МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ**

**АКАДЕМИЯ**

Институт прикладной математики и информационных технологий

Кафедра «Общая информатика»

С.Х. Биджиева

МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ И МАТЕМАТИКИ В ШКОЛЕ

ЧАСТЬ 2

Лабораторный практикум

для обучающихся 1 курса по направлению подготовки

09.03.03. Прикладная информатика

Черкесск 2022

УДК 681.3:51:378

ББК 32.81:22:74.58

Б59

Рассмотрено на заседании кафедры «Общая информатика»

Протокол № 1 от «02» 09.2021г.

Рекомендовано к изданию редакционно-издательским советом СКГА

Протокол №20 от «20» 09.2021г.

**Рецензенты: Эльканова Л.М**.- к. ф-м. н., доцент кафедры «Общая информатика»

Б59  **Биджиева, С.Х. М**етодика преподавания информатики и математики в школе (Часть 2): лабораторный практикум для обучающихся 1 курса по направлению подготовки 09.03.03. Прикладная информатика / С.Х Биджиева. –Черкесск: БИЦ СКГА, 2022.-92с.

Влабораторном практикуме представлен материал для освоения обучающимися дисциплины «Методика преподавания информатики и математики в школе». Практикум состоит из двух частей: в теоретической части представлена теоретическая информация; практическая часть включает задания, направленные на закрепление и расширение знаний студентов в области информационных процессов, кодирования информации, алгоритмизации, программирования, моделирования и информационных технологий.

**УДК 681.3:51:378**

**ББК 32.81:22:74.58**

©  С.Х. Биджиева, 2022

©  ФГБОУ ВО СКГА, 2022

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение | 4 |
| Информация и информационные процессы. Системы счисления | 5 |
| Кодирование числовой информации в компьютере | 14 |
| Кодирование текстовой информации в компьютере | 20 |
| **Кодирование графической и звуковой информации** | 24 |
| Логические операции. Построение таблиц истинности логических функций | 30 |
| Построение функциональных схем | 41 |
| Алгоритмизация и основные технологии программирования | 47 |
| Моделирование | 65 |
| Технология хранения, отбора и сортировки информации. Базы данных | 73 |
| Список литературы | 90 |

**Введение**

Данный лабораторный практикум предназначен для обучающихся по направлению подготовки 09.03.03. Прикладная информатика. Лабораторный практикум составлен на основе рабочей программы по дисциплине «Методика преподавания информатики и математики в школе», в соответствии с требованиями Государственного образовательного стандарта к минимуму содержания и уровню подготовки выпускника по дисциплине.

Цели освоения дисциплины «Методика преподавания информатики и математики в школе»:

* формирование критического мышления и развитие у обучающихся
* прочного интереса к проблемам теории и методики информатики и математики;
* ознакомление с новыми технологиями обучения;
* формирование и развитие практических умений репродуктивного и локально-моделирующего характера на основе рефлексивной предметной деятельности.

Основные задачи дисциплины:

* освоение обучающимися теоретических основ содержания школьного образования по информатике и ИКТ;
* освоение методики преподавания школьных курсов математики и информатики;
* выработка у обучающихся практических навыков проведения учебной работы на уровне требований, предъявляемых реформой общеобразовательной и профессиональной школы;
* формирование навыков самостоятельного процесса обучения.

Знание основных разделов дисциплины способствует повышению эффективности учебной деятельности обучающихся и их будущей профессиональной деятельности, а также положительному восприятию процесса информатизации общества.

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 1**

**Информация и информационные процессы. Системы счисления**

**Цель**: закрепить знания обучающихся об основных понятиях, видах, свойствах и измерении информации; развитие практических навыков преобразования чисел в различные системы счисления.

Теоретическая часть

**Информатика** - это дисциплина, изучающая структуру и свойства информации, закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

Понятие **информация** точно и однозначно не определяется, хотя используется повсеместно. Оно вводится путём объяснения, ко­торое опирается на интуицию, здравый смысл или бытовое примене­ние этого термина.

Основные свойства информации



Актуальность

Доступность (понятность)

Ценность

(достоверность)

Рисунок 1.1

Классификации информации.

1. По форме представления:
* дискретная информация;
* аналоговая информация:

2.По области возникновения выделяют информацию:

* механическую, которая отражает процессы и явления не­одушевленной природы;
* биологическую, которая отражает процессы животного и растительного мира;
* социальную, которая отражает процессы человеческого общества.
1. По способу передачи и восприятия различают следующие виды информации:
* визуальную, передаваемую видимыми образами и символами;
* аудиальную, передаваемую звуками;
* тактильную, передаваемую ощущениями прикосновений;
* органолептическую, передаваемую запахами и вкусами;
* машинную, выдаваемую и воспринимаемую средствами вычислительной техники.

4.Информацию, создаваемую и используемую человеком, по общественному назначению можно разбить на три вида:

* личную, предназначенную для конкретного человека;
* массовую, предназначенную для любого желающего ею пользоваться (общественно-политическая, научно-популярная и т.д.);
* специальную, предназначенную для использования узким кругом лиц, занимающихся решением сложных специальных задач в области науки, техники, экономики.

5.По способам кодирования выделяют следующие типы ин­формации:

* символьную;
* текстовую;
* графическую.

Подход к информации как к мере уменьшения неопределенности наших знаний позволяет количественно измерять информацию, полученную через некоторое сообщение.

Клод Шеннон предложил в 1948 году формулу для определения количества информации, которую мы получаем после получения одного из N возможных сообщений:

**I = — ( p1log2 p1 + p2 log2 p2 + . . . + pN log2 pN)**

Здесь pi - вероятность того, что будет получено именно i-е сообщение. Если все сообщения равновероятны, то все Pi=1/N, и из этой формулы получается формула Хартли:

**I =** $log\_{2}N$

Из формулы Хартли следует: если I=1, то N=2, то есть в качестве единицы измерения информации можно взять тот объём информации, который мы получаем при принятии сигнала о том, что же произошло в ситуации с двумя возможными исходами. Такая единица названа ***битом***.

Наряду с единицей бит иногда используют в качестве единиц информации количества, взятые по логарифмам с другими основаниями: д***ит*** - по десятичному логарифму (за единицу информации выбирается количество информации, необходимой для различения десяти равновероятных сообщений), ***нат*** - по натуральному основанию.

Используя формулу Хартли можно, зная количество информации, пришедшее с одним из равновероятных сообщений, оп­ределить, сколько сообщений вообще можно было ожидать в данной ситуации. Решив это уравнение относительно N, получим при равно­вероятных исходах:

**I =** $log\_{2}N$**=** **N = 21**

При алфавитном подходе к определению количества информации отвлекаются от содержания информации и рассматривают информационное сообщение как последовательность знаков определенной знаковой системы.

В простейшем случае, когда длина кода сообщения составляет один знак, отправитель может послать N разных сообщений. Количество информации I, которое несет каждое сообщение, то есть один знак, можно рассчитать по формуле Хартли.

**I =** $log\_{2}N$

Эта величина называется ***информационной емкостью знака***. С помощью этой формулы можно, например, определить информационную емкость знака двоичной знаковой системы:

**I =** $log\_{2}2$**= 1 бит**

Бит - это минимальная единица измерения количества информации. Первой более крупной, чем бит, единицей измерения информации, выбран ***байт***:

1 байт = 8 бит = 23 бит.

1 Килобайт (Кбайт) = 210 байт = 1024 байт;

1 Мегабайт (Мбайт) = 210 Кбайт = 1024 Кбайт=1 048 576 байт;

1 Гигабайт (Гбайт) = 210 Мбайт = 1024 Мбайт = 1 073 741 824 байт;

1 Терабайт (Тбайт)= 210 Гбайт = 1024 Гбайт = 240 байт;

1 Петабайт (Пбайт) =210 Тбайт =1024 Тбайт;

1 Эксабайт (Эбайт) = 210 Пбайт =1024 Пбайт;

1 Зеттабайт (Збайт) =210 Эбайт =1024 Эбайт;

1 Йоттабайт (Йбайт) =210 Збайт =1024 Збайт и т.д.

В позиционных системах количественное значение цифры зависит от её положения в числе. Обычно при записи числа в позиционных системах используют арабские цифры. Количество цифр, которое используется при этом, называется основанием системы. Оно определяет, во сколько раз различаются количества, соответствующие одинаковым цифрам, стоящим в соседних позициях числа, и указывается нижним индексом после последней цифры числа. Если основание системы, по которой записано число, не указано, по умолчанию считается, что оно равно десяти.

Количество, соответствующее числу, можно представить в виде многочлена по степеням основания. Цифры, из которых составляется число, это коэффициенты, на которые надо умножить соответствующие степени основания. Первая цифра справа - коэффициент при нулевой степени основания. Далее справа налево перечисляются коэффициенты при первой, второй и т. д. степенях. Примеры:

33310 = 3 \* 102 + 3 \* 101 +3 \* 100;

33312 = 3 \* 122 + 3 \*121 + 3 \* 120 = 3 \* 144 + 3 \* 12 + 3 = 47110

1F3D16 = 1 \* 163 + 15 \* 162 + 3 \* 161 + 13 \* 160 = 799710

378 = 3 \* 81 + 7 \* 80 = 3110

01102 = 0 \* 23 +1 \* 22 + 1 \* 21 + 0 \* 20 = 610

1Кб = 210 байт = 100000000002 байт = 102410 байт

Дробная часть числа раскладывается в многочлен по отрицательным степеням основания.

Базовой системой счисления в вычислительной технике является двоичная система. Так как коды чисел и команд в ней слишком длинные, в документации используют более компактную запись по родственным основаниям: в восьмеричной или шестнадцатеричной системе. В восьмеричной системе для записи числа используются цифры 0, 1, 2,.., 7. В шестнадцатеричной системе арабские цифры 0, 1,2,..., *9* дополняются начальными буквами латинского алфавита A,B,C,D,E,F.

Из табл. 1.1 видно, что если добавить слева незначащие ноли, то значение каждой цифры восьмеричной системы можно представить тремя, а шестнадцатеричной - четырьмя цифрами двоичной системы.

Таблица 1.1

Таблица соответствия между начальными двоичными,

восьмеричными, шестнадцатеричными и десятичными числами

|  |  |
| --- | --- |
| Основание системы счисления | Вид числа |
| 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| 8 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| 16 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 10 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 8 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 8 | 9 | A | B | C | D | E | F |

**Правила перевода чисел из одной системы счисления в другую**

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:



При переводе удобно пользоваться таблицей степеней двойки.

Таблица 1.2

Степени числа 2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n (степень) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Описание: Описание: http://inf.e-alekseev.ru/extra/ris10.gif | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 |

Пример . Число 11101000 (2) перевести в десятичную систему счисления.



2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

****

При переводе удобно пользоваться таблицей степеней восьмерки:

Таблица 1.3

 Степени числа 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n (степень) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Описание: Описание: http://inf.e-alekseev.ru/extra/ris8.gif | 1 | 8 | 64 | 512 | 4096 | 32768 | 262144 |

Пример .Число 75013 (8) перевести в десятичную систему счисления.

****

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:



При переводе удобно пользоваться таблицей степеней числа 16:

Таблица 1.4

 Степени числа 16

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n (степень) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Описание: Описание: http://inf.e-alekseev.ru/extra/ris9.gif | 1 | 16 | 256 | 4096 | 65536 | 1048576 | 16777216 |

Пример. Число FDA1 (16) перевести в десятичную систему счисления.



4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 22(10) перевести в двоичную систему счисления.





5. Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или равный 7. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 571 (10) перевести в восьмеричную систему счисления.





6. Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший или равный 15. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 7467(10) перевести в шестнадцатеричную систему счисления.



7. Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой.

Пример. Число 1001011 (2)  перевести в восьмеричную систему счисления.



8. Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей восьмеричной цифрой.

Пример. Число 1011100011 (2)  перевести в шестнадцатеричную систему счисления.



9. Для перевода восьмеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число 531 (8) перевести в двоичную систему счисления.



10. Для перевода шестнадцатеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число ЕЕ8 (16) перевести в двоичную систему счисления.



11. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число FEA (16) перевести в восьмеричную систему счисления.



Пример 2. Число 6635(8) перевести в шестнадцатеричную систему счисления.



Практическая часть

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

1. 948;
2. 763;
3. 994,125;
4. 523,25;
5. 203,82.
6. 563;
7. 264;
8. 234,25;
9. 53,125;
10. 286,16.
11. 279;
12. 281;
13. 841,375;
14. 800,3125;
15. 208,92.
16. 737;
17. 92;
18. 934,25;
19. 413,5625;
20. 100,94.

Задание 2. Переведите числа в десятичную систему счисления.

1. 10001112;
2. 1000110112;
3. 1001100101,10012;
4. 1001001,0112;
5. 335,78;
6. 14C,A16.
7. 11000100102;
8. 100110112;
9. 1111000001,012;
10. 10110111,012;
11. 416,18;
12. 215,7116.
13. 11001110012;
14. 100111012;
15. 1111011,0012;
16. 110000101,012;
17. 1601,568;
18. 16E,B416.
19. 665,428;
20. 246,1816.

Задание 3. Выполните сложение чисел.

1. 11101010102+101110012;
2. 101110102+100101002;
3. 111101110,10112+1111011110, 12;
4. 1153,28+1147,328;
5. 40F,416+160,416.
6. 101111112+1100100002;
7. 1100101002+10111000012;
8. 1000000101,01012+1010000110,012;
9. 1512,48+1015,28;
10. 274,516+DD,416
11. 10001000012+10111001112;
12. 11011100112+1110001012;
13. 1011011,012+1000101110,10012;
14. 665,18+1217,28;
15. 30C,716+2А1,816;
16. 111101002+1101000012;
17. 11011102+1010010002;
18. 1100110011, 12+111000011,1012;
19. 1455,048+203,38;
20. 14Е,816+184,316.

Задание 4. Выполните вычитание чисел.

1. 10000001002-1010100012;
2. 10101111012-1110000102;
3. 1101000000,012-1001011010,0112;
4. 2023,58-527,48;
5. 25E,616-1B1,516.
6. 10000010012-1111101002;
7. 11110001012-11001101012;
8. 1100110101, 12-1011100011,012;
9. 1501,348-1374,58;
10. 12D,316-39,616.
11. 111100102-101010012;
12. 11101000012-10110010012;
13. 1101001010, 12-1011101001,110112;
14. 166,148-143,28;
15. 287,А16-62,816;
16. 10000101012-1001010002;
17. 10010110112-1010011102;
18. 111111011,1012-100000010,012;
19. 341,28-275,28;
20. 249,516-62, А16.

Задание 5. Выполните умножение чисел.

1. 10010112\*10101102;
2. 1650,28\*120,28;
3. 19,416\*2F,816.
4. 1111012\*10101112;
5. 1252,148\*76,048;
6. 66,6816\*1E,316.
7. 10010012\*1000102;
8. 324,28\*122,128;
9. F,416\*38,616.
10. 10010002\*10100112;
11. 412,58\*13,18;
12. 3B,A16\*10,416
13. 10000101012\*1001010002;
14. 10010110112\*1010011102;
15. 111111011,1012\*100000010,012;
16. 341,28\*275,28;
17. 249,516\*ЕЕ, А16.
18. 1512,48\*1015,28;
19. 274,516\*DD,416;
20. 11011102+1010010002.

ЛАБОРАТОРНОЕ ЗАНЯТИЕ 2

Кодирование числовой информации в компьютере

**Цель**: сформировать понимание процесса кодирования числовой информации; закрепить навыки сложения и вычитания в прямом, обратном и дополнительном кодах.

Теоретическая часть

Кодирование числовой информации в компьютере

Прямой, обратный и дополнительный коды двоичного числа - это способы представления двоичных чисел с фиксированной запятой в компьютерной (микроконтроллерной) арифметике, предназначенные для записи отрицательных и неотрицательных чисел.

**Прямой код**

Прямой код - способ представления двоичных чисел с фиксированной запятой. Главным образом используется для записи неотрицательных чисел.

Прямой код используется в двух вариантах.

В первом (основной) - для записи только неотрицательных чисел (таблица 2.1).

Таблица 2.1

|  |  |
| --- | --- |
| **Десятичное число** | **Двоичное число в прямом коде****(в 8-битном представлении)** |
| 0 | 0000 0000 |
| 10 | 0000 1010 |
| 100 | 0110 0100 |
| 255 | 1111 1111 |

Прямой код используется главным образом для представления неотрицательных чисел.

Использование прямого кода для представления отрицательных чисел является неэффективным - очень сложно реализовать арифметические операции и, кроме того, в прямом коде два представления нуля - положительный ноль и отрицательный ноль (чего не бывает).

**Обратный код**

Обратный код - метод вычислительной математики, позволяющий вычесть одно число из другого, используя только операцию сложения.

Обратный двоичный код положительного числа состоит из одноразрядного кода знака (битового знака) - двоичной цифры 0, за которым следует значение числа.

Обратный двоичный код отрицательного числа состоит из одноразрядного кода знака (битового знака) - двоичной цифры 1, за которым следует инвертированное значение положительного числа.

Для неотрицательных чисел обратный код двоичного числа имеет тот же вид, что и запись неотрицательного числа в прямом коде.

Для отрицательных чисел обратный код получается из неотрицательного числа в прямом коде, путем инвертирования всех битов (1 меняем на 0, а 0 меняем на 1).

Для преобразования отрицательного числа записанное в обратном коде в положительное достаточного его проинвертировать.

При 8-битном двоичном числе - знаковый бит (как и в прямом коде) старший (8-й).

