РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

Соколова О.И.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ И САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

МДК.02.02 «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

для специальности
09.02.09 Веб-разработка

Ростов-на-Дону
2025

СОДЕРЖАНИЕ

Введение 4

Лабораторная работа №1 «Представление чисел в позиционных системах счисления» 6

Лабораторная работа №2 «Арифметические действия в системах счислениях» 13

Лабораторная работа №3 «Битовая длина числа и диапазон значений» 21

Лабораторная работа №4 «Определение количества бит для кодирования чисел» 25

Лабораторная работа №5 «Определение количества возможных значений при заданной разрядности» 28

Лабораторная работа №6 «Определение объёма памяти для хранения числовых данных» 30

Лабораторная работа №7 «Кодирование символов в ASCII и Unicode» 34

Лабораторная работа №8 «Определение объёма текстовой информации» 41

Лабораторная работа №9 «Определение количества символов по объёму файла» 47

Лабораторная работа №10 «Расчёт объёма графической информации» 49

Лабораторная работа №11 «Расчёт объёма звуковой информации» 55

Лабораторная работа №12 «Шифр Цезаря» 61

Лабораторная работа №13 «Шифры замены» 65

Лабораторная работа №14 «Шифры перестановки» 70

Лабораторная работа №15 «Сравнение методов шифрования» 75

Лабораторная работа №16 «Определение хэш-сумм файлов» 79

Лабораторная работа №17 «Проверка целостности данных» 82

Лабораторная работа №18 «Анализ свойств файлов» 84

Лабораторная работа №19 «Работа с архивами и защитой паролем» 86

Лабораторная работа №20 «Настройка параметров безопасности» 90

Лабораторная работа №21 «Антивирусная защита» 99

2 ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ 104

3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ 107

4 МЕТОДИКА ВЫПОЛНЕНИЯ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ 116
5 МЕТОДЫ КОНТРОЛЯ И ОЦЕНКА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ 116
СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ 117

Введение

Методические указания к лабораторным работам и по выполнению самостоятельной работы студентов составлены в соответствии с ФГОС СПО и рабочей программой профессионального модуля МДК.02.02 «Информационная безопасность», которые являются частью программы подготовки специалистов среднего звена специальности 09.02.09 Веб-разработка.

Рабочей программой дисциплины МДК.02.02 «Информационная безопасность» предусмотрено на выполнение практических работ – 42 часа и самостоятельной работы студентов – 32 часа.

При выполнении лабораторных работ и при организации самостоятельной работы студентов используются активные и интерактивные формы обучения - просмотр и обсуждение учебных видеофильмов, групповая дискуссия, лекция - консультация, моделирование производственных процессов и ситуаций, обсуждение в группах, тренинг, кейс-метод, защита практических и лабораторных работ и другие.

Цель методических рекомендаций - оказание методической помощи студентам в выполнении лабораторных работ и в организации их самостоятельной работы по изучению учебного материала, для расширения, углубления и закрепления знаний и умений, а также формирования профессиональных (ПК) компетенций.

Код и содержание компетенции	Уметь	Знать
ПК 2.3 - Настраивать права пользователей в соответствии с функциональными задачами (ролями) и на основании информации о поведенческих факторах.	Настраивать учетные записи пользователей; разграничивать права доступа; применять методы аутентификации и авторизации.	Принципы управления доступом; модели разграничения прав пользователей; основные методы аутентификации и идентификации.
ПК 2.4 - Применять программные средства обеспечения безопасности информации веб приложений	Использовать базовые средства защиты веб-приложений; применять методы хэширования и шифрования данных; выявлять типовые	Основные угрозы информационной безопасности веб-приложений; принципы защиты данных при хранении и передаче; основы

	угрозы и уязвимости веб-ресурсов.	криптографических методов защиты информации.

Лабораторная работа №1 «Представление чисел в позиционных системах счисления»

Цель работы: сформировать практические навыки представления числовой информации в системах счисления и определения разрядности данных, необходимых для понимания принципов хранения информации в памяти компьютера и обеспечения её целостности.

Теоретические сведения

В современных информационных системах вся информация независимо от её вида представляется в **цифровой форме**. Тексты, изображения, видео, звук и числовые данные в конечном итоге преобразуются в наборы двоичных кодов. Это обусловлено принципом работы электронных устройств, которые надёжно различают только два устойчивых состояния: наличие электрического сигнала и его отсутствие. Эти состояния кодируются цифрами **1** и **0**.

Минимальной единицей информации является **бит**. Один бит может принимать только два значения: 0 или 1. Восемь бит образуют **байт**, который является базовой единицей измерения объёма информации в вычислительной технике.

Любое число, вводимое пользователем в десятичной системе счисления, перед обработкой преобразуется компьютером в двоичную форму. Корректность такого представления имеет прямое отношение к информационной безопасности, так как:

- искажение одного бита может привести к повреждению данных;
- неправильный выбор разрядности приводит к переполнению памяти;
- ошибки хранения чисел могут нарушить целостность баз данных и программ.

Таким образом, понимание принципов представления числовой информации является основой надёжной обработки и защиты данных.

Система счисления – это способ записи чисел с помощью определённого набора символов (цифр). Компьютерная техника использует позиционные системы счисления, в которых значение цифры определяется не только её величиной, но и позицией в числе.

Общее правило записи числа в позиционной системе:

$$\chi = a_n \times p^n + a_{n-1} \times p^{n-1} + \cdots + a_0 \times p^0$$

где

p – основание системы счисления;

a_i – цифры числа.

На практике используются следующие системы:

Система	Основание	Цифры
Десятичная	10	0–9
Двоичная	2	0, 1
Восьмеричная	8	0–7
Шестнадцатеричная	16	0–9, A–F

Пример разложения числа:
Десятичное число 345:

$$345_{10} = 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

Двоичное число 10110_2 :

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{10}$$

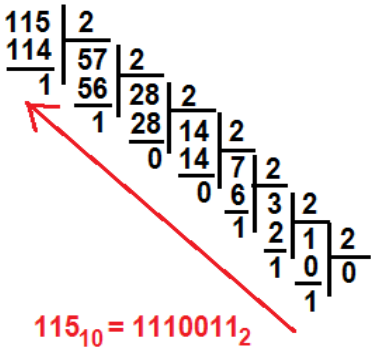
Перевод целого числа из десятичной системы счисления в двоичную.

Алгоритм

1. Последовательно выполнить деление исходного целого десятичного числа и получаемых целых частных на основание системы (на 2) до тех пор, пока не получится частное, меньшее делителя (т.е. меньшее 2).

2. Записать полученные остатки в обратной последовательности.

Перевод числа 115 в двоичную систему.



Перевод дробного числа из двоичной системы счисления в десятичную.

Пример.

$10110{,}01_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 16 + 0 + 4 + 2 + 0 + 0 + \frac{1}{2} = 22{,}05$

1.

Перевод дробного числа из десятичной системы счисления в двоичную.

Алгоритм.

1. Последовательно умножать (в исходной системе счисления) данное число и получаемые дробные части произведений на основание новой системы (на 2) до тех пор, пока дробная часть произведения не станет равной нулю или будет достигнута требуемая точность представления данного числа.

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

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

0,		5625
	x	2
1		1250
	x	2
0		2500
	x	2
0		5000
	x	2
1		0000

Пример.

Перевести число **$0{,}375_{1\ 0}$ в двоичную систему счисления.**

Целая часть	*Дробная часть*	*× 2*
0	0,375	0,75
0	0,75	1,5
1	0,5	1,0
1	0,0	—

Записываем целые части **сверху вниз в порядке получения**:

$$0{,}375_{10} = 0{,}011_2$$

Процесс умножения продолжается до тех пор, пока дробная часть не станет равной нулю либо не будет достигнута требуемая точность.

Разрядность данных – это количество бит, выделенных для хранения одного числа в памяти.

Если число хранится в n битах, то количество различных комбинаций равно: **2^n различных значений**

Каждая комбинация соответствует одному возможному значению.

Для **беззнаковых целых чисел** диапазон значений:

$$0 \le x \le 2^n - 1$$

Пример 1

Сколько значений можно закодировать 8 битами?

$$2^8 = 256$$

Диапазон: от 0 до 255.

Минимальное количество бит для хранения числа

Чтобы определить, сколько бит нужно для хранения числа, необходимо найти минимальное n, при котором:

$$2^n > X$$

Пример 2

Какое минимальное количество бит нужно для хранения числа 200?

$$2^7 = 128;\ 2^8 = 256$$

$$128 < 200 < 256$$

Ответ: требуется **8 бит.**

В системе счисления с основанием q=4 используют цифры 0–3; с q=8 – цифры 0–7; с q=16– цифры 0–9 и буквы A, B, C, Д, E, F. Здесь A, B, C, Д, E, F обозначают соответственно цифры 10, 11, 12, 13, 14, 15.

