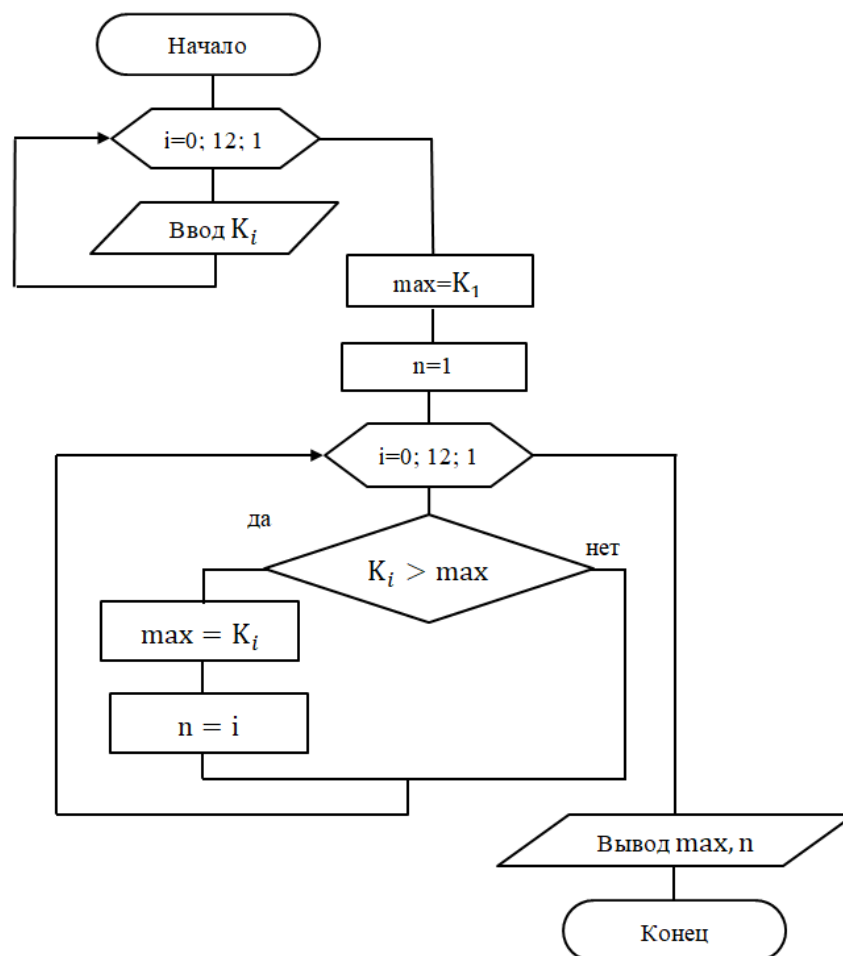
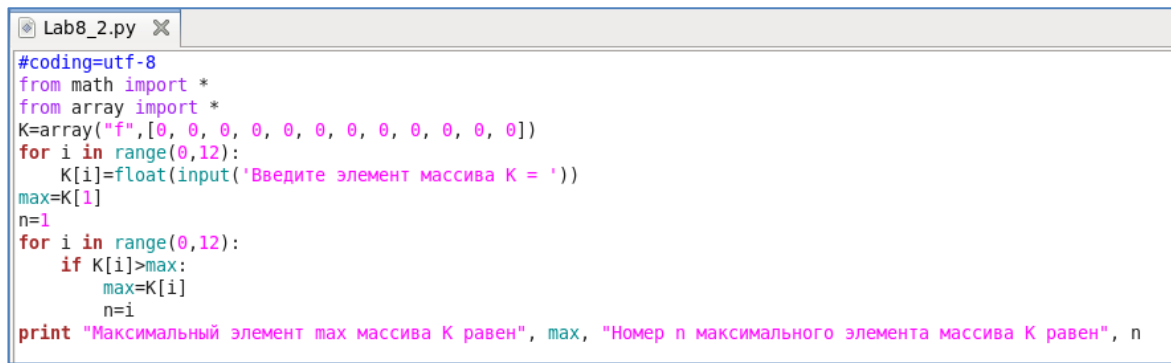


Рисунок 14.4 – Блок-схема алгоритма

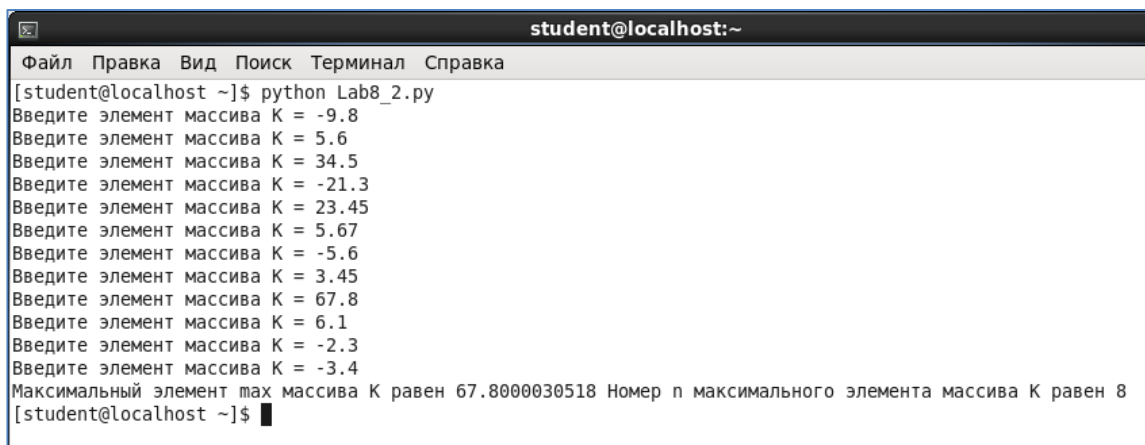
2. Создайте файл с именем **Lab14_2.py**.
3. Откройте созданный файл и наберите текст программы ([рисунок 14.5](#)).



```
#coding=utf-8
from math import *
from array import *
K=array("f",[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
for i in range(0,12):
    K[i]=float(input('Введите элемент массива K = '))
max=K[1]
n=1
for i in range(0,12):
    if K[i]>max:
        max=K[i]
        n=i
print "Максимальный элемент max массива K равен", max, "Номер n максимального элемента массива K равен", n
```

Рисунок 14.5 – Текст программы

4. Сохраните файл.
5. Запустите программу на выполнение.
6. Сравните результат выполнения программы с [рисунок 14.6](#).



```
student@localhost:~
Файл Правка Вид Поиск Терминал Справка
[student@localhost ~]$ python Lab8_2.py
Введите элемент массива K = -9.8
Введите элемент массива K = 5.6
Введите элемент массива K = 34.5
Введите элемент массива K = -21.3
Введите элемент массива K = 23.45
Введите элемент массива K = 5.67
Введите элемент массива K = -5.6
Введите элемент массива K = 3.45
Введите элемент массива K = 67.8
Введите элемент массива K = 6.1
Введите элемент массива K = -2.3
Введите элемент массива K = -3.4
Максимальный элемент max массива K равен 67.8000030518 Номер n максимального элемента массива K равен 8
[student@localhost ~]$
```

Рисунок 14.6 – Результат выполнения программы

4 УПРАЖНЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Выполните задание, согласно варианту, указанному преподавателем. Имя файла – **Lab14_3.py**.

- 1.1. Составьте схему алгоритма и напишите программу для нахождения среднего арифметического элементов массива X, состоящего из 16 вещественных чисел.
- 1.2. Составьте схему алгоритма и напишите программу. Дан массив A, состоящий из 10 вещественных чисел. Найдите количество отрицательных элементов массива A.
- 1.3. Составьте схему алгоритма и напишите программу. Дан массив C, состоящий из 15 вещественных чисел. Найдите количество нулевых элементов массива C.

- 1.4. Составьте схему алгоритма и напишите программу. Дан массив В, состоящий из 12 вещественных чисел. Найдите сумму положительных элементов массива В.
- 1.5. Составьте схему алгоритма и напишите программу. Дан массив D, состоящий из 20 вещественных чисел. Найдите произведение элементов массива D, значение которых меньше заданного числа n.
- 1.6. Составьте схему алгоритма и напишите программу. Дан одномерный массив вещественных чисел, состоящий из 20 элементов. Вывести на экран произведение положительных элементов массива.
- 1.7. Составьте схему алгоритма и напишите программу для нахождения максимального элемента массива V(14) и его индекса.
- 1.8. Составьте схему алгоритма и напишите программу для нахождения минимального элемента массива Z(10) и его индекса.
- 1.9. Составьте схему алгоритма и напишите программу для подсчета числа элементов, равных максимальному в числовом массиве из 9 элементов.
- 1.10. Составьте схему алгоритма и напишите программу для вычисления разности между максимальным и минимальным значениями элементов заданного целочисленного массива, состоящего из 11 вещественных чисел.
- 1.11. Составьте схему алгоритма и напишите программу для нахождения в массиве С, состоящего из 7 вещественных чисел, минимального из элементов, имеющих четный индекс.
- 1.12. Составьте схему алгоритма и напишите программу для нахождения наименьшего положительного элемента целочисленного массива A(29).

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать блок-схему алгоритма, листинг программы, исходные данные, результат выполнения программы и ответы на контрольные вопросы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется массивом?
2. Что такое размерность массива?
3. Что называется размером массива?
4. Как обратиться к элементу массива?
5. С какого значения начинается нумерация элементов массива в Python?
6. Какой модуль необходимо использовать для работы с массивами?
7. Какой командой импортируется содержимое модуля для работы с массивами?
8. Какая функция в Python возвращает количество элементов массива?

ЛИТЕРАТУРА

1. Технология программирования: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, М.П. Беляев, Ю.В. Минин. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 172 с.
2. Руденко, Н.Б., Информатика: Понятие информации. Общая характеристика процессов сбора, передачи, обработки и накопления информации. Позиционные системы счисления. Количество информации. Логические основы работы компьютера: методические указания / Н.Б. Руденко, Н.Н. Грачева, В.Н. Литвинов. – Зерноград: РИО ФГБОУ ВПО ДГАУ АЧИИ, 2015. – 58 с.
3. Грачева, Н.Н. Информатика. Алгоритмизация и программирование. Языки программирования высокого уровня: сборник задач / Н.Н. Грачева, Н.Б. Руденко. – Зерноград: РИО ФГБОУ ВПО АЧГАА, 2013. – 74 с.
4. Руденко, Н.Б. Системы счисления. Количество информации. Логические основы работы компьютера: сборник задач по информатике. / Н.Б. Руденко, Н.Н. Грачева. – Зерноград: РИО ФГБОУ ВПО АЧГАА, 2012. – Часть 1. – 63 с.

Установка и настройка виртуальной машины VirtualBox

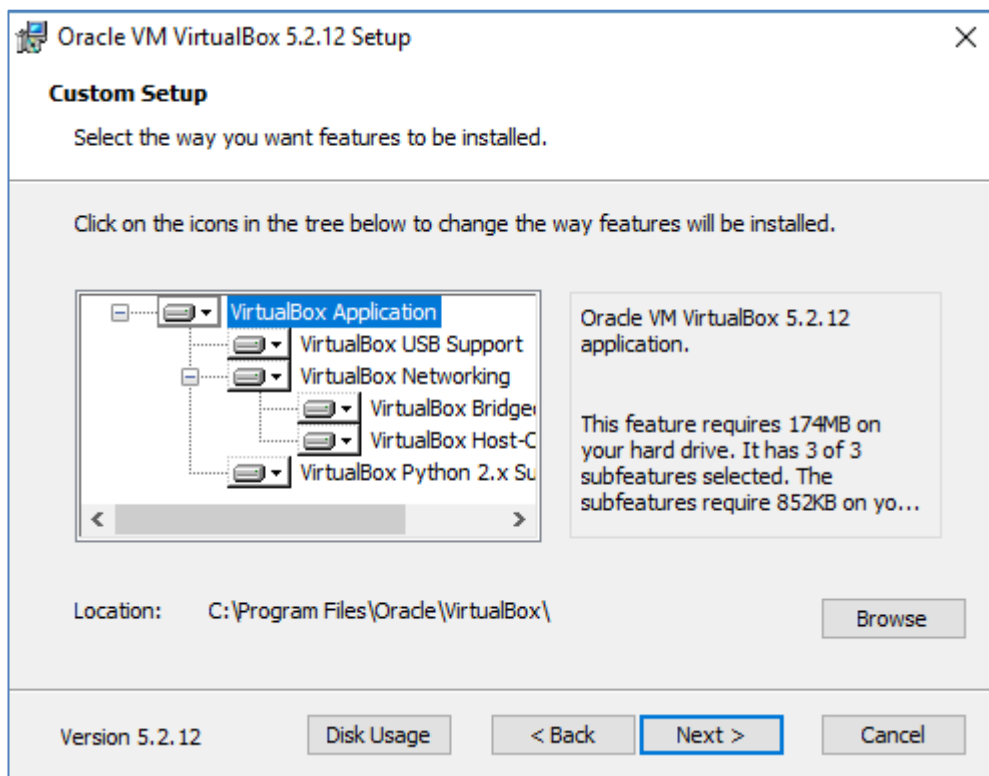
1. Скачайте установочный файл по адресу

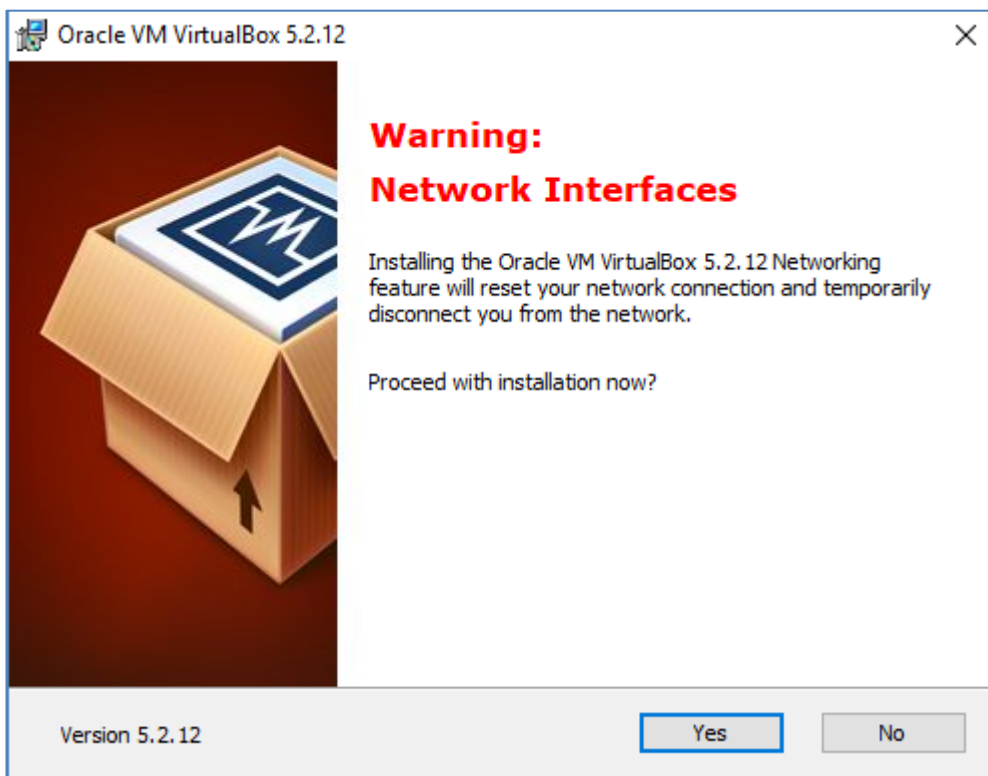
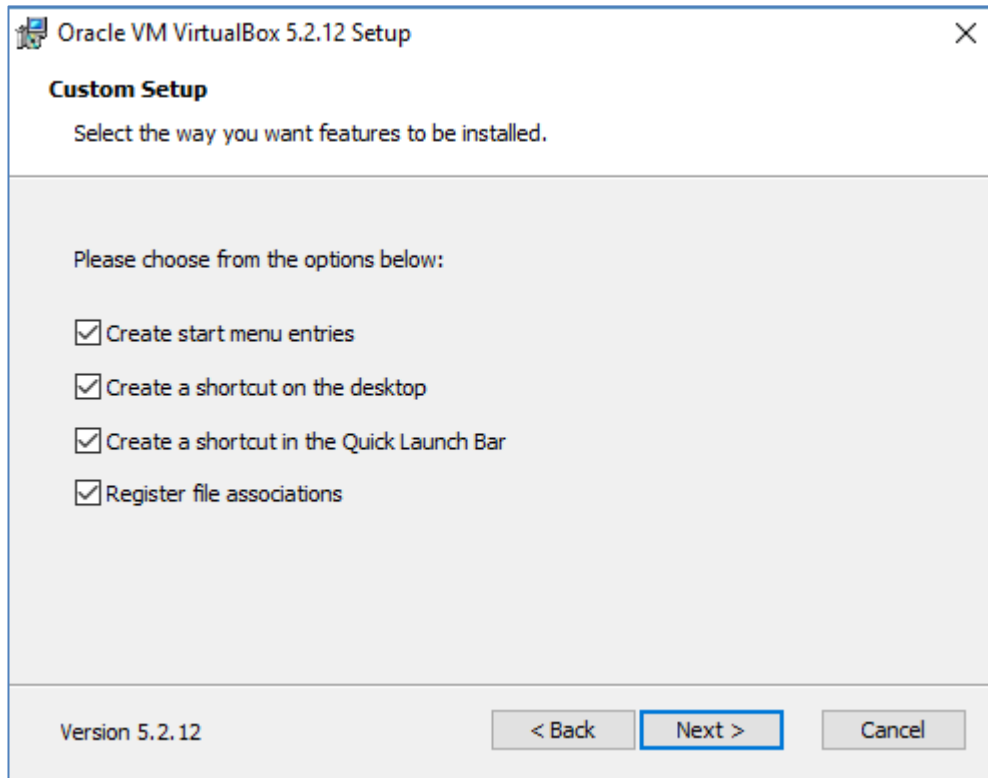
<https://www.virtualbox.org/wiki/Downloads>