Таблица 2.2

**Диапазон десятичных чисел, который можно записать в обратном коде от -127 до + 127**

|  |  |  |  |
| --- | --- | --- | --- |
| **Положительное десятичное число** | **Двоичное число в обратном коде** | **Отрицательное десятичное число** | **Двоичное число в обратном коде** |
| 0 | 0000 0000 | - 0 | 1111 1111 |
| 10 | 0000 1010 | - 10 | 1111 0101 |
| 100 | 0110 0100 | - 100 |  1001 1011 |
| 127 | 0111 1111 | - 127 | 1000 0000 |

**Арифметические операции с отрицательными числами в обратном коде**

[**(Арифметические операции с двоичными числами)**](https://microkontroller.ru/programmirovanie-mikrokontrollerov-avr/dvoichnaya-i-shestnadtsatirichnaya-sistemyi-schisleniya/)

**1-й пример (для положительного результата)**

Даны два числа:

100 = 0110 0100

-25 = - 0001 1001

Необходимо их сложить:

100 + (-25) = 100 - 25 = 75

**1-й этап**

Переводим число -25 в двоичное число в обратном коде:

25 = 0001 1001

-25= 1110 0110

и складываем два числа:

0110 0100 (100) + 1110 0110 (-25) = 1 0100 1010, отбрасываем старшую 1 (у нас получился лишний 9-й разряд - переполнение), = 0100 1010

**2-й этап**

Отброшенную в результате старшую единицу прибавляем к результату:

0100 1010 + 1 = 0100 1011 (знаковый бит =0, значит число положительное), что равно 75 в десятичной системе

**2-й пример (для отрицательного результата)**

Даны два числа:

5 = 0000 0101

-10 = - 0000 1010

Необходимо их сложить:

5 + (-10) = 5 - 10 = -5

**1-й этап**

Переводим число -10 в двоичное число в обратном коде:

10 = 0000 1010

-10= 1111 0101

и складываем два числа:

0000 0101 (5) + 1111 0101 (-10) = 1111 1010 (знаковый бит =1, значит число отрицательное)

**2-й этап**

Раз результат получился отрицательный, значит число представлено в обратном коде. Переводим результат в прямой код (путем инвертирования значения, знаковый бит не трогаем):1111 1010 ----> 1000 0101

Таблица 2.3

|  |  |
| --- | --- |
| **Десятичные числа** | **Двоичные числа (8 бит)** |
| **Прямой код** | **Обратный код** | **Дополнительный** |
| 127 | 0111 1111 | 0111 1111 | 0111 1111 |
| 1 | 0000 0001 | 0000 0001 | 0000 0001 |
| 0 | 0000 0000 | 0000 0000 | 0000 0000 |
| -0 | 1000 0000 | 1111 1111 |  |
| -1 | 1000 0001 | 1111 1110 | 1111 1111 |
| -2 | 1000 0010 | 1111 1101 | 1111 1110 |
| 10 | 0000 1010 | 0000 1010 | 0000 1010 |
| -10 | 1000 1010 | 1111 0101 | 1111 0110 |
| -127 | 1111 1111 | 1000 0000 | 1000 0001 |

Дополнительный код отрицательного числа можно получить двумя способами.

**1-й способ**.

Инвертируем значение отрицательного числа, записанного в прямом коде (знаковый бит не трогаем) - к полученной инверсии прибавляем 1.

Пример: Дано десятичное число -10.

Переводим в прямой код:10 = 0000 1010 ----> -10 = 1000 1010

Инвертируем значение (получаем обратный код):1000 1010 ----> 1111 0101

К полученной инверсии прибавляем 1:1111 0101 + 1 = 1111 0110 - десятичное число -10 в дополнительном коде

**2-й способ.**

Вычитание числа из нуля

Дано десятичное число 10, необходимо получить отрицательное число (-10) в дополнительном двоичном коде

Переводим 10 в двоичное число:

10 = 0000 1010

Вычитаем из нуля:

0 - 0000 1010 = 1111 0110 - десятичное число -10 в дополнительном коде.

**Арифметические операции с отрицательными числами в дополнительном коде**

Дано: необходимо сложить два числа -10 и 5

-10 + 5 = -5

Решение:

5 = 0000 0101-10 = 1111 0110 (в дополнительном коде)

Складываем:

1111 0110 + 0000 0101 = 1111 1011, что соответствует числу -5 в дополнительном коде

Как мы видим на этом примере - дополнительный код отрицательного двоичного числа наиболее подходит для выполнения арифметических операций сложения и вычитания отрицательных чисел.

Таким образом, для арифметических операций сложения и вычитания положительных двоичных чисел наиболее подходит применение прямого кода, для арифметических операций сложения и вычитания отрицательных двоичных чисел наиболее подходит применение дополнительного кода.

В формате с плавающей запятой представляются вещественные числа (предполагается, что они могут содержать дробную часть). В этом формате число заносится в память компьютера в экспоненциальной форме, то есть в виде двух сомножителей: мантиссы (дроби, в которой первая значащая цифра стоит сразу после запятой) и основания системы счисления в соответствующей степени (порядке). Примеры для десятичной системы счисления приведены в таблице 2.4.

Таблица 2.4

**Примеры перевода вещественных чисел в экспоненциальную форму**

|  |  |  |  |
| --- | --- | --- | --- |
| Вещественное число | Экспоненциальная форма | Мантисса | Порядок |
| 98567 | 0,98567\*105 | 0,98567 | 5 |
| - 98567 | - 0,98567\*105 | -0,98567 | 5 |
| 98,567 | 0,98567\*102 | 0,98567 | 2 |
| - 0,0009856 | - 0,9856\*10-3 | - 0,9856 | -3 |

В байтах, отведенных для записи числа, выделяются определенные разряды для хранения всех фрагментов числа: знаков мантиссы и порядка, их абсолютных значений. Пример (код максимального положительного числа)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 1 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Знак и значение порядка | Знак и значение мантиссы |

**Модифицированные обратный и дополнительный коды**

Переполнение разрядной сетки может привести к переносу единицы в знаковый разряд, что приведет к неправильному результату. Положительное число, получившееся в результате арифметической операции, может восприниматься как отрицательное, так как в знаковом разряде появится «1», и наоборот. Например:

Х= 0,1011110

Y= 0.1101100

X + Y= 1,1001010

Х и Y — коды положительных чисел, но в процессе сложения в знаковом разряде появилась «1», что означает код отрицательного числа. Чтобы распознать переполнение разрядной сетки, вводятся модифицированные коды.

 Модифицированный обратный код характеризуется тем, что под знак числа отводится не один, а два разряда. Форма записи чисел в модифицированном обратном коде выглядит следующим образом:

• для положительного числа

Х = Хп Хп-1 ... Х2 Х1 Х0 ... => Xмодобр = 00,ХпХп-1...Х2Х1Х0 ;

• для отрицательного числа

Х = Хп Хп-1 ... Х2 Х1 Х0 ... => Xмодобр = 11, Х ̅пХ ̅п-1... Х ̅2Х ̅1Х ̅0 ;

(XХ ̅ — обозначение логической операции отрицания «не X», если

Х= 0, то Х ̅ =1; Х= 1, Х ̅ = 0).

В модифицированных обратном и дополнительном коде под знак числа отводится не один, а два разряда: «00» соответствует знаку «плюс», «11» — знаку «минус». Любая другая комбинация («01» или «10»), получившаяся в знаковых разрядах, является признаком переполнения разрядной сетки. Сложение чисел в модифицированных кодах ничем, не отличается от сложения в обычных обратном и дополнительном кодах.

Практическая часть

**Задание 1**. Запишите следующие числа в прямом, обратном и дополнительном кодах.

а) 1101011; б) -101011; в) -101101; г) -1100111.

**Задание 2.** Запишите числа в прямом, обратном и дополнительном коде.

1. 27, -39;
2. 30, -10;
3. 45, -5;
4. -99, -1;
5. -88,-2;
6. -17, -13;
7. -22,20;
8. -33,-17;
9. 34, 3;
10. 9,-11.

**Задание 3**. Сложите числа в прямом, обратном и дополнительном коде.

1. -27, -39;
2. 30, -10;
3. 45, -5;
4. -99, -1;
5. -88,-2;
6. -17, -13;
7. -22,20;
8. -33,-17;
9. 34, 3;
10. 9,-11.

**Задание 4**. Переведите числа *X* и *Y* в прямой, обратный и дополнительный коды. Выполните сложение в обратном и дополнительном кодах. Результат переведите в прямой код. Полученный результат проверьте, используя правила двоичной арифметики.

а) *X* = -11010; Y= 100111;

б) *X*=-11101; Y = -10011;

в) *X*=111010; *Y* = -10111;

г) Х = -101110; Y = -11101;

д) Х= 1101011; Y = -1001110;

е) *X*=-11011; *Y*=-10111.

**Задание 5.** Сложите числа X и Y в модифицированном обратном и модифицированном дополнительном восьмиразрядных кодах. При обнаружении переполнения увеличьте число разрядов в кодах и повторите суммирование. Результат переведите в прямой код. Полученный результат проверьте, используя правила двоичной арифметики.

1. Х= 1101011;  *Y* = -1001110;
2. Х= 1001111;  *Y* = -1011010;
3. Х = -101110; Y = -11101;
4. Х = -1010101; Y = -10001;
5. *X*=1110110;Y = -11111;
6. *X*=111010;Y = -10111;
7. *X*=-11101; Y = -10011;
8. *X*=-11101; Y = -10011;
9. *X*=-11011; *Y*=-10111;
10. *X*=101110;Y = -10100;
11. *X*=-10101; Y = -10010;
12. *X*=-10010; *Y*=-10011;
13. *X* = -110100; Y= 101110;
14. *X* = -11010; Y= 100111;
15. *X* = -10010; Y= 101111;

**Задание 6.** Запишите числа –N\*2+12, N+18, где N – номер варианта обучающегося, в прямом коде.

**Задание 7.** Запишите числа –N\*2+12, N+18, где N – номер варианта обучающегося, в обратном коде.

**Задание 8.** Запишите числа –N\*2+12, N+18, где N – номер варианта обучающегося, в дополнительном коде.

ЛАБОРАТОРНОЕ ЗАНЯТИЕ 3

Кодирование текстовой информации в компьютере

**Цель**: сформировать понимание процесса кодирования текстовой информации; показать различные виды кодирования текстовой информации.

Теоретическая часть

Компьютерная обработка текстовой информации начала ис­пользоваться с середины 60-х годов. Помимо преимуществ, которые появляются при автоматическом внесении текстовых комментариев в результаты расчетных программ, создание, обработка и хранение текстовых документов в файловом виде представляет массу удобств.

При кодировании текста в память последовательно заносятся коды символов, составляющих текст, и команд, управляющих внешним видом и размещением этих символов. То есть если мы определяем числа 69 и 96 как текстовую информацию, коды этих чисел будут отличаться только порядком следования кодов цифр 6 и 9. Если же мы определяем их как числовую информацию, их коды будут совершенно различны, так как они представляют разные по величине числа.

Первоначально для кодирования одного символа использовался 1байт. В байт можно записать в 28 = 256 разных кодов (состояний). Эти состояния перенумерованы, и каждому сопоставляется какой-либо буквенный символ, графический элемент или команда, необходимая при оформлении текстовой информации. Такое соответствие называется кодовой таблицей.

В настоящее время существуют и применяются разные варианты 8-битных кодовых таблиц. Наиболее популярные из них:

* *ASCII* - American Standart Code for Information Interchange - американский стандартный код для обмена информацией;
* *КОИ8-Р* - Код Обмена Информацией 8-битный с кириллицей;
* *CP1251* - (Code Page) - кодировка с кириллицей в Microsoft Windows;
* *CP866* - кодировка MSDOS;

*ISO 8859****-5*** - International Standards Organization - Международная организация по стандартизации. Ещё один стандарт для кодов для кириллицы.

Множество кодовых таблиц вызвано тем, что с учетом разнообразия естественных языков и фирм, выпускающих программное обеспечение, 256 состояний одного байта недостаточно для того, чтобы закодировать все встречающиеся символы и способы форматирования текста.

При разработке всех кодовых таблиц использовано следующее соглашение: первая половина таблицы - это коды с 0 по 127 - интер­национальна, т. е одинакова во всех вариантах кодировок. Первые 33 состояния (0-32) - это коды операций с текстом (перевод на новую строку, пробел, удаление последнего символа и т. п.). Затем состояния с 33 по 127 - это коды знаков препинания, арифметических действий, цифр, строчных и прописных букв латинского алфавита. Вторая половина кодовых таблиц отводится под знаки национальных и специальных алфавитов и ввода в текст графических элементов для оформления таблиц.

Ограниченность 8-битной кодировки, не позволяющей одновременно пользоваться несколькими языками, а также трудности, связанные с необходимостью преобразования одной кодировки в другую, привели к разработке нового кода. В 1991 году был разработан новый стандарт кодирования символов, получивший название Unicode (Юникод), позволяющий использовать в текстах любые символы любых языков мира.

Unicode — это «уникальный код для любого символа, независимо от платформы, независимо от программы, независимо от языка»

В Unicode на кодирование символов отводится 31 бит. Первые 128 символов (коды 0-127) совпадают с таблицей ASCII. Далее размещены основные алфавиты современных языков: они полностью умещаются в первой части таблицы, их коды не превосходят 65 536 = 216.

Стандарт Unicode описывает алфавиты всех известных, в том числе и «мёртвых», языков. Для языков, имеющих несколько алфавитов или вариантов написания (например, японского и индийского), закодированы все варианты.

Всего современная версия Unicode позволяет закодировать более миллиона различных знаков, но реально используется чуть менее 110 000 кодовых позиций.

Для представления символов в памяти компьютера в стандарте Unicode имеется несколько кодировок.

В операционных системах семейства Windows используется кодировка UTF-16. В ней все наиболее важные символы кодируются с помощью 2 байт (16 бит), а редко используемые — с помощью 4 байт.

В операционной системе Linux применяется кодировка UTF-8, в которой символы могут занимать от 1 (символы, входящие в таблицу ASCII) до 4 байт. Если значительную часть текста составляют цифры и латинские буквы, то это позволяет в несколько раз уменьшить размер файла по сравнению с кодировкой UTF-16.

Кодировки Unicode позволяют включать в один документ символы самых разных языков, но их использование ведёт к увеличению размеров текстовых файлов.

**Практическая часть**

**Задание 1**.Закодируйте следующие слова, используя таблицы ASCII-кодов:

1. Информатизация, Микропроцессор
2. Моделирование, Мультимедиа
3. Разрядность, Массовость
4. Достоверность, Своевременность
5. Форматирование, Фильтрация
6. Тестирование, Сканирование
7. Результативность, Детерминированность
8. Программирование, Маршрутизатор
9. Криптография, Контроллер
10. Компьютеризация, Гиперссылка
11. Винчестер, Архитектура
12. Антивирус, Конъюнктур
13. Объективность, Компьютер
14. Майнфрейм, Компьютер
15. Технология, Адекватность

**Задание** 2.Раскодируйте следующие слова, используя таблицы ASCII-кодов:

1. 208 152 208 189 209 132 208 190 209 128 208 188 208 176 209 130 208 184 208 183 208 176 209 134 208 184 209 143
2. 208 156 208 184 208 186 209 128 208 190 208 191 209 128 208 190 209 134 208 181 209 129 209 129 208 190 209 128
3. 208 156 208 190 208 180 208 181 208 187 208 184 209 128 208 190 208 178 208 176 208 189 208 184 208 181
4. 208 156 209 131 208 187 209 140 209 130 208 184 208 188 208 181 208 180 208 184 208 176
5. 208 160 208 176 208 183 209 128 209 143 208 180 208 189 208 190 209 129 209 130 209 140
6. 208 156 208 176 209 129 209 129 208 190 208 178 208 190 209 129 209 130 209 140
7. 208 148 208 190 209 129 209 130 208 190 208 178 208 181 209 128 208 189 208 190 209 129 209 130 209 140
8. 208 161 208 178 208 190 208 181 208 178 209 128 208 181 208 188 208 181 208 189 208 189 208 190 209 129 209 130 209 140
9. 208 164 208 190 209 128 208 188 208 176 209 130 208 184 209 128 208 190 208 178 208 176 208 189 208 184 208 181
10. 208 164 208 184 208 187 209 140 209 130 209 128 208 176 209 134 208 184 209 143
11. 208 162 208 181 209 129 209 130 208 184 209 128 208 190 208 178 208 176 208 189 208 184 208 181
12. 67 208 186 208 176 208 189 208 184 209 128 208 190 208 178 208 176 208 189 208 184 208 181
13. 208 162 208 181 209 133 208 189 208 190 208 187 208 190 208 179 208 184 209 143
14. 208 160 208 181 208 183 209 131 208 187 209 140 209 130 208 176 209 130 208 184 208 178 208 189 208 190 209 129 209 130 209 140
15. 208 148 208 181 209 130 208 181 209 128 208 188 208 184 208 189 208 184 209 128 208 190 208 178 208 176 208 189 208 189 208 190 209 129 209 130 209 140

**Задание** 3.Открыть текстовый редактор Блокнот.

а) Используя клавишу Alt и малую цифровую клавиатуру раскодировать фразу:

145 170 174 224 174 255 170 160 173 168 170 227 171 235;

*Методические указания.*

*При удерживаемой клавише Alt, набрать на малой цифровой клавиатуре указанные цифры. Отпустить клавишу Alt, после чего в тексте появится буква, закодированная набранным кодом.*

б) Используя ключ к кодированию, закодировать слово  – ***алгоритмΚΚΚо***;

*Методические указания.*

*Из предыдущего задания выяснить, каким кодом записана буква а. Учитывая, что буквы кодируются в алфавитном порядке, выяснить коды остальных букв*.

**Задание** 4. Открыть текстовый процессор  MS Word.

1. Закодировать слово Forest
2. Выбрать шрифт Courier New, кодировку ASCII(дес.)
3. Выбрать шрифт Courier New, кодировку Юникод(шест.)
4. Выбрать шрифт Times New Roman, кодировку Кирилица(дес.)
5. Выбрать шрифт Times New Roman, кодировку ASCII(дес.)

*Методические указания.*

*Рассмотрим на примере: представить в различных кодировках слово*Кодировка*.*

*Решение:*

 *Создать новый текстовый документ в Word.*

1. *Выбрать  – Команда –  Вставка – Символ.*
2. *В открывшемся окне «Символ» установить из Юникод (шестн.).*
3. *В наборе символов находим букву*К*и щелкнем на ней левой кнопкой мыши.*
4. *В строке код знака  появится код выбранной буквы* 041А *(незначащие нули тоже записываем).*
5. *У буквы*о*код –* 043Е *и так далее:* д – 0434, и – 0438, р – 0440, о – 043Е, в – 0432, к – 043А, а – 0430*.*
6. *Установить Кириллица (дес.)*
7. К – 0202, о – 0238, д – 0228, и – 0232, р – 0240, о – 0238, в –0226, к – 0202, а –0224*.*

**Задание** 5. Буква Z  имеет десятичный код 90, а z – 122. Записать в десятичном коде слово:

1. laptop
2. keyboard
3. monitor
4. drive
5. motherboard
6. modem
7. software
8. router
9. memory
10. display
11. mouse
12. copier
13. overload
14. window
15. account

**Задание** 7. Записать последовательность десятичных кодов для слова своего варианта из предыдущего задания, но записанного заглавными буквами.

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 4**

**Кодирование графической и звуковой информации**

**Цель**: сформировать понимание процесса кодирования графической и звуковой информации; выявить преимущества двоичного кодирования информации.