Вес разряда pi числа в позиционной системе счисления есть отношение вида pi=qi/q0=qi, где i-номер разряда справа налево.

Если разряд имеет вес pi=qi=, то следующий старший разряд будет иметь вес pi+1=qi−1. Таким образом, в позиционной системе счисления вес разряда определяется его положением (позицией) в числе.

Десятичная	Двоичная	Четверичная	Восьмеричная	Шестнадцатеричная
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8

9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F

Практическое задание

Для своего варианта выполнить все пункты задания.

Необходимо:

1. Определить минимальное количество бит, достаточное для хранения указанного числа.
2. Определить максимальное число, которое можно закодировать этим количеством бит.
3. Перевести исходное число в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 1

Исходное число: **85** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 85.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 85 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 2

Исходное число: **300** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 300.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 300 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 3

Исходное число: **1000** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 1000.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 1000 в двоичную систему счисления.
4. Указать, сколько различных значения можно представить с найденной разрядностью.

Вариант 4

Исходное число: **17** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 17.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 17 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 5

Исходное число: **511** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 511.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 511 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 6

Исходное число: **65** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 65.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 65 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 7

Исходное число: **1023** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 1023.

2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 1023 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 8

Исходное число: **255** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 255.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 255 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 9

Исходное число: **4095** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 4095.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 4095 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Вариант 10

Исходное число: **31** (в десятичной системе)

Задание:

1. Определить минимальное количество бит, достаточное для хранения числа 31.
2. Определить максимальное десятичное число, которое можно закодировать этим количеством бит.
3. Перевести число 31 в двоичную систему счисления.
4. Указать, сколько различных значений можно представить с найденной разрядностью.

Контрольные вопросы

Лабораторная работа №2 «Арифметические действия в системах счислениях»

Цель работы:

1. Выполнение арифметических операций в двоичной системе счисления.

Для двоичной системы счисления правила выполнения арифметических операций над двоичными числами остаются такими же, как и в привычной для всех десятичной системе счисления.

Основой выполнения арифметических операций являются следующие таблицы сложения, вычитания и умножения одноразрядных чисел:

Таблица сложения

	0	**1**
0	0	1
1	1	10 ←

единица переноса в старший разряд

Таблица вычитания

	0	**1**
0	0	1 → 10-1=1
1	1	0

с учетом заёма единицы из старшего разряда

. Таблица умножения

	0	**1**
0	0	0
1	0	1

Сложение двух чисел в двоичной системе можно выполнить столбиком, начиная с младших разрядов. При этом в каждом разряде складываются две цифры одноименных разрядов (в соответствии с таблицей сложения) и единицы переноса из соседнего младшего разряда, если он имел место. В результате сложения получим цифру соответствующего разряда суммы и возможную единицу переноса в старший соседний разряд.

Вычитание чисел, как и сложение, также выполняется столбиком (в соответствии с таблицей вычитания). Особым случаем является тот, когда необходимо занимать единицу из соседнего старшего разряда, которая равна двум единицам данного разряда.

Умножение двоичных многоразрядных чисел осуществляется последовательным сложением частичных произведений, каждое из которых (в соответствии с таблицей умножения) равно множителю, сдвинутому на соответствующее число разрядов, если в разряде множителя стоит единица, или нулю, если в разряде множителя стоит 0.

Деление двоичных чисел производится аналогично делению десятичных чисел, но с учетом специфики операции вычитания двоичных чисел. Положение запятой результата умножения и деления определяется так же, как и для десятичных чисел.

```
  101100          101,1001
+   1011        +   1,1111
--------        ----------
  110111          111,1000
```

Рисунок 3 – Пример сложения чисел в двоичной системе счисления

```
  110111          111,1000
-   1011        -   1,1111
--------        ----------
  101100          101,1001
```

Рисунок 4 – Пример вычитания чисел в двоичной системе счисления

```
   101
 × 11
 ----
   101
+ 101
 ----
  1111
```

Рисунок 5 – Пример умножения чисел в двоичной системе счисления

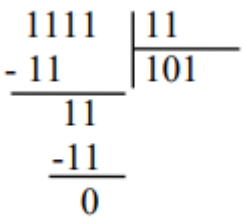


Рисунок 6 – Пример деления чисел в двоичной системе счисления

Восьмеричная и шестнадцатеричная системы счисления относятся к классу двоично-кодированных систем, так как основание этих систем представляют целые степени двойки: 2^3 – для восьмеричной и 2^4 – для шестнадцатеричной системы счисления.

Для систем счисления с основанием q ≤10 для изображения цифровых символов используются цифры от 0 до (q−1), а для q>10помимо цифр используются первые шесть букв латинского алфавита.

Большим достоинством восьмеричной и шестнадцатеричной систем счисления является, во-первых, возможность более компактно представить запись двоичного числа, а именно, запись одного и того же двоичного числа в восьмеричной и шестнадцатеричной системах будет соответственно в 3 и 4 раза короче двоичной. Во-вторых, сравнительно просто осуществляется преобразование чисел из двоичной в восьмеричную и шестнадцатеричную системы и наоборот. Действительно, так как для восьмеричного числа каждый разряд представляется группой из трех двоичных разрядов (триад), а для шестнадцатеричного – группой из четырех двоичных разрядов (тетрад), то для такого преобразования достаточно объединить двоичные цифры в группы по 3 и 4 бита соответственно, продвигаясь от раздельной запятой вправо и влево. При этом в случае необходимости добавлять нули в начале и в конце числа и каждую такую группу – триаду или тетраду – заменяют эквивалентной восьмеричной или шестнадцатеричной цифрой.

Указанные достоинства восьмеричных и шестнадцатеричных систем счисления определили использование их при составлении программ для более короткой и удобной записи двоичных чисел, команд и специальных двоичных слов, с которыми оперирует ЭВМ. Особенно оказалось удобным использование шестнадцатеричной системы, когда разрядность чисел и команд выбрана кратной байту, при этом каждый двоичный код байта запишется в виде 2-разрядного шестнадцатеричного числа.

Использование шестнадцатеричной системы счисления в ЭВМ общего назначения, как будет видно из дальнейшего изложения, позволяет расширить допустимый диапазон представления нормализованных чисел.

Представим таблицы сложения и умножения для шестнадцатеричной системы счисления.

Таблица сложения в шестнадцатеричной системе счисления.

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

Таблица умножения в шестнадцатеричной системе счисления.

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Практическое задание

Задание 1. Арифметические операции в двоичной системе

Выполните операции столбиком:

1. 1011+ 1101
2. 10010−101

3. 101×11
4. 1110÷10

Задание 2. Преобразование систем счисления

1. Перевести двоичное число 101110_2 в восьмеричное и шестнадцатеричное.
2. Перевести восьмеричное число 57_8 в двоичное и шестнадцатеричное.
3. Перевести шестнадцатеричное число $3F_{16}$ в двоичное и восьмеричное.

Задание 3. Арифметические действия с числами в разных системах

1. Сложите двоичное число 1011_2 с восьмеричным 7_8 (сначала переведите в одну систему).
2. Вычтите шестнадцатеричное число A_{16} из двоичного числа 11010_2
3. Умножьте восьмеричное число 12_8 на шестнадцатеричное число 3_{16}.

Задание 4. Сравнение методов записи

1. Запишите двоичное число 11010101 в восьмеричной и шестнадцатеричной системах.
2. Проанализируйте, на сколько разрядов короче запись числа в восьмеричной и шестнадцатеричной системах по сравнению с двоичной.

Задание 5. 1. Перевести данное число из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления. 2. Перевести данное число в десятичную систему счисления.

Вариант 1

1. а) 777; б) 305; в) 153,25; г) 162,25; д) 248,46.
2. а) 1100111011_2; б) 10000000111_2; в) $10110101,1_2$; г) $100000110,10101_2$; д) $671,2_4$; е) $41A,6_{16}$.

Вариант 2

1. а) 164; б) 255; в) 712,25; г) 670,25; д) 11,89.
2. а) 1001110011_2; б) 1001000_2; в) $1111100111,01_2$; г) $1010001100,101101_2$; д) $413,41_8$; е) $118,8C_{16}$.

Вариант 3

1. а) 273; б) 661; в) 156,25; г) 797,5; д) 53,74.
2. а) 1100000000_2; б) 1101011111_2; в) $1011001101,00011_2$; г) $1011110100,0111_2$; д) $1017,2_8$; е) $111,B_{16}$.