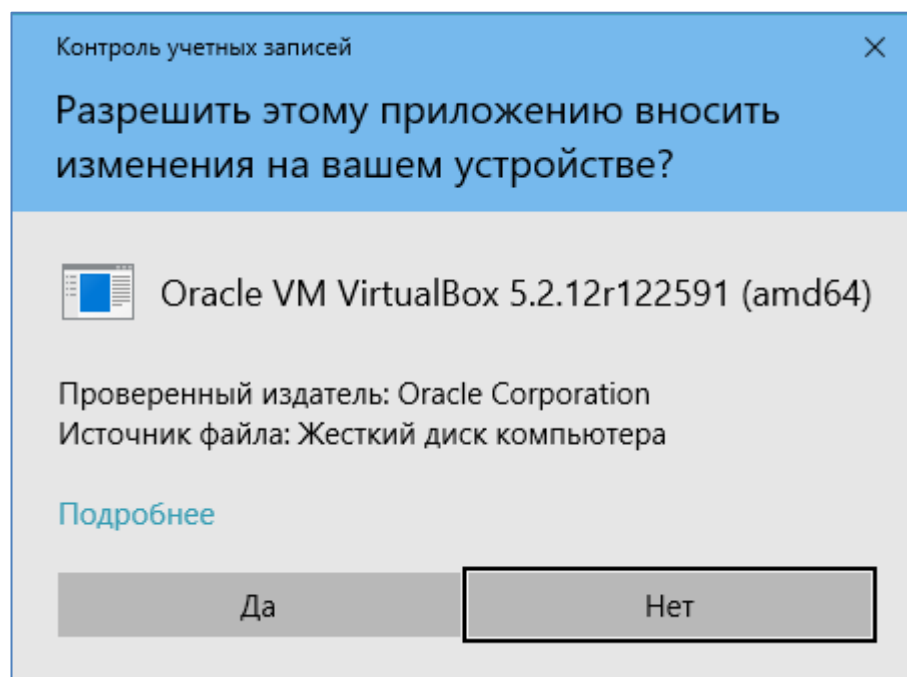
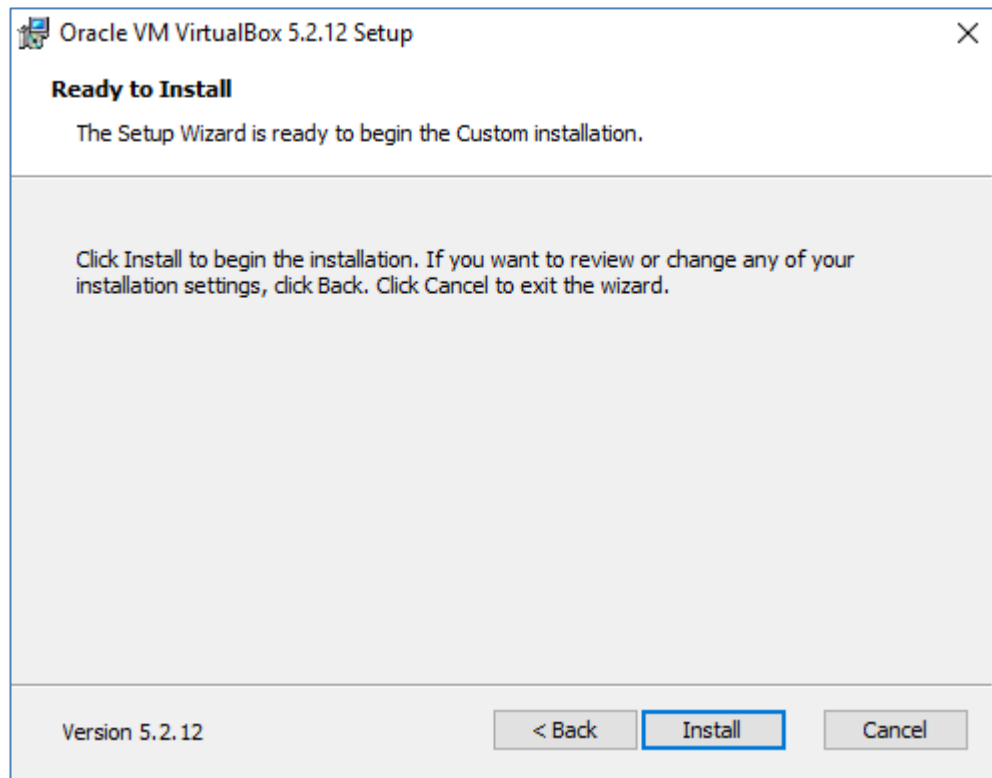


2. Щелкаем по ссылке **Windows hosts** и в появившемся диалоговом окне нажимаем **Сохранить**.

3. Находим сохраненный файл и запускаем на установку. Выполняем настройки как показано на рисунках ниже.



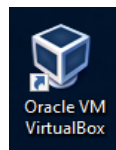




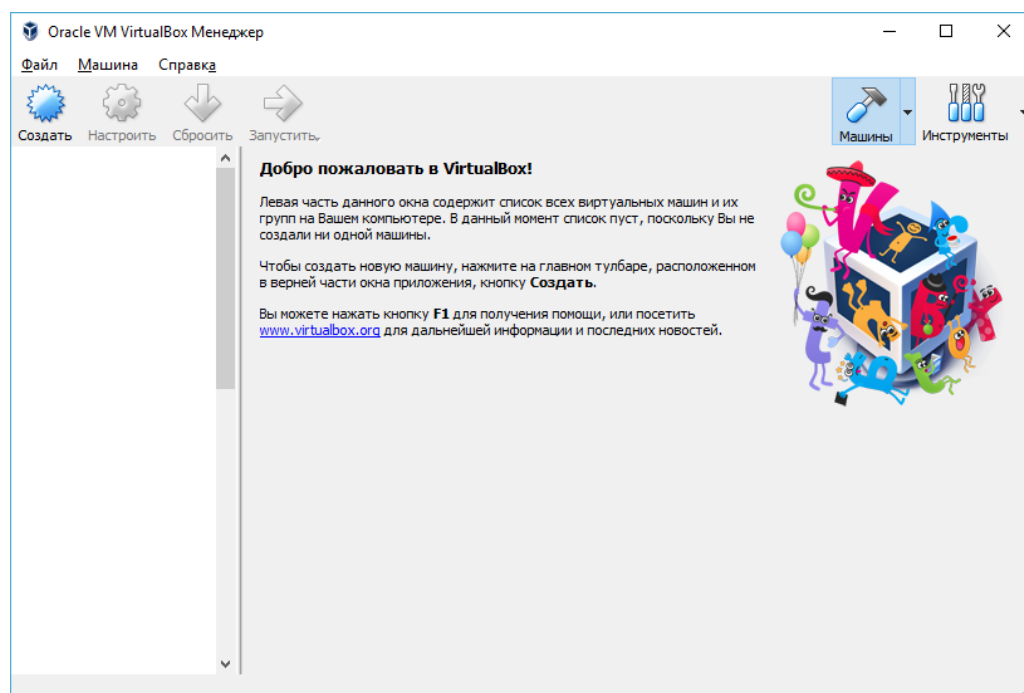
Разрешите внести изменения, нажмите **Да**.

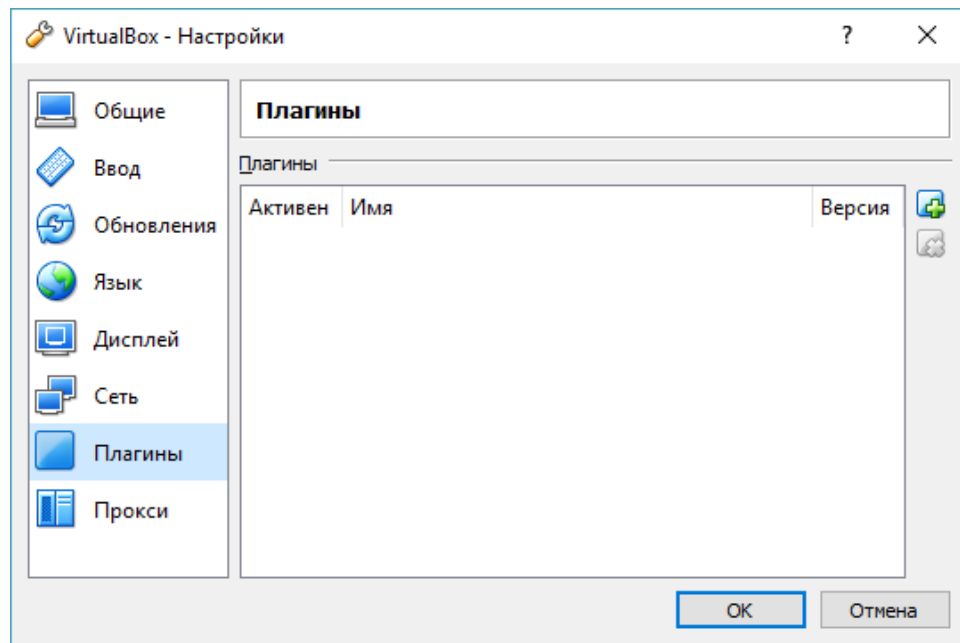
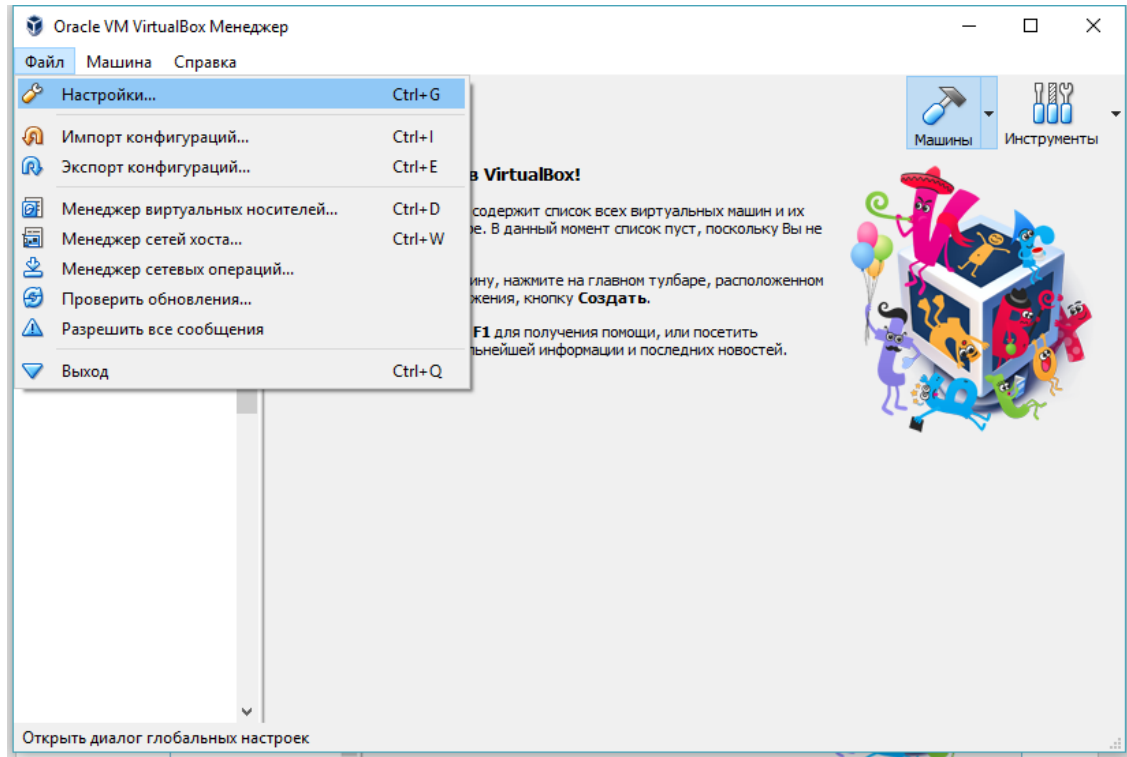


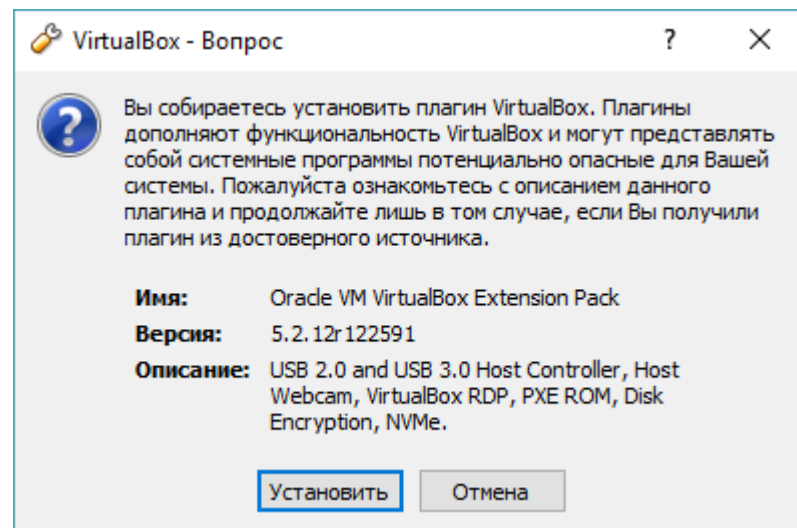
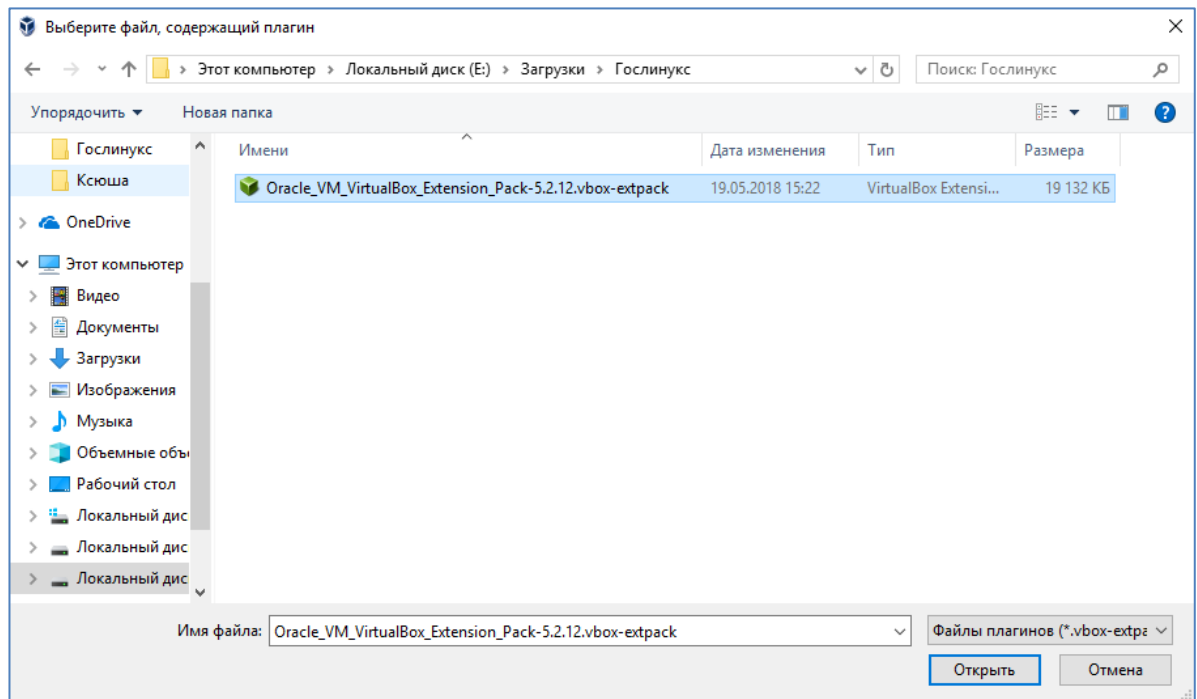
После установки в меню Пуск и на рабочем столе появится ярлык установленной программы.