**Теоретическая часть**

Графическая информация в докомпьютерную эпоху регистрировалась и воспроизводилась в аналоговой форме. Чертежи, рисунки создавались с помощью сплошных линий и мазков разной величины и цвета.

Дискретное представление графики получается за счет того, что экран монитора разбивается на строки и колонки. Совокупность получившихся клеточек (точек) называется растром, каждая точка - пикселем. Количество строк и колонок в растре - это разрешение (разрешающая способность) экрана. Типовые разрешения: 1024X768 , 800X600 пикселей. Первым указывается количество колонок, вторым - количество строк в растре.

Для монохромных изображений общепринятым считается кодирование цвета одного пикселя в 1байте. Это позволяет передать 254 оттенка серого плюс черный и белый цвета (всего 256 вариантов). Цветные изображения могут кодироваться разными способами в зависимости от того, для какой цели создаётся рисунок.

Метод (система) RGB (True color) - от слов Red, Green, Blue удобен для изображений, рассматриваемых на экране, выводи­мых на устройство записи на киноплёнку. Оттенки цвета создаются смешением лучей трёх базовых цветов разной интенсивности. Под значение интенсивности каждого луча отводится 1 байт, т. е. различают 256 уровней интенсивности. Для совокупности трёх лучей получается 2563=16 777 216^17 млн. разных вариантов, каж­дый из которых создает свой оттенок цвета.

Метод (система) CMYK - от слов голубой (Cyan), пурпурный (Magenta), жёлтый (Yellow), чёрный (black): удобен для изображе­ний, которые предполагается печатать на бумаге. Он учитывает осо­бенности полиграфии, в которой цвет получается смешением четырёх красок. Для кодирования одного пикселя требуется 4 байта и можно передать 2564-4 млрд. оттенков.

Для WEB-документов учитывать такое обилие оттенков не­удобно, так как это приводит к файлам очень больших размеров, и их неудобно пересылать по сети. Поэтому в них используются так назы­ваемые индексированные цвета: из всего обилия возможных комби­наций выбрано 256 базовых оттенков. Это позволило для запомина­ния цвета каждого пикселя использовать только 1 байт. Каждому со­стоянию байта сопоставляется определенная комбинация интенсив­ностей базовых цветов. \*.JPG, \*.GIF, \*.PNG- кодировки объединяют области рисунка, закрашенные близкими оттенками, и сохраняют для них усреднённый цвет. За счет этого размеры графических файлов существенно уменьшаются.

В компьютерной документации коды RGB- и CMYK-цвета представляются так:

1. Интенсивности каждого базового цвета перечисляются в том порядке, который использован в аббревиатуре используемой сис­темы. Эти интенсивности представляются в 16-ричной позиционной системе
2. Перед полученным кодом оттенка размещают символ #.

Примеры составления кодов для некоторых цветов:

* #000000 - чёрный цвет - нулевая интенсивность каждого луча, т. е. нет ни одного цвета;
* #B5B5B5 - какой-то оттенок серого цвета, т. к. интенсивности всех лучей одинаковы;
* #FFFFFF - белый цвет - все цвета в максимальной интенсив­ности;
* #CF35D1 - номера интенсивностей базовых лучей в десятич­ной системе: красного CF = 12 \*16 + 15 = 207; зелёного - 35 =3 \* 16 + 5 = 53;
* синего - D1 = 13 \*16 + 1 = 209.

Звук - это колебания физической среды. В повседневной жизни такой средой является воздух. Чаще всего звуковые колебания преобразуют в электрические с помощью микрофона. Представление о форме этого сигнала можно получить через программу Windows Player.

Звуковой (аудио) сигнал имеет аналоговую природу. Для того чтобы преобразовать его в дискретную форму используют специальный блок, входящий в состав звуковой карты компьютера, АЦП (аналого-цифровой преобразователь). Основной принцип его работы за­ключается в том, что интенсивность звукового сигнала фиксируется не непрерывно, а периодически, в определенные моменты времени. Частоту, характеризующую периодичность измерения, называют частотой дискретизации. Считается, что для хорошего воспроизведения звука она должна, по крайней мере, в два раза превышать максимальную частоту волны, входящей в спектр звукового сигнала. Человеческое ухо воспринимает как звук колебания в диапазоне частот до 22 000 Гц. Следовательно, для хорошего воспроизведения музыки частота дискретизации должна быть не менее 44 000 Гц. При записи речи такое высокое качество воспроизведения не нужно. Определено, что речь воспринимается вполне разборчиво уже при частоте дискретизации 8 000 Гц.

Помимо дискретизации по времени АЦП проводит дискретизацию и по интенсивности звука, т. е. по амплитуде звукового сигнала. В АЦП закладывается сетка стандартных интенсивностей - глубина кодирования (256 или 65 536 уровней), и реальная интенсив­ность округляется до уровня, ближайшего по сетке.

Обратное преобразование закодированного таким образом звука в аналоговую форму, воспринимаемую человеческим ухом, производится блоком ЦАП (цифро-аналоговый преобразователь). По закодированным точкам время-интенсивность с помощью интерпо­ляции рассчитывается гладкая непрерывная кривая, которая исполь­зуется при восстановлении звукового сигнала. Для проведения расче­тов, восстанавливающих вид звукового сигнала, выпускаются спе­циализированные микропроцессоры, DSP (Digital Signal Processor).

Для определения объема графической информации используются те же формулы, что и для измерения текстовой информации, только обозначения изменяют свой смысл:

**2i= N  I = K\*i,**

где  i – битовая глубина (информационный объем одного пикселя)

       N – количество цветов палитры

       К – разрешение экрана

       I – информационный объем видеофайла или видеопамяти

**Решение задач на измерение графической информации**

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

**Дано:**                   **Решение:**  2i = N  i = 1

К = 640×200          I = K × i = 640 × 200 × 1 = 128000 бит = 16000 байт ≈ 16 Кб

N = 2

Найти:

I -?

**Ответ**: необходим объем 16 Кбайт.

**Для определения объема звуковой информации используется следующая  формула**:

**I = K\*i\*t\*n,**

где  i – глубина кодирования звука (разрядность регистра аудиоадаптера, в битах)

       N – количество различных уровней звукового сигнала

       К – частота дискретизации аналогового звукового устройства

       t – время записи в секундах

       I – информационный объем аудиофайла

       n – количество каналов (стереозапись – 2, монозапись – 1)

**Решение задач на измерение звуковой информации**

На студии при четырехканальной (квадро) звукозаписи с 32-битным разрешением за 30 секунд был записан звуковой файл. Сжатие данных не производилось. Известно, что размер файла оказался 7500 Кбайт.

С какой частотой дискретизации (в кГц) велась запись? В качестве ответа укажите только число, единицы измерения указывать не нужно.

**Решение**

По формуле объема звукового файла получим:

I = К \* t \* i \* n

I= 7500 Кбайт

К= 32 бита

t= 30 секунд

n= 4 канала

ƒ — частота дискретизации — неизвестна, выразим ее из формулы:

ƒ=I n∗K∗t=7500∗210∗23бит27∗30Гц=750∗261000КГц=24=16

24 = 16 КГц

Результат: 16

**Ответ:16**

**Практическая часть**

**Задание 1**. Определить установленное на вашем компьютере разрешение экрана монитора, измеренное в dpi. Варианты выполнения работы:

* использование мониторов различного размера;
* использование различных разрешений экрана монитора.

1. В операционной системе Windows щелкнуть правой кнопкой мыши по Рабочему столу, появится диалоговое окно Свойства: Экран. Выбрать вкладку Параметры и с помощью ползунка Разрешение экрана узнать установленное разрешение экрана монитора в количестве точек по горизонтали и по вертикали. Разрешение по горизонтали = 1024 точки.



2. Измерить с помощью линейки размер изображения на экране монитора по горизонтали (например, для 17"-монитора L — 31,5 см).

3. Определить, чему равен горизонтальный размер изображения на экране монитора в дюймах: L — 31,5 см = 31,5 см/2,54 см/дюйм  12,4 дюйма.

4. Определить разрешение экрана монитора в dpi: Разрешение по горизонтали в dpi = 1024 точки / 12,4 дюйма а 82,5 dpi.

**Задание 2**. Свойства документа

Открыть любой документ и по примеру расписать свой документ, измерить его параметры. Записать в блокноте.

Размер файла: 0.16 Mb; Страниц определено: 6

Тип: DOC; ID: ru-16207

Страниц теоретически по объему текста (в стандартном форматировании): 6

Название документа: Базовый курс Информатика и ИКТ основная школа

Поисковое название: Информатика 9 практическая работа 1 кодирование графической информации

Автор: comp N 1

Компания: школа 6

Время добавления документа в каталог: 21-09-2011 // 01:49:05

Слов: 2626

Символов в тексте: 21877

Строк: 124

Количество параграфов: 35

Символов вместе с пробелами: 17564

**Задание 3**.Определить требуемый объем видеопамяти для различных графических режимов экрана монитора, если известна глубина цвета на одну точку.

|  |  |
| --- | --- |
| Режим экрана | Глубина цвета (бит на точку) |
| 4 | 8 | 16 | 24 | 32 |
| 640 на 480 |  |  |  |  |  |
| 800 на 600 |  |  |  |  |  |
| 1024 на 768 |  |  |  |  |  |
| 1280 на 1024 |  |  |  |  |  |

**Задание 4.** Черно-белое (без градаций серого) растровое графическое изображение имеет размер 10×10 точек. Какой объем памяти займет это изображение?

**Задание 5**.Для хранения растрового изображения размером 128×128 пикселей отвели 4 КБ памяти. Каково максимально возможное число цветов в палитре изображения?

**Задание 6**.Сколько бит видеопамяти занимает информация об одном пикселе на ч/б экране (без полутонов)?

**Задание 7**.Определить объем видеопамяти компьютера, который необходим для реализации графического режима монитора High Color с разрешающей способностью 1024 × 768 точек и палитрой цветов из 65536 цветов***.***

**Задание 8**. 3. Монитор позволяет получать на экране 1024 различного цвета. Сколько бит памяти занимает один пиксель?

**Задание 9**.

Какой объем видеопамяти необходим для хранения двух страниц изображения при условии, что разрешающая способность дисплея равна 640350 пикселей, а количество используемых цветов – 16?

**Задание 10.**

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.

* 1. 128
	2. 512
	3. 1024
	4. 2048

**Задание 11.**

Объем видеопамяти равен 1Мбайт. Разрешающая способность дисплея – 800 х 600. Какое максимальное количество цветов можно использовать при условии, что видеопамять делится на две страницы?

**Задание 12.**

Две минуты записи цифрового аудиофайла занимают на диске 5,1 Мбайт. Частота дискретизации – 22050 Гц. Какова глубина кодирования звука?

**Задание 13.**

Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 32-битным разрешением. Запись длится 4 минуты, ее результаты записываются в файл, сжатие данных не производится. Определить размер файла в Мегабайтах.

**Задание 14.**

Производилась двухканальная (стерео) звукозапись с частотой дискретизации 64 кГц и 24-битным разрешением. В результате был получен файл размером 72 Мбайт, сжатие данных не производилось. Определите приблизительно, сколько времени (в минутах) проводилась запись. В качестве ответа укажите ближайшее к времени записи целое число.

**Задание 15.**

Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 15 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 2 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 5**

**Логические операции. Построение таблиц истинности логических функций**

**Цель:** дать представление об основных логических операциях; сформировать навыки построения таблиц истинности.

**Теоретическая часть**

Высказывание – это повествовательное предложение, про которое можно определенно сказать истинно оно или ложно.
Логические операции – мыслительные действия, результатом которых
является изменение содержания или объема понятий, а также образование новых понятий.

Таблица 5.1

**Основные логические операции (иерархия сверху вниз)**

|  |  |  |
| --- | --- | --- |
| Название | Обозначение | Результат |
| Отрицание, инверсия (связка «не») | А, |А | А = 0—> ~|А = 1А = 1 —> ~|А = 0 |
| Конъюнкция, логическое умножение (связ­ка «и») | А • В, А & В, А и В | А = 1, В = 1 —> А • В =1, в остальных случаях - = 0 |
| Дизъюнкция, логическое сложение (связка «или») | А v В, А + В | А = 0, B = 0—> AvB =0, в остальных случаях -= 1 |
| Импликация (связки «если..., то»,«из... следует», ... влечёт.» | В | А = 1, В = 0, то А —> В =0, в остальных случаях =1 |
| Эквиваленция, двойная импликация (связки «тогда и только тогда», «необходимо и достаточно», «равносильно», | А - В; А о В | А= 1,В= 1 —>А~В = 1А = 0, В = 0 —> А - В = 1А = 1, В = 0 —> А - В = 0А = 0, В = 1 —> А - В = 0 |

Для изменения указанного порядка выполнения логических операций используются скобки.

| А | В | F =А\*В |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Анализ составных логических высказываний удобно делать с помощью ***таблиц истинности***. В них представляют все возможные комбинации значений элементарных высказываний, которые входят в составное, и его результирующее значения для каждой из них. Примеры:

Конъюнкция

| А | В | F =А+В |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Дизъюнкция

| А | В | F = A-»B |
| --- | --- | --- |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Импликация

| А | В | F = А~В |
| --- | --- | --- |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Эквиваленция

Законы логики отражают наиболее важные закономерности логического мышления. В алгебре высказываний законы логики записываются в виде формул, которые позволяют проводить эквивалентные преобразования логических выражений в соответствие с законами логики.

Таблица 5.2

**Основные законы логики**

|  |  |
| --- | --- |
| **Закон логики** | **Название закона** |
| А̿ = А | Закон двойного отрицания |
| A ∨ B = B ∨ AA&B = B&A | Переместительный (коммутативный) закон |
| (A ∨ B) ∨ C = A ∨ (B ∨ C)(A&B) &C = A& (B&C) | Сочетательный (ассоциативный) закон |
| (A ∨ B) &C = (A&C) ∨ (B&C)(A&B) ∨ C = (A ∨ C) & (B ∨ C) | Распределительный (дистрибутивный) закон |
| (̅̅̅A̅̅∨̅̅̅B̅̅)̅ = A̅&B̅(̅̅A̅̅&̅̅B̅̅) = A̅ ∨ B̅ | Закон общей инверсии (законы де Моргана) |
| A ∨ A = AA&A = A | Закон идемпотентности |
| A ∨ 1 = 1A ∨ 0 = AA& 1 = AA& 0 = 0 | Законы исключения констант |
| A& A̅ = 0 | Закон противоречия |
| A ∨ A̅ = 1 | Закон исключения третьего |
| A ∨ (A&B) = AA& (A ∨ B) = A | Закон поглощения |
| (A&B) ∨ (A̅&B) = B(A ∨ B) & (A̅ ∨ B) = B | Закон исключения (склеивания) |

**Практическая часть**

**Задание 1.**

Напишите следующие высказывания в виде логических выражений.

1. Число 17 нечетное и двузначное.
2. Если число делится на 2 и не делится на 3, то оно не делится на 6.
3. Если Маша – сестра Саши, то Саша – брат Маши.
4. Произведение трех чисел равно нулю тогда и только тогда, когда одно из них равно нулю
5.
6. Голова думает тогда и только тогда, когда язык отдыхает.
7. Если производная функции в точке равна нулю и вторая производная этой функции в той же точке отрицательна, то данная точка есть точка локального максимума функции.
8. На уроке физики ученики выполняли лабораторную работу и сообщали результаты исследований учителю.
9. Если прямая параллельна каждой из двух пересекающихся плоскостей, то она параллельна и линии их пересечения
10. Ты можешь купить в магазине продукты, если у тебя есть деньги.
11. Если прямая *l* перпендикулярна двум прямым *a* и *b*, лежащим в плоскости *L*, и прямые *a* и *b* не параллельны *a* $∦$ *b*, то прямая *l* перпендикулярна всякой прямой *с*, лежащей в плоскости *L*
12. При замерзании воды выделяется тепло
13. Если прямая *l* перпендикулярна двум прямым *a* и *b*, лежащим в плоскости *L*, и неперпендикулярна некоторой прямой *с*, лежащей в этой же плоскости, то прямые *a* и *b* параллельны
14. Неверно, что корова – хищное животное.
15. Если две прямые *a* и *b*, лежащие в плоскости *L*, непараллельны *a*$∦$*b* и прямая *l* неперпендикулярна некоторой прямой *с*, лежащей в плоскости *L*, то *l* неперпендикулярна одной из прямых *a* или *b*.
16. На уроке информатике необходимо соблюдать правила техники безопасности.
17. Если какие-либо два из трех векторов *а, b ,* *с* коллинеарны, то их смешанное произведение равно нулю.
18. Если на улице дождь, то асфальт мокрый.
19. Логарифм некоторого положительного числа будет положительным, если основание логарифма и логарифмируемое число будут больше 1 или если основание логарифма и логарифмируемое число будут заключены между 0 и 1
20. Если компьютер включен, то можно на нем работать.
21. Если в параллелограмме не все углы прямые или не все стороны равны между собой, то этот параллелограмм не прямоугольник или не ромб
22. Катя любит писать сочинения или решать задачи.
23. Если в треугольнике любая его медиана не является высотой и биссектрисой, то этот треугольник не равнобедренный и не равносторонний
24. Тише едешь – дальше будешь.
25. Если производная функции в точке равна нулю и вторая производная этой функции в той же точке отрицательна, то данная точка есть точка локального максимума функции.
26.
27. Если число делится на 2, то оно – четное.
28. Если прямая *l* перпендикулярна двум прямым *a* и *b*, лежащим в плоскости *L*, и прямые *a* и *b* не параллельны *a* $∦$ *b*, то прямая *l* перпендикулярна всякой прямой *с*, лежащей в плоскости *L*
29.
30. Земля движется по круговой или эллиптической орбите.
31. Если какие-либо два из трех векторов *а, b ,* *с* коллинеарны, то их смешанное произведение равно нулю
32. Водительские права можно получить тогда и только тогда, когда тебе исполниться 18 лет.
33. Если прямая параллельна каждой из двух пересекающихся плоскостей, то она параллельна и линии их пересечения

**Задание 2.**

1. Даны высказывания: А = {3+3=7}, B = {3+3=6} Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {2+3=5}, B = {2\*2=4}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {5\*5=25}, B = {5+5=11}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {7+3=10}, B = {7-3=4}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В

1. Даны высказывания: А = {10-3=7}, B = {10-7=3}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = «9 делится на 3» B = «8 делится на 3» Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {Этот треугольник равнобедренный}, B = {Этот треугольник равносторонний}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {Это число целое}, B = {Это число положительное}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

1. Даны высказывания: А = {Это число делится на 3}, B = {Это число простое}. Определить истинность высказываний:

А, В, А&В, А̅, В̅, АvВ, А→В, А↔В.