Вариант 4

1. а) 105; б) 358; в) 377,5; г) 247,25; д) 87,27.
2. а) 1100001001_2 ; б) 1100100101_2 ; в) $1111110110{,}01_2$; г) $11001100{,}0111_2$; д) $112{,}04_8$; е) $334{,}A_{1\ 6}$.

Вариант 5

1. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.
2. а) 111000100_2 ; б) 1011001101_2 ; в) $10110011{,}01_2$; г) $1010111111{,}011_2$; д) $1665{,}3_8$; е) $FA{,}7_{1\ 6}$.

Вариант 6

1. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.
2. а) 1111000111_2 ; б) 11010101_2 ; в) $1001111010{,}010001_2$; г) $1000001111{,}01_2$;
   д) $465{,}3_8$; е) $252{,}38_{1\ 6}$.

Вариант 7

1. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.
2. а) 110010001_2 ; б) 100100000_2 ; в) $1110011100{,}111_2$; г) $1010111010{,}1110111_2$;
   д) $704{,}6_8$; е) $367{,}38_{1\ 6}$.

Вариант 8

1. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.
2. а) 1000110110_2 ; б) 111100001_2 ; в) $1110010100{,}1011001_2$; г) $1000000110{,}00101_2$;
   д) $666{,}16_8$; е) $1C7{,}68_{1\ 6}$.

Вариант 9

1. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.
2. а) 101111111_2 ; б) 1111000110_2 ; в) $10011000{,}1101011_2$; г) $1110001101{,}1001_2$;
   д) $140{,}22_8$; е) $1DE{,}54_{1\ 6}$.

Вариант 10

1. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.
2. а) 111010000_2 ; б) 1010001111_2 ; в) $1101101000{,}010001_2$; г) $1000000101{,}01011_2$;
   д) $1600{,}14_8$; е) $1E9{,}4_{1\ 6}$.

Вариант 11

1. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.
2. а) 10000011001_2 ; б) 10101001_2 ; в) $1101100100,01_2$; г) $1110001100,1_2$; д) $1053,2_8$; е) $200,6_{1\ 6}$.

Вариант 12

1. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.
2. а) 1101111101_2 ; б) 1110011101_2 ; в) $1110010000,01_2$; г) $1100111001,1001_2$;
   д) $1471,17_8$; е) $3EC,5_{1\ 6}$.

Вариант 13

1. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.
2. а) 10100010_2 ; б) 1110010111_2 ; в) $110010010,101_2$; г) $111101100,10011_2$; д) $605,02_8$; е) $3C8,8_{1\ 6}$.

Вариант 14

1. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.
2. а) 1001101001_2 ; б) 110011101_2 ; в) $1000001101,01_2$; г) $1010001001,11011_2$; д) $247,18_8$; е) $81,4_{1\ 6}$.

Вариант 15

1. а) 917; б) 477; в) 74,5; г) 792,25; д) 84,33.
2. а) 1110011100_2 ; б) 1111011111_2 ; в) $111101100,101_2$; г) $110011110,1000011_2$; д) $1446,62_8$; е) $9C,D_{1\ 6}$.

Вариант 16

1. а) 477; б) 182; в) 863,25; г) 882,25; д) 75,2.
2. а) 101011100_2 ; б) 1000010011_2 ; в) $11100011,1_2$; г) $100101010,00011_2$; д) $1762,78_8$; е) $1B5,6_{1\ 6}$.

Вариант 17

1. а) 804; б) 157; в) 207,625; г) 435,375; д) 30,43.
2. а) 10010000_2 ; б) 11001010_2 ; в) $1110101100,1011_2$; г) $110110101,10111_2$; д) $1164,36_8$; е) $1D5,C8_{1\ 6}$.

Вариант 18

1. а) 753; б) 404; в) 111,1875; г) 907,0625; д) 62,88.
2. а) 11100011_2 ; б) 1111001111_2 ; в) $101111111,01001_2$; г) $100101101,0111_2$; д) $615,72_8$; е) $3DA,5_{1\ 6}$.

Вариант 19

1. а) 571; б) 556; в) 696,25; г) 580,375; д) 106,67.
2. а) 110011010_2 ; б) 111001010_2 ; в) $1000010011{,}00101_2$; г) $110101100{,}00001_2$; д) $1343{,}66_8$; е) $3C3{,}6_{16}$.

Вариант 20

1. а) 244; б) 581; в) 351,6875; г) 1027,375; д) 151,44.
2. а) 1001100111_2 ; б) 1100010010_2 ; в) $1100110010{,}1101_2$; г) 10010110101_2 ; д) $171{,}38_8$; е) $3A3{,}4_{16}$.

Лабораторная работа №3 «Битовая длина числа и диапазон значений»

Цель работы: освоить методы определения битовой длины чисел, понять взаимосвязь между системой счисления, битовой длиной и диапазоном представления чисел в различных форматах (беззнаковом, знаковом в дополнительном коде). Научиться выполнять расчёты диапазонов значений для заданной разрядности и наоборот – определять необходимую разрядность для заданного диапазона. Получить практические навыки анализа представления чисел в двоичной, восьмеричной, шестнадцатеричной и десятичной системах счисления.

Теоретические сведения

1. Битовая длина – фундаментальная характеристика числового представления в цифровых системах. Это минимальное количество двоичных разрядов (битов), необходимых для записи числа в двоичной системе счисления. Каждый бит может принимать значение 0 или 1, что соответствует физическим состояниям электронных компонентов: отсутствию или наличию сигнала, низкому или высокому напряжению.

Исторически битовая длина тесно связана с архитектурой процессоров и организацией памяти. Первые ЭВМ использовали 4-битные и 8-битные слова, современные процессоры оперируют 64-битными словами. Увеличение битовой длины напрямую влияет на вычислительную мощность и максимальный размер обрабатываемых данных.

Для неотрицательного целого числа N битовая длина L определяется как минимальное целое число, удовлетворяющее неравенству:

$$2^{L-1} \leq N < 2^{L} \text{ при } N > 0$$

Для N=0 принимается L=1, хотя технически для представления нуля достаточно одного бита (значения 0).

Альтернативно битовую длину можно выразить через двоичный логарифм:

$$L = [log_2 N] + 1$$

Пример

1. Для числа 100_{10}: $log_2 100 \approx$ **6.64**, L=**6**+1=7 бит.
   100_{10}=1100100_{10} – 7 цифр.
2. Для числа 255_{10}: $log_2 255 \approx$ **7.99**, L=**7**+1=8 бит. Максимальное 8-битное число.
3. Для числа 256_{10}: $log_2 256 \approx$ **8**, L=**8**+1=9 бит. Показывает, что для степеней двойки требуется дополнительный разряд.

2. Диапазоны значений для различных представлений

Это фундаментальное ограничение любой системы представления чисел в цифровой технике. В отличие от математических чисел, которые могут быть сколь угодно большими или малыми, в компьютерах числа хранятся в ограниченном количестве битов, что накладывает жесткие границы на их возможные значения. Понимание этих диапазонов критически важно для

предотвращения ошибок переполнения, правильного выбора типов данных и проектирования надежных систем.

Каждое представление чисел – будь то беззнаковое целое, знаковое целое в дополнительном коде, число с плавающей точкой или фиксированной точкой – имеет свои уникальные диапазоны, определяемые количеством битов и способом их интерпретации.

Примеры диапазонов для различных битов

Биты (N)	**Максимальное значение (2^N-1)**	**Диапазон**	**Количество значений**	**Типичное использование**
4	15	0..15	16	Полубайт (nibble)
8	255	0..255	256	Байт, коды ASCII
16	65,535	0..65,535	65,536	Порты TCP/UDP, Unicode BMP
32	4,294,967,295	0..~4.3×10^9	~4.3×10^9	IPv4 адреса, файловые размеры
64	~1.84×10^{19}	0..~1.84×10^{19}	~1.84×10^{19}	Современные счетчики, хэши

Беззнаковое представление (unsigned)

Беззнаковое представление – наиболее простое и интуитивное: все биты интерпретируются как положительное целое число. Каждый бит представляет степень двойки, начиная с младшего разряда как 2^0 . Это представление используется, когда гарантируется, что значение никогда не будет отрицательным.

$$\text{Диапазон}=[0,\ 2^n-1]$$

Где n – битовая длина.

Каждая из 2^n комбинаций битов соответствует уникальному числу. Например, при n=3:

$$000_2 = 0_{10}$$
$$001_2 = 1_{10}$$
$$\dots$$
$$111_2 = 7_{10}$$

Невозможно представить отрицательные числа, что ограничивает применение в арифметических вычислениях.