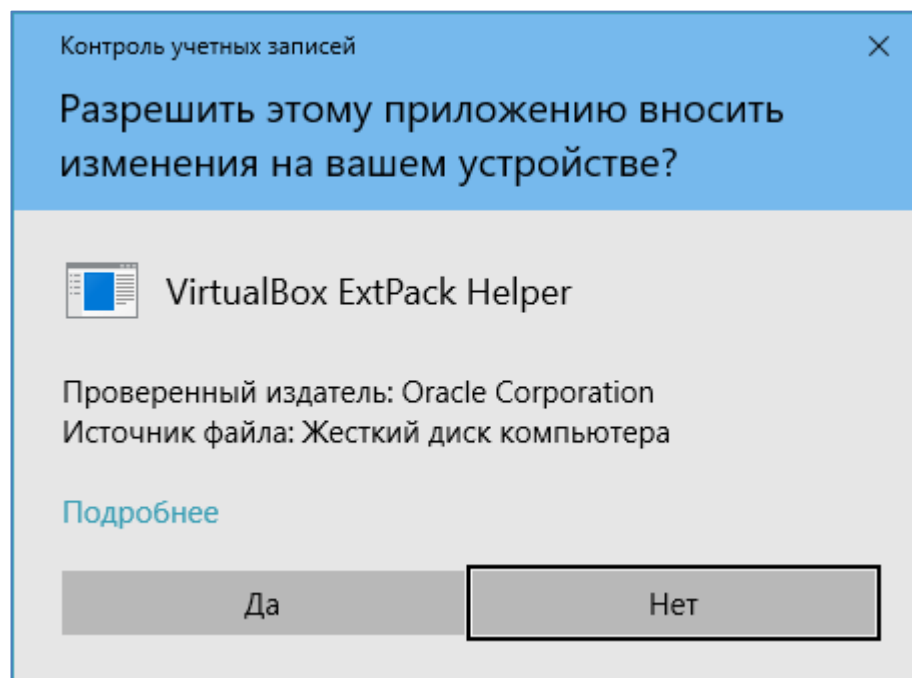
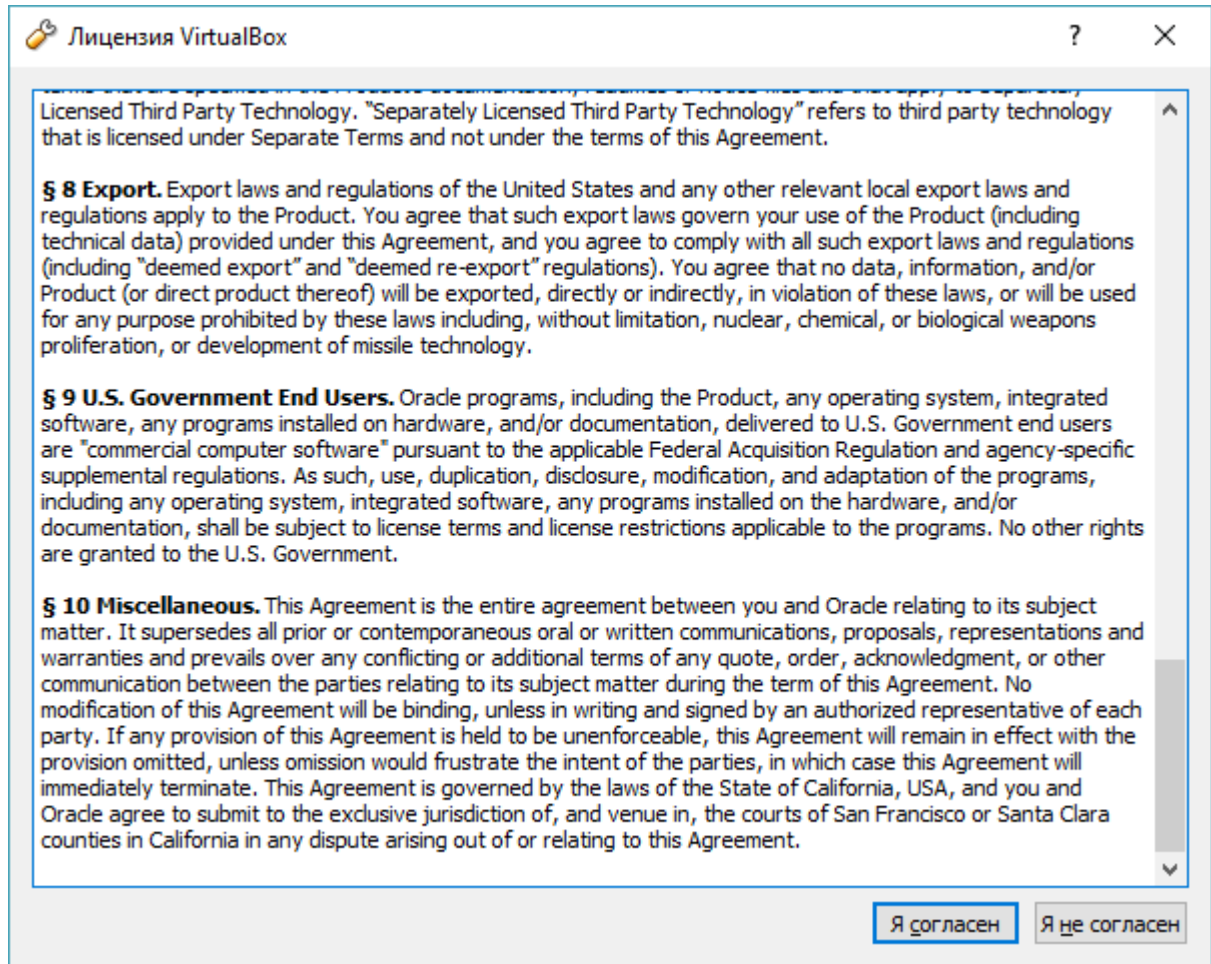
4. Запустите VirtualBox. Выполните настройки виртуальной машины как показано на рисунках ниже.



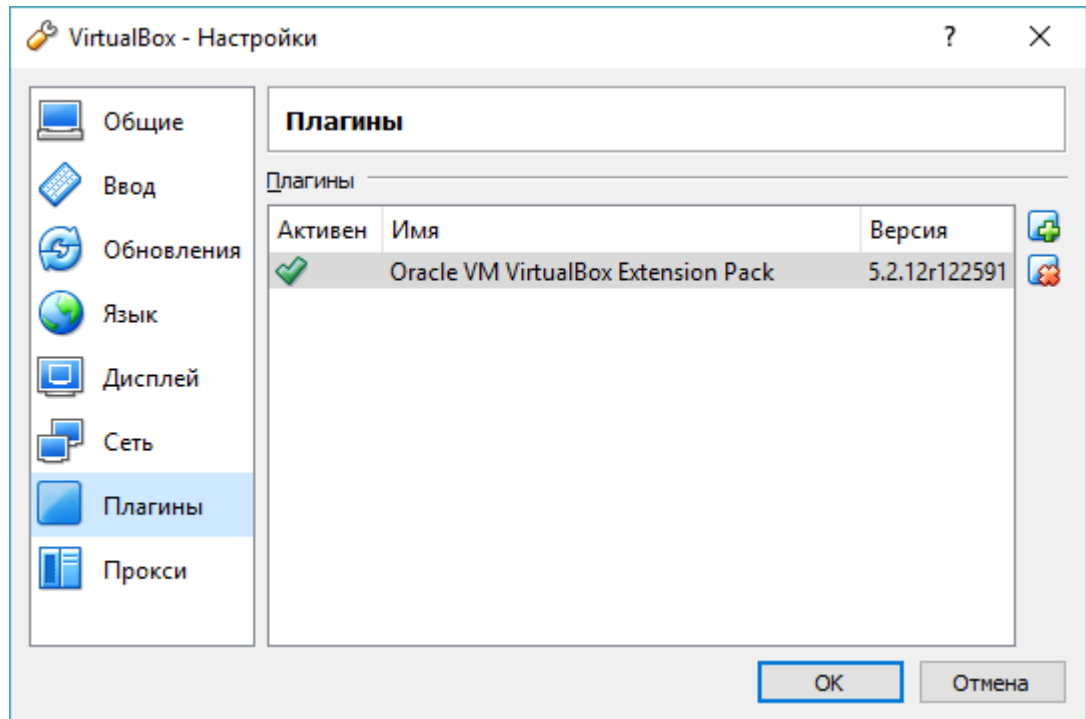
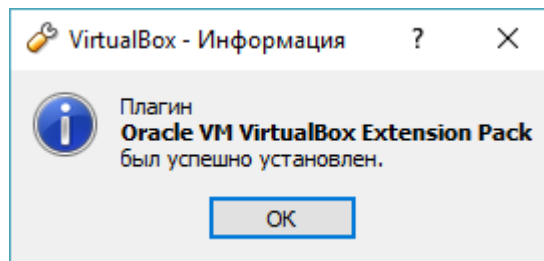




Прокрутить текст лицензии до конца. Нажать «Я согласен»



Разрешите вносить изменения, нажмите Да.



Закройте настройки нажатием «ОК». Виртуальная машина готова к работе.

Установка и настройка ОС GosLinux на виртуальной машине VirtualBox

1. Скачайте установочный файл ОС GosLinux по адресу

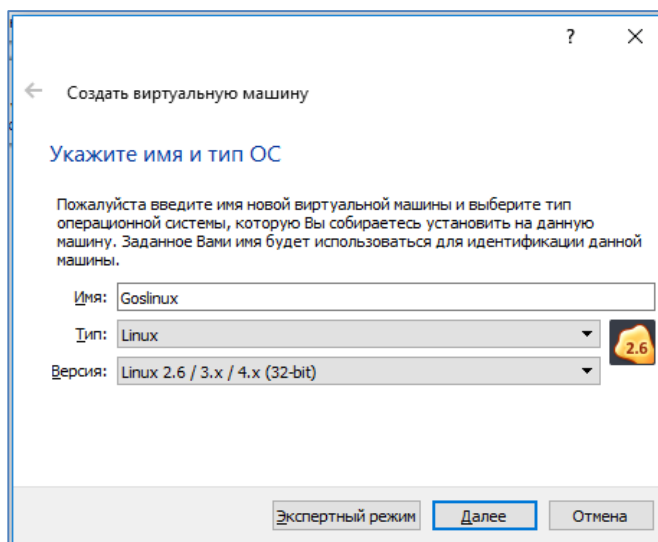
http://fap.fssp.ru/goslinux_fssp/6/iso/IC3/

Index of /goslinux_fssp/6/iso/IC3/			
Name	Last Modified	Size	Type
Parent Directory/		-	Directory
GosLinux-IC3-RTM-i686-DVD1.iso	2017-Apr-27 16:20:18	2.8G	application/octet-stream
GosLinux-IC3-RTM-i686-DVD1.iso.sig	2017-Apr-27 16:20:25	5.6K	application/pgp-signature
GosLinux-IC3-RTM-x86_64-DVD1.iso	2017-Apr-27 16:21:58	2.9G	application/octet-stream
GosLinux-IC3-RTM-x86_64-DVD1.iso.sig	2017-Apr-27 16:22:43	5.6K	application/pgp-signature

Для 32-х битных систем необходимо скачать файл [GosLinux-IC3-RTM-i686-DVD1.iso](#).

Для 64-х битных - [GosLinux-IC3-RTM-x86_64-DVD1.iso](#).

2. Запустите виртуальную машину. В окне нажмите кнопку Создать. Появится диалоговое окно



3. Выполните настройки как показано на рисунках ниже.

?

×

←

Создать виртуальную машину

Укажите объём памяти

Укажите объём оперативной памяти (RAM) выделенный данной виртуальной машине.

Рекомендуемый объём равен **512 МБ**.

4 МБ

32768 МБ

1024

МБ

Далее

Отмена

?

×

←

Создать виртуальную машину

Жесткий диск

При желании к новой виртуальной машине можно подключить виртуальный жёсткий диск. Вы можете создать новый или выбрать из уже имеющихся.

Если Вам необходима более сложная конфигурация Вы можете пропустить этот шаг и внести изменения в настройки машины после её создания.

Рекомендуемый объём нового виртуального жёсткого диска равен **8.00 ГБ**.

☐ Не подключать виртуальный жёсткий диск
☒ Создать новый виртуальный жёсткий диск
☐ Использовать существующий виртуальный жёсткий диск

Пусто

Создать

Отмена

? ×

← Создать виртуальный жёсткий диск

Укажите тип

Пожалуйста, укажите тип файла, определяющий формат, который Вы хотите использовать при создании нового жёсткого диска. Если у Вас нет необходимости использовать диск с другими продуктами программной виртуализации, Вы можете оставить данный параметр без изменений.

☒ VDI (VirtualBox Disk Image)

☐ VHD (Virtual Hard Disk)

☐ VMDK (Virtual Machine Disk)

? ×

← Создать виртуальный жёсткий диск

Укажите формат хранения

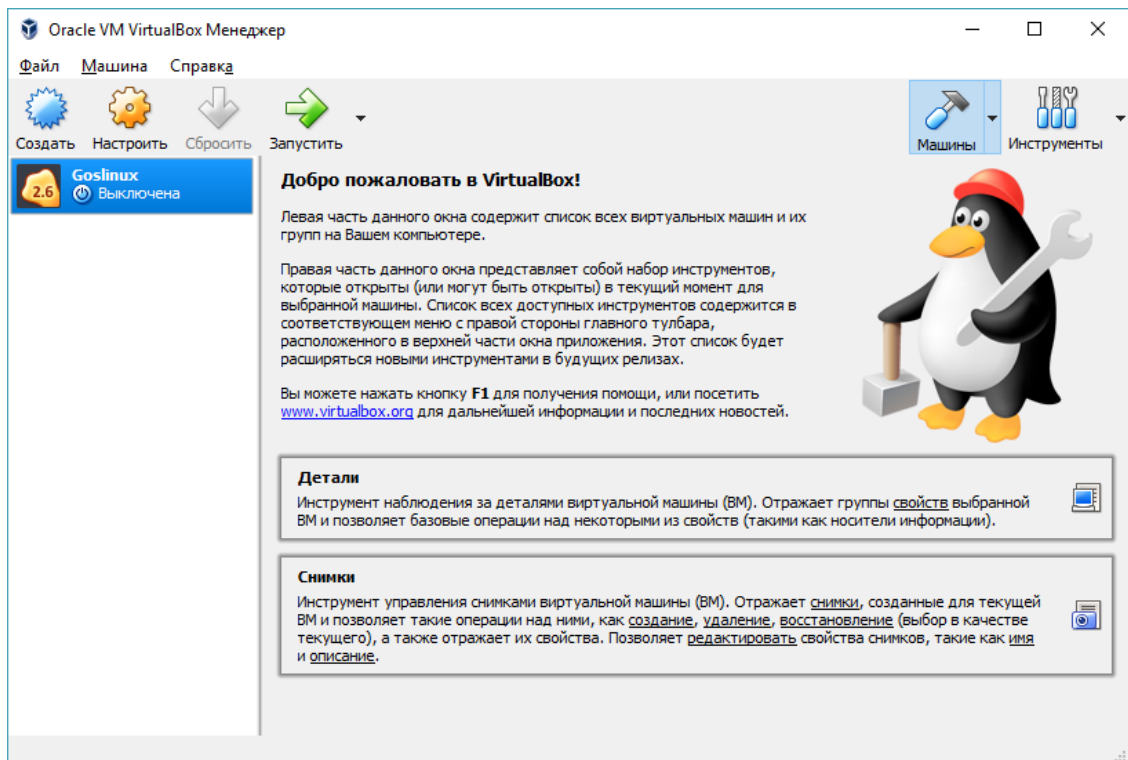
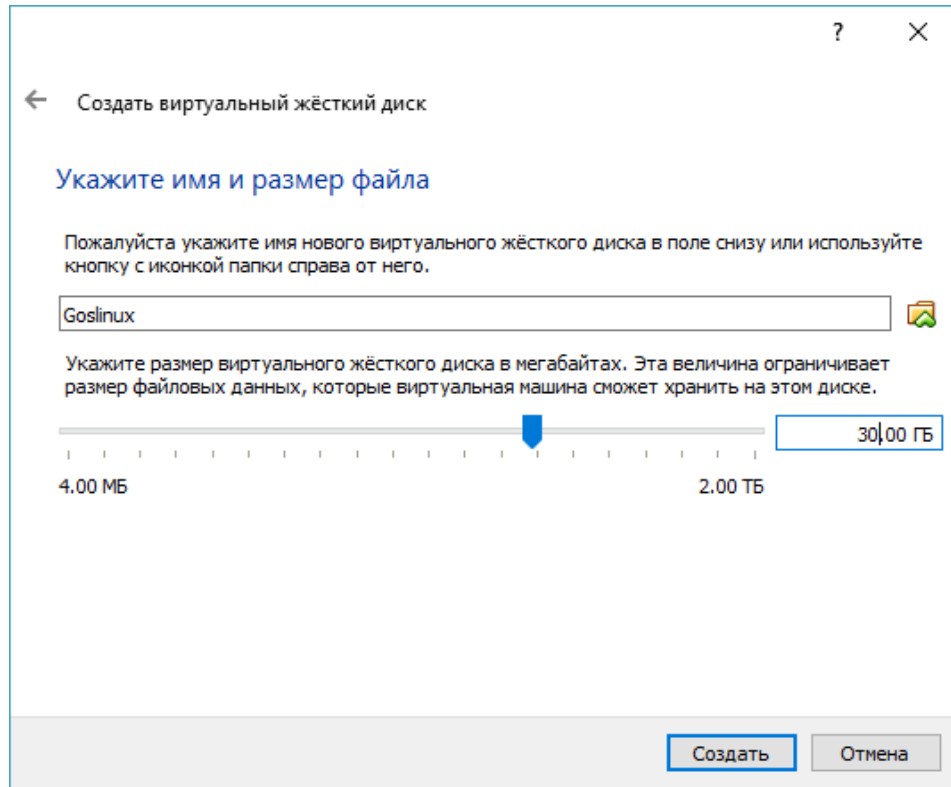
Пожалуйста уточните, должен ли новый виртуальный жёсткий диск подстраивать свой размер под размер своего содержимого или быть точно заданного размера.