**Задание 3.** Постройте отрицание для высказываний

1. Все ребята умеют плавать.
2. Невозможно создать вечный двигатель.
3. Каждый человек – художник.
4. Человек все может.
5. Сегодня в театре идет опера «Евгений Онегин».
6. Волга впадает в Каспийское море.
7. Число 28 не делится на число 7.
8. 6 > 3.
9. Все простые числа нечетны.
10. 4≤5.
11. √2 рациональное число.
12. Африка – остров.
13. Все слова можно разделить на слоги.
14. Некоторые грибы несъедобны.
15. Железо тяжелее свинца.

**Задание 4.** Составить таблицы истинности для следующих логических выражений

1. F = А̅&(A&B) ∨ (A → B)
2. F = А̅ ∨ B&(A&A) ∨ B
3. F = (̅̅А&̅̅B̅̅) ∨ (A → B) ∨ A
4. F = А̅&(A ∨ B) ∨ (A ↔ B)
5. F = (A → B)&(А̅ ∨ B̅)
6. F = (̅̅(̅X̅̅∨̅̅̅Y̅̅)̅& ̅̅̅(̅Z̅̅̅↔̅̅̅̅X̅)̅̅)& (Z ∨ Y)
7. F = (X&Y) &( ̅̅X ̅̅̅∨̅̅̅X̅̅)&(Z ↔ Y)
8. F = (X ↔ Z) & (X ̅ ∨ X) & (Z ∨ Y)
9. F = (̅̅X̅̅∨̅̅̅Y̅̅) ∨ (Z → X) & (Z ↔ Y)
10. F = ( ̅̅(̅X̅̅∨̅̅̅Z̅̅)̅&̅̅̅(̅Z̅̅̅↔̅̅̅̅X̅)̅̅)& (Z → Y)
11. F = (A ∨ B) &( ̅̅C̅&̅̅̅D̅̅̅)
12. F = (̅̅A̅̅̅→̅̅̅B ̅̅̅) ∨ C&D̅
13. F = (A ↔ B) &( ̅̅C∨D̅̅)
14. F = A ∨ B̅& (C → D̅)
15. F = (A → B) ∨ A̅& (C ↔ D)

**Задание 5.** Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F. Какое выражение соответствует F?

1.

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |

1) ¬(X ∧ Y) ∨ (X ≡ Z)

2) (X ∧ Y) ∨ (X ≡ Z)

3) (¬X ∧ Y) ∧ (X ≡ Z)

4) ¬(X ∧ Y) ∧ (X ≡ Z)

2

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |

1) ¬X → Z ∧ Y

2) Z → X ∨ Y

3) (¬X ∨ Y) ∧ Z

4) X ∨ Y → ¬Z

3

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |

1) X → (Y →Z)

2) (X → Y) → Z

3) X ∨ Y ∧ ¬Z

4) Х ∨ Y → Z

4

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |

1) X ∧ Y ∨ Z

2) ¬X ∨ ¬Y ∨ Z

3) (X ∨ Y) ∧ ¬Z

4) ¬(X ∨ Y) → Z

5

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

1) ¬X ∨ ¬Y ∨ Z

2) X ∧ Y ∧ Z

3) X ∨ Y ∨ Z

4) ¬X ∧ ¬Y ∧ ¬Z

6

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |

1) X ∨ Y → Z

2) ¬X ∨ Y → Z

3) ¬X ∧ Z → Y

4) X ∨ ¬Z → Y

7

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

1) (X ≡ Z) ≡ Y

2) (X ≡ Z) ≡ (¬Y)

3) (X ≡ Z) ∧ Y

4) (X ≡ Z) ∨ (¬Y)

8

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |

1) ¬X ∧ ¬ Y

2) (X ≡ Y) ∧ Z

3) (X ≡ Y) ∨ Z

4) (¬X ≡ Y) ∨ Z

9

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |

1) X → Z ∧ Y

2) ¬Z →(X → Y)

3) ¬(X ∨ Y) ∧ Z

4) ¬X ∨ ¬(Y ∧ Z)

10

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |

1) X ∨ Y ∨ Z

2) ¬X ∨ ¬Y ∨ ¬Z

3) X ∧ ¬Y ∧ Z

4) ¬X ∧ ¬Y ∧ ¬Z

11

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |

1) (0 ∧ Y) ∧ (X ≡ Z)

2) (1 ∧ Y) ∧ (X ≡ Z)

3) (0 ∨ ¬Z) ∧ (X ≡ Y)

4) (¬1 ∧ Y) ∧ (X ≡ Z)

12

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |

1) X ∨ Y ∨ Z

2) X ∨ ¬Y ∧ ¬Z

3) X ∨ ¬Y ∨ Z

4) ¬X ∧ Y ∧ ¬Z

13

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | F |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

1) X ∨ Y ∨ Z

2) X ∧ Y ∧ ¬Z

3) ¬X ∧ Y ∧ ¬Z

4) X ∨ ¬Y ∨ Z

14

|  |  |  |  |
| --- | --- | --- | --- |
|  | Y | Z | F |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |

1) (X ∨ ¬Y) ∧ Z

2) (X ∧ ¬Y) ∨ Z

3) (X ∨ ¬Y) ∨ ¬Z

4) X ∧ ¬Y ∧ ¬Z

15

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |

1) (A → ¬B) ∨ C

2) (¬A ∨ B) ∧ C

3) (A ∧ B) → C

4) (A ∨ B) → C

**Задание 6.**

|  |
| --- |
| 1. Для какого имени ложно высказывание: (Первая буква имени гласная → Четвертая буква имени согласная)?
2. 1) Елена
3. 2) Вадим
4. 3) Антон
5. 4) Федор
 |
| 1. Для какого имени ложно высказывание: (Первая буква гласная) \/ (Четвёртая буква согласная)?
2. 1) Пётр
3. 2) Алексей
4. 3) Наталья
5. 4) Елена
 |
| 1. Какое из приведенных названий животных удовлетворяет логическому условию: (В слове пять букв) & (Четвертая буква гласная)?
2. 1) Зебра
3. 2) Слон
4. 3) Кабан
5. 4) Олень
 |
| 1. Для какого из названий животных ложно высказывание: Четвёртая буква гласная → (Вторая буква согласная)?
	1. Кошка
	2. Жираф
	3. Верблюд

4) Страус |
| 1. Какое из приведённых имён удовлетворяет логическому условию:
2. (вторая буква гласная → первая буква гласная) & (последняя буква согласная)?
3. 1) Алексей
4. 2) Павел
5. 3) Ксения
6. 4) Марина
 |
| 1. Для какого из названий животных ложно высказывание: (Заканчивается на согласную букву) & (B слове 6 букв) → (Четвертая буква согласная)?
2. 1) Страус
3. 2) Леопард
4. 3) Верблюд
5. 4) Кенгуру
 |
| 1. Какое из приведённых имен удовлетворяет логическому условию: (Первая буква гласная) & ((Четвёртая буква согласная) ∨ (B слове четыре буквы))?
2. 1) Сергей
3. 2) Вадим
4. 3) Антон
5. 4) Илья
 |
| 1. Для какого символьного набора истинно высказывание: Вторая буква согласная & (В слове 3 гласных буквы \/ Первая буква согласная)?
	1. УББОШТ
	2. ТУИОШШ
	3. ШУБВОИ

4) ИТТРАО |
| 1. Какое из приведенных названий домашних животных удовлетворяет следующему логическому условию: ((первая буква согласная) → (последняя буква согласная)) & (название содержит букву «к»)?

1) Кролик2) Корова3) Коза4) Свинья |

**Задание 7.** Упростить логические выражения:

1) 

2) 

3) 

4) 

5)

6) 

7) 

8) 

9) 

10) 

11) 

12) 

13) 

14) 

15) 

**Задание 8.**

Обозначим через *m&n* поразрядную конъюнкцию неотрицательных целых чисел *m* и *n*. Так, например, 14&5 = 11102&01012 = 01002 = 4.

Для какого наименьшего неотрицательного целого числа *А* формула тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной *х*) ?

1. *x&25 ≠ 0 → (x&17 = 0 → x&А ≠ 0)*
2. *x* & 29 ≠ 0 → (*x* & 12 = 0 → *x* & *А* ≠ 0)
3. 
4. x&25 ≠ 0 → (x&9 = 0 → x&А ≠ 0)
5. x&25 ≠ 0 → (x&19 = 0 → x&А ≠ 0)
6. x&77 ≠ 0 → (x&12 = 0 → x&А ≠ 0)
7. x&33 = 0 → (x&45≠0 → x&А ≠ 0)
8. x&49 ≠ 0 → (x&41 = 0 → x&А ≠ 0)
9. x&41 ≠ 0 → (x&33 = 0 → x&А ≠ 0)
10. *х&А* → (*x*&10 = 0 → *х*&3)
11. *х&А* → (*x*&36 = 0 → *х*&6)
12. x&9 = 0 → (x&19 ≠ 0 → x&А ≠ 0)
13. x&17 = 0 → (x&29 ≠ 0 → x&А ≠ 0)
14. *x*&51 = 0 ∨ (*x*&41 = 0 → *x*&*А* = 0)
15. *x*&51 = 0 ∨ (*x*&41 = 0 → *x*&*А* = 0)

**Задание 9.**

|  |
| --- |
| а) Три одноклассника – Влад, Тимур и Юра, встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой физиком, а третий юристом. Один полюбил туризм, другой бег, страсть третьего – регби. Юра сказал, что на туризм ему не хватает времени, хотя его сестра – единственный врач в семье, заядлый турист. Врач сказал, что он разделяет увлечение коллеги. Забавно, но у двоих из друзей в названии их профессий и увлечений не встречается ни одна буква их имен. Определите, кто, чем любит заниматься в свободное время и у кого какая профессия. |
| б) Три дочери писательницы Дорис Кей – Джуди, Айрис и Линда, тоже очень талантливы. Они приобрели известность в разных видах искусств – пении, балете и кино. Все они живут в разных городах, поэтому Дорис часто звонит им в Париж, Рим и Чикаго. Известно, что: * Джуди живет не в Париже, а Линда – не в Риме,
* парижанка не снимается в кино,
* та, кто живет в Риме, певица,
* Линда равнодушна к балету

Где живет Айрис, и какова ее профессия? |
| в) Пятеро одноклассников: Ирена, Тимур, Камилла, Эльдар и Рамиль стали победителями олимпиад школьников по физике, математике, информатике, литературе и географии. Известно, что:* победитель олимпиады по информатике учит Ирену и Тимура работе на компьютере,
* Камилла и Эльдар тоже заинтересовались информатикой,
* Тимур всегда побаивался физики,
* Камилла, Тимур и победитель олимпиады по литературе занимаются плаванием,
* Тимур и Камилла поздравили победителя олимпиады по математике,
* Ирена сожалеет о том, что у нее остается мало времени на литературу.

Победителем какой олимпиады стал каждый из этих ребят? |
| г) В педагогическом институте преподаватели Афанасьева, Бирюкова, Климова, Дунин, Исаев и Федоров преподают немецкий язык, английский язык, французский язык, историю, физическое воспитание и математику.Известно, что:* преподаватель немецкого языка и преподаватель математики в студенческие годы занимались танцами,
* Исаев старше Федорова, но стаж работы у него меньше, чем у преподавателя физкультуры,
* будучи студентками, Афанасьева и Бирюкова учились вместе в одном университете,
* все остальные окончили педагогический институт,
* Федоров – отец преподавателя французского языка,
* преподаватель английского языка – самый старший из всех по возрасту и имеет самый большой стаж работы. Он работает в этом институте с тех пор, как окончил его. Преподаватели математики и истории - его бывшие студенты.
* Афанасьева старше преподавателя немецкого языка.

Назовите, кто какой предмет преподает? |
| д) Как – то случай свел в одном купе известного астронома, поэта, прозаика и драматурга. Это были Алексеев, Борисов, Константинов и Дмитриев. Оказалось, что каждый из них взял с собой книгу, написанную одним из пассажиров купе. Алексеев и Борисов углубились в чтение, предварительно обменявшись купленными книгами. Поэт читал пьесу. Прозаик, очень молодой человек, выпустивший свою книгу, говорил, что он никогда ничего не читает по астрономии. Борисов купил в дорогу одно из произведений Дмитриева. Никто из пассажиров не покупал и не читал книги, написанные им самим. Что читал каждый из них? Кто кем был? |

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 6**

**Построение функциональных схем**

**Цель:** развитие знаний обучающихся о логических элементах, комбинационной логике и способах построения сложных логических схем на их основе.

**Теоретическая часть**

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера. Любая информация при обработке на компьютере представляется в двоичной форме, т. е. кодируется некоторой последовательностью 0 и 1. Обработку двоичных сигналов, соответствующих 0 и 1, выполняют в компьютере логические элементы. Логические элементы, которые выполняют основные логические операции **И, ИЛИ, НЕ,** представлены на рис.6.1,6.2, 6.3.

**Логический элемент «И» (конъюнктор)**

Логический элемент «И» (конъюнктор) выдает на выходе значение
логического произведения входных сигналов: на выходе выдает 1 тогда и только тогда, когда на все входы поданы 1.

&

Рисунок 6.1. Условное обозначение конъюнктора

**Логический элемент «ИЛИ» (дизъюнктор)**

Логический элемент ИЛИ (дизъюнктор) выдает на выходе значение суммы входных сигналов: на выходе выдает 1, если хотя бы на один из входов подается 1.

1

Рисунок 6.2. Условное обозначение дизъюнктора

**Логический элемент «НЕ» (инвертор)**

Логический элемент «НЕ» выдает на выходе сигнал, противоположный
сигналу на входе, т.е. на его выходе будет 1, если на вход поступит 0 и наоборот.

Рисунок 6.3. Условное обозначение инвертора

На рис. 6.4 приведена логическая схема триггера. Это устройство может хранить 1 бит информации. Триггеры используются как разряды оперативной памяти и памяти процессора. В обычном состоянии триггер хранит сигнал 0. Для записи 1 на вход S подается сигнал 1. Про­дя по схеме он формирует на выходе Q сигнал 1 и устойчиво хранит его после того, как сигнал S исчезнет. Для того, чтобы сбросить этот сигнал и подготовиться к приему нового на вход R подается сигнал 1, который приводит триггер к «нулевому» состоянию.

Рисунок 6.4.Логическая схема триггера

Условные обозначения логических элементов являются стандартными и используются при составлении логических схем компьютера. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу компьютера.

Технически компьютерный логический элемент реализуется в виде электрической схемы, которая представляет собой соединение различных деталей: диодов, транзисторов, резисторов, конденсаторов. На вход логического элемента, который называют также вентилем, поступают электрические сигналы высокого и низкого уровней напряжения, на выход выдается один выходной сигнал также либо высокого, либо низкого уровня. Эти уровни соответствуют одному из состояний двоичной системы: 1 — 0; ИСТИНА — ЛОЖЬ.

Из логических элементов составляются электронные логические схемы, выполняющие более сложные логические операции. Набор логических элементов, состоящий из элементов НЕ, ИЛИ, И, с помощью которых можно построить логическую структуру любой сложности, называется *функционально полным*.

Логические схемы необходимо строить из минимально возможного количества элементов, что в свою очередь, обеспечивает большую скорость работы и увеличивает надежность устройства.

**Алгоритм построения логических схем:**

1) Определить число логических переменных.

2) Определить количество базовых логических операций и их порядок.

3) Изобразить для каждой логической операции соответствующий ей вентиль.

4) Соединить вентили в порядке выполнения логических операций.

**Пример1.**

Дана логическая функция: F(А, В) = ¬(А/\В). Постройте соответствующую ей функциональную схему.

 **Решение:**

Функциональная схема будет содержать 2 входа А и В. Рассмотрим логическое выражение и определим порядок действий в нем:

1) первым выполняется логическое умножение А /\ В, следовательно, сигналы с входов А и В подаются на конъюнктор;

2) далее выполняется логическое отрицание ¬(А /\ В), следовательно, сигнал, полученный на выходе из конъюнктора должен быть инвертирован, т.е. подан на инвертор.

Выход инвертора является выходом функциональной схемы.

Изобразим схему, следуя данным действиям



**Пример 2.**

Постройте логическую схему, соответствующую логическому выражению F=X&Yv¬(YvX).

Вычислить значения выражения для X=1, Y=0.

1)   Переменных две: X и Y.

2)   Логических операций четыре: конъюнкция, две дизъюнкции и отрицание.

Определяем порядок выполнения операций:

       1  4  3   2

     X&Yv ¬(YvX).

3)   Схему строим слева направо в соответствии с порядком выполнения логических операций:



4) Вычислим значение выражения: F=1&0v¬(0v1)=0.

Итак, логическим элементом называется преобразователь, который, получая сигналы об истинности отдельных высказываний, обрабатывает их и в результате выдает значение логического отрицания, логической суммы или логического произведения этих высказываний.

Логическое устройство – это цепочка из логических элементов, в которой выходы одних элементов являются входами других. Функциональная схема – это схема соединения логических элементов, реализующая логическую функцию.