Знаковое представление (signed)

Дополнительный код стал стандартом де-факто для представления знаковых целых чисел в современных компьютерах. Его развитие было обусловлено поиском представления, которое:

- Имело бы одно представление нуля (в отличие от прямого и обратного кода);
- Позволяло бы использовать одну и ту же схему сложения для положительных и отрицательных чисел;
- Минимизировало бы аппаратную сложность АЛУ;

1. Прямой код:

Старший бит указывает знак (0 это "+", 1 это "–"); Остальные биты – модуль числа; Диапазон = [-(2^{n-1}-1), 2^{n-1}–1]

Недостаток: два представления нуля (+0 и –0), сложность арифметики

2. Обратный код:

Для положительных чисел – как прямой код. Для отрицательных – инверсия всех битов положительного числа; Диапазон = [-(2^{n-1}-1), 2^{n-1}–1], также имеет два нуля

3. Дополнительный код (two's complement):

Стандарт в современных компьютерах; положительные числа – как в прямом коде; отрицательное число −X представляется как 2^n-X; диапазон = [-2^{n-1}, 2^{n-1}–1];

Примеры диапазонов для различных битов

Биты (N)	**Диапазон**	**Максимум**	**Минимум**	**Асимметрия**	**Типичное применение**
8	-128..127	127	-128	1	Байтовые данные, аудиосэмплы
16	-32,768..32,767	32,767	-32,768	1	Короткие целые, индексы массивов
32	-2.1×10^9 ..2.1×10^9	2,147,483,647	-2,147,483,648	1	Стандартные целые (int)
64	-9.2×10^{18} ..9.2×10^{18}	~9.22×10^{18}	~-9.22×10^{18}	1	Длинные целые, большие счетчики

Переполнение в дополнительном коде происходит, когда:

1. При сложении двух положительных – результат отрицательный
2. При сложении двух отрицательных – результат положительный
3. При вычитании – аналогичные условия

Формально переполнение происходит, когда знаковый бит результата не соответствует знакам операндов при учете переноса.

Практическое задание

Задание 1. Определение битовой длины

Дано число в десятичной системе. Определить минимальную битовую длину, необходимую для его представления:

- В беззнаковом формате
- В знаковом формате (дополнительный код)

Задание 2. Диапазоны значений

Для заданной битовой длины N определить:

- Диапазон представляемых чисел в беззнаковом формате
- Диапазон представляемых чисел в знаковом формате (дополнительный код)

Задание 3. Анализ переполнения

Даны два числа. Определить:

- Произойдет ли переполнение при их сложении в беззнаковом N-битном представлении
- Произойдет ли переполнение при их сложении в знаковом N-битном представлении

Вариант	Число для заданий 1 (десятичное)	Биты N для заданий 2, 3	Числа A и B для задания 4 (десятичные)
1	423	12 бит	A=1800, B=900
2	1025	10 бит	A=500, B=600
3	77	8 бит	A=100, B=30
4	4095	16 бит	A=30000, B=10000
5	-150 (для знакового)	9 бит	A=200, B=100
6	65000	20 бит	A=400000, B=100000
7	255	7 бит	A=80, B=50
8	-500	12 бит	A=1500, B=800
9	10000	14 бит	A=12000, B=5000
10	32768	15 бит	A=20000, B=15000
11	777	11 бит	A=1200, B=900
12	-1024	11 бит	A=800, B=400
13	16383	16 бит	A=40000, B=20000
14	511	10 бит	A=700, B=400
15	-32767	16 бит	A=20000, B=15000
16	9999	13 бит	A=6000, B=5000
17	2047	12 бит	A=2500, B=1000
18	-8191	14 бит	A=5000, B=4000
19	65535	17 бит	A=80000, B=20000
20	129	8 бит	A=200, B=100

Лабораторная работа №4 «Определение количества бит для кодирования чисел»

Цель работы: закрепить навыки представления числовой информации в двоичной форме, научиться определять минимальное количество бит для кодирования чисел, вычислять диапазон кодируемых значений и анализировать корректность выбора разрядности при хранении данных в информационных системах.

В современных информационных системах вся числовая информация хранится и обрабатывается в двоичном виде. Это связано с особенностями работы вычислительной техники, в которой используются два устойчивых состояния электрического сигнала. Любое число в памяти компьютера представляется в виде последовательности нулей и единиц, называемой двоичным кодом.

Для хранения числовых данных используется определённое количество двоичных разрядов, называемое разрядностью. Разрядность определяет, сколько различных числовых значений может быть закодировано. Если для представления числа используется n бит, то общее количество различных значений равно 2^n. При этом максимальное число, которое может быть представлено, равно $2^n - 1$.

При проектировании информационных систем важно правильно выбирать разрядность числовых данных. Недостаточная разрядность приводит к переполнению и искажению информации, а избыточная — к нерациональному использованию памяти. В системах информационной безопасности ошибки кодирования числовых данных могут вызывать сбои в идентификаторах пользователей, журналах событий и системах контроля доступа.

Минимальное количество бит, необходимое для хранения заданного числа, определяется путём подбора наименьшей степени двойки, превышающей это число. Для этого последовательно сравниваются значения 2^n с заданным числом до тех пор, пока не будет найдено первое значение, большее исходного. Найденное n и будет минимальной разрядностью.

Например, для числа 150: $2^7 = 128$ меньше 150, а $2^8 = 256$ больше 150, следовательно, минимальная разрядность равна 8 битам. Это означает, что для хранения числа 150 требуется не менее одного байта памяти.

Для своего варианта необходимо определить минимальное количество бит, достаточное для кодирования заданного числа, максимальное значение, которое можно представить с найденной разрядностью, а также записать исходное число в двоичной системе счисления.

Вариант	Заданное число	Минимальное количество бит	Максимальное кодируемое значение	Двоичное представление числа
1	73			
2	95			
3	128			
4	156			
5	201			
6	245			
7	256			
8	312			
9	389			
10	447			
11	512			
12	625			
13	700			
14	845			
15	999			
16	1023			
17	1100			
18	1350			
19	1500			
20	1800			

Контрольные вопросы

1. В каком виде представляется числовая информация в памяти компьютера?
2. Что называется разрядностью двоичного числа?
3. Как определить количество различных значений, кодируемых n битами?

4. Как вычисляется максимальное число, представимое при заданной разрядности?
5. Как определяется минимальное количество бит для хранения заданного числа?
6. К каким последствиям приводит недостаточная разрядность числовых данных?
7. Почему избыточная разрядность приводит к неэффективному использованию памяти?

Лабораторная работа №5 «Определение количества возможных значений при заданной разрядности»

Цель работы: сформировать умение определять количество различных значений, которые могут быть закодированы при заданной разрядности, вычислять максимальное кодируемое число и анализировать влияние разрядности на возможности хранения числовой информации в информационных системах.

В информационных системах числовые данные хранятся в двоичном виде с использованием фиксированного количества бит. Количество бит, отведённых для хранения одного числа, называется разрядностью. Разрядность определяет диапазон возможных значений, которые могут быть представлены в памяти компьютера.

Если для хранения числа используется n бит, то каждый бит может принимать два значения: 0 или 1. Следовательно, общее количество различных комбинаций бит равно 2^n. Это число показывает, сколько различных значений можно закодировать при данной разрядности.

Максимальное значение, которое может быть представлено при разрядности n бит, вычисляется по формуле $2^n - 1$. Это связано с тем, что отсчёт значений начинается с нуля. Таким образом, диапазон кодируемых чисел составляет от 0 до $2^n - 1$.

Например, при разрядности 8 бит можно закодировать $2^8 = 256$ различных значений, а максимальное число равно 255. При разрядности 10 бит возможно 1024 различных значения, а максимальное число равно 1023.

Выбор разрядности имеет важное практическое значение. Недостаточная разрядность приводит к невозможности хранения всех необходимых значений, что вызывает переполнение и ошибки обработки данных. Избыточная разрядность приводит к нерациональному использованию памяти, увеличению объёма хранимых данных и снижению эффективности работы информационных систем.

В системах информационной безопасности корректный выбор разрядности важен при хранении идентификаторов пользователей, номеров событий безопасности, сетевых адресов и других числовых параметров. Ошибки в расчётах могут привести к потере данных и нарушению целостности информации.

Практическое задание

Для своего варианта необходимо определить количество различных значений, которые можно закодировать при заданной разрядности, вычислить максимальное кодируемое число и указать диапазон возможных значений.