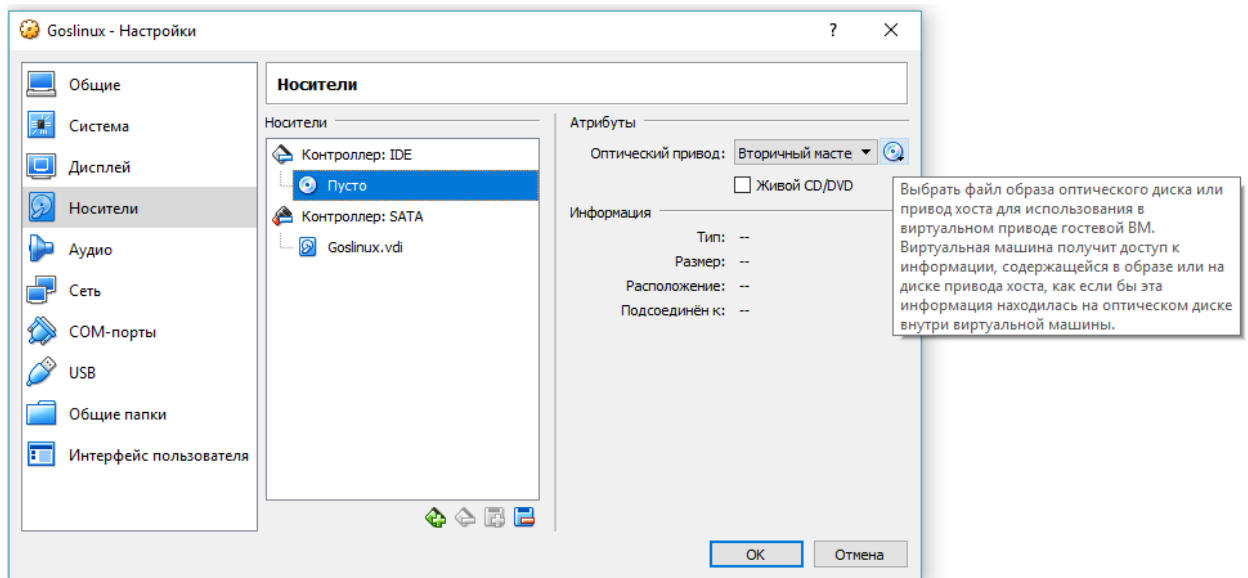
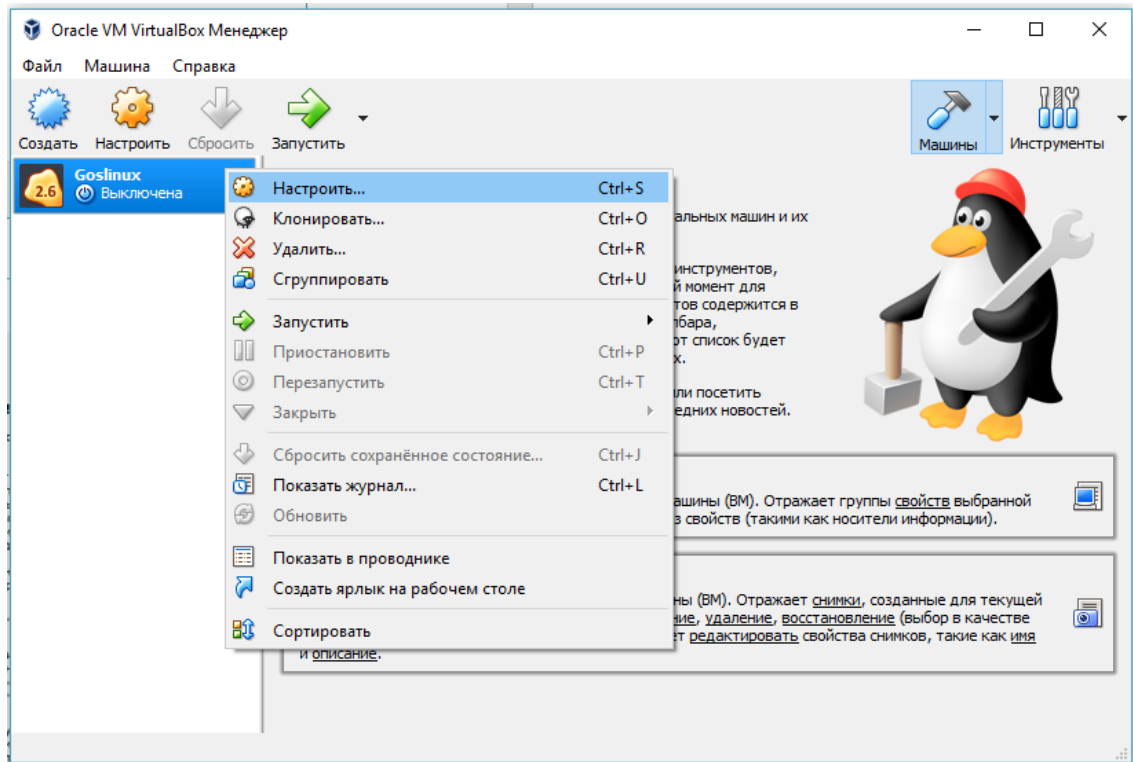
Файл **динамического** жёсткого диска будет занимать необходимое место на Вашем физическом носителе информации лишь по мере заполнения, однако не сможет уменьшиться в размере если место, занятое его содержимым, освободится.

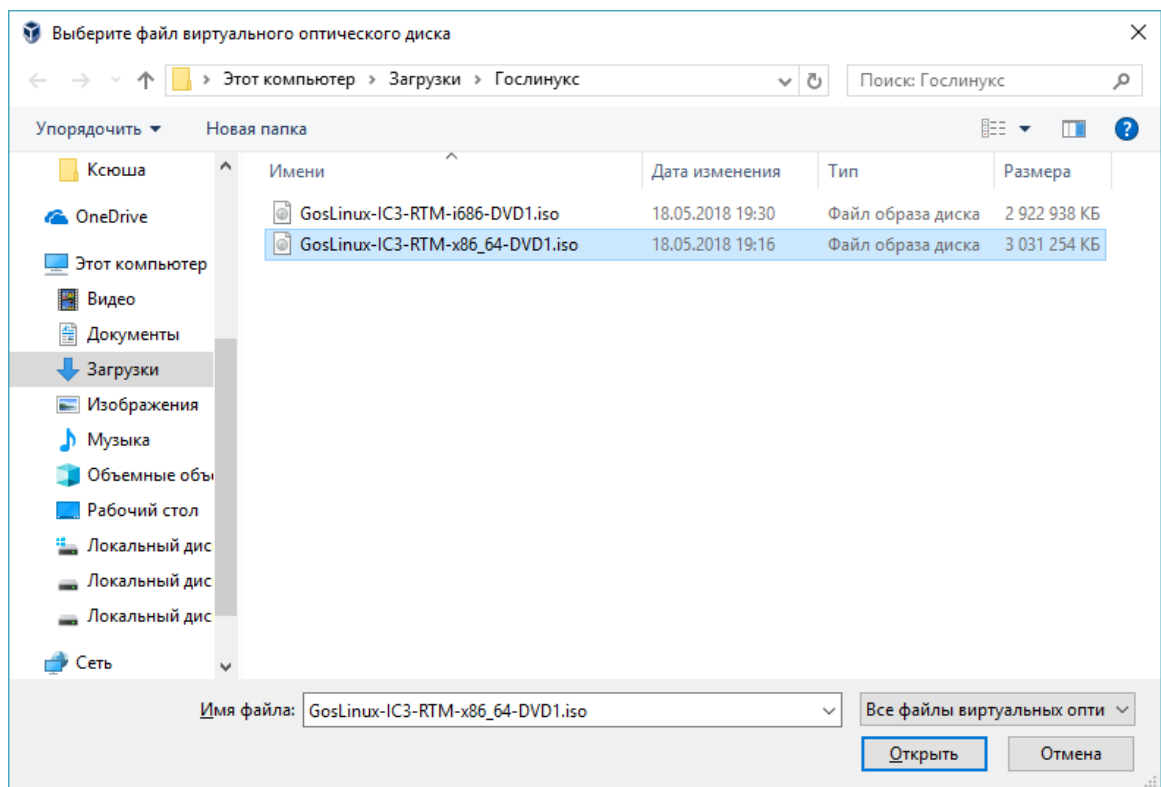
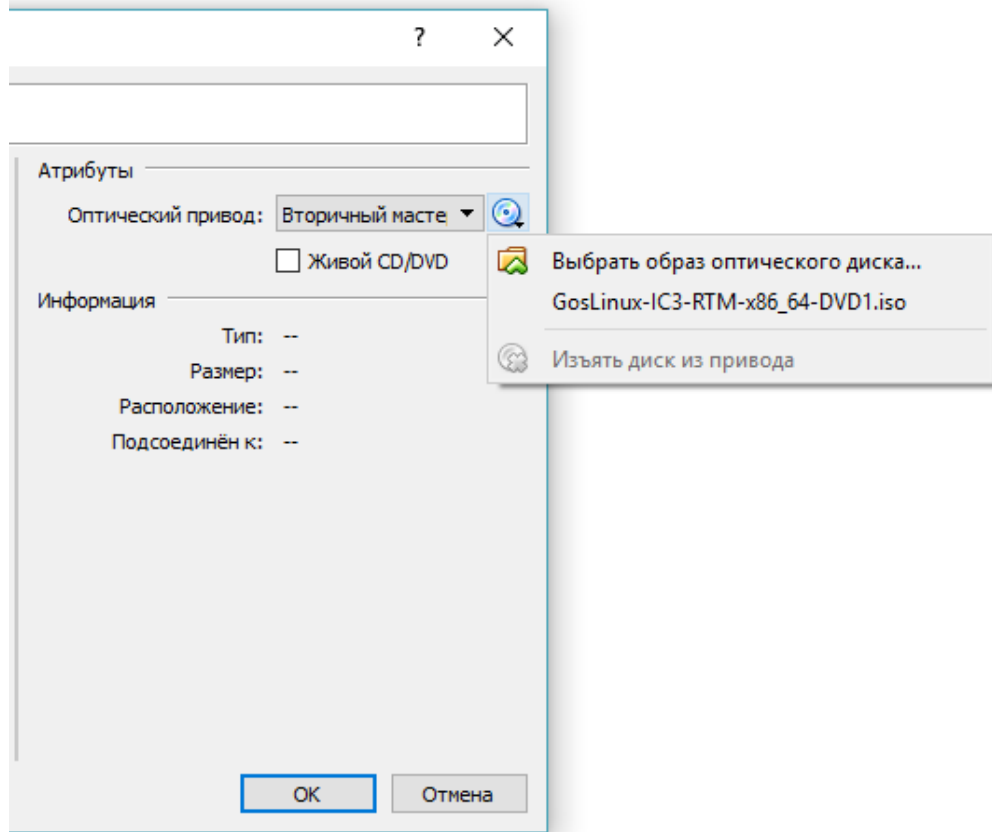
Файл **фиксированного** жёсткого диска может потребовать больше времени при создании на некоторых файловых системах, однако, обычно, быстрее в использовании.

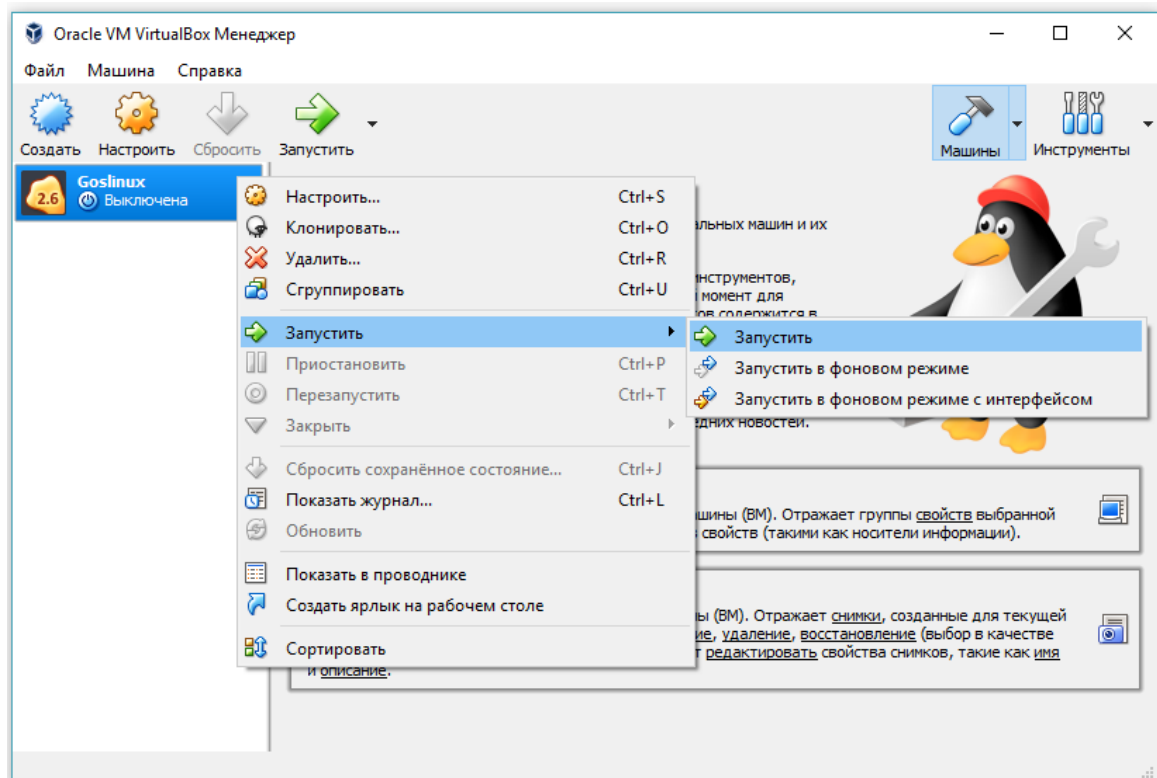
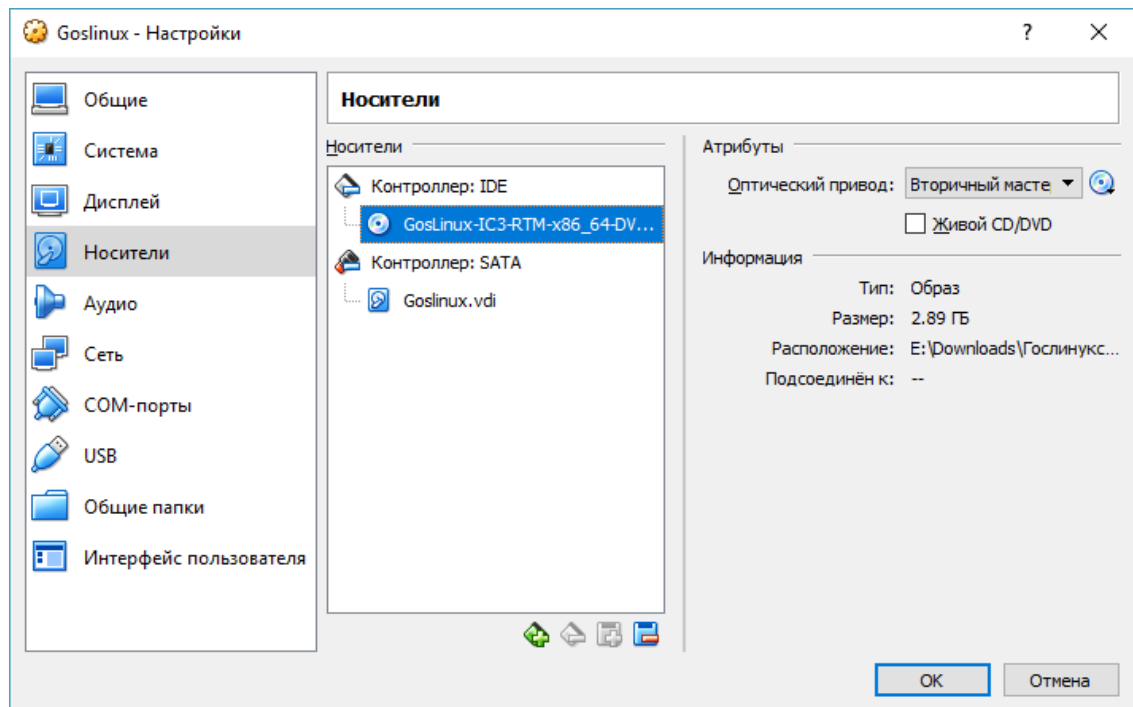
☒ Динамический виртуальный жёсткий диск