**Практическая часть**

|  |
| --- |
| **Задание 1.**Построить логическую схему, соответствующую логическому выражению. |
| 1. F = A ∨ B&C̅
 |
| 1. F = A̅ ∨ B&C
 |
| 1. F = X&Y ∨ Z̅
 |
| 1. F = A&B ∨ C̅
 |
| 1. F = X ̅&Y ∨ Z
 |
| 1. F = (A̅&B̅) ∨ (B̅̅̅&̅̅B̅̅)
 |
| 1. F = (A̅∨̅̅̅B̅̅)&( A̅ &̅̅C̅)
 |
| 1. F = (X ̅̅̅&̅̅Y̅)&(X ̅̅̅∨̅̅̅Z̅)
 |
| 1. F = (X ̅̅̅&̅̅Y̅) ∨ (Z̅̅&̅̅̅X̅)
 |
| 1. F = (A̅ ∨̅̅̅C̅)&(B ̅̅̅&̅̅C̅)
 |
| 1. F = (A&( ̅̅B̅̅∨̅̅̅C̅̅)) ∨ (C&A̅)
 |
| 1. F = (A ∨ B̅) ∨ (C&B̅&A̅)
 |
| 1. F = (A̅ &̅̅B̅)&(A ∨ B)&C̅)
 |
| 1. F = (A̅ &( A̅̅∨̅̅̅B̅))&(C ∨ B̅)
 |
| 1. F = (A̅&C̅) ∨ (A̅∨̅̅̅B̅) ∨ A
 |
| **Задание 2.** Построить логическую схему, соответствующую логическому выражению, и найти значение логического выражения. |
| 1. F = (X ∨ Y)&(Z ∨ Y); если X = 0; Y = 1; Z = 0
 |
| 1. F = (X&Y) ∨ (Z ∨ Y); если X = 1; Y = 0; Z = 1
 |
| 1. F = (A ∨ B)&(C&B); если A = 1; B = 1; C = 0
 |
| 1. F = (X&Y) ∨ (Z ∨ Y); если X = 1; Y = 0; Z = 0
 |
| 1. F = (A&B)&(C ∨ B); если A = 0; B = 1; C = 0
 |
| 1. F = (A̅&B ∨ C)&( ̅̅C&̅̅B); если A = 1; B = 0; C = 0
2. F = (X ∨ Y ̅)&(Z̅ ∨ Y) ∨ X; если X = 1; Y = 0; Z = 1
3. F = ( ̅̅X∨̅̅̅Y)&(Z̅ ∨ Y ̅&X); если X = 1; Y = 0; Z = 0
4. F = A ∨ (̅̅A&̅̅B)&(C̅ ∨ B); если A = 1; B = 1; C = 0
5. F = (X&Y ̅) ∨ Z ∨ (̅̅Z∨̅̅̅Y); если X = 0; Y = 1; Z = 1
6. F = А ̅̅̅&̅̅B ∨ B̅̅̅∨̅̅̅C, если A = 1; B = 0; C = 0
7. F = X ̅&( ̅̅Y ∨X̅̅) ; если X = 0; Y = 0; Z = 1
8. F = (A ̅̅̅&B̅) ∨ (A̅̅&̅̅B), если A = 1; B = 0; C = 1
9. F = A&B ∨ C ∨ (A ̅̅̅∨B̅̅), если A = 0; B = 1; C = 0
10. F = (X ∨ Z)&(X ∨ Z̅)&Y ̅; если X = 1; Y = 0; Z = 1
 |
| Примеры решения задач "Логические основы работы компьютера" 2 |
|  Z&1 Y X |
|  Примеры решения задач "Логические основы работы компьютера" |
| 4)https://konspekta.net/studopedianet/baza4/2059217023376.files/image034.jpg |





7)



8)



|  |
| --- |
| **Задание 4.** По заданной логической формуле построить логическую схему и таблицу истинности |
| 1. F = (A ̅̅̅&B̅) ∨ (B̅̅̅&C̅)
 |
| 1. F = (A ̅̅̅∨B̅)&(A ̅̅̅&C̅)
 |
| 1. F = (X ̅̅̅&Y̅)&(X ̅̅̅∨Z̅)
 |
| 1. F = (X ̅̅̅&Y̅) ∨ (Z̅̅&̅X̅)
 |
| 1. F = (A ̅̅̅∨C̅)&( B ̅̅&C̅)
 |
| 1. F = А ̅̅̅&B̅ ∨ B̅̅̅∨C̅
 |
| 1. F = X ̅&( ̅̅Y ∨X̅̅)
 |
| 1. F = (A ̅̅̅&B̅) ∨ (A̅̅&B̅̅)
 |
| 1. F = A&B ∨ C ∨ (A ̅̅̅∨B̅̅)
 |
| 1. F = (X ∨ Z)&(X ∨ Z̅)&Y ̅
 |
| 1. F = (X&Y ̅) ∨ Z ∨ (̅Z̅∨Y̅̅)
 |
|

|  |
| --- |
| 1. F = (A&( ̅̅B∨̅̅̅C)) ∨ (C&A̅)
 |
| 1. F = (A ∨ B̅) ∨ (C&B̅&A̅)
 |
| 1. F = (A ̅̅̅&B)&(A ∨ B)&C̅)
 |
| 1. F = (A ̅̅&A̅̅∨B̅))&(C ∨ B̅)
 |
|  |

 |

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 7**

**Алгоритмизация и** **основные технологии программирования**

**Цель:** формирование понятия алгоритмов, свойств и видов алгоритмов, развитие алгоритмического мышления.

**Теоретическая часть**

Алгоритм - это чётко определённая последовательность действий, описывающих процесс преобразования объекта из начального состояния в конечное с помощью понятной исполнителю последовательности команд.

Свойства алгоритма:

* дискретность, т.е. алгоритм представлен в виде последовательности команд, которые исполнитель должен выполнять одну за другой;
* понятность, т. е. алгоритм должен содержать только те команды, которые входят в систему выполняемых команд исполнителя;
* детерминированность, т. е. алгоритм должен быть представлен таким образом, чтобы, выполняя очередную команду, исполнитель точно знал, какую команду следует выполнять следующей;
* результативность, т. е. алгоритм должен обеспечить пре­образование от начальных данных к результату за конечное число команд.

Для графического представления алгоритмов компьютерных программ используют специальные обозначения.

Основные типы алгоритмических структур и их блок-схемы

- линейная - каждое действие должно быть выполнено после­довательно одно за другим:

► > ► ►

Рисунок 7.1

- ветвление - в зависимости от выполнения или невыполнения указанного условия реализуется та или иная последовательность команд:



Рисунок 7.2

- цикл - серия команд (тело цикла) выполняется многократно. Разновидности циклов:

Цикл со счётчиком - выполняется заранее определённое количество раз:



Рисунок7.3

Цикл с предусловием - проверяется условие и, если оно вы­полняется, то тело цикла (тц) повторяется, если нет - происходит переход к действию, следующему за телом цикла. Если условие не выполняется при первой проверке, то тело цикла не выполняется ни одного раза:



Рисунок 7.4

Цикл с постусловием *-* тело цикла (тц) выполняется, затем проверяется условие и, если оно не выполняется, то тело цикла повторяется, если выполняется, то происходит переход к действию, следующему за телом цикла. В этом варианте тело цикла выполняется хотя бы один раз всегда.



Рисунок 7.5

**Примеры**

1. Задан одномерный массив *Х: xb x2, ..., xn .* Приведённый фрагмент программы определяет сумму положительных элементов массива *Х*.



Рисунок 7.6

1. Представленный фрагмент блок-схемы вычисляет количество нулевых элементов в массиве *Х*, состоящем из *n* элементов.



Рисунок 7.7

Сложные алгоритмы удобнее записывать на ***псевдокоде,*** который включает фразы естественного языка и общепринятые математические обозначения, оформленные по стандартам, принятым в алгоритмических языках.

1. После выполнения представленного фрагмента программы переменная *у* приняла значение 10. Каким было значение *х* перед входом в этот фрагмент?

*y* := *x* - 1; *x* :=*y* + 2; *y* := *x* + *y ;* вывод *y*

Ответ: х = 5.

1. Задан фрагмент алгоритма. Какими будут значения d и c на выходе, если на входе а = 8; в =3?
2. Если а < в, то с = в - а иначе с = 2 \* (а - в)
3. d = 0
4. Пока с > А выполнять: d = d + 1;c = c - 1

Ответ. При на выходе из этого фрагмента переменные d и c примут значения: d = 2; c=8

1. Представлен алгоритм (div(x,y) - целочисленное деление х на у, mod(x,y) - целый остаток от деления):

k = 50

Выбор

при div(k,12) = 4: d := k при mod(k,12) < 5: d := 2 при mod(k,12) > 9: d := 3 иначе d := 1

Всё

Ответ*.* Значение переменной *d* после выполнения алгоритма равно 2.

1. Представлен алгоритм (mod(.*x*,*y*) - целый остаток от деле­ния):

*k* = 70

Выбор

при mod(*k,*12) = 7: *d* := *k* при mod(*k*,12) < 5: *d* := 2 при mod(*k*,12) > 9: *d* := 3 иначе *d* := 1

Всё

Ответ. Значение переменной d после выполнения алгоритма равно 3

При разработке компьютерных моделей для задач, решаемых по сложным и громоздким алгоритмам, получаются длинные программы, которые очень сложно отлаживать и тестировать. Для облегчения этих работ были разработаны разные технологии программирования, которые упрощают структуру программы и ее отладку. Основные из них:

* модульное программирование;
* программирование «снизу вверх» (восходящее программи­рование);
* программирование «сверху вниз» (нисходящее программи­рование).

Согласно этим технологиям в алгоритме задачи выделяются модули. Модуль - это либо логически законченный фрагмент общей задачи, либо часто повторяющийся блок расчётов.

Каждый модуль оформляется по определённым правилам в виде подпрограммы, и в текст основного алгоритма вместо него вставляется короткая инструкция вызова подпрограммы. Когда выполнение программы доходит до этой инструкции, выполняется фрагмент, заложенный в тело подпрограммы, после чего управление передаётся на команду, следующую за инструкцией вызова подпрограммы.

Различают подпрограммы функции (используются как операнды в выражениях) и процедуры (используются как операторы). Общую структуру подпрограмм можно представить следующим образом:



Тело подпрограммы - это операторы, которые программируют ту часть алгоритма, которая выделена в данный модуль. Формальные параметры - это те переменные в теле подпрограммы, значения которых надо будет перед началом работы этого модуля получить из основной программы.

При использовании технологии «сверху вниз» разработка про­граммы начинается с последовательной детализации алгоритма на всё более мелкие части до тех пор, пока получатся такие модули, для которых можно написать конкретные команды. Затем составляется текст основной программы, в которой вместо фрагментов, выделенных в подпрограммы, ставят «заглушки». Это подпрограммы, в которых вместо реально нужных операторов ставят сигнальные печати или ввод результатов, которые должна была сосчитать эта подпрограмма. Таким образом, проверяют и отлаживают последовательность действий в основном алгоритме. Затем подпрограммы-заглушки по очереди заменяют на соответствующие алгоритму подпрограммы, отлаживают и тестируют их. Такая технология облегчает создание программы, уменьшает количество ошибок и облегчает нахождение допущенных ошибок. Считается, что программа оптимального по размерам модуля целиком должна помещаться на экране дисплея.

При использовании технологии «снизу вверх» в первую очередь определяются и разрабатываются вспомогательные модули, которые потребуются для проектируемой программы. После того, как все модули отлажены, из них, как из кубиков, в соответствии с начальным алгоритмом собирается основная программа.

Объектно-ориентированное программирование (ООП) применяют при программировании разных манипуляций над объектами. Например, при составлении программ управления размерами и положением окон Windows, листами книги Excel, файлами и т. п.

Основные термины и понятия объектно-ориентированного программирования:

* моделируемая система состоит из объектов. Объекты могут быть вложены друг в друга, например, объект «лист Excel» - это часть объекта «книга Excel»;
* объекты каким-то образом взаимодействуют между собой;
* каждый объект характеризуется своим состоянием и поведением. Состояние объекта задаётся значением некоторых его свойств.

Класс объектов - шаблон, определяющий основные свойства, методы и события группы объектов, объединяемых в класс. Это же можно сформулировать другими словами: это множество объектов, имеющих общее поведение и общую структуру.

События - ситуации, в которых надо программировать какой- то отклик объекта.

Наследование - порождает иерархию объектов. Оно может быть смоделировано с помощью таксонометрической классификационной схемы (иерархии). В основном классе (родителе) можно выделять подклассы (потомки). Они состоят из объектов, входящих в класс родителя и обладают наряду со всеми его характеристиками дополнительной группой свойств, которых у других объектов класса - родителя нет.

Инкапсуляция - сокрытие деталей программ, создающих и манипулирующих объектами. Создание объектов, манипулирование ими осуществляется программами языка ООП. Программист указывает в своей программе только то, что и с каким объектом нужно сделать, или какой результат нужно получить. То есть объекты рассматриваются как «чёрные ящики». Такой способ упрощает разработку программы и её модификацию.

Полиморфизм - для выполнения одного и того же типа действий в разных подклассах одного класса можно использовать одно и то же имя, хотя это действие реализуется разными внутренними методами (программами).

Этапы решения задач на компьютере

1.Постановка задачи:

* сбор информации о задаче;
* описание исходных данных и конечных целей;
* определение формы выдачи результатов.

2.Анализ и исследование модели задачи:

* анализ существующих аналогов;
* анализ технических и программных средств;
* разработка математической модели;
* разработка структур данных.

3.Разработка алгоритма:

* выбор метода проектирования алгоритма;
* выбор формы записи алгоритма (блок-схема, псевдокод и т. п.);
* выбор тестов и метода тестирования;
* проектирование алгоритма.

4.Программирование:

* выбор языка программирования;
* уточнение способа организации данных;
* запись алгоритма на выбранном языке.

5.Отладка и тестирование:

* синтаксическая отладка: исправление ошибок в форме за­писи конструкций;
* отладка семантики и логической структуры: семантика - система правил истолкования отдельных конструкций языка, напри­мер проверка правильности организации циклов, ветвлений и т. п., соответствия типов переменных в выражениях, логическая структу­ра - правильная последовательность обработки данных;
* тестовые расчёты и анализ результатов тестирования;
* совершенствование программы.

Деятельность, направленная на исправление ошибок в программной системе, называется отладкой. Тестирование - прогон отлаженной программы на эталонных вариантах исходных данных, для которых заранее известны результаты.

6.Анализ результатов тестирования и, если нужно, уточне­ние модели и повторение п. п. 2 - 5.

7.Сопровождение программы: составление документации по математической модели, алгоритму, программе, набору тестов, использованию готовой программы и т. п.

**Практическая часть**

**Задание 1.** Дайте ответ на вопрос

1. Перечислите основные свойства алгоритма.
2. При какой организации цикла может случиться, что тело цикла не выполнится ни разу.
3. Ошибка в форме записи программы - это ошибка.
4. Укажите типовые алгоритмические структуры, используемые при структурном программировании.
5. Ошибка в последовательности обработки данных, орга­низации циклов, ветвлений, соответствий типов переменных, исполь­зуемых в выражениях, - это ошибка.
6. Укажите различие между формальными и фактически­ми параметрами подпрограмм.
7. Укажите, какие из приведенных терминов являются ба­зовыми понятиями в объектно-ориентированном подходе к програм­мированию:
8. объект
9. свойство
10. метод обработки
11. инструкции
12. данные
13. модель
14. событие
15. класс объектов
16. Какой методологии программирования присуще понятие «Иерархия классов»?
17. Набор операторов, выполняющих заданное действие и не зависящих от других частей исходного кода, - это .
18. Выделите принципы, которые можно использовать при разработке программ методом структурного программирования:
19. принцип модульной разработки сложных программ;
20. использование композиции трёх базовых структур при записи алгоритма - линейных, ветвлений и циклов;
21. использование композиции двух базовых структур при записи алгоритма - ветвлений и циклов;
22. использование большого количества подпрограмм.
23. Объясните смысл понятий, используемых в методологии объектно-ориентированного программирования: объект, класс, метод, наследование, полиморфизм, инкапсуляция.
24. Выберите правильные варианты: при проектировании программного обеспечения используют подходы:
25. сверху-вниз
26. снизу-вверх
27. слева-направо
28. справа-налево
29. Укажите вариант описания, соответствующий циклу с постусловием:
30. пока условие истинно, выполнять оператор;
31. выполнять оператор, пока условие ложно;
32. выполнять оператор заданное число раз;
33. если условие истинно, выполнять оператор, иначе - остановиться.
34. Имя переменной определяет:
35. данные, хранящиеся в выделенной области оперативной памяти;
36. выделенную область оперативной памяти;
37. количество выделяемых ячеек оперативной памяти;
38. диапазон значений переменной.
39. .Значение переменной определяет:
40. данные, хранящиеся в выделенной области оперативной памяти;
41. выделенную область оперативной памяти;
42. количество выделяемых ячеек оперативной памяти;
43. диапазон значений переменной.

**Задание 2**.

1. Приведён алгоритм процедуры «ABCD» в псевдокоде:

Начать

Писать («Введите A,B,C,D»)

Читать(A,B,C,D)

Если A=B то

Если С<В то

Х:=1

Иначе

Х:=2

Иначе

Х:=3

Конец

Запишите этот алгоритм математическим языком на бумаге.

1. Определите, какими будут значения Х и Y на выходе из этого алгоритма:



1. Определите значение переменной D после выполнения алгоритма (mod(x,y) - целый остаток от деления):

k=70

Выбор

при mod(k,12)=7: D:=k при mod(k,12)<5: D:=2 при mod(k,12)>9: D:=3 иначе D:=1

Всё

1. Представлен фрагмент программы:

Y:=X+5; X:=Y; Y:=X+Y; вывод Y

Определите значение переменной Х перед входом в этот фрагмент, если известно, что после выполнения этого фрагмента переменная Y приняла значение 14.

1. Определите, сколько раз тело цикла выполняется в представленном фрагменте:

B := 10; D := 40

Начало цикла: пока D >= B

D := D - B

Конец цикла.

1. Определите вид вывода результатов в предложенном фрагменте программы:

X:=5

Z:=7

вывод ("X=",X," X=",Z,Z+X)

1. X=5 X=712



3. X=5 X=7 12

4. "X=",X," X=",Z,Z+X

1. Записать алгоритмическую структуру «ветвление» на одном из языков программирования: Паскаль, Python, С++.



1. Записать алгоритмическую структуру «выбор» на од­ном из языков программирования: Паскаль, Python, С++.
2. 
3. Записать алгоритмическую структуру «цикл со счет­чиком» на одном из языков программирования: Паскаль, Python, С++.
4. ****
5. Записать алгоритмическую структуру «цикл с предусловием» на одном из языков программирования: Паскаль, Python, С++.



1. Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной *X* будет напечатано в результате выполнения алгоритма.



1)8; 2) 10; 3) 15;….4)18

12) Фрагмент алгоритма изображен в виде блок-схемы. Определить, какое значение переменной *X* будет на­печатано в результате выполнения алгоритма



1)5; 2)6; 3)7; 4)8.

13)Алгоритм изображен в виде блок-схемы, в графичес­ких элементах которой приведен программный код на языке Visual Basic .NET. В текстовое поле введено це­лое десятичное число 10. Определить, какое целое двоичное число будет выведено на надпись в результа­те выполнения алгоритма.



14) Алгоритм изображен в виде блок-схемы, в графичес­ких элементах которой приведен программный код на языке Visual С#. В текстовое поле введено целое десятичное число 20. Определить, какое целое восьмеричное число будет выведено на надпись в результате вы­полнения алгоритма.