Вариант	**Разрядность (бит)**	**Количество возможных значений**	**Максимальное кодируемое число**	**Диапазон значений**
1	5			
2	6			
3	7			
4	8			

5	9			
6	10			
7	11			
8	12			
9	13			
10	14			
11	15			
12	16			
13	17			
14	18			
15	19			
16	20			
17	21			
18	22			
19	23			
20	24			

Контрольные вопросы

1. Что называется разрядностью двоичного числа?
2. Почему количество возможных значений равно 2^n?
3. Как определяется максимальное кодируемое число при заданной разрядности?
4. Почему диапазон значений начинается с нуля?
5. К каким последствиям приводит недостаточная разрядность данных?

Лабораторная работа №6 «Определение объёма памяти для хранения числовых данных»

Цель работы: сформировать умение рассчитывать объём памяти, необходимый для хранения числовых данных в информационных системах, на основе разрядности чисел и количества элементов, а также применять данные расчёты при проектировании и анализе компьютерных систем хранения информации.

Теоретические сведения

В процессе работы информационных систем постоянно выполняется хранение, обработка и передача данных. Вся информация в компьютере представляется в двоичном виде, так как электронные устройства используют два устойчивых состояния сигнала. Основной единицей измерения информации является бит. Для удобства практического использования восемь бит объединяются в один байт.

Числовые данные в памяти компьютера хранятся с использованием фиксированного количества бит, называемого разрядностью. Разрядность определяет, какие значения могут быть представлены и сколько памяти требуется для хранения информации. Если число кодируется с помощью n бит, то возможно представить 2^n различных значений, а максимальное число равно $2^n - 1$.

При хранении массивов числовых данных важно учитывать не только разрядность одного числа, но и общее количество элементов. Объём памяти, необходимый для хранения данных, определяется произведением количества элементов на информационный вес одного элемента. Таким образом, если N чисел кодируются с использованием i бит каждое, то общий объём памяти V вычисляется по формуле:

$$V = N * i$$

Полученный результат выражается в битах и при необходимости переводится в байты, килобайты или мегабайты.

Например, если необходимо сохранить 200 чисел, каждое из которых занимает 10 бит, общий объём памяти составит $200 \cdot 10 = 2000$ бит, что соответствует 250 байтам.

Корректный расчёт объёма памяти имеет большое значение в области информационной безопасности. Ошибки в оценке объёма данных могут привести к переполнению памяти, потере информации, сбоям в работе баз данных и журналов событий безопасности.

Практическое задание

Вариант 1

1. Какое минимальное количество бит потребуется для кодирования одного неотрицательного целого числа из диапазона от 0 до 63 включительно?

2. Автомобильный номер состоит из 6 символов: сначала три буквы из набора {A, B, C, D}, затем три цифры. Каждый такой номер в компьютерной программе должен быть представлен как минимально возможное целое число (одинаковое для всех номеров). Сколько байт необходимо выделить для хранения одного номера?

3. В школьной базе данных хранится информация об учениках: уникальный целый идентификатор (ID) и год рождения. Известно, что в школе 850 учеников, а ID нумеруются с 1. Для хранения года рождения выделено 2 байта. Каков минимальный достаточный объём в байтах для хранения этих двух полей об одном ученике?

Вариант 2

1. Датчик температуры передаёт показания в виде целого числа от -50 до +40 градусов. Какое минимальное количество бит потребуется для кодирования одного показания этого датчика?

2. Для хранения растрового изображения размером 128x256 пикселей отвели 64 Кбайт памяти. Каково максимально возможное количество цветов в палитре изображения? *(Связь с графикой)*

3. В программе используется 200 различных символов. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль должен быть длиной ровно 12 символов. Определите объём памяти в байтах, необходимый для хранения 50 паролей.

Вариант 3

1. Шахматная доска состоит из 64 полей. Для кодирования координаты одного поля на доске (например, «e2») используется минимально возможное количество бит. Сколько бит потребуется?

2. Световое табло состоит из лампочек, каждая из которых может находиться в одном из трёх состояний («выключено», «горит красным», «горит зелёным»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 90 различных сигналов?

3. В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым спортсменом промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Какой объём памяти в битах будет

использован устройством, когда промежуточный финиш пройдут 70 велосипедистов?

Вариант 4

1. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 0, для буквы Б — кодовое слово 110. Какова наименьшая возможная суммарная длина кодовых слов для букв В и Г?

2. Каждое показание счётчика, фиксируемое в памяти компьютера, занимает 10 бит. Запись показаний начинается с адреса 3200. Какой будет адрес памяти, по которому будет записано 30-е показание?

3. Метеорологическая станция ведёт наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100 процентов, которое записывается при помощи минимально возможного количества бит. Станция сделала 4096 измерений. Определите информационный объём результатов наблюдений в Кбайтах.

Вариант 5

1. В некоторой стране автомобильный номер состоит из 7 символов: сначала 3 буквы из 26-буквенного алфавита, затем 4 цифры. При этом цифры не могут повторяться. Для хранения каждого номера используется одинаковое и минимально возможное целое число байт. Сколько байт нужно выделить для хранения 50 таких номеров?

2. Для хранения произвольного растрового изображения размером 512×512 пикселей отведено 256 Кбайт памяти, при этом для каждого пикселя хранится двоичное число – код цвета этого пикселя. Для каждого пикселя для хранения кода выделено одинаковое количество бит. Сжатие данных не производится. Какое максимальное количество цветов можно использовать в изображении?

3. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: A, B, C, D, E, F, G, H, K, L, M, N. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено

12 байт. Определите объём памяти (в байтах), необходимый для хранения сведений о 100 пользователях.

Контрольные вопросы

1. Как определить минимальное количество бит, необходимое для кодирования N различных целых чисел?
2. В чём разница между расчётом объёма для хранения одного числа и массива чисел?
3. Как повлияет на требуемый объём памяти, если для хранения чисел использовать не минимально возможное, а стандартное для системы количество байт (например, всегда 4 байта)?
4. Что такое условие Фано и как оно применяется при расчёте длины кода?
5. Как решать задачи, в которых данные представлены в смешанных форматах (например, текст + числа)?

Лабораторная работа №7 «Кодирование символов в ASCII и Unicode»

Кодирование текстовой информации в ЭВМ основано на использовании таблицы кодировки, в которой устанавливается соответствие между символом (буквой, знаком препинания, цифрой, графическим символом) и его уникальным десятичным кодом.

Базовым стандартом кодирования текстовой информации является стандарт ASCII (American Standard Code for Information Interchange), разработанный в США в Национальном институте ANSI (American National Standards Institute). В стандарте ASCII на кодирование одного символа отводится 1 байт, поэтому, используя кодировку ASCII, можно закодировать 256 различных символов.

В стандарте ASCII описываются две таблицы — базовая и расширенная. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255. При этом первые 33 кода (с 0 до 32) соответствуют не символам, а операциям (перевод строки, ввод пробела и т. д.). Коды с 33 по 127 являются международными и соответствуют символам латинского алфавита, цифрам, знакам арифметических операций и знакам препинания. Коды же с 128 по 255 являются национальными, т. е. в национальных кодировках одному и тому же коду соответствуют различные символы.

Кодирование символов русского алфавита осуществляется на основе нескольких кодовых таблиц (КОИ-8R, CP-1251, CP-866, Mac, ISO-8859-5), которые несовместимы друг с другом.

Название кодировки						
	ASCII	**ISO**	**CP-866**	**CP-1251**	**КОИ-8R**	**UNICODE**
Байт/символ	1	1	1	1	1	2

Кодировка CP 866 основана на реализации таблицы ASCII, разработанной фирмой IBM. Эта кодировка в верхней половине расширенной кодовой таблицы содержит псевдографические символы, а специфические европейские символы в верхней половине кодовой таблицы заменены на кириллицу. Данная кодировка создана в ВЦ АН СССР, для которого впервые в СССР была закуплена партия IBM PC.

Кодировка КОИ 8R разработана при адаптации операционной системы UNIX к русскому языку. В ней символы русской кириллицы размещены в верхней части расширенной ASCII таблицы так, что позиции кириллических символов соответствуют их фонетическим аналогам в английском алфавите в нижней части таблицы.

Семейство кодировок ISO-8859-X определено международной организацией по стандартизации (ISO). Это семейство представляет собой совокупность 8-битных кодировок, где младшая половина кодовой таблицы (символы с кодами 0-127) соответствует таблице ASCII, а в верхней половине определены символы для различных языков. Например, кодовая страница для кириллицы определена в стандарте ISO-8859-5.

Кодировка CP 1251 – стандартная 8-битная кодировка для всех русских версий Microsoft Windows, которая создана на базе кодировок, использовавшихся в ранних русификаторах Windows.