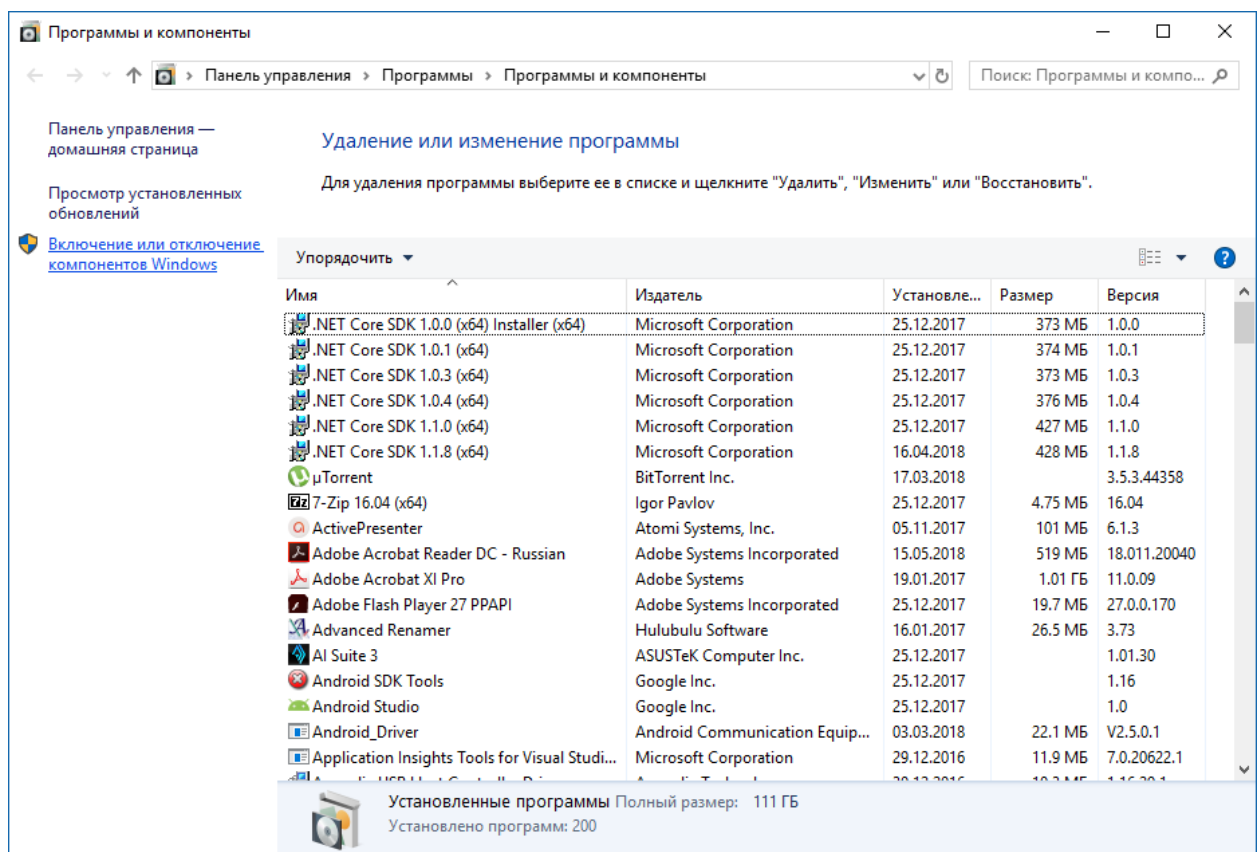
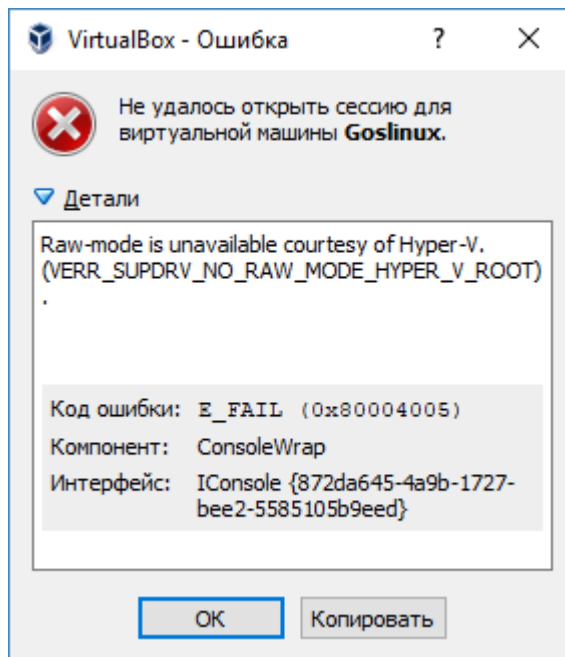
☐ Фиксированный виртуальный жёсткий диск

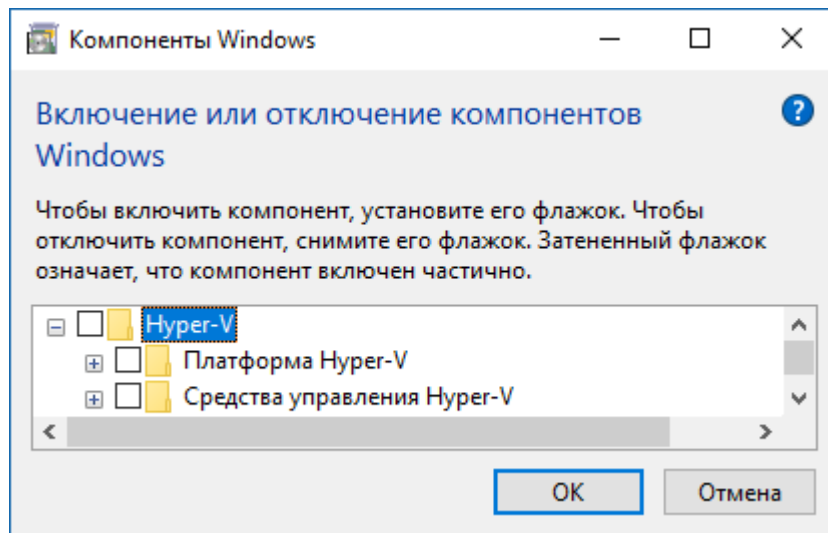




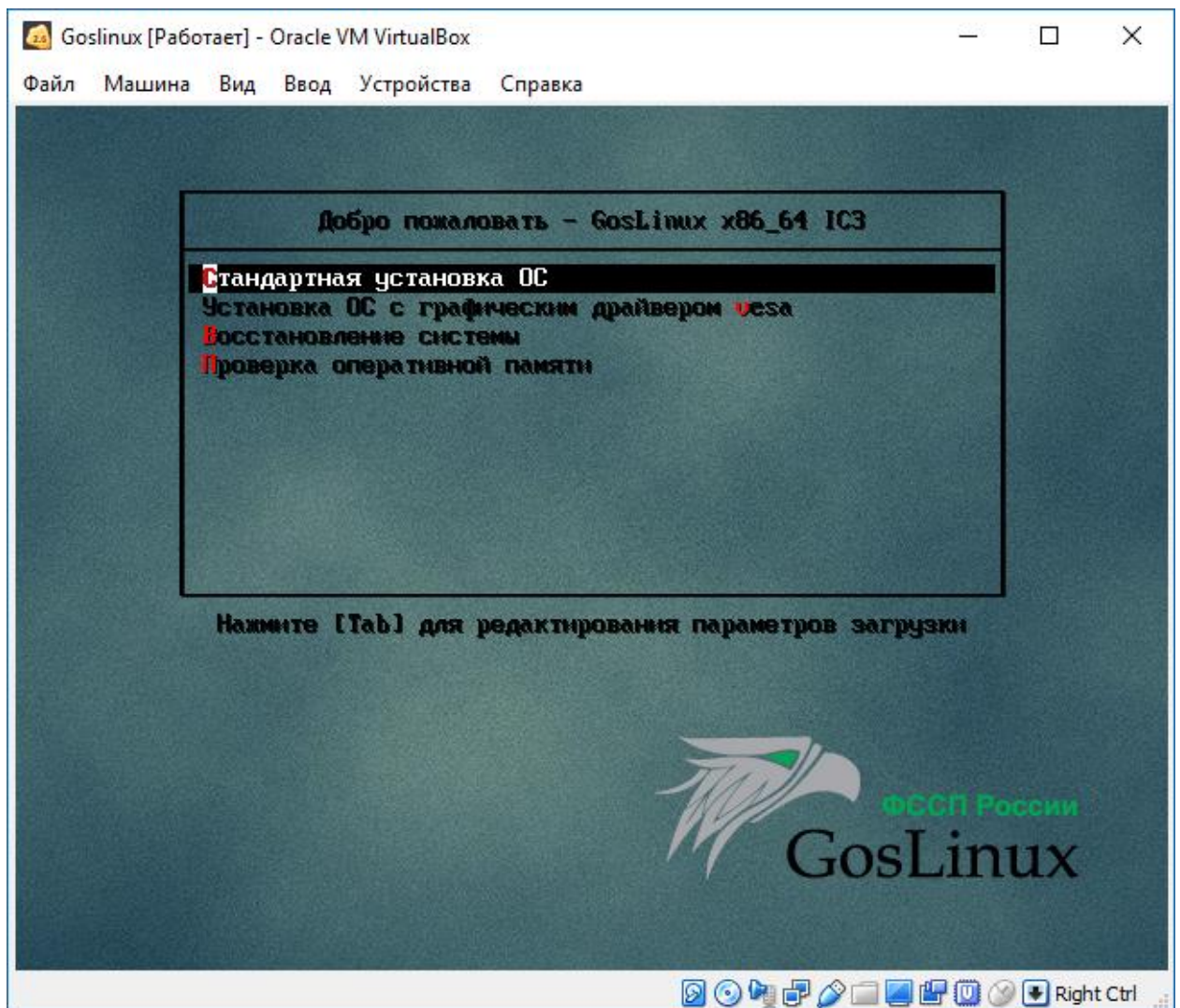








4. Далее устанавливаем ОС Goslinux. Выполните настройки как показано на рисунках ниже.

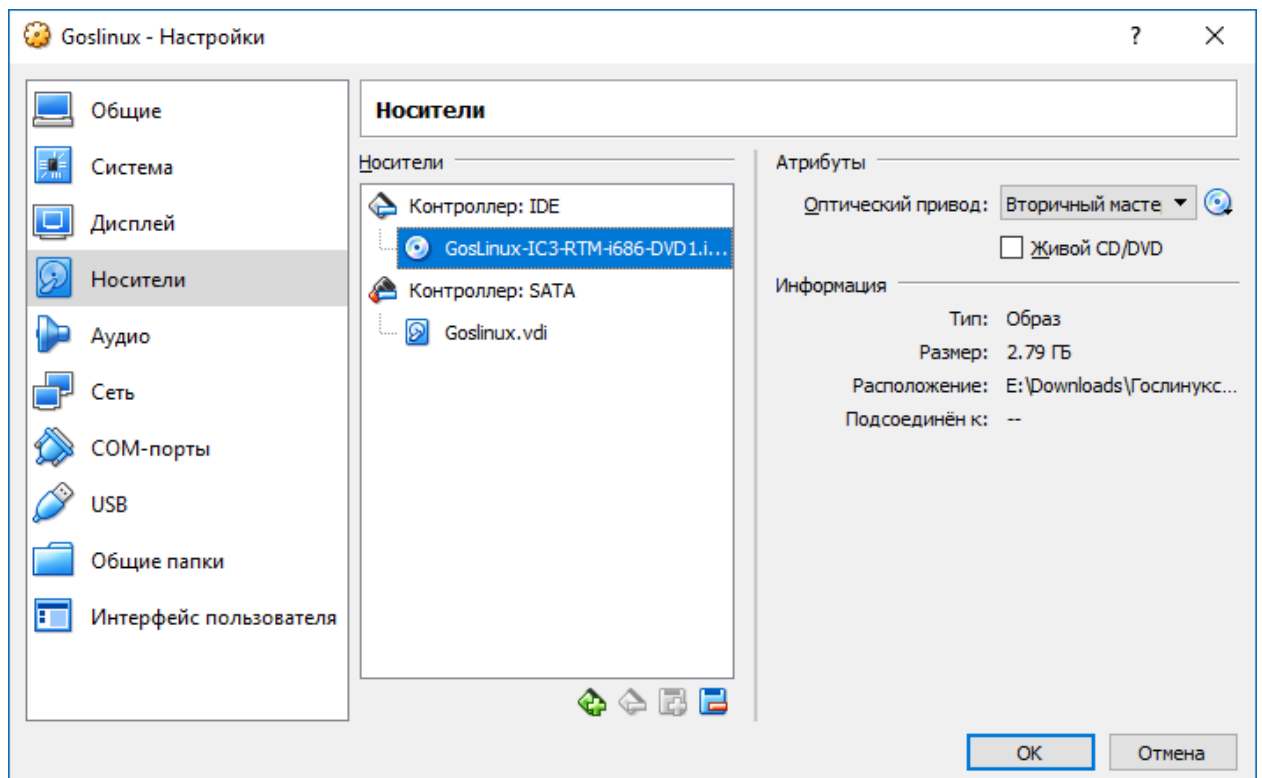
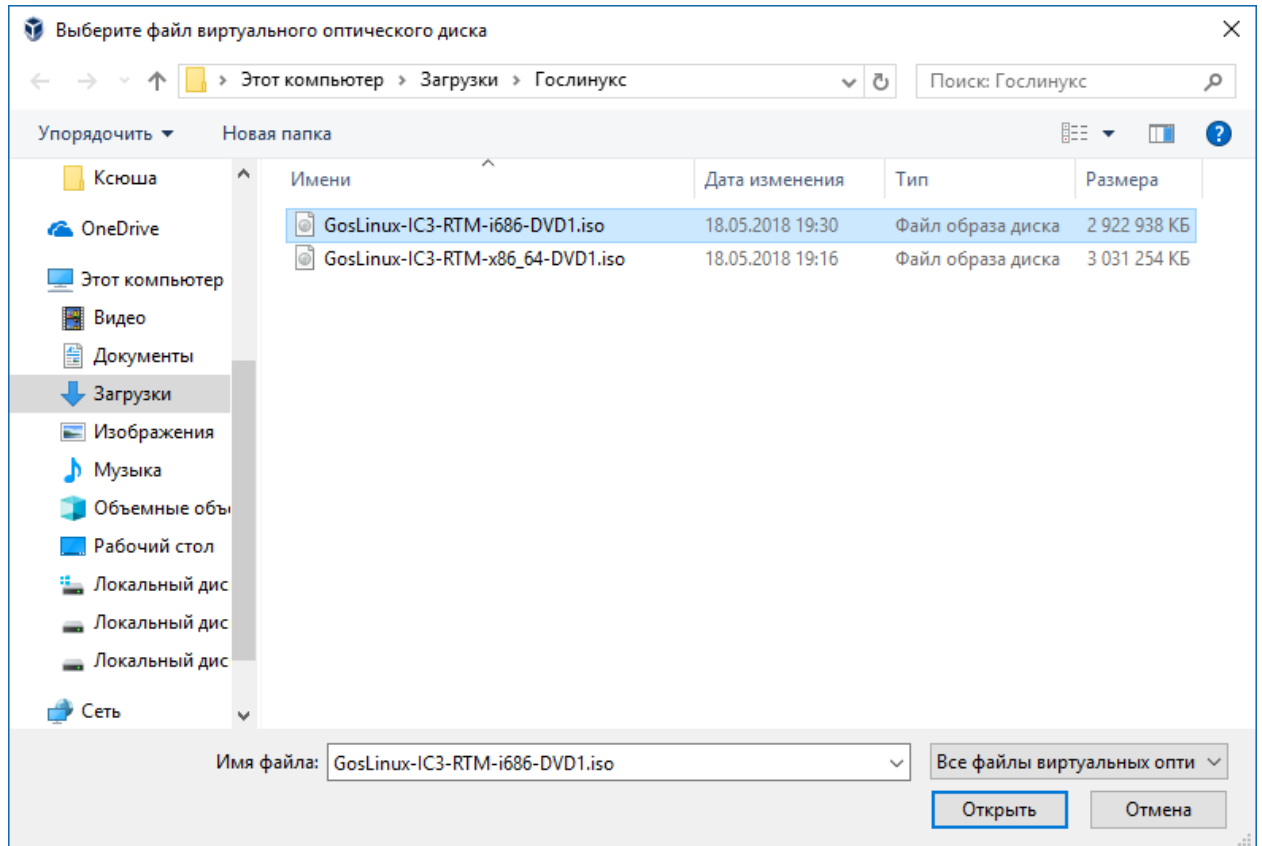