15) Ниже на языке программирования записаны две рекурсивные функции: *F* и *G*.

|  |
| --- |
| function F(n: integer): integer;begin if n > 2 then F := F(n - 1) + G(n - 2) else F := 1;end;function G(n: integer): integer;begin if n > 2 then G := G(n - 1) + F(n - 2) else G := 1;end; |

Чему будет равно значение, вычисленное при выполнении вызова *F*(7)?

**Задание 3.**  В приведенных программах допущены ошибки. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

**1.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число x, не превосходящее 1000, и выводится количество значащих цифр в двоичной записи этого числа.

|  |  |  |
| --- | --- | --- |
| **Паскаль** | **Алгоритмический язык** | **Python** |
| var x,cnt: integer;beginreadln(x);cnt := 0;while x > 0 dobegincnt:=cnt + x mod 2;x := x div 10end;writeln(cnt);end. | алгначцел x, cntввод xcnt := 0нц пока x > 0cnt := cnt+mod(x,2)x := div(x, 10)кцвывод cntкон | x = int(input())cnt = 0while x > 0:    cnt = cnt+x % 2    x = x // 10print(cnt) |

.

**2.**На обработку поступает натуральное число, не превышающее 109. Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 3. Если в числе нет цифр, кратных 3, требуется на экран вывести «NO».

|  |  |  |
| --- | --- | --- |
| **Паскаль** | **Алгоритмический язык** | **Python** |
| var N,digit,maxDigit: longint;begin    readln(N);    maxDigit := N mod 10;    while N > 0 do    begin        digit := N mod 10;        if digit mod 3 = 0 then            if digit > maxDigit then                maxDigit := digit;        N := N div 10;    end;    if maxDigit = 0 then        writeln('NO')    else        writeln(maxDigit)end. | алгнач    цел N, digit, maxDigit    ввод N    maxDigit := mod(N,10)    нц пока N > 0        digit := mod(N,10)            если mod(digit, 3) = 0 то            если digit > maxDigit то                maxDigit := digit            все        все        N := div(N,10)    кц    если maxDigit = 0 то        вывод "NO"    иначе    вывод maxDigit    всекон | N = int(input())maxDigit = N % 10while N > 0:    digit = N % 10    if digit % 3 == 0:        if digit > maxDigit:            maxDigit = digit    N = N // 10if maxDigit == 0:    print("NO")else:    print(maxDigit) |

**3.**На обработку поступает положительное целое число, не превышающее 109. Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0.

|  |  |  |
| --- | --- | --- |
| **Паскаль** | **Алгоритмический язык** | **Python** |
| var N, digit, sum: longint;begin    readln(N);    sum := 0;    while N > 0 do    begin        digit := N mod 10;        if digit < 7 then            sum := sum + 1;        N := N div 10;    end;    writeln(digit)end. | алгнач    цел N, digit, sum    ввод N    sum := 0    нц пока N > 0        digit := mod(N,10)        если digit < 7 то            sum := sum + 1        все        N := div(N,10)    кц    вывод digitкон | N = int(input())sum = 0while N > 0:    digit = N % 10    if digit < 7:        sum = sum + 1    N = N // 10print(digit) |

**4.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится количество цифр этого числа.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел N, sumввод Nsum := 1нц пока N > 1N := div(N, 10)sum := sum + 1кцвывод sumкон | var N: longint;sum: integer;beginreadln(N);sum := 1;while N > 1 dobeginN := N div 10;sum := sum + 1;end;writeln(sum);end. | n = int(input())sum = 1while n > 1:    n //= 10    sum += 1print(sum) |

* 1. Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится произведение цифр этого числа.

|  |  |  |
| --- | --- | --- |
| **Python** | **Паскаль** | **Алгоритмический язык** |
| n = int(input())product = n % 10while n >= 10:    digit = n % 10    product = product\*digit    n //= 10print(product) | var N, product: longint;    digit: integer;begin    readln(N);    product := N mod 10;    while N >= 10 do    begin        digit := N mod 10;        product := product\*digit;        N := N div 10;    end;    writeln(product);end. | алгнач    цел N, digit, product    ввод N    product := mod(N, 10)    нц пока N >= 10        digit := mod(N, 10)        product := product\*digit        N := div(N, 10)    кц    вывод productкон |

**6.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится максимальная цифра этого числа.

|  |  |  |
| --- | --- | --- |
|  **Алгоритмический язык** | **Паскаль** | **Python** |
| алгнач    цел N, digit, max\_digit    ввод N    max\_digit := 9    нц пока N > 9         digit := mod(N, 10)        если max\_digit < digit то            max\_digit := digit        все    N := div(N, 10)    кц    вывод max\_digitкон | var N: longint;    digit, max\_digit: integer;begin    readln(N);    max\_digit := 9;    while N > 9 do    begin        digit := N mod 10;        if max\_digit < digit then            max\_digit := digit;        N := N div 10;    end;    writeln(max\_digit);end. | n = int(input())max\_digit = 9while n > 9:    digit = n % 10    if max\_digit < digit:        max\_digit = digit    n //= 10;print(max\_digit) |

**7.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится сумма цифр этого числа.

|  |  |  |
| --- | --- | --- |
|  **Алгоритмический язык** | **Паскаль** | **Python** |
| алгнач    цел N, digit, sum    ввод N    sum := 0    нц пока N >= 9        digit := mod(N, 10)        sum := sum + digit        N := div(N, 10)    кц    вывод sumкон | var N: longint;    digit, sum: integer;begin    readln(N);    sum := 0;    while N >= 9 do    begin        digit := N mod 10;        sum := sum + digit;        N := N div 10;    end;    writeln(sum);end. | n = int(input())sum = 0while n >= 9:    digit = n % 10    sum += digit    n //=10;print(sum) |

**8.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N, не превосходящее 109, и выводится максимальная цифра этого числа.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгнач    цел N, digit, max\_digit    ввод N    max\_digit := 9    нц пока N >= 10        digit := mod(N, 10)        если digit > max\_digit то            max\_digit := digit        все        N := div(N, 10)    кц    вывод max\_digitкон | var N: longint;    digit, max\_digit: integer;begin    readln(N);    max\_digit := 9;    while N >= 10 do    begin        digit := N mod 10;        if digit > max\_digit then            max\_digit := digit;        N := N div 10;    end;    writeln(max\_digit);end. | n = int(input())max\_digit = 9while n >= 10:    digit = n % 10    if digit > max\_digit:        max\_digit = digit    n //= 10print(max\_digit) |

**9.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится сумма чётных цифр в десятичной записи этого числа или 0, если чётных цифр в записи нет.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический язык** | **Паскаль** | **Python** |
| алгначцел N, sввод Ns := 0нц пока N > 1если mod(N,2) = 0 тоs := mod(N,10)всеN := div(N,10)кцвывод sкон | var N: longint;s: integer;beginreadln(N);s := 0;while N > 1 do beginif N mod 2 = 0 then begins := N mod 10;end;N := N div 10;end;write(s);end. | N = int(input())s = 0while N > 1:    if N % 2 == 0:        s = N % 10    N //= 10print(s) |

**10.**Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число, не превосходящее 108, и выводится его первая (старшая) цифра.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел nввод nнц пока n>10n := mod(n,10)кцвывод nкон | var n: longint;beginread(n);while n>10 do beginn := n mod 10end;write(n);end. | n = int(input())while n > 10:    n %= 10print(n) |

 **11.**Требовалось написать программу, которая получает на вход натуральное число *N*, не превосходящее 109, и выводит число, которое получается из *N* после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры — единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел N, R, T, dввод NR := 0T := 1нц пока N > 0d := mod(N, 10)если d <> 1 тоR := R + d\*TT := T+1всеN := div(N, 10)кцвывод Tкон | var N, R, T: longint;d: integer;beginreadln(N);R:=0;T:=1;while N>0 do begind := N mod 10;if d<>1 then beginR := R + d\*T;T := T+1end;N := N div 10;end;writeln(T);end. | N = int(input())R=0T=1while N > 0:    d = N % 10    if d != 1:        R = R + d\*T        T = T+1    N //= 10;print(T) |

**12.**Дано натуральное число *N*, не превосходящее 108. Необходимо найти и вывести число, которое получится при записи *N* справа налево и удалении всех единиц. Ведущие нули выводить не надо. Если в числе *N* нет цифр кроме единиц и нулей, необходимо вывести 0. Например, при вводе числа 1984 нужно вывести 489, а при вводе 2001 нужно вывести 2.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел n, mцел dввод nm := 0нц пока n>=1d := mod(n,10)если d>1 тоm := 10\*d + mвсеn := div(n-d,10)кцвывод mкон | var n, m: longint;d: integer;beginread(n);m := 0;while n>=1 do begind := n mod 10;if d > 1 then beginm := 10\*d + m;end;n := (n – d) div 10;end;write(m);end. | n = int(input())m = 0while n >= 1:    d = n %10    if d > 1:        m = 10\*d + m    n = (n – d) // 10print(m) |

**13.**Дано натуральное число *N*, не превосходящее 108. Необходимо найти и вывести число, которое получится при записи *N* справа налево и удалении всех девяток. Ведущие нули выводить не надо. Если в числе *N* нет цифр кроме девяток и нулей, необходимо вывести 0. Например, при вводе числа 1984 нужно вывести 481, а при вводе 2009 нужно вывести 2.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел n, mцел dввод nm := 0нц пока n>1d := mod(n,10)если d<9 тоm := m + dвсеn := div(n,10)кцвывод mкон | var n, m: longint;d: integer;beginread(n);m := 0;while n>1 do begind := n mod 10;if d < 9 then beginm := m + d;end;n := n div 10;end;write(m);end. | n = int(input())m = 0while n > 1:    d = n % 10    if d < 9:        m = m + d    n //= 10print(m) |

**14.**Программист писал программу, для определения количества цифр в целом положительном числе, не большем 10^9.

|  |  |  |
| --- | --- | --- |
| **Алгоритмический** | **Паскаль** | **Python** |
| алгначцел N, sumввод Nsum := 1нц пока N > 1N := div(N, 10)sum := sum + 1КЦвывод sumКОН  | var N: longint;sum: integer;beginreadln(N);sum := 1;while N > 1 dobeginN := N div 10;sum := sum + 1;end;writeln(sum);end. | n = int(input())sum = 1while n > 1:    n //= 10    sum += 1print(sum) |

**15.**Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число *N*, не превосходящее 109, и выводится максимальная цифра этого числа.

|  |  |  |
| --- | --- | --- |
|  **Алгоритмический язык** | **Паскаль** | **Python** |
| алгнач    цел N, digit, max\_digit    ввод N    max\_digit := 10    нц пока N >= 10        digit := mod(N, 10)        если max\_digit < digit то            max\_digit := digit        все        N := div(N, 10)    кц    вывод max\_digitкон | var N: longint;    digit, max\_digit: integer;begin    readln(N);    max\_digit := 10;    while N >= 10 do    begin        digit := N mod 10;        if max\_digit < digit then            max\_digit := digit;        N := N div 10;    end;    writeln(max\_digit);end. | n = int(input())max\_digit = 10while n >= 10:    digit = n % 10    if max\_digit < digit:        max\_digit = digit    n //= 10print(max\_digit) |

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 8**

**Моделирование**

Цель: формирование основных понятий, видов и способов моделирования.

Теоретическая часть

Модель - упрощённое подобие реального объекта, процесса или явления, которое отражает его существенные особенности.

Сущность - обобщённое название объекта, явления или процесса, которое изучается с помощью моделирования.

Атрибуты (параметры) - характеристики сущности, которые учитываются в её модели.

Моделирование - метод познания, состоящий в создании и ис­следовании моделей изучаемых сущностей.

Классификация моделей по области использования

Учебные - используются при обучении.

Опытные - это уменьшенные или увеличенные копии проектируемого объекта. Используют для исследования и прогнозирования его будущих характеристик (аэродинамическая труба).

Научно-технические - для исследования процессов и явлений.

Игровые - репетиция поведения объекта в различных условиях.

Имитационные - отражение реальности в той или иной степени (это метод проб и ошибок).

**Классификация моделей по фактору времени**

Статические - описывают состояние системы в определенный момент времени (единовременный срез информации по данному объекту). Примеры: классификация животных, строение молекул, список посаженных деревьев, отчет об обследовании состояния зубов и т. д.

Динамические - описывают процессы изменения и развития системы (изменения объекта во времени). Примеры: моделирование движения тел, развития организмов, процесс химических реакций.

Классификация моделей по отрасли знаний - это классификация по отрасли деятельности человека (математические, биологические, химические, социальные, экономические, исторические и т. д.).

Классификация моделей по форме представления:

* материальные (предметные, физические) - это модели, которые имеют реальное воплощение и отражают внешние свойства или внутреннее устройство моделируемых сущностей, суть процессов и явлений в объекте-оригинале. Материальное моделирование - это экспериментальный метод познания окружающей среды. Примеры: детские игрушки, скелет человека, чучело, макет солнечной системы, школьные пособия, физические и химические опыты, авиамодель истребителя, полоса препятствий.
* информационные - это целенаправленно отобранная информация о моделируемой сущности, которая отражает ее свойства, наиболее существенные для исследователя. В информационных моделях реальный объект или процесс заменяется его формальным описанием. Такая процедура называется формализацией.

Например, информационной моделью движения поездов является расписание их движения, а материальной - макет железной дороги с движущимися паровозиками.

По уровню формализации различают:

* хорошо формализованные модели. Их можно решить средствами, принятыми в данной предметной области, не используя субъективные мнения экспертов;
* плохо формализованные модели. Их нельзя решить без при­влечения эксперта в данной предметной области.

**Типы информационных моделей**

Абстрактные (мысленные) - при построении модели используются понятия, не существующие в реальной жизни.

Вербальные - мысленные модели, выраженные в разговорной форме с помощью естественных языков.

Знаковые (формализованные) - выражены специальными символами, применяемыми в изучаемой предметной области. Например, компьютерная модель реализована средствами программной среды, математическая - формулами, которые описывают изучаемую сущность. Знаковая формализованная модель музыкального произведения - запись с помощью нот и т. д.

В знаковых информационных моделях выделяют класс образно-знаковых моделей. Например, к таким моделям относятся:

* геометрические - рисунок, пиктограмма, чертеж, карта, план, объемное изображение;
* структурные - таблица, граф, схема, диаграмма;
* алгоритмические - нумерованный список действий, пошаговое перечисление, блок-схема.

По способу организации данных информационные модели делятся на:

* реляционные (табличные}: перечень объектов и их свойств оформляется в виде связанных между собою таблиц. Каждая строка таблицы содержит информацию об одном экземпляре (сущности) предметной области, каждый столбец - значения одной и той же характеристики (атрибута) для разных сущностей. Пример: расписание движения поездов - это табличная информационная модель реального перемещения поездов по железной дороге;
* иерархические: объекты распределены по уровням. Каждый элемент высокого уровня состоит из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня.

Такие модели представляются ориентированным графом («деревом»), у которого начальная вершина не подчинена никакой другой, а все остальные подчинены только одной, но могут иметь в своём подчинении сколько угодно объектов нижнего уровня. Если из каждого узла выходит только два потомка, то такая структура называется бинарным деревом.

Примеры: файловая структура в компьютере (система каталогов), система доменных имён в Интернете, структура почтовых адресов, классификации животных, растений. В иерархической модели две любые вершины могут быть соединены только одним путём. Пример: относительный путь к файлу имеет только один вариант.

* сетевые: между объектами моделируемой системы существуют множественные связи. Такие модели представляются графом, в котором имеются связи между вершинами, позволяющие создать разные пути перехода между ними.

В настоящее время при моделировании плохо формализованных задач используют базы знаний. Базы знаний в отличие от баз данных содержат в себе не только фактическую информацию, которая является постоянной для данной предметной области, но и правила вывода, допускающие [автоматические](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B4%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE_%D1%82%D0%B5%D0%BE%D1%80%D0%B5%D0%BC) [умозаключения](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B4%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE_%D1%82%D0%B5%D0%BE%D1%80%D0%B5%D0%BC) о вновь вводимых фактах, то есть моделируют [«осмысленную»](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_%28%D0%BB%D0%B8%D0%BD%D0%B3%D0%B2%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0%29) обработку информации с помощью правил логики. Компьютерные модели таких задач называются интеллектуальными системами. Математические методы, по которым работают ин­теллектуальные системы, называются методами искусственного интеллекта. В их основе лежат эвристические приёмы. Эвристика - это неформализованная процедура, сокращающая количество шагов поиска решения за счет некоторого способа направленного поиска решения, а не простого перебора всех возможных вариантов. Часто считается, что база знаний отличается от [базы данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) именно наличием [механизма вывода](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4%D0%B0) новых знаний из старых. Еще одна особенность, которая закладывается в базы знаний, это выдача системой «объяснения» хода ее рассуждений при поиске ответа.

Наиболее известный класс интеллектуальных систем — это [экспертные системы](http://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) (ЭС). Эти системы рассматриваются как модели поведения [экспертов](http://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D1%86%D0%B5%D0%BD%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) в определенной области знаний. База знаний ЭС создается при помощи трех групп специалистов:

* эксперты той проблемной области, к которой относятся задачи, решаемые ЭС;
* инженеры по знаниям, являющиеся специалистами по разработке интеллектуальной информационной системы [(ИИС)](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0);
* программисты, осуществляющие реализацию ЭС.

ЭС может функционировать в 2-х режимах: ввод фактов по данной предметной области или проведение консультаций. В режиме ввода знаний эксперт с помощью инженера по знаниям вводит из­вестные ему сведения о предметной области в базу знаний ЭС. В режиме консультации пользователь ведет диалог с ЭС, сообщая ей сведения о текущей задаче и получает рекомендации ЭС, сопровождающиеся «объяснением хода ее рассуждений». Например, в сфере тушения лесных пожаров на основе информации о пожаре (описание растительности в данной области, сводка погоды и т. п.), выдаётся некая рекомендация относительно стратегии борьбы с данным пожаром (оценка требуемых ресурсов, рекомендации по размещению техники и т. п.)

Создание компьютерной модели разбивается на несколько этапов:

1. Постановка задачи: описание сущности, подлежащей моделированию, цели исследования и той информации о моделируе­мом объекте, которую необходимо учитывать в связи с поставленной целью.
2. Разработка модели: выбор типа создаваемой модели (предметная, компьютерная, математическая и т. п.) и ее реализация. Если выбран компьютерный тип модели, создается соответствующая программа.
3. Тестирование модели: проведение компьютерного или натурного эксперимента на тестах. Тест - набор данных, для которых заранее известен результат.