Кодировка UNICODE была разработана для создания единой кодировки символов всех современных и многих древних письменных языков. Каждый 10 символ в этом стандарте кодируется 16 битами, что позволяет охватить гораздо большее количество символов, чем принятые ранее 7- и 8-битовые кодировки.

В кодировке UNICODE с каждым символом связан уникальный код и определены характеристики этого символа, например:

- тип символа (прописная буква, строчная буква, цифра, знак препинания и т. д.);

- атрибуты символа (отображение слева направо или справа налево, пробел, разрыв строки и т. д.);

- соответствующая прописная или строчная буква (для строчных и прописных букв);

- соответствующее числовое значение (для цифровых символов).

Объем информации, содержащийся в тексте из K символов, равен:

$It = K \cdot I$, где It — объем информации в тексте, I — информационный вес 1 символа текста (количество байтов, отводимых на хранение 1 символа).

ЗАДАЧА 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем в битах следующего предложения: «Мой дядя самых честных правил. Когда не в шутку занемог, Он уважать себя заставил И лучше выдумать не мог».

Решение.

Для оценки информационного объема предложения необходимо сосчитать количество входящих в предложение символов: букв, знаков препинания (кавычек, точек, запятых и др.) и пробелов.

В заданном тексте всего букв – 85, пробелов – 18, знаков препинания – 5.

Получаем, что предложение состоит из 108 символов.

Учитывая, что каждый символ кодируется одним байтом, можно записать: $108 \cdot 8 = 864$ бита.

Ответ: 864 бита.

ЗАДАЧА 2. Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 480 бит. Какова длина сообщения в символах?

Решение.

Пусть x – количество символов в информационном сообщении. Тогда информационный объем сообщения составляет: - в кодировке UNICODE – 16 · x битов, - в кодировке КОИ-8 – 8 · x битов. Исходя из условия задачи, запишем следующее уравнение:

16 · x – 8 · x = 480

Тогда 8 · x = 480, откуда x = 480 / 8 = 60 символов.

Ответ: 60 символов.

ЗАДАЧА 3. В таблице 3 представлена часть кодовой таблицы ASCII. Определите шестнадцатеричный код символа «q». Таблица 3 – Характеристики кодировок символов Символ 1 5 A B Q a b Десятичный код 49 53 65 66 81 97 98 Шестнадцатеричный код 31 35 41 42 51 61 62

Решение.

Заглавный и строчный символы находятся в таблице ASCII на одном расстоянии, следовательно

Кa − КA = 97 − 65 = 32,

Кb − КB = 98 − 66 = 32.

Тогда Кq − КQ = Кq − 81 = 32, откуда получаем Кq = 11310 = 7116.

Ответ: 7116.

ЗАДАЧА 4. В одной из кодировок Unicode каждый символ кодируется 16 битами. Ученик написал текст (в нём нет лишних пробелов): «Ёж, лев, слон, олень, тюлень, носорог, крокодил, аллигатор — дикие животные». Затем он вычеркнул из списка название одного из животных. Заодно он вычеркнул ставшие лишними запятые и пробелы — два пробела не должны идти подряд. При этом размер нового предложения в данной кодировке оказался на 16 байт меньше, чем размер исходного предложения. Напишите в ответе вычеркнутое название животного.

Решение.

Поскольку один символ кодируется двумя байтами, из текста удалили 16 : 2 = 8 символов.

Из этих 8 символов 2 символа – это лишняя запятая и пробел. Тогда удаленное из текста название животного должно состоять из 6 букв. Из всего списка только одно название животного состоит из 6 букв – тюлень.

Ответ: тюлень.

Практическое задание

Вариант 1

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем текста:
«Программист пишет код, проверяет ошибки и тестирует программу».

Задача 2. Сообщение в Unicode (16 бит на символ) перекодировали в CP-1251 (8 бит на символ). При этом информационный объем уменьшился на 256 бит. Сколько символов в сообщении?

Задача 3. В таблице ASCII: A – 65, B – 66, a – 97, b – 98, Q – 81.

Найдите шестнадцатеричный код символа «s».

Задача 4. Текст в Unicode: «Собака, кот, мышь, лиса, волк». После удаления одного животного с запятой и пробелом объем уменьшился на 14 байт. Какое животное удалили?

Вариант 2

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем текста: «Учёба приносит знания, а практика закрепляет умения».

Задача 2. Сообщение в Unicode перекодировали в КОИ-8. Информационный объем уменьшился на 400 бит. Сколько символов в исходном сообщении?

Задача 3. В таблице ASCII: C – 67, D – 68, c – 99, d – 100, Q – 81. Определите шестнадцатеричный код символа «t».

Задача 4. Текст Unicode: «Ёж, лев, слон, тигр, медведь». После удаления одного животного с запятой и пробелом объем уменьшился на 12 байт. Какое животное удалили?

Вариант 3

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем текста:

«Книга — источник знаний и вдохновения для каждого человека».

Задача 2. Сообщение в Unicode перекодировали в CP-866. После перекодировки объем уменьшился на 192 бита. Определите количество символов.

Задача 3. В таблице ASCII: E – 69, e – 101, F – 70, f – 102, R – 82. Найдите шестнадцатеричный код символа «u».

Задача 4. Текст Unicode: «Лев, тигр, слон, жираф, крокодил». После удаления одного животного и связанных с ним знаков объем уменьшился на 16 байт. Какое животное удалили?

Вариант 4

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем текста: «Математика — это язык науки, логики и точности».

Задача 2. Сообщение в Unicode перекодировали в CP-1251. Объем уменьшился на 288 бит. Сколько символов содержится в тексте?

Задача 3. В таблице ASCII: G – 71, g – 103, H – 72, h – 104, Q – 81. Найдите шестнадцатеричный код символа «v».

Задача 4. Текст Unicode: «Слон, тигр, лев, жираф, крокодил, носорог». После удаления одного животного с запятой и пробелом объем уменьшился на 20 байт. Какое животное удалили?

Вариант 5

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста:

«Компьютерная графика позволяет создавать удивительные изображения».

Задача 2. Unicode → KOI-8R. Объем уменьшился на 224 бита. Определите число символов.

Задача 3. В таблице ASCII: I – 73, i – 105, J – 74, j – 106, R – 82. Найдите шестнадцатеричный код символа «x».

Задача 4. Текст Unicode: «Кот, собака, мышь, ёж, волк». После удаления одного животного объем уменьшился на 10 байт. Какое животное удалили?

Вариант 6

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста: «Природа вдохновляет художников на создание шедевров».

Задача 2. Unicode → CP-866. Объем уменьшился на 128 бит. Сколько символов в тексте?

Задача 3. В таблице ASCII: K – 75, k – 107, L – 76, l – 108, Q – 81. Найдите шестнадцатеричный код символа «z».

Задача 4. Текст Unicode: «Лев, слон, тигр, волк, медведь». После удаления одного животного с запятой и пробелом уменьшился объем на 18 байт. Какое животное удалили?

Вариант 7

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста:

«История науки полна открытий и удивительных фактов».

Задача 2. Unicode → CP-1251. Объем уменьшился на 352 бит. Найдите количество символов.

Задача 3. В таблице ASCII: M – 77, m – 109, N – 78, n – 110, R – 82. Найдите шестнадцатеричный код символа «y».

Задача 4. Текст Unicode: «Кот, собака, лиса, волк, ёж». После удаления одного животного уменьшился объем на 12 байт. Какое животное удалили?

Вариант 8

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста:

«Программирование требует внимательности и усидчивости».

Задача 2. Unicode → KOI-8. Объем уменьшился на 192 бита. Определите число символов.

Задача 3. В таблице ASCII: O – 79, o – 111, P – 80, p – 112, Q – 81. Найдите шестнадцатеричный код символа «w».

Задача 4. Текст Unicode: «Тигр, лев, слон, медведь, волк». После удаления одного животного и связанных знаков объем уменьшился на 16 байт. Какое животное удалили?

Вариант 9

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста:

«Информационные технологии развиваются очень быстро».

Задача 2. Unicode → CP-866. Объем уменьшился на 160 бит. Определите число символов.

Задача 3. В таблице ASCII:

R – 82, r – 114, S – 83, s – 115, Q – 81.

Найдите шестнадцатеричный код символа «p».

Задача 4. Текст Unicode: «Слон, тигр, лев, носорог, крокодил». После удаления одного животного уменьшился объем на 14 байт. Какое животное удалили?

Вариант 10

Задача 1. Считая, что каждый символ кодируется одним байтом, оцените объем текста:

«Учёба, практика и опыт делают специалиста профессионалом».

Задача 2. Unicode → CP-1251. Объем уменьшился на 288 бит. Определите число символов.