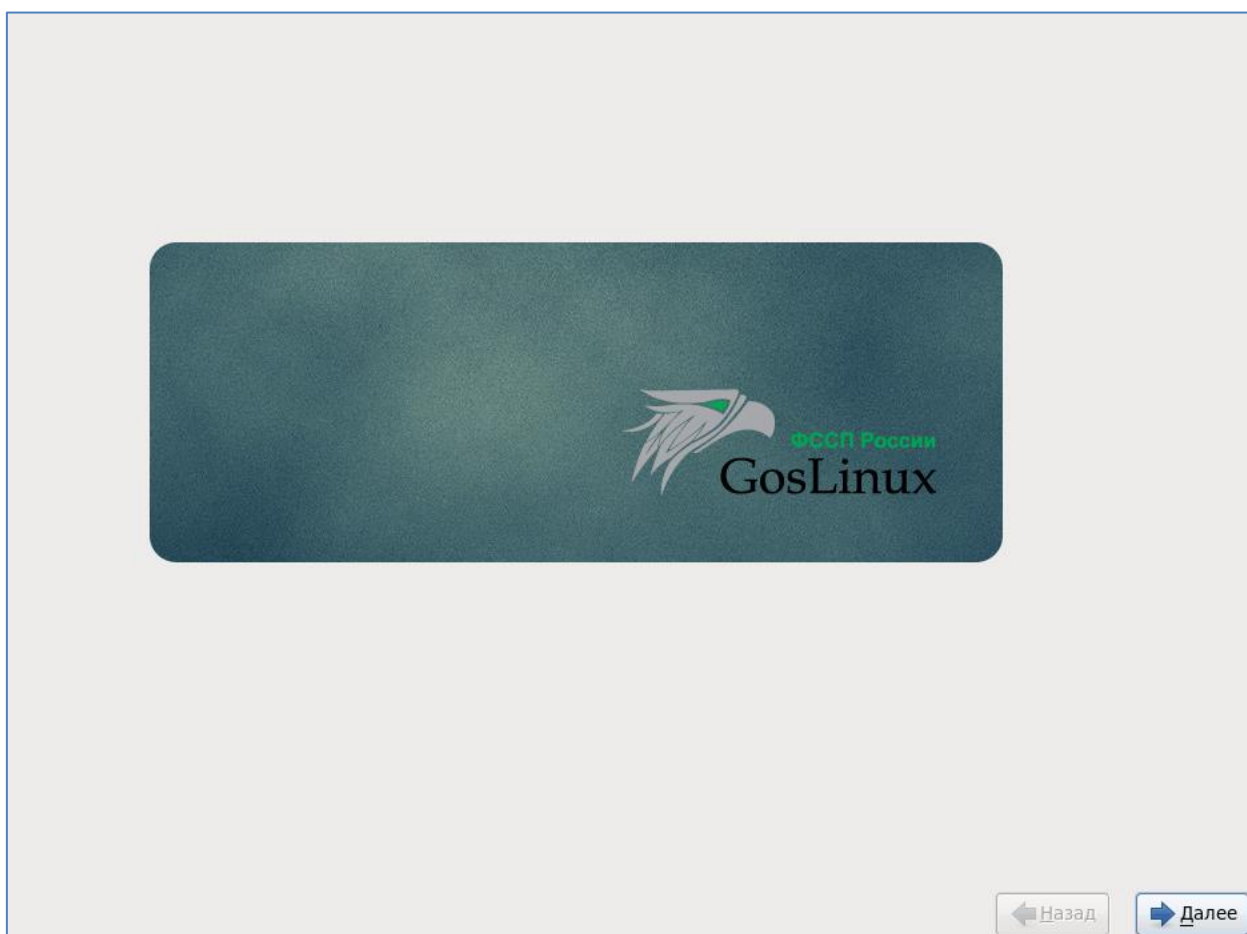
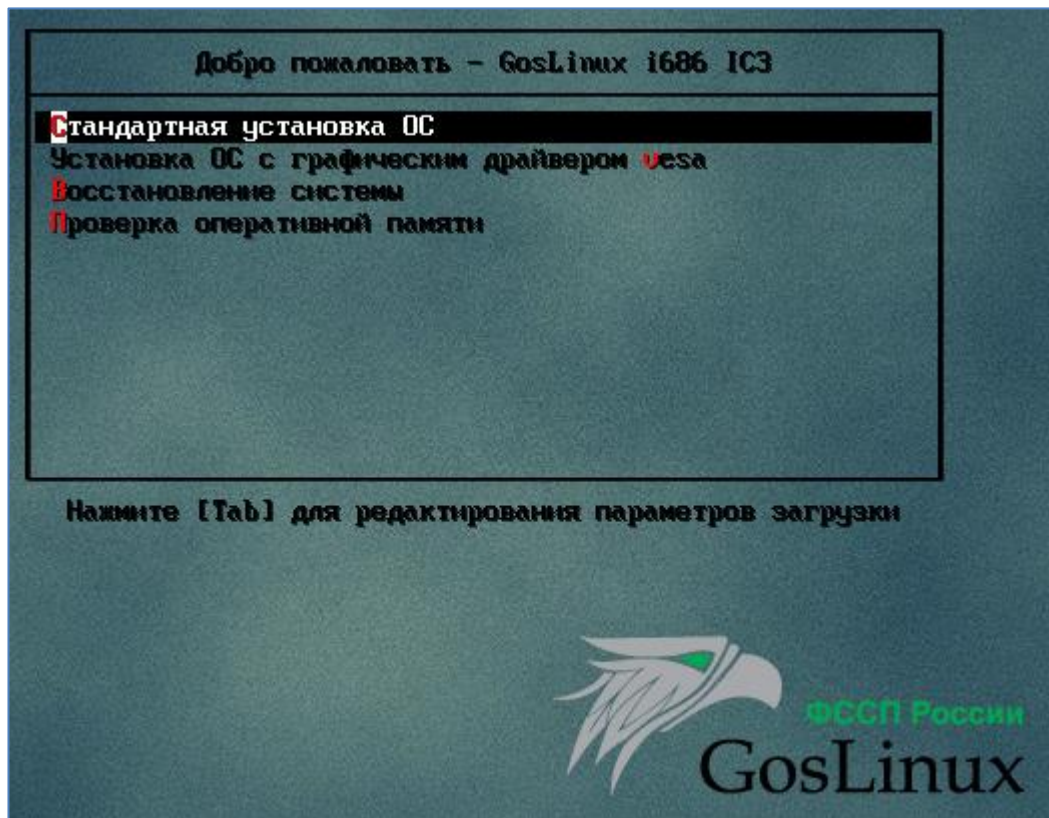
```

Loading vmlinuz.....
Loading initrd.img.....ready.
This kernel requires an x86-64 CPU, but only detected an i686 CPU.
Unable to boot - please use a kernel appropriate for your CPU.

```

Нужен GosLinux-IC3-RTM-i686-DVD1.iso





Какой тип устройств будет использоваться при установке?

Стандартные накопители

- ☒ Установка или обновление на стандартных накопителях. Этот выбор подходит, если вы не уверены, какой вариант следует выбрать.

Специальные накопители

- ☐ Позволяет установить и обновить устройства SAN (Storage Area Network), добавить диски FCoE, iSCSI, zFCP и задать устройства, которые установщик должен будет пропустить.

← Назад

Далее →

Предупреждение по поводу устройств хранения



Устройство может содержать данные.



ATA VBOX HARDDISK

30720.0 MB pci-0000:00:0d.0-scsi-0:0:0:0

На устройстве не обнаружены разделы и файловые системы.

Возможно, это устройство **пустое, неразмеченное** или **виртуальное**.

Если это не так, при продолжении установки данные, находящиеся на этом устройстве, могут быть утрачены. Для предотвращения потери данных это устройство можно исключить из установки.

Вы уверены, что это устройство не содержит важные данные?


- ☒ Применить мой выбор ко всем устройствам с нераспознанными разделами и файловыми системами

Да, удалить данные

Нет, сохранить данные


Да, удалить данные.

На вашем компьютере можно оставить предложенное имя узла.

 Присвойте этому компьютеру имя, которое будет использоваться для его идентификации в сети.

Имя узла:

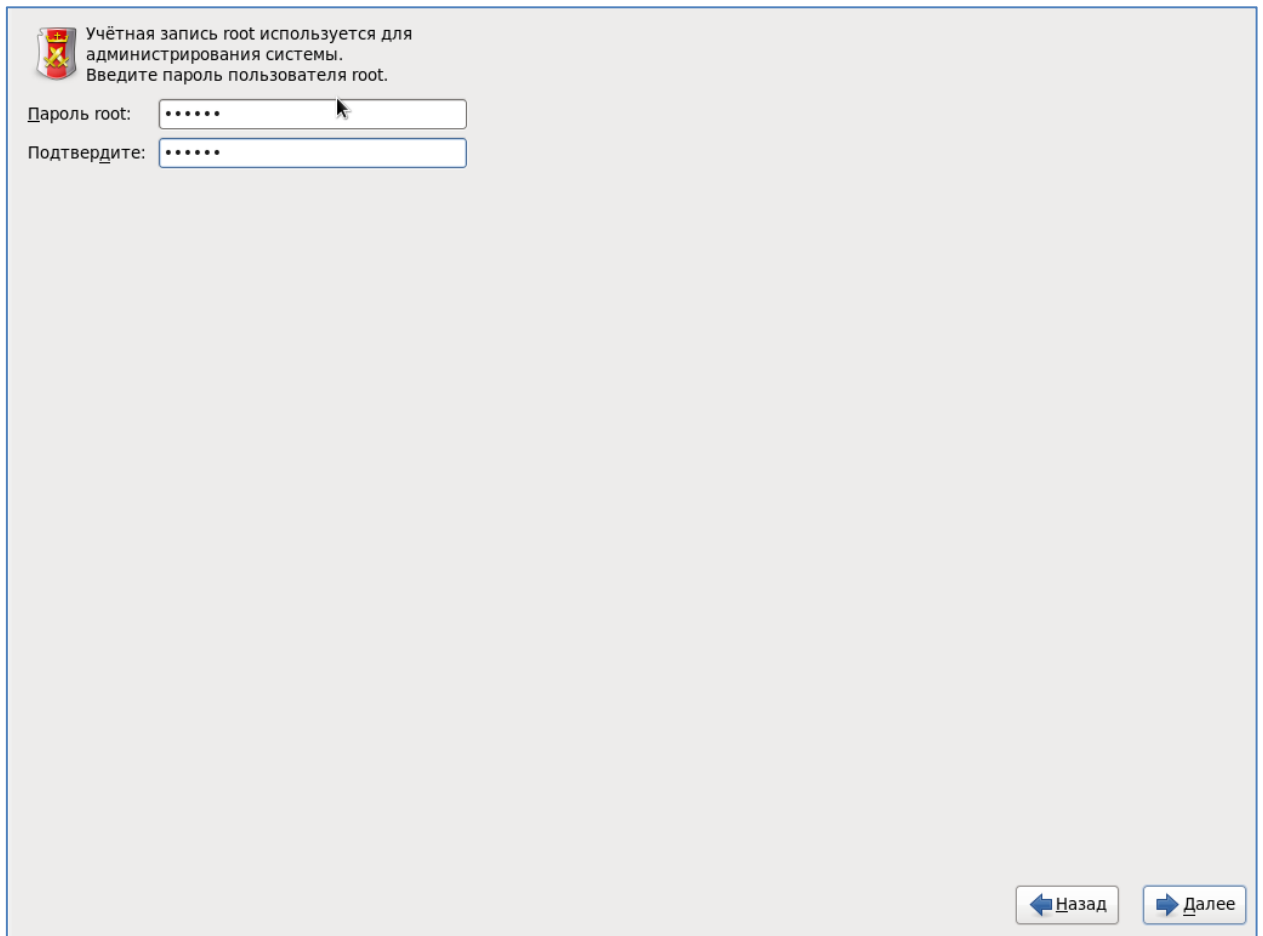
Укажите ближайший город в вашем часовом поясе:



Выбранный город: Москва, Европа (Москва+00 - Западная часть России)

☐ Системные часы используют UTC

Придумайте пароль для пользователя root и введите его.

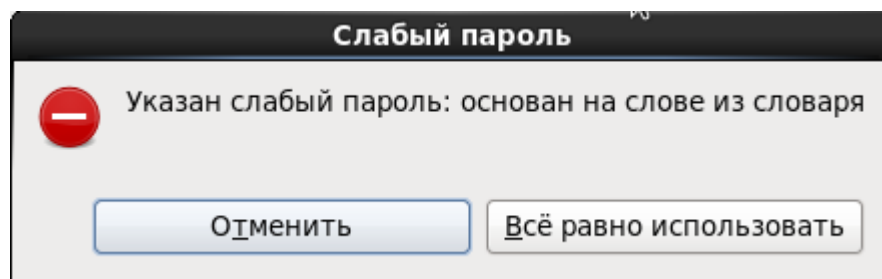


Учётная запись root используется для администрирования системы.
Введите пароль пользователя root.

Пароль root:

Подтвердите:

Назад Далее



Всё равно использовать.

Какой тип установки вы предпочитаете?

☒

Всё пространство
 Удаляет все разделы на выбранных устройствах, в том числе и разделы, созданные другими операционными системами.
Совет: Этот выбор приведёт к удалению данных на выбранных устройствах. Убедитесь, что у вас есть резервные копии.

☐

Заменить существующую систему Linux
 Удаляет только разделы Linux, созданные в ходе предыдущей установки Linux. При этом другие разделы (например, VFAT или FAT32) не будут удалены.
Совет: Этот выбор приведёт к удалению данных на выбранных устройствах. Убедитесь, что у вас есть резервные копии.

☐

Уменьшить существующую систему
 Сокращает существующие разделы с целью освобождения места для стандартного разбиения.

☐

Использовать свободное пространство
 Сохраняет текущие данные и разделы и использует только нераспределённое пространство на выбранных устройствах, предполагая, что этого будет достаточно.


☐

Создать собственное разбиение
 Позволяет создать собственное разбиение на выбранных устройствах с помощью специальной утилиты создания разделов.

☐ Зашифровать систему
☐ Просмотр и изменение структуры разделов

← Назад
Далее →

Выполняется запись информации о хранилище на диск



Выбранные параметры разбиения будут записаны на диск. Данные в удаляемых и форматируемых разделах будут удалены.

Вернуться
Сохранить изменения на диск

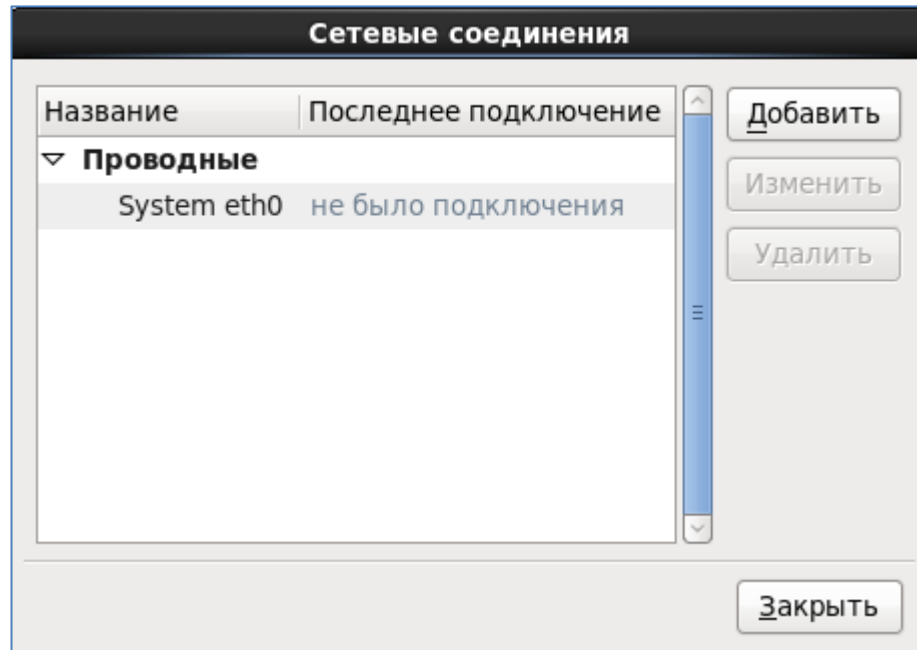
Сохранить изменения на диск.