Для компьютерных программ применяются разные технологии тестирования. Если тестируется программа, которая получена путем внесения изменений, улучшающих ранее работавшую программу, то, чтобы убедиться, что внесенные изменения не испортили ее, в первую очередь ее проверяют на тех тестах, по которым проверяли предыдущую программу. Такое тестирование называется регрессионным. Тестирование новых сложных программ разбивается на два этапа: альфа-тестирование - это проверка программы на тестах, ко­торые составил ее разработчик. Бета-тестирование - передача про­граммы заказчикам в бесплатное пользование с условием, что при возникновении ошибок в работе разработчик будет уведомлен об этом.

1. Анализ результатов моделирования: процесс проверки правильности модели на совокупности тестов, охватывающих все диапазоны моделируемых параметров. Если результаты соответствуют цели, построение модели закончено, и можно использовать модель для прогнозирования реальных результатов. В противном случае модель уточняется за счет введения в нее новых данных, не учтенных на первом этапе, или изменяется ее реализация.

**Практическая часть**

**Заданиe 1.**

Коммивояжеру, живущему в городе A1, надо посетить города A2, A3, A4 причем каждый город надо посетить только один раз, и затем вернуться обратно в A1. Известно, что все города попарно соединены между собой дорогами, причем длины дорог bij между городами Ai и Aj (i=1,2,3,4 j=1,2,3,4) таковы: b12=30, b14=20, b23=50, b24=40, b13=70, b34=60.

Надо определить порядок посещения городов, при котором длина пути будет минимальна.

Изобразим каждый город точкой на плоскости и пометим ее соответствующей меткой Ai (i=1,2,3,4). Соединим эти точки отрезками прямых, которые будут изображать дороги между городами. Для каждой «дороги» укажем ее протяженность в километрах (рис. 8.1). Получился граф.

Рассматриваемая задача заключается в отыскании такого цикла на графе, проходящего через все четыре вершины, для которого сумма всех весов ребер минимальна. В нашем случае число вершин n=4 и число циклов равно 3.



Рисунок 8.1. Пример графа.

Найдем перебором все различные циклы, проходящие через четыре вершины и начинающиеся в А1:

1) A1 A4 A3 A2 A1;

2) A1 A3 A2 A4 A1;

3) A1 A3 A4 A2 A1;

Найдем теперь длины этих циклов (в км): L1=160, L2=180, L3=200. Итак, маршрутнаименьшей длины — это первый.

**Задание 2.**

На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова протяжённость дороги из пункта Д в пункт Е. В ответе запишите целое число - так, как оно указано в таблице.



Рисунок 8.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  |  | 9 |  |  | 7 |
|  |  |  |  | 5 |  | И |  |
|  |  |  |  |  |  | 12 |  |
|  | 9 | 5 |  |  | 4 | 13 | 15 |
|  |  |  |  | 4 |  | 10 | 8 |
|  |  | И | 12 | 13 | 10 |  |  |
|  | 7 | 7 |  | 15 | 8 |  |  |

**Задание 3.** На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, 3, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города А в город М, проходящих через город Ж?



Рисунок 8.3

**Задание 4.**

Между населёнными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | Z |
| A |  | 4 | 6 |  |  |  | 39 |
| B | 4 |  | 1 |  |  |  |  |
| C | 6 | 1 |  | 11 |  |  | 28 |
| D |  |  | 11 |  | 4 | 6 | 10 |
| E |  |  |  | 4 |  |  | 8 |
| F |  |  |  | 6 |  |  | 2 |
| Z | 39 |  | 28 | 10 | 8 | 2 |  |

Определите длину кратчайшего пути между пунктами A и Z (при условии, что передвигаться можно только по построенным дорогам).

**Задание 5.**

Сережа и Андрей играют в следующую игру. Перед ними лежат две кучи камней. Игроки ходят по очереди, первый ход делает Сережа. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в три раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций:

(10, 7), (30, 7), (10, 8), (10, 21). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 68. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 68 или больше камней.

В начальный момент в первой куче было 6 камней, во второй куче - S камней; 1 < S < 61.

Будем говорить, что игрок имеет выигрышную стратегию*,* если он может выиграть при любых ходах противника. Описать стратегию игрока - значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следуетвключать ходы играющего по этой стратегии игрока, не являющиеся для него, безусловно, выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания.

 **1.** а) Укажите все такие значения числа *S,* при которых Сережа может выиграть за один ход.

б) Известно, что Андрей выиграл своим первым ходом после неудачного первого хода Сережи. Укажите минимальное значение *S,* когда такая ситуация возможна.

 **2.**Укажите такое значение *S,* при котором у Сережи есть выигрышная стратегия, причём одновременно выполняются два условия:

* Сережа не может выиграть за один ход;
* Сережа может выиграть своим вторым ходом независимо от того, как будет ходить Андрей.

Для указанного значения S опишите выигрышную стратегию Сережи.

 **3**.Укажите значение *S,* при котором одновременно выполняются два условия:

* у Андрея есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Сережи;
* у Андрея нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Андрея.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Андрея (в виде рисунка или таблицы).

В узлах дерева указывайте позиции, на рёбрах рекомендуется указывать ходы. Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

**Задание 6.**

Два игрока. Света и Маша, играют в такую игру. Перед ними кучакамней. Игроки ходят по очереди, первой ход делает Света. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ход, неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 38. Если при этом в куче оказалось не более 66 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче 35 камней и Света удвоит количество камней в куче, то игра закончится и победителем будет Маша.

В начальный момент в куче было S каменей 1≤S ≤37.

Игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая может встретиться при различной игре противника.

Выполните следующие задания.

**1.** При каких значениях числа S света может выиграть в один ход?

Укажите все такие значения и соответствующие ходы Светы.

б) У кого из игроков есть выигрышная стратегия при S -36, 35,34?

Опишите выигрышные стратегии для этих случаев.

**2.** У кого из игроков есть выигрышная стратегия при S=18, 17? Опишите соответствующие выигрышные стратегии.

**3.** У кого из игроков есть выигрышная стратегия при S=16?

Постройте дерево всех партий, возможных при выигрышной стратегии (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ 9**

**Технология хранения, отбора и сортировки информации. Базы данных**

**Цель:** формирование представлений о базах данных и их видах; развитие навыков создания реляционных таблиц.

**Теоретическая часть**

Для структурированного хранения и обработки, связанных между собой данных используются базы данных.

База данных является информационной моделью организации данных предметной области.

Базы данных можно классифицировать по различным признакам. Выделяются реляционные, иерархические и сетевые модели данных. Реляционная модель базы данных была предложена в 1969 г. математиком, научным сотрудником фирмы IBM Э. Ф. Коддом.

 Таблицы состоят из столбцов и строк и содержат данные.

Строка таблицы соответствует некоторому объекту моделируемой предметной области. Строки таблицы называются записями. Записи разбиты на поля столбцами таблицы. Столбец таблицы описывает некоторый атрибут объектов и содержит значения этого атрибута. Столбцы таблицы называются полями. Таким образом, каждая запись представляет собой набор значений атрибутов данного объекта; значение каждого атрибута — в своем столбце. Каждое поле характеризуется своим именем (именем соответствующего атрибута) и типом данных, представляющих собой значения данного атрибута.

Каждая строка в таблице должна быть уникальной, т. е. иметь значения полей, отличающие ее от всех других строк. Набор полей, значения в которых уникальны для каждой строки, называется первичным ключом. Каждая таблица должна иметь первичный ключ, что позволяет однозначно идентифицировать каждую запись в таблице. Если такой ключ не объявлен, то им считаются все поля таблицы в совокупности.

В качестве первичного ключа можно использовать поле, содержащее данные типа счетчик, а также другие поля таблицы: код товара, инвентарный номер и т. п.

Тип поля определяется типом данных, которые оно содержит. Поля могут содержать данные следующих основных типов:

* Текстовый. Содержит до 255 символов.
* Числовой. Число.
* Счетчик. Вид числового типа. Последовательность целых чисел, которые задаются автоматически при вводе записей. Эти числа не могут быть изменены пользователем.
* Денежный. Вид числового типа. Число в денежном формате.
* Дата/Время. Дата и/или время.
* Логический. Значение Истина (Да) или Ложь (Нет ).
* Гиперссылка (например, в интерфейсе БД Microsoft Access). Ссылка на информационный ресурс в Интер­нете (например, Web-сайт).

Поле каждого типа имеет свой набор свойств. Наиболее важными свойствами полей являются:

* Размер поля. Определяет максимальную длину текстового или числового поля.
* Формат поля. Устанавливает формат данных.
* Непустое поле. Указывает на то, что данное поле обязательно надо заполнить.

 Иерархическая модель данных графически может быть представлена как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень) занимает один объект, второй — объекты второго уровня и т. д.

Сетевая модель данных. Сетевая модель данных является обобщением иерархической за счет допущения объектов, имеющих более одного предка, т. е. каждый элемент вышестоящего уровня может быть связан одновременно с любыми элементами следующего уровня. Вообще, на связи между объектами в сетевых моделях не накладывается никаких ограничений.

 Создание баз данных, а также обработка данных (в том числе операции отбора и сортировки данных) выполняются специальными комплексами программ — системами управления базами данных (СУБД).

Реляционные СУБД. В качестве примеров СУБД, использующих реляционную модель данных, можно привести в операционной системе Windows приложение Access, входящее в Microsoft Office, а в операционной системе Linux — систему управления базами данных Knoda.

Таблица-это базовый объект БД, все остальные объекты (запросы, формы, отчеты и пр.) создаются на основе существующих таблиц (производные объекты).

В реляционных СУБД запросы являются важным инструментом для пользователя. Главное предназначение запросов — это отбор данных, удовлетворяющих определенным условиям, в отсортированном (упорядоченном) по желанию пользователя виде.

 Формы позволяют отображать данные, содержащиеся в таблицах или запросах, в более удобном для восприятия виде. При помощи форм можно добавлять в таблицы новые данные, а также редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты.

Отчеты предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде.

Записи базы данных можно просматривать и редактировать в виде таблицы или в виде формы.

СУБД Access позволяет производить отбор записей, в которых значения определенных полей удовлетворяют заданным условиям. Отбор данных из базы данных можно производить с помощью фильтров. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и т. д.).

Запросы позволяют осуществлять отбор данных из баз данных так же, как и фильтры. Различие между ними состоит в том, что запросы являются самостоятельными объектами базы данных, а фильтры привязаны к конкретной таблице или форме.

В процессе создания запроса можно отбирать не только записи, но и поля, которые будут присутствовать в запросе.

Сортировка записей при выдаче их пользователю — это упорядочение записей по значениям одного или нескольких полей. Можно осуществлять печать непосредственно таблиц, форм и запросов. Однако для красивой печати документов целесообразно использовать отчеты. Отчеты являются производными объектами баз данных и создаются на основе таблиц, форм и запросов.

Межтабличная связь обеспечивает целостность данных. Связанные таблицы представляют собой единую базу данных, в которой можно создавать новые таблицы, а также запросы и отчеты, содержащие данные из связанных таблиц.

**Практическая часть**

**Решите тестовые задания**

**Задание 1.**

Представлена следующая база данных. Какой тип имеет поле «Отдел»

Изображение:

|  |  |  |
| --- | --- | --- |
| Отдел | Кол сотр | Нач. стд |
| 310а | 27 | Шпак |
| ТОГ в | 26 | Антонов |
| 215 | 30 | Чеботарев |
| 101 г | 18 | Ракитский |
| 112 | 24 | Кабанов |

1. счетчик;
2. в типе содержится ошибка;
3. числовой, с плавающей точкой;
4. символьный;
5. целочисленный.

**Задание 2.**

Сколько записей в таблице удовлетворяют условию Место <= 5 и (В>4 или МЗ>12)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Место | Команда | Б | Н | П | О | М3 | МП |
| 1 | Боец | 5 | 3 | 1 | 18 | 9 | 5 |
| 2 | Авангард | 6 | 0 | 3 | 18 | 13 | 7 |
| 3 | Опушка | 4 | 1 | 4 | 16 | 13 | 7 |
| 4 | Звезда | 3 | 6 | 0 | 15 | 5 | 2 |
| 5 | Химик | 3 | 3 | 3 | 12 | 14 | 17 |
| 6 | Пират | 3 | 2 | 4 | 11 | 13 | 7 |

* 1. 4
	2. 1
	3. 5
	4. 3
	5. 2

**Задание 3.**

По какому полю в следующей базе данных упорядочены записи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер | Фамилия | Улица | Дом | Квартира | Телефон |
| 1 | Иванов | Советска | 42 | 15 | 258-36-19 |
| 2 | Петров | Пушкина | 15/2 | 366 | 366-56-98 |
| 3 | Сидоров | Гоголя | 35 | 25 | 255-41-88 |
| 4 | Кузьмин | Гафури | 69 | 38 | 564-89-71 |

1. квартира;
2. фамилия;
3. улица;
4. дом;
5. телефон.

**Задание 4.**

Поле, значение которого не повторяется в различных записях, называется:

1. уникальным полем;
2. ключевым полем;
3. составным ключом;
4. типом поля;
5. именем поля.

**Задание 5.**

Условие поиска может задаваться с помощью:

1. только арифметического выражения;
2. символьной строки;
3. вызова справки;
4. простого или сложного логического выражения.

**Задание 6.**

Столбец однотипных данных в Access называется:

1. отчетом;
2. записью;
3. бланком;
4. полем.

**Задание 7.**

После проведения сортировки по полю «Фамилия И.О.» в порядке возрастания запись, содержащая номер телефона 568-98-00 переместится на:

|  |  |
| --- | --- |
| Фамилия И.О. | Телефон |
| Иванов И.И. | 234-56-98 |
| Иванова А.П. | 235-60-07 |
| Кедров А.К. | 435-88-78 |
| Иванов И.К. | 568-98-00 |
| Иванников Р.Т | 384-15-1 5 |

1. одну строку вверх;
2. не переместится;
3. две строки вверх;
4. одну строку вниз.

**Задание 8.**

Система управления базами данных - это:

1. программная система, поддерживающая наполнение и манипулирование данными в файлах баз данных;
2. набор программ, обеспечивающий работу всех аппаратных устройств компьютера и доступ пользователя к ним;
3. прикладная программа для обработки текстов и различных документов;
4. оболочка операционной системы, позволяющая более комфортно работать с файлами.

**Задание 9.**

Имеется база данных. Сколько в ней числовых полей:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Фамилия | Имя | Отчество | Год рождения | Класс | Школа |
| 1 | Сидоров | Павел | Ильич | 1993 | 7 | 105 |
| 2 | Смирнов | Станислав | Алексеевич | 1991 | 9 | 49 |
| 3 | Ефремов | Василии | Олегович | 1989 | 11 | 2 |
| 4 | Катин | Андрей | Никитич | 1990 | 10 | 5 |

1. 1
2. 5
3. 4
4. 20
5. 16

**Задание 10.**

В таблице представлен фрагмент БД «Расписание консультаций». Укажите первичный ключ таблицы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Дата | Время | Класс | Предмет | Учитель | Кабинет |
| 1 | 2июня | 9 | 11 | литература | Граина | 25 |
| 2 | 2 июня | 10 | 9 | алгебра | Кошкина | 31 |
| 3 | 2 июня | 10 | 11 | алгебра | Егоров | 29 |
| 4 | 4 июня | 9 | 9 | физика | Волков | 20 |
| 5 | 4 июня | 9 | 9 | история | Ларина | 34 |
| 6 | 4июня | 11 | 11 | физика | Кравова | 20 |

1. учитель;
2. время + класс;
3. предмет;
4. дата + время;
5. дата.

**Задание 11.**

Как называют организованную совокупность структурированных данных в определенной предметной области:

1. базой данных;
2. организованным структурированным списком;
3. организованным списком;
4. списком;
5. электронной таблицей.

**Задание 12.**

В поле файла реляционной базы данных могут быть записаны:

1. только номера записей;
2. только обозначение ключа;
3. данные только одного типа;
4. как числовые, так и текстовые данные одновременно;
5. только время создания записей.

**Задание 13.**

Записью в реляционных базах данных называют:

1. всю таблицу;
2. столбец таблицы;
3. строку таблицы;
4. ячейку таблицы;
5. имя поля.

**Задание 14.**

В представленной базе данных провели сортировку по возрастанию по полю «Класс». Какую строку стал занимать после сортировки Иванов?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Фамилия | Имя | Отчество | Год рожд. | Класс | Школа |
| 1 | Беляев | Иван | Петрович | 1985 | 11 | 45 |
| 2 | Катаев | Сергей | Иванович | 1986 | 9 | 195 |
| 3 | Иванов | Петр | Олегович | 1988 | 7 | 135 |
| 4 | Носков | Антон | Павлович | 1986 | 10 | 4 |

1. 4
2. 2
3. 3
4. 1

**Задание 15.**

С помощью системы управления базами данных (Microsoft Access или Open Office.org Base) создать базу данных «Моя группа», включающую столбцы (Фамилия — текстовый тип, Год рождения — дата, Возраст — числовой тип) и состоящую не менее чем из семи записей. Сохранить в папке Result файл базы данных. Произвести сортировку записей по первому столбцу.

**Задание 16.**

С помощью системы управления базами данных (Microsoft Access или Open Office.org Base) создать базу данных «Процессоры», включающую столбцы (Название — текстовый тип, Год выпуска — дата и Разрядность — числовой тип) и состоящую не менее чем из шести записей. Сохранить в папке. Result файл базы данных. Произвести сортировку записей по второму столбцу.

**Задание 17.**

С помощью системы управления базами данных (Microsoft Access или Open Office.org Base) создать базу данных «Переменные», включающую столбцы (Название — текстовый тип, Тип переменной — текстовый тип, и Количество занимаемых ячеек в оперативной па­мяти — числовой тип) и состоящую не менее чем из пяти записей. Сохранить в папке Result файл базы данных. Произвести сортировку записей по третьему столбцу.

**ИТОГОВЫЕ ТЕСТЫ**

1. Определите, сколько бит содержит сообщение: «На улице идёт дождь».

2. Определите, сколько бит может содержать сообщение: «Вылет самолёта задерживается».

3. Выберите верное утверждение:

1. в качестве материального носителя информации могут вы­ступать знания, сведения или сообщения;
2. в качестве носителя информации могут выступать только световые и звуковые волны;
3. в качестве носителя информации могут выступать материальные предметы;
4. информационные процессы являются материальным носителем информации.

4*.* Выберите верное утверждение: энтропия максимальна, если:

* 1. информация засекречена;
	2. события детерминированы;
	3. события равновероятны;
	4. информация точна.