Задача 3. В таблице ASCII:

T – 84, t – 116, U – 85, u – 117, R – 82.

Найдите шестнадцатеричный код символа «m».

Задача 4. Текст Unicode: «Кот, лиса, собака, волк, ёж». После удаления одного животного уменьшился объем на 12 байт. Какое животное удалили?

Контрольные вопросы

1. Что такое ASCII и Unicode, и чем они отличаются по объему кодируемой информации на один символ?
2. Что такое расширенные кодировки (CP-1251, KOI8-R, CP-866) и для чего они нужны?
3. Как вычислить информационный объем текста в битах, если известен его размер в символах и кодировка?
4. Почему при перекодировке текста из Unicode в 8-битную кодировку информационный объем уменьшается?
5. Как определить шестнадцатеричный код символа в ASCII, если известен его десятичный код?

Лабораторная работа №8 «Определение объёма текстовой информации»

Цель работы: изучить способы определения количества информации на синтаксическом уровне и научиться вычислять объём текстовой информации, представленной в символьной форме.

Понятие информации может рассматриваться на трёх уровнях: синтаксическом, семантическом и прагматическом.

В данной практической работе используется **синтаксический уровень,** при котором учитывается только форма представления сообщения, без анализа его смысла и полезности.

На синтаксическом уровне количество информации оценивается с помощью вероятностных методов. Такой подход был разработан в середине XX века и основан на идее уменьшения неопределённости знаний: если сообщение уменьшает неопределённость, то оно содержит информацию.

Математическую формулу для определения количества информации предложил Клод Шеннон в 1948 году:

$$I = -\sum_{i=1}^{N} p_i \log_2 p_i, \qquad (1.1)$$

где I –количество информации; N –количество возможных событий (сообщений); pi –вероятность отдельных событий (сообщений); Σ – математический знак суммы чисел.

При работе с текстами отдельные символы алфавита рассматриваются как возможные сообщения.

Количество различных символов, которые могут использоваться в тексте, называется мощностью алфавита.

Тогда количество информации, которое несёт один символ, определяется по формуле:

$$I = \log_2 N, \qquad (1.2)$$

где N - количество возможных сообщений.

Применение формулы к текстовой информации

При работе с текстами отдельные символы алфавита рассматриваются как возможные сообщения.

Количество различных символов, которые могут использоваться в тексте, называется мощностью алфавита.

Тогда количество информации, которое несёт один символ, определяется по формуле:

$$I = \log_2 N,$$

Чем больше мощность алфавита, тем больше информации несёт один символ.

В формулах (1.1) и (1.2) отношение между количеством информации и соответственно вероятностью, или количеством, отдельных событий выражается с помощью логарифма. Применение логарифмов в формулах (1.1) и (1.2) можно объяснить следующим образом. Для простоты рассуждений воспользуемся соотношением (1.2). Будем последовательно присваивать аргументу N значения, выбираемые, например, из ряда чисел: 1, 2, 4, 8, 16, 32, 64 и т. д. Чтобы определить, какое событие из N равновероятных событий произошло, для каждого числа ряда необходимо последовательно производить операции выбора из двух возможных событий.

Так, при N =1 количество операций будет равно 0 (вероятность события равна 1), при N =2, количество операций будет равно 1, при N =4 количество операций будет равно 2, при N =8, количество операций будет равно 3 и т. д. Таким образом получим следующий ряд чисел: 0, 1, 2, 3, 4, 5, 6 и т. д., который можно считать соответствующим значениям функции I в соотношении (1.2). Последовательность значений чисел, которые принимает аргумент N, представляет собой ряд, известный в математике как ряд чисел, образующих геометрическую прогрессию, а последовательность значений чисел, которые принимает функция I, будет являться рядом, образующим арифметическую прогрессию. Таким образом, логарифм в формулах (1.1) и (1.2) устанавливает соотношение между рядами, представляющими геометрическую и арифметическую прогрессии, что достаточно хорошо известно в математике.

В информатике, изучающей процессы получения, обработки, передачи и хранения информации с помощью вычислительных (компьютерных) систем, в основном используется двоичное кодирование, при котором используется знаковая система, состоящая из двух символов 0 и 1. По этой причине в формулах (1.1) и (1.2) в качестве основания логарифма используется цифра 2.

Исходя из вероятностного подхода к определению количества информации эти два символа двоичной знаковой системы можно рассматривать как два различных возможных события, поэтому за единицу количества информации принято такое количество информации, которое содержит сообщение, уменьшающее неопределенность знания в два раза (до получения событий их вероятность равна 0,5, после получения – 1, неопределенность уменьшается соответственно: 1/0,5 = 2, т. е. в 2 раза). Такая единица измерения информации называется битом (от англ. слова binary digit –двоичная цифра). Таким образом, в качестве меры для оценки количества информации на синтаксическом уровне, при условии двоичного кодирования, принят один бит.

Следующей по величине единицей измерения количества информации является байт, представляющий собой последовательность, составленную из восьми бит, т. е.

1 байт = 23 бит = 8 бит.

В информатике также широко используются кратные байту единицы измерения количества информации, однако в отличие от метрической системы мер, где в качестве множителей кратных единиц применяют коэффициент 10n, где п =3, 6, 9 и т. д., в кратных единицах измерения количества информации используется коэффициент 2n. Выбор этот объясняется тем, что компьютер в основном оперирует числами не в десятичной, а в двоичной системе счисления.

Кратные байту единицы измерения количества информации вводятся следующим образом:

1 Килобайт (Кбайт) = 2^{10} байт = 1024 байт,

1 Мегабайт (Мбайт) = 2^{10} Кбайт = 1024 Кбайт,

1 Гигабайт (Гбайт) = 2^{10} Мбайт = 1024 Мбайт,

1 Терабайт (Тбайт) = 2^{10} Гбайт = 1024 Гбайт,

1 Петабайт (Пбайт) = 2^{10} Тбайт = 1024 Тбайт,

1 Экзабайт (Эбайт) = 2^{10} Пбайт = 1024 Пбайт.

Единицы измерения количества информации, в названии которых есть приставки «кило», «мега» и т. д., с точки зрения теории измерений не являются корректными, поскольку эти приставки используются в метрической системе мер, в которой в качестве множителей кратных единиц используется коэффициент 10n, где п =3, 6, 9 и т. д. Для устранения этой некорректности международная организацией International Electrotechnical Commission, занимающаяся созданием стандартов для отрасли электронных технологий, утвердила ряд новых приставок для единиц измерения количества информации: киби (kibi), меби (mebi), гиби (gibi), теби (tebi), пети (peti), эксби (exbi). Однако пока используются старые обозначения единиц измерения количества информации, и требуется время, чтобы новые названия начали широко применяться.

Вероятностный подход используется и при определении количества информации, представленной с помощью знаковых систем. Если рассматривать символы алфавита как множество возможных сообщений N, то количество информации, которое несет один знак алфавита, можно определить по формуле (1.1). При равновероятном появлении каждого знака алфавита в тексте сообщения для определения количества информации можно воспользоваться формулой (1.2).

Количество информации, которое несет один знак алфавита, тем больше, чем больше знаков входит в этот алфавит. Количество знаков, входящих в алфавит, называется мощностью алфавита. Количество информации (информационный объем), содержащееся в сообщении, закодированном с помощью знаковой системы и содержащем определенное количество знаков (символов), определяется с помощью формулы:

$$V = I \cdot K, \quad (1.3)$$

где V – информационный объем сообщения; I = log2N, информационный объем одного символа (знака); К – количество символов (знаков) в сообщении; N –мощность алфавита (количество знаков в алфавите).

Информационный объём текста

Текст представляет собой последовательность символов. Если текст состоит из K символов, то его информационный объём определяется формулой:

$$V = K \times I = K \times log_2 N$$

Где V — информационный объём сообщения, K — количество символов в тексте, N — мощность алфавита.

Следовательно, объём текстовой информации зависит от двух факторов: длины текста (количества символов); мощности используемого алфавита.

Порядок определения объёма текстовой информации

1. Определить, из каких символов состоит текст (алфавит).
2. Найти мощность алфавита N.
3. Вычислить количество информации в одном символе:
   $I = \log_2 N,$
4. Подсчитать количество символов в тексте K (с пробелами).
5. Определить информационный объём текста:
   $V = I \cdot K,$
6. Перевести результат в байты, килобайты.

Практическое задание

Задание 1. Алфавит и один символ

Определите, какое количество информации несёт один символ алфавита, если текст набран:

а) десятичными цифрами (0–9);

б) латинскими буквами (26 символов);

в) русскими строчными буквами без «ё» (32 символа).