Выбор сетевого интерфейса

Для этого необходимо наличие активного сетевого соединения на протяжении процесса установки. Выполните настройку сетевого интерфейса.

eth0 - Intel Corporation 82540EM Gigabit Ethernet Controller - 08:00:4

Отменить
ОК



Стандартная установка GosLinux это графический рабочий стол. Вы также можете выбрать другой набор устанавливаемого ПО.

- ☐ Рабочая станция ЗЛВС ФССП
- ☐ Рабочая станция ОЛВС ФССП
- ☐ Сервер ФССП (минимальный)
- ☐ Сервер ФССП (базовый)
- ☐ Стандартный сервер
- ☒ Стандартная рабочая станция

Стандартная установка GosLinux это графический рабочий стол. Вы также можете выбрать другой набор устанавливаемого ПО.

☐ Рабочая станция ЗЛВС ФССП
☐ Рабочая станция ОЛВС ФССП
☐ Сервер ФССП (минимальный)
☐ Сервер ФССП (базовый)
☐ Стандартный сервер
☒ Стандартная рабочая станция

Выберите дополнительные репозитории для установки ПО.

☒ GosLinux
☒ GosLinux-6.6 - Base
☐ GosLinux-6.6 - Extras
☐ GosLinux-6.6 - Extras

Можно изменить набор пакетов сейчас или после завершения установки с помощью специальной программы управления пакетами.

☒ Настроить позже
 ☐ Настроить сейчас

Изменить репозиторий

Укажите настройки для этого программного репозитория.

Имя репозитория: GosLinux-6.6 - Base

Тип репозитория: HTTP/FTP

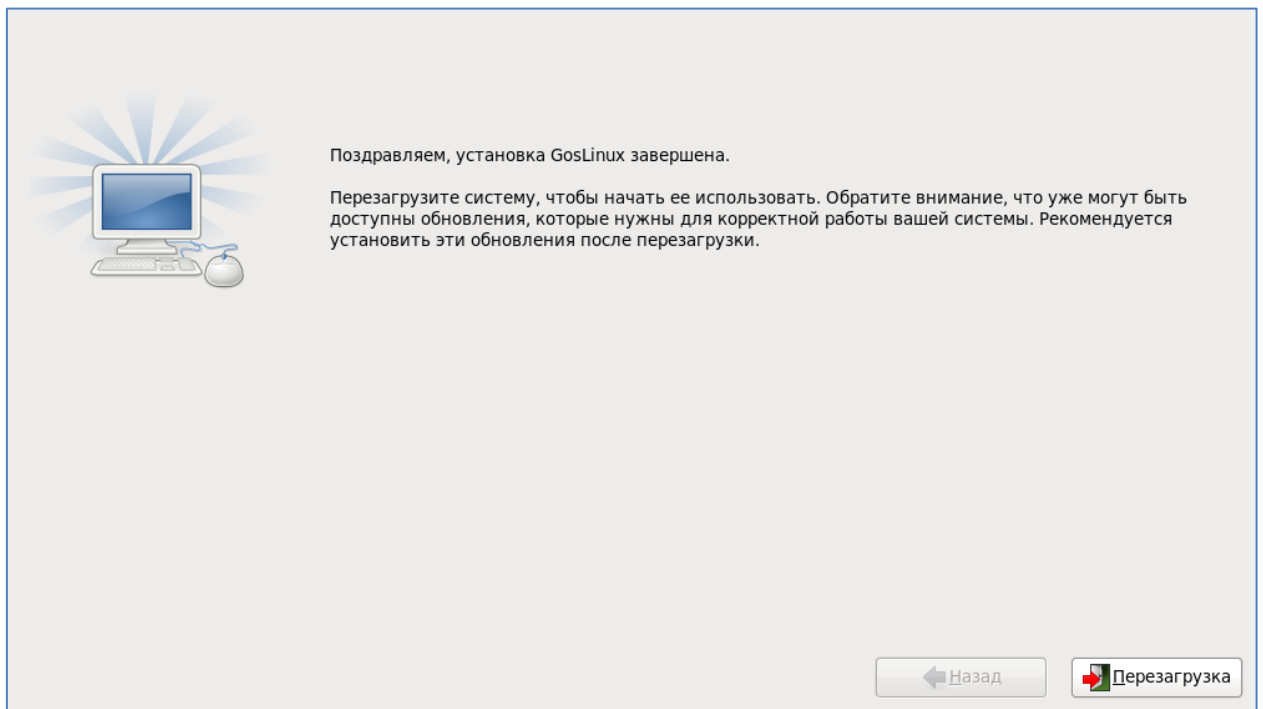
URL репозитория http://fap.fssprus.ru/goslinux_fssp/6.6/os/i386

☐ URL содержит список зеркал
☐ Настройка прокси

Адрес прокси (узел:порт)

Имя пользователя прокси

Пароль прокси



Добро
пожаловать
Информация о
лицензии
Пользователь
Дата и время
Kdump

Добро пожаловать

Осталось ещё несколько шагов, после чего ваша система будет готова к работе. В этом вам поможет помощник по настройке. Нажмите кнопку продолжения.

[Назад](#)[Вперёд](#)

Добро
пожаловать
Информация о
лицензии
Пользователь
Дата и время
Kdump

Информация о лицензии

Пользовательское соглашение Операционной системы типового дистрибутива АИС ФССП России (ОС ТД)

Операционная система типового дистрибутива АИС ФССП России (ОС ТД) распространяется без каких-либо гарантий и обязательств любого вида - письменных или подразумеваемых.

Дистрибутив распространяется под лицензией GPL версии 2. Отдельные пакеты в дистрибутиве имеют свою собственную лицензию.

Оригинал GPL версии 2 и его перевод на русский язык включены в распространяемый образ.

☒ Да, я принимаю условия соглашения

☐ Нет, я не согласен

[Назад](#)[Вперёд](#)

Создайте пользователя, например student и придумайте пароль.

Добро пожаловать

Информация о лицензии

Пользователь

Дата и время

Kdump

Пользователь

Требуется создать пользователя для повседневного (не административного) использования системы. Для этого введите необходимые данные.

Имя пользователя:

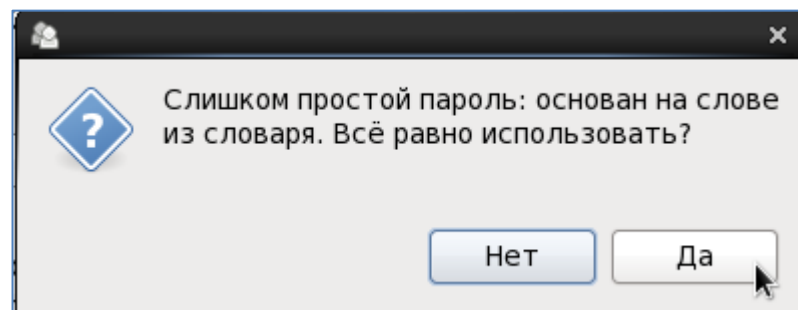
Полное имя:

Пароль:

Подтвердите пароль:

Если требуется использовать проверку подлинности по сети, например Kerberos или NIS, нажмите кнопку «Сетевая аутентификация».

Для настройки других параметров (домашнего каталога, UID) нажмите кнопку «Дополнительно».



Нажмите да.

Добро
пожаловать
Информация о
лицензии
Пользователь
› Дата и время
Kdump

Дата и время

Пожалуйста, установите дату и время системы.

Дата и время

Текущие дата и время: Сбт 19 Май 2018 13:40:18

☐ Синхронизация даты и времени по сети

Установка системной даты и времени вручную:

Дата

< Май >		< 2018 >				
Пнд	Втр	Срд	Чтв	Птн	Сбт	Вск
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Время

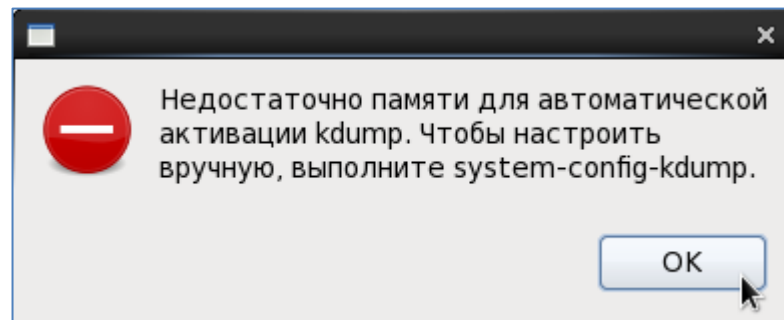
Часы : 13

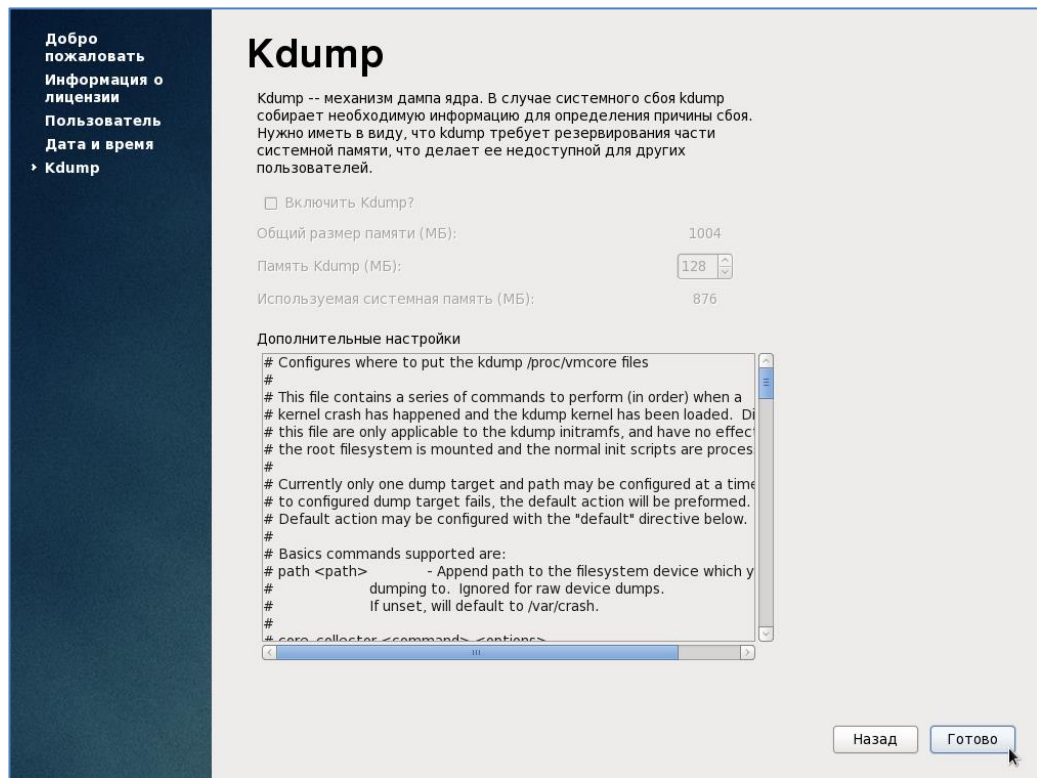
Минуты : 36

Секунды : 56

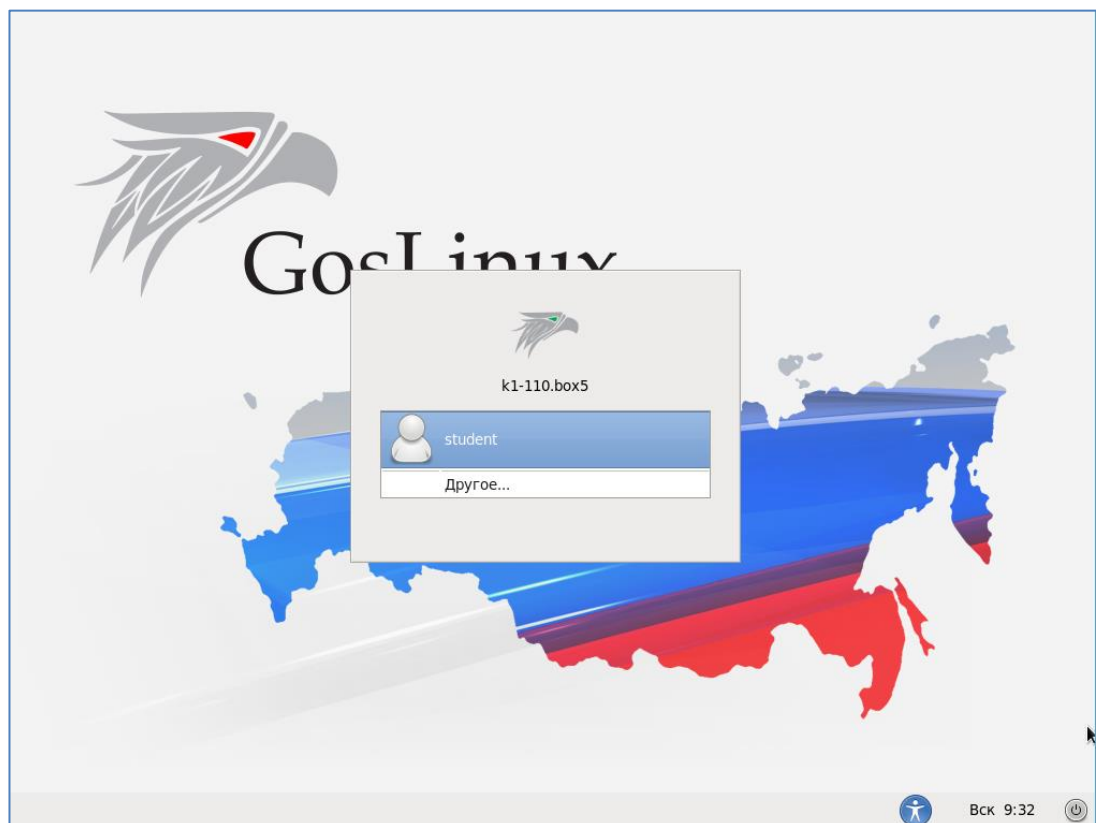
Назад

Вперёд





ОС GosLinux установлена.



В таблице приведены стандартные каталоги и их содержимое

Каталог	ОС	Содержимое
/bin	Все	Команды операционной системы ядра
/boot	LS	Ядро и файлы для его загрузки
/dev	Все	Файлы устройств: дисков, принтеров, псевдотерминалов и т.д.
/etc	Все	Важные файлы запуска и конфигурации системы
/home	Все	Стандартные домашние каталоги пользователей
/kernel	S	Компоненты ядра
/lib	Все	Библиотеки, совместно используемые библиотеки и компоненты компилятора языка C
/media	LS	Точки монтирования файловых системы на съемных носителях
/mnt	LSA	Временные точки монтирования
/opt	Все	Программные пакеты необязательных приложения (которые пока не находят широкого применения)
/proc	LSA	Информация о всех выполняющихся процессах
/root	LS	Домашний каталог суперпользователя (часто просто /)
/sbin	Все	Команды, необходимые для обеспечения минимальной работоспособности системы
/stand	H	Автономные утилиты, средства диагностики и форматирования дисков
/tmp	Все	Временные файлы, которые могут удаляться при перезагрузке
/usr	Все	Иерархия дополнительных файлов и программ
/usb/bin	Все	Содержимое
/usr/include	Все	Файлы заголовков, предназначенные для компиляции C-программ

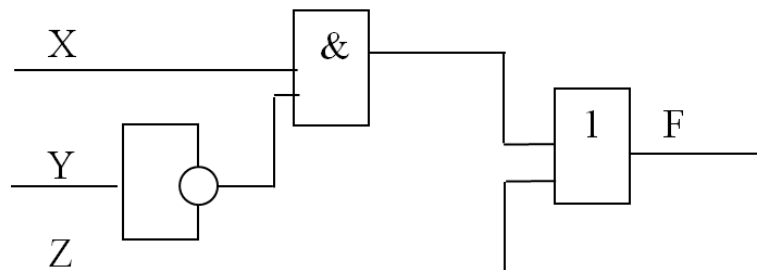
/usr/lib	Все	Библиотеки и вспомогательные файлы для стандартных программ
/usr/lib64	L	64-разрядные библиотеки для 64-разрядных дистрибутивов Linux
/usr/local	Все	Локальные программы (программы, создаваемые или устанавливаемые локальными пользователями)
/usr/sbin	Все	Менее важные файлы системного администрирования
/usr/share	Все	Элементы, общие для различных систем
/usr/share/man	Все	Страницы интерактивной документации
/usr/src	LSA	Исходные коды нелокальных программных пакетов (не находят широкого применения)
/usr/tmp	Все	Дополнительный каталог для временных файлов, которые могут сохраняться при перезагрузке
/var	Все	Системные данные и конфигурационные файлы
/var/adm	Все	Разное: журнальные файлы, записи об установке системы, административные компоненты
/var/log	LSA	Системные журнальные файлы
/var/spool	Все	Буферные каталоги для принтеров, электронной почты и т.д.
/var/tmp	Все	Каталог для временного хранения файлов

Примечание: L = Linux, S = Solaris, H = HP-UX, A = AIX

ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ

Вариант 1

1. Вычислите сумму двух чисел 22_8 и 10100101_2 . Ответ переведите в десятичную систему счисления.
2. Решите задачу. Сколько дискет объемом 1,44 Мбайт необходимо для хранения энциклопедии из 60 томов по 500 страниц, на каждой из которых по две полосы, содержащей 80 строк из 45 знаков?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. В массиве $Z(6,8)$ найти максимальный элемент 4 столбца.