5. Выберите верное утверждение: Прагматический аспект информации рассматривает:

* 1. информацию с точки зрения её практической полезности для получателя;
	2. даёт возможность раскрыть её содержание;
	3. показать отношения между смысловыми значениями её элементов;
	4. показывает количество информации в утверждении.

6*.* Выберите верное утверждение: свойство информации, заключающееся в достаточности данных для принятия решения, есть:

1. объективность;
2. полнота;
3. содержательность;
4. достоверность.

 7*.* Выберите верное утверждение: информация достовер­на, если она:

* 1. полезна;
	2. отражает истинное положение дел;
	3. достаточна для принятия решений;
	4. используется в современной системе обработки информации.

8*.* Выберите верное утверждение: сообщением в теории кодирования является:

1. электрический импульс, распространяемый в канале связи телефонной линии;
2. воспринятая, осознанная и ставшая личностно значимой ин­формация;
3. набор данных, объединённых смысловым содержанием и пригодных для обработки и передачи по каналам связи;
4. процесс переноса или копирования данных по некоторым признакам с одного места на другое с целью сортировки, фор­мирования результирующих документов.

9*.* Выберите верное утверждение: цепочка костров, зажигающаяся при необходимости оповещения «Горит – да», «Не горит – нет» - это:

1. линия передачи сообщения;
2. неадекватное поведение людей;
3. способ обработки информации;
4. шифрование информации.

10*.* Выберите верное утверждение: в вычислительной технике в качестве основной используется система счисления:

1. шестнадцатеричная;
2. десятичная;
3. восьмеричная;
4. двоичная.

11. Выберите верное утверждение: сканирование книги является операцией данных:

1. верификации;
2. преобразования;
3. архивирования;
4. транспортировки.

12. Представьте десятичное число 1023 в двоичной системе.

13. Расположите в возрастающей последовательности следующие единицы измерения информации: 1Кб, 1Мб, 1Тб, 1Гб.

14. Определите, сколько двоичных разрядов необходимо для кодирования двадцати состояний.

15. Сколько информации содержится в одном разряде двоичного числа?

16. Какой объем памяти потребуется для хранения 32 символов в кодировке KOI-8?

17. Определите последнюю цифру суммы чисел 558 и 568 в шестнадцатеричной системе счисления.

18. Отсортируйте по возрастанию последовательность текстовых величин: 8б; 8а; 10а; 10б; 11а.

19. Упорядочьте по убыванию последовательность чисел: 10 бит, 20 бит, 2 байта.

20. Определите объём памяти, который потребуется для кодировки фразы «Я помню чудное мгновенье» в Unicode.

21. Представьте в десятичной системе результат суммирования 1112 и 1112.

22. Числа в двоичной системе имеют вид: 101012 и 10002. Какой вид имеет их разность?

23. Представьте в двоичной системе результат вычисления 27+24+1.

24. Упорядочьте по возрастанию последовательность чисел: 558, 5516, 557.

25. Упорядочить по убыванию последовательность чисел: 10 бит, 20 бит, 2 байта.

26. Укажите, к какому типу относится информация, представленная в виде слов.

27*.*Определить информационный объём фразы «Я помню чудное мгновенье» при алфавитном подходе к определению количества информации, если считать, что информационная емкость буквы русского алфавита равна 5 битам.

28. Модель есть замещение изучаемого объекта другим объектом, который отражает:

а) все стороны данного объекта;

б) некоторые стороны данного объекта;

в) существенные стороны данного объекта;

г) несущественные стороны данного объекта.

29. Результатом процесса формализации является:

1. описательная модель;
2. графическая модель;
3. математическая модель;
4. предметная модель.

30. Информационной моделью организации занятий в школе является:

а) свод правил поведения учащихся;

б) расписание уроков;

в) список класса;

г) перечень учебников.

31. Материальной моделью является:

а) макет самолеты;

б) чертеж;

в) карта;

г) диаграмма.

32. Генеалогическое дерево семьи является:

а) табличной информационной моделью;

б) иерархической информационной моделью;

в) сетевой информационной моделью;

г) словесной информационной моделью.

33. Знаковой моделью является:

а) анатомический муляж;

б) модель корабля;

в) макет здания;

 г) диаграмма.

34. Укажите в моделировании процесса исследования температурного режима комнаты объект моделирования:

а) конвекция воздуха в комнате;

б) исследование температурного режима комнаты;

в) комната;

г) температура.

35. Правильный порядок указанных этапов математического моделирования процесса:

1) анализ результата; 3) определение целей моделирования;

2) проведение исследования; 4) поиск математического описания.

Соответствует последовательности:

а) 3 – 4 – 2 – 1; в) 2 – 1 – 3 – 4;

б) 1 – 2 – 3 – 4; г) 3 – 1 – 4 – 2;

36. Из скольких объектов, как правило, состоит система?

а) из нескольких; в) из бесконечного числа;

б) из одного; г) она не делима.

37. Как называется граф, предназначенный для отображения вложенности, подчиненности, наследования и т.п. между объектами?

а) схемой; в) таблицей;

б) сетью; г) деревом.

38. Отличительной чертой интеллектуальных систем является \_\_\_.

39. Упорядочение информации по определенному признаку называется:

а) сортировкой; в) систематизацией;

б) формализацией; г) моделированием.

40. Дана таблица моделирования:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | моделируемый процесс | A | ракета |
| 2 | моделируемый объект | B | исследование траектории |
| 3 | цель моделирования | C | полёт ракеты |
| 4 | моделируемые характеристики | D | координаты местоположения |

Укажите правильный порядок установки соответствия в этой таблице.

41. Как называется упрощенное представление реального объекта?

а) оригинал; в) модель;

б) прототип; г) система.

42. Процесс построения моделей называется:

а) моделирование; в) экспериментирование;

б) конструирование; г) проектирование

43. Информационная модель, состоящая из строк и столбцов, называется:

а) таблица; в) схема;

б) график; г) чертеж.

44. Каково общее название моделей, которые представляют собой совокупность полезной и нужной информации об объекте?

а) материальные; в) предметные;

б) информационные; г) словесные.

45. Схема электрической цепи является:

а) табличной информационной моделью;

б) иерархической информационной моделью;

в) графической информационной моделью;

г) словесной информационной моделью

46. Знаковой моделью является:

а) карта; в) глобус;

б) детские игрушки; г) макет здания.

47. Укажите в моделировании процесса исследования температурного режима комнаты цель моделирования:

а) конвекция воздуха в комнате;

б) исследование температурного режима комнаты;

в) комната;

г) температура.

48. Правильные определения понятий приведены в пунктах

1) моделируемый параметр – признаки и свойства объекта – оригинала, которыми должна обязательно обладать модель;

2) моделируемый объект-предмет или группа предметов, структура или поведение которых исследуется с помощью моделирования;

3) закон – поведение моделируемого объекта.

а) 1 – 2 – 3; в) 1 – 3;

б) 2 – 3; г) 1 – 2.

49. Инструментом для компьютерного моделирования является:

а) сканер; в) принтер;

б) компьютер; г) монитор.

50. Как называется средство для наглядного представления состава и структуры системы?

а) таблица; в) текст;

б) граф; г) рисунок.

51. Какие методы используют для плохо формализованных задач?

52. Решение задачи автоматизации продажи билетов требует использования:

а) графического редактора; в) операционной системы;

б) текстового редактора; г) языка программирования.

53. Дана таблица моделирования:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | моделируемый процесс | A | человек |
| 2 | моделируемый объект | B | разработка метода лечения |
| 3 | цель моделирования | C | температура и давление |
| 4 | моделируемыехарактеристики | D | влияние лекарства на состояние больного организма |

54.Укажите, какие модели относятся к классу предметных:

1. модель молекулы в виде кристаллической решётки;
2. алгоритм работы станка с числовым программным управлением;
3. макет нефтяной вышки;
4. электрическая схема радиоприемника.

55. Укажите типы моделей в классификации по фактору времени.

56. Эвристика - это .

57. Аэродинамическая труба это \_\_\_.

58.Укажите типы тестирования компьютерных моделей\_\_\_\_\_\_\_\_

59. Информационной (знаковой) моделью является:

1. анатомический муляж;
2. макет здания;
3. модель корабля;
4. химическая формула.

60. Материальной моделью является:

1. анатомический муляж;
2. техническое описание компьютера;
3. рисунок функциональной схемы компьютера;
4. программа на языке программирования.

61.Какие пары объектов находятся в отношении «объ­ект - модель»?

1. компьютер - данные;
2. компьютер - его функциональная схема;
3. компьютер - программа;
4. компьютер - алгоритм.

62. Какая модель является статической (описывающей состояние объекта)?

1. формула равноускоренного движения;
2. формула химической реакции;
3. формула химического соединения;
4. второй закон Ньютона.

63. Укажите отличие модели от реального объекта, явления или процесса.

64. Как называются модели, в которых на основе анализа различных условий принимается решение?

а) словесные;

б) графические;

в) табличные;

 г) логические.

65. Устное представление информационной модели называется:

а) графической моделью;

б) словесной моделью;

в) табличной моделью;

 г) логической моделью.

66. Дана таблица моделирования:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | моделируемый процесс | A | ракета |
| 2 | моделируемый объект | B | исследование траектории |
| 3 | цель моделирования | C | полёт ракеты |
| 4 | моделируемые характеристики | D | координаты местоположения |

Укажите правильный порядок установки соответствия в этой таблице.

67. База данных это…

1. прикладная программа позволяющая обрабатывать большие объемы различной информации;
2. разнообразные данные, организованные в виде таблицы;
3. совокупность данных из некой предметной области, организованных специальным образом на внешней памяти компьютера.

68. Программы для работы с базами данных:

1. OpenOffice Calc
2. OpenOffice Base
3. Microsoft Word.

69. СУБД имеющие своим недостатком слабую защиту от неправомерного доступа к данным:

1. файл-серверные СУБД;
2. клиент-серверная СУБД.

70. Транзакция это…

1. операции выполняются по порядку, одна за другой;
2. многошаговая операция, которая производит сохранение данных после каждого шага;
3. многошаговая операция по выполнению единого действия, которая выполняется целиком или не выполняется вообще.

71. Столбцы таблицы называются:

1. записи;
2. поля;
3. индексы;
4. ключи.

72. Идея реляционной теории Кодда подразумевает:

1. порядок записей в таблице определен;
2. в таблице нет двух одинаковых записей;
3. в таблице описывается несколько классов объектов.

73. Каких объектов нет в базе данных?

1. отчет;
2. запрос;
3. фильтр;
4. таблица.

74. Для хранения номера телефона в базе данных используется тип данных:

1. текстовое;
2. логическое;
3. вещественное;
4. памятка.

75. Для создания связей между таблицами можно использовать:

1. любые ключевые поля;
2. поля одинакового типа;
3. связи устанавливаются между записями.

76. Сортировка это…

1. расстановка данных в определенном порядке;
2. отбор данных удовлетворяющих выбранному условию;
3. отбор записей или выполнение других операций с данными.

77. Для эффективной работы БД должно выполняться условие:

1. непротиворечивости данных;
2. достоверности данных;
3. объективности данных.

78. Поле «Счетчик» отличается тем, что:

1. обязательно должны вводиться целые числа;
2. в поле хранится только значение, а сами данные в другом поле;
3. в нем происходит автоматическое наращивание.

79. Какая функция позволяет выбрать несколько атрибутов сразу из нескольких таблиц и получить новую таблицу с результатом?

1. форма;
2. запрос;
3. отчет.

80. Для чего предназначены формы в MS Access?

1. для ввода данных в удобном порядке;
2. для вывода данных в удобном формате;
3. для представления конечной информации в удобном виде.

81. Преимущество локальных информационных систем:

1. переносимость;
2. автономность;
3. состыкованность изменений, вносимых пользователями.

82. Программа не предназначенная для работы с базами данных:

1. OpenOffice Calc;
2. OpenOffice Base;
3. Microsoft Access.

83. СУБД имеющие своим недостатком создание лишней нагрузки на сеть:

1. файл-серверные СУБД;
2. клиент-серверная СУБД.

84. Первичный ключ это…

1. запись таблицы;
2. многошаговая операция, которая производит сохранение данных после каждого шага;
3. поле, однозначно определяющее запись.

85. Строки таблицы называются:

1. записи;
2. поля;
3. индексы;
4. ключи.

86. К достоинству однотабличной базы данных относится:

1. защита от ошибок ввода данных (опечаток);
2. простой поиск информации;
3. нет дублирования данных.

87. Каких объектов нет в базе данных?

1. таблицы;
2. формы ;
3. запросы;
4. связи.

88. Для хранения номера автомобиля в базе данных используется тип данных:

1. текстовое;
2. логическое;
3. вещественное;
4. памятка.

89. Созданные связи между двумя таблицами не могут иметь тип:

1. один к одному;
2. многие ко многим;
3. один ко многим.

90. Нормализация это…

1. разделение единой таблицы базы данных на несколько, для дальнейшего связывания таблиц;
2. добавление, изменение и удаление записей и таблицу;
3. изменение структуры базы данных с целью устранения избыточности и нарушения целостности.

91. Какой символ заменяет все при запросе в БД?

1. символ \*;
2. символ ";
3. символ &.

92. Что позволяет автоматизировать ввод данных в таблицу?

1. шаблон;
2. значение по умолчанию;
3. список подстановки.

93. Запросы создаются с помощью:

1. мастера запросов;
2. службы запросов;
3. клиента запросов.

94. Основные понятия иерархической БД:

1. таблица, столбец, строка;
2. уровень, узел, связь;
3. отношение, атрибут, кортеж.

95. Базы данных -это:

1. сложная программа, направленная учет входящей информации;
2. наборы данных, находящиеся под контролем систем управления;
3. бесконечный объем данных, постоянно управляющийся с помощью СУБД.

96. Основное отличие реляционной БД:

1. данные организовываются в виде отношений;
2. строго древовидная структура;
3. представлена в виде графов.

97. Расширением файла БД является:

1. .f2;
2. .mdb, .db;
3. .mcs.

98. Слово Null в БД используется для обозначения:

1. неопределенных значений;
2. пустых значений;
3. нуля.

99. Что такое кортеж?

1. совокупность атрибутов;
2. множество пар атрибутов и их значений;
3. схема отношений данных.

100. Мощность отношений - это:

1. количество веток в графовой системе;
2. порядок подчинения данных в древовидной структуре БД;
3. количество кортежей в отношении.

101. Главное условие сравнимых отношений:

1. одинаковая схема отношений;
2. точное количество сравнимых признаков;
3. наличие количественности признаков.

102. Операция проекции направлена на:

1. накладывание данных одной БД на данные другой БД;
2. выборку данных согласно заданным атрибутам;
3. сравнение БД на основе схожести.

103. В отличие от пользовательского типа данных базовые типы данных:

1. присутствуют в БД изначально;
2. должны быть в любой БД;
3. имеют более простую структуру.

104. Если а - это цена, б - масса, то атрибут с, обозначающий стоимость будет:

1. базовым атрибутом;
2. виртуальным атрибутом;
3. сложным атрибутом.

105. Подсхема исходной схемы, состоящая из одного или нескольких атрибутов, для которых декларируется условие уникальности значений в кортежах отношений называется…

1. глобальная схема отношений;
2. ключ;
3. отчет.

106. Индекс для подсхемы, состоящей из нескольких атрибутов называется…

1. составной;
2. неуникальный;
3. сложный.

107. В MS Access нельзя осуществить запрос на:

1. обновление данных;
2. создание данных;
3. добавление данных.

108. MS Access при закрытии программы:

1. предлагает сохранить БД;
2. автоматически сохраняет при вводе данных;
3. автоматически сохраняет при закрытии программы.

**СПИСОК ЛИТЕРАТУРЫ**

1. Егупова, М.В. Практические приложения математики в школе [Электронный ресурс]: учебное пособие для студентов педагогических вузов / М.В. Егупова. — Электрон. текстовые данные. — М.: Прометей, 2015. — 248 c. — 978-5-9906264-5-4. — Режим доступа: http://www.iprbookshop.ru/58178.html
2. Информатика [Текст]: учебник/ под ред. В.В. Трофимова.- М.: Юрайт, 2011.- 911 с.
3. Информатика [Электронный ресурс]: учебное пособие/ — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 178 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/66024.html
4. Королев, В.Т. Математика и информатика. Часть первая. Математика [Электронный ресурс]/ В.Т. Королев, Д.А. Ловцов, В.В. Радионов. — Электрон. текстовые данные. — М.: Российский государственный университет правосудия, 2015. — 248 c. — 978-5-93916-462-7. — Режим доступа: http://www.iprbookshop.ru/45225.html
5. Кучугурова, Н.Д. Интенсивный курс общей методики преподавания математики [Электронный ресурс]: учебное пособие/ Н.Д. Кучугурова. — Электрон. текстовые данные. — М.: Московский педагогический государственный университет, 2014. — 152 c. — 978-5-4263-0169-6. — Режим доступа: http://www.iprbookshop.ru/70123.html
6. Методика изучения математики в основной школе [Электронный ресурс]: курс лекций для организации самостоятельной работы студентов по вопросам частных методик/ ГЛ. Васильева [и др.]. — Электрон. текстовые данные. — Пермь: Пермский государственный гуманитарно-педагогический университет, 2011. — 96 c. — 978-5-85218-547-1. — Режим доступа: http://www.iprbookshop.ru/32214.html
7. Попов, А.М. Информатика и математика [Электронный ресурс]: учебное пособие для студентов вузов/ А.М. Попов, В.Н. Сотников, Е.И. Нагаева. — Электрон. текстовые данные. — М.: ЮНИТИ-ДАНА, 2017. — 302 c. — 978-5-238-01396-1. — Режим доступа: <http://www.iprbookshop.ru/71195.html>
8. Рихтер, Т.В. Избранные вопросы методики преподавания информатики [Электронный ресурс]: методическое пособие/ Т.В. Рихтер. — Электрон. текстовые данные. — Соликамск: Соликамский государственный педагогический институт, 2010. — 115 c. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47868.html>

БИДЖИЕВА Сапият Ханапиевна

МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ И МАТЕМАТИКИ В ШКОЛЕ

ЧАСТЬ 2

Лабораторный практикум

для обучающихся 1 курса по направлению подготовки

09.03.03. Прикладная информатика

Корректор Чагова О.Х.

Редактор Чагова О.Х.

Сдано в набор 04.04.2022г.

Формат 60х84/16

Бумага офсетная

Печать офсетная

Усл.печ. 5,34

Заказ № 4568

Тираж 100 экз.

Оригинал –макет подготовлен

В библиотечно-издательском центре СКГА

369000, г. Черкесск, ул. Ставропольская, 36