Вариант 2

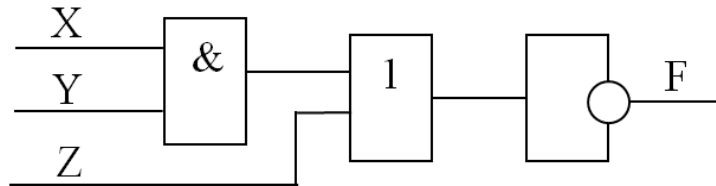
1. Вычислите разность двух чисел $AB6_{16}$ и 10101_2 . Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. В студенческой группе 16 студентов, 15 юношей и 1 девушка. Сколько бит информации содержится в сообщении «староста группы – девушка»?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

<i>A</i>	<i>B</i>	<i>F</i>
0	0	0
0	1	1
1	0	1
1	1	0

4. Составьте схему алгоритма и напишите программу. В массиве $A(5,7)$ найти минимальный элемент 3 строки.

Вариант 3

1. Вычислите сумму двух чисел $76,45_{10}$ и $77,6_8$. Ответ переведите в двоичную систему счисления.
2. Решите задачу. Объем звукового файла 5,25 Мбайт, разрядность звуковой платы – 16. Какова длительность звучания этого файла, записанного с частотой дискретизации 22,05 кГц?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. В массиве D(6,6) замените все элементы 3-й строки и 5-го столбца на противоположные по знаку (элемент, стоящий на пересечении, не изменять).

Вариант 4

1. Вычислите разность двух чисел 1000000_2 и $11,95_{10}$. Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Информационное сообщение объемом 2,5 кбайт передается со скоростью 2 560 бит/мин. За сколько минут будет передано данное сообщение?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

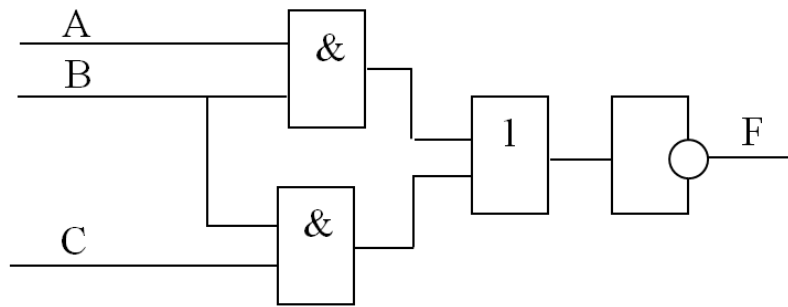
<i>A</i>	<i>B</i>	<i>F</i>
0	0	1
0	1	1
1	0	0
1	1	1

4. Составьте схему алгоритма и напишите программу. В массиве D(6,6) к элементам 3-го столбца прибавить элементы 6-го столбца.

Вариант 5

1. Вычислите сумму двух чисел $45,2_{10}$ и $F80_{16}$. Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. Сколько информации содержится в картинке экрана с разрешающей способностью 800×600 пикселей и 16 цветами?

3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу для вывода на экран значений тех элементов массива $N(3,4)$, сумма индексов которых меньше 6.

Вариант 6

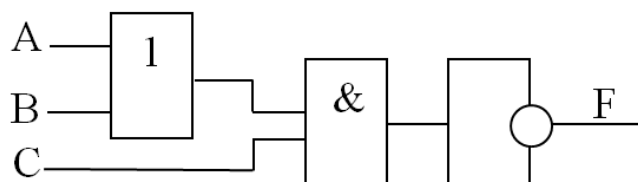
1. Вычислите разность двух чисел BA_{16} и 75_8 . Ответ переведите в двоичную систему счисления.
2. Решите задачу. Объем сообщения, содержащего 2048 символов, составил $1/512$ Мбайт. Каков размер алфавита, с помощью которого записано сообщение?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу для нахождения среднего арифметического элементов массива $X(3,4)$.

Вариант 7

1. Вычислите сумму двух чисел $77,6_8$ и $10,001001_2$. Ответ переведите в десятичную систему счисления.
2. Решите задачу. Чему равно количество бит информации в сообщении «выбранный из урны шар является черным» (в урне 8 черных и 56 белых шаров)?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. Дан массив $C(K, K)$ и вещественное число m . Вычислить сумму элементов главной диагонали массива C , удовлетворяющих условию $C(i, i) \geq m$.

Вариант 8

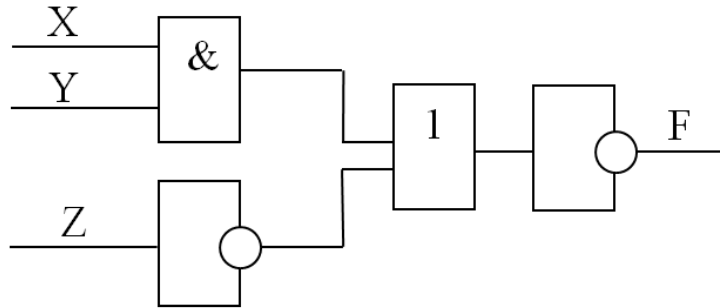
1. Вычислите разность двух чисел 165_8 и 11101_2 . Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Определить длительность звукового файла, который уместится на диске 3,5" (учтите, что для хранения данных на такой диске выделяется 2 847 секторов объемом 512 байт каждый) при низком качестве звука: моно, 8 битов, 8 000 измерений в секунду.
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу. Дан двумерный массив $A(n, m)$. Вычислить количество элементов массива, значение которых удовлетворяет условию: $5 \leq A_{ij} < 7,6$. Результат вывести на экран.

Вариант 9

1. Вычислите сумму двух чисел $1110,1_2$ и $34,25_{10}$. Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. Скорость передачи данных через модемное соединение равна 51 200 бит/с. Передача текстового файла через это соединение заняла 10 секунд. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу, в которой найдите сумму элементов каждой строки массива $A(8,9)$.

Вариант 10

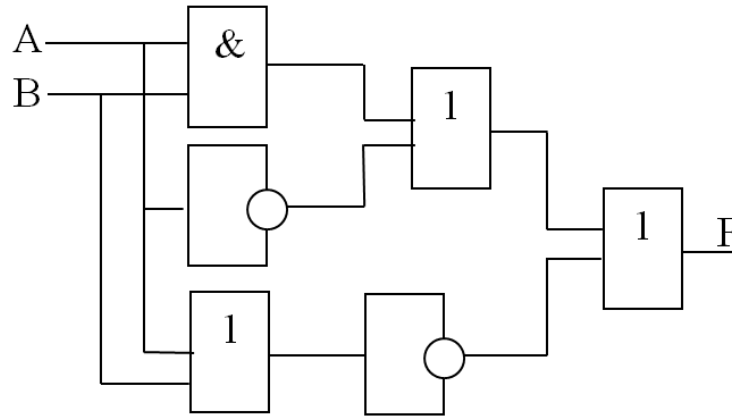
1. Вычислите разность двух чисел 111101_2 и $17,17_8$. Ответ переведите в двоичную систему счисления.
2. Решите задачу. Цветное растровое графическое изображение, палитра которого включает в себя 65 536 цветов, имеет размер 100×100 точек (пикселей). Какой объем видеопамяти компьютера (Кбайт) занимает это изображение?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу, в которой посчитайте количество элементов одномерного массива $K(6,6)$, кратных 5, или выдать сообщение, что таковых нет.

Вариант 11

1. Вычислите сумму двух чисел $100,0001_2$ и 190_{16} . Ответ переведите в десятичную систему счисления.
2. Решите задачу. Для записи текста использовался 256-символьный алфавит. Каждая страница содержит 30 строк по 70 символов в строке. Какой объем информации содержат 5 страниц текста?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. Найдите сумму положительных элементов второй строки массива $B(3,5)$.

Вариант 12

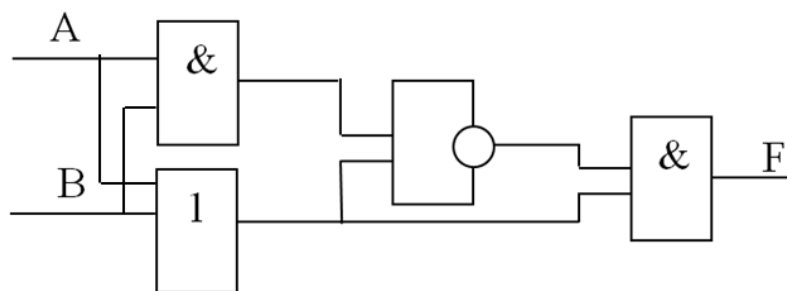
1. Вычислите разность двух чисел 52_8 и $86,2_{10}$. Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Одна минута записи цифрового аудиофайла занимает на диске 1,3 Мбайт, разрядность звуковой платы – 8. С какой частотой дискретизации записан звук?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу. Найдите сумму и количество элементов каждого столбца массива $C(7,5)$, которые попадают в промежуток от 1 до 4.

Вариант 13

1. Вычислите сумму двух чисел $CC3,5_{16}$ и $11,2_8$. Ответ переведите в двоичную систему счисления.
2. Решите задачу. Скорость передачи данных через модемное соединение равна 4 096 бит/с. Передача текстового файла через это соединение заняла 10 секунд. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. В массиве $S(5,6)$ найдите количество тех элементов в каждой строке, которые больше среднего арифметического элементов данной строки.

Вариант 14

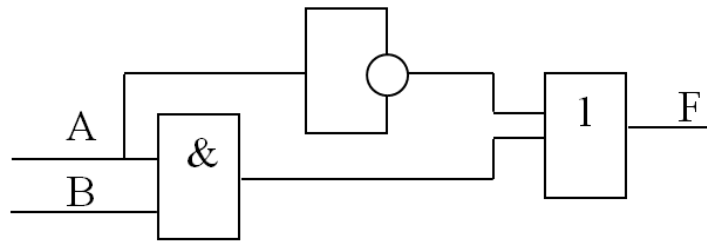
1. Вычислите разность двух чисел $1,0001_2$ и $44,9_{10}$. Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. Сколько байт памяти необходимо, чтобы закодировать изображение на экране компьютерного монитора, который может отображать 1 280 точек по горизонтали и 1 024 точек по вертикали при 256 цветах?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу для нахождения в массиве $C(7,7)$ минимального из элементов, стоящих ниже главной диагонали.

Вариант 15

1. Вычислите сумму двух чисел 1000101_2 и $F80_{16}$. Ответ переведите в десятичную систему счисления.
2. Решите задачу. Сообщение занимает 3 страницы по 25 строк. В каждой строке записано по 60 символов. Сколько символов в использованном алфавите, если все сообщение содержит 1 125 байт?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. В двумерном массиве $D(2,4)$ найти сумму произведений элементов второго и третьего столбцов. Результат вывести на экран.

Вариант 16

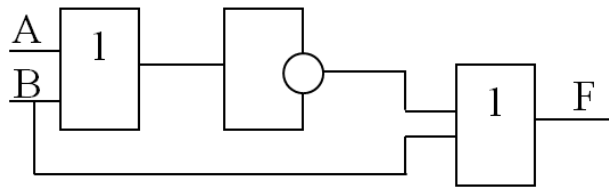
1. Вычислите разность двух чисел $10,001001_2$ и 555_{16} . Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Рассчитать время звучания моно аудиофайла, если при 8-битном кодировании и частоте дискретизации 32 кГц его объем равен 350 Кбайт.
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу. В массиве целых чисел $D(3,4)$ вычислить количество отрицательных элементов. Вывести на экран элементы, значение которых больше вычисленного количества.

Вариант 17

1. Вычислите сумму двух чисел $66,5_{10}$ и 1001100_2 . Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать 100 страниц текста в 30 строк по 16 символов каждая, при условии, что каждый символ кодируется 1 байтом?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. В массиве $X(5,7)$ последний отрицательный элемент каждого столбца заменить нулем.

Вариант 18

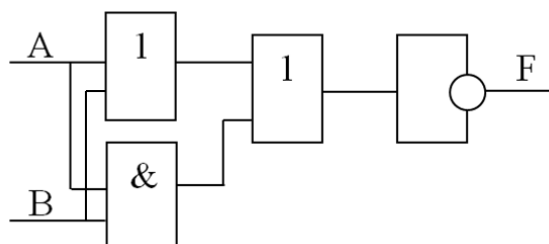
1. Вычислите разность двух чисел 11100_2 и 900_{16} . Ответ переведите в десятичную систему счисления.
2. Решите задачу. Объем видеопамати 2 Мб, битовая глубина – 24, разрешающая способность дисплея – 640×480 . Какое максимальное количество страниц можно использовать при этих условиях?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу. Дан массив $A(7,6)$. Найдите количество отрицательных элементов массива A .

Вариант 19

1. Вычислите сумму двух чисел $91,5_{10}$ и $F09_{16}$. Ответ переведите в двоичную систему счисления.
2. Решите задачу. Сообщение занимает 2 страницы и содержит 1/16 Кбайт информации. На каждой странице записано 256 символов. Какова мощность использованного алфавита?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу, в которой найдите количество элементов массива $K(2,7)$, больших среднего арифметического его элементов.

Вариант 20

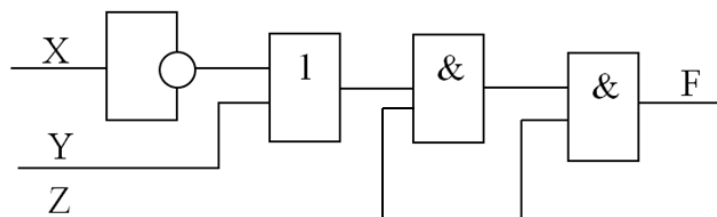
1. Вычислите разность двух чисел 1100100_2 и 900_8 . Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Объем свободной памяти на диске – 5,25 Мб, разрядность звуковой платы – 16. Какова длительность звучания цифрового аудиофайла с частотой дискретизации 22,05.
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу для нахождения наименьшего положительного элемента и его номера строки целочисленного массива $A(3,6)$.

Вариант 21

1. Вычислите сумму двух чисел $78,05_{10}$ и 1000001_2 . Ответ переведите в восьмеричную систему счисления.
2. Решите задачу. Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать цветное растровое изображение размером 800×600 пикселей, при условии, что в палитре 16 миллионов цветов?
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. Дан целочисленный двумерный массив $C(4,3)$. Вычислить сумму максимальных элементов каждой строки массива. Результат вывести на экран.

Вариант 24

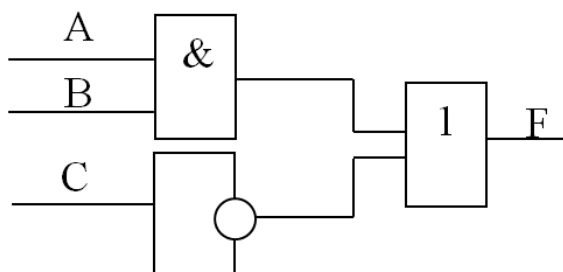
1. Вычислите разность двух чисел $55,55_8$ и $10,001001_2$. Ответ переведите в шестнадцатеричную систему счисления.
2. Решите задачу. Информационное сообщение объемом 1,5 Кбайт содержит 3072 символа. Сколько символов содержит алфавит, при помощи которого было записано это сообщение?
3. По заданной таблице истинности запишите логическую функцию, составьте логическую схему:

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

4. Составьте схему алгоритма и напишите программу. В массиве $Z(5,6)$ найдите сумму элементов в столбцах с 3-го по 6-й.

Вариант 25

1. Вычислите сумму двух чисел 45_8 и 111_{16} . Ответ переведите в десятичную систему счисления.
2. Решите задачу. Известно, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 минут. Определите максимальный размер файла (в килобайтах), который может быть передан за время такого подключения, если модем передает информацию в среднем со скоростью 32 Кбит/с.
3. По заданной логической схеме составьте соответствующее ей логическое выражение и упростите его:



4. Составьте схему алгоритма и напишите программу. Дан целочисленный двумерный массив $D(4,4)$. Вычислить сумму минимальных элементов каждой строки массива. Результат вывести на экран.

Литвинов Владимир Николаевич
кандидат технических наук, доцент

Грачева Наталья Николаевна
кандидат технических наук, доцент

Руденко Нелли Борисовна
кандидат технических наук, доцент

Каун Олег Юрьевич
кандидат технических наук

ИНФОРМАТИКА

Учебное пособие

Издается в авторской редакции

Объем ЭИ: 6,52 Мб

Формат ЭИ: Portable Document Format (PDF)

[На титульную страницу](#)