Задание 2. Короткое сообщение

Сообщение «ИНФОРМАТИКА» записано заглавными русскими буквами без пробелов.

1. Определите мощность алфавита.
2. Найдите количество информации в одном символе.
3. Определите информационный объём всего сообщения в битах и байтах.

Здание 3. Определение объёма информации в текстовом сообщении

Для своего варианта:

1. Определите мощность алфавита.
2. Подсчитайте количество символов в тексте (с пробелами).
3. Найдите количество информации в одном символе.
4. Определите информационный объём сообщения в битах и байтах.

Вариант	Текст сообщения	Алфавит	Вариант	Текст сообщения	Алфавит
1	Информация — это знания.	Русские буквы, пробел, точка, тире	11	Компьютер — это устройство.	Русские буквы, пробел, точка, тире
2	Компьютер обрабатывает данные.	Русские буквы, пробел, точка	12	Данные — основа информатики.	Русские буквы, пробел, точка, тире
3	Информатика изучает информацию.	Русские буквы, пробел, точка	13	Информация — важный ресурс.	Русские буквы, пробел, точка, тире
4	Данные передаются по сети.	Русские буквы, пробел, точка	14	Код — способ представления данных.	Русские буквы, пробел, точка, тире
5	Сообщение содержит информацию.	Русские буквы, пробел, точка	15	Текст — форма представления информации.	Русские буквы, пробел, точка, тире
6	Кодирование используется в компьютере.	Русские буквы, пробел, точка	16	Алфавит состоит из символов.	Русские буквы, пробел, точка
7	Информация хранится в памяти.	Русские буквы, пробел, точка	17	Информация измеряется в битах.	Русские буквы, пробел, точка
8	Алгоритм решает задачу.	Русские буквы, пробел, точка	18	Сообщения передаются по каналам связи.	Русские буквы, пробел, точка
9	Программа выполняет команды.	Русские буквы, пробел, точка	19	Двоичное кодирование используется в компьютерах.	Русские буквы, пробел, точка
10	Файл содержит текстовую информацию.	Русские буквы, пробел, точка	20	Объём информации зависит от алфавита.	Русские буквы, пробел, точка

Задание 4. Лабораторная работа за компьютером

1. Наберите в текстовом редакторе любой абзац объёмом не менее 300 символов.
2. Определите, какие символы реально используются в тексте.
3. Найдите мощность алфавита.
4. Вычислите объём информации в тексте.

5. Сравните полученный результат с размером файла на диске. Объясните различие.

Контрольные вопросы

1. Почему при определении объёма текста не учитывается его смысл?
2. Что называется мощностью алфавита?
3. По какой формуле определяется информационный объём текста?
4. Как изменится объём сообщения при увеличении мощности алфавита?

Лабораторная работа №9 «Определение количества символов по объёму файла»

Цель работы: научиться определять количество символов в текстовом сообщении по известному объёму файла, используя формулы теории информации.

На прошлом практическом занятии рассматривалась задача определения объёма текстовой информации по известному количеству символов. При этом использовался синтаксический уровень рассмотрения информации, при котором учитывается только форма представления сообщения, без анализа его смыслового содержания. Текст рассматривался как последовательность символов некоторого алфавита, а каждый символ считался равновероятным.

Было установлено, что количество информации, которое несёт один символ алфавита, определяется по формуле Хартли:

$$I = \log_2 N,$$

где N — мощность алфавита, то есть количество различных символов, используемых в тексте.

Информационный объём текстового сообщения вычислялся по формуле:

$$V = I \cdot K,$$

где V — объём информации, K — количество символов, I — количество информации в одном символе.

В данной практической работе решается обратная задача: по известному объёму информации требуется определить количество символов в тексте. Для этого формула объёма сообщения преобразуется следующим образом:

$$K = \frac{V}{I} = \frac{V}{\log_2 N}$$

Таким образом, количество символов в тексте зависит от двух факторов: объёма информации и мощности используемого алфавита.

На практике объём текстового сообщения задаётся размером файла на диске, который измеряется в байтах. Для применения формул теории информации необходимо перевести объём файла в биты, так как количество информации в одном символе выражается в битах. Перевод осуществляется по соотношению:

1 байт=8 бит

Следовательно, для определения количества символов в текстовом файле необходимо сначала перевести размер файла из байтов в биты, затем определить мощность алфавита, вычислить количество информации в одном символе и после этого рассчитать количество символов по формуле:

$$K = \frac{V}{\log_2 N}$$

Используемый в работе подход основан на синтаксическом уровне измерения информации и не учитывает особенности кодировок и служебной информации файловой системы, поэтому полученное количество символов

является теоретическим и может незначительно отличаться от реального значения.

Лабораторная работа

Задание 1

Текстовый файл имеет объём 4 Кбайт. Текст записан русскими буквами, используется пробел и точка.

1. Определите мощность алфавита.
2. Найдите количество информации в одном символе.
3. Определите количество символов в тексте.

Задание 2

Файл объёмом 2 Кбайт содержит текст, набранный латинскими буквами и пробелами.

Определите количество символов в файле.

Задание 3 (варианты)

Определите количество символов в тексте по известному объёму файла.

Вариант	Объём файла	Используемый алфавит
1	1 Кбайт	Русские буквы, пробел
2	2 Кбайт	Русские буквы, пробел
3	4 Кбайт	Русские буквы, пробел
4	8 Кбайт	Русские буквы, пробел
5	16 Кбайт	Русские буквы, пробел
6	1 Кбайт	Латинские буквы, пробел
7	2 Кбайт	Латинские буквы, пробел
8	4 Кбайт	Латинские буквы, пробел
9	8 Кбайт	Латинские буквы, пробел
10	16 Кбайт	Латинские буквы, пробел

Задание 4. Практическое исследование

1. Создайте текстовый файл.
2. Наберите в нём произвольный текст.
3. Определите размер файла на диске.
4. Определите мощность алфавита.
5. Вычислите предполагаемое количество символов.
6. Сравните результат с реальным количеством символов в редакторе.
7. Объясните возможные расхождения.

Контрольные вопросы

1. Как связаны объём файла и количество символов?
2. По какой формуле определяется число символов по объёму информации?
3. Почему необходимо переводить байты в биты?
4. Как мощность алфавита влияет на длину текста при фиксированном объёме файла?
5. Почему реальный размер файла может отличаться от теоретического?

Лабораторная работа №10 «Расчёт объёма графической информации»

Графическая информация может быть представлена в аналоговой или дискретной форме. Примером аналогового представления графической информации является, например, картина на холсте, а примером дискретного представления – изображение на экране монитора, состоящее из отдельных точек – пикселей (pixel) разного цвета.

Получение цифрового представления изображения основано на выполнении пространственной дискретизации аналогового изображения (осуществлении аналого-цифрового преобразования). Данный процесс заключается в разбиении непрерывного (аналогового) изображения на отдельные мелкие фрагменты, после чего цвет каждого фрагмента (а точнее – 13 код цвета, например, в цветовой системе RGB) записывается в ячейку таблицы с координатами, соответствующими координатам фрагмента исходного изображения.

Одним из устройств, которое выполняет дискретизацию изображения, является сканер. К основным параметрам, определяющим результат работы сканера, относятся следующие.

1. Оптическое разрешение, которое измеряется в точках на дюйм (dots per inch – dpi). Обычно указывается два значения, например 600 x 1200 dpi, где горизонтальное разрешение (первая цифра) определяется CCD-матрицей сканера, а вертикальное (вторая цифра) определяется количеством шагов двигателя на дюйм.

2. Глубина цвета определяется качеством CCD-матрицы и разрядностью АЦП. Измеряется количеством оттенков, которые устройство способно распознать (например, 24 бита соответствуют 16 777 216 оттенкам). В настоящее время сканеры выпускают с глубиной цвета 24, 30 и 36 бит.

В зависимости от выбранного способа кодирования графической информации различают векторные и растровые изображения.

Векторное изображение содержит объекты, определяемые математическими уравнениями, которые содержат информацию о размере, форме, цвете, границе и местоположении каждого объекта. Достоинства векторных изображений – малый объем файлов и отсутствие потерь качества при масштабировании. Возможные форматы файлов векторных изображений: wmf, svg и др.

Растровое изображение представляет собой совокупность (матрицу) пикселей, каждый из которых имеет определенный цвет в заданной цветовой модели. Растровая графика наиболее распространена там, где требуется создание детализированных реалистичных изображений со множеством оттенков. Для хранения растровых изображений требуется больший объем памяти по сравнению с векторными изображениями, так как хранятся данные о каждом 14 пикселе изображения. При масштабировании растрового изображения ухудшается его качество. Возможные форматы файлов растровых изображений: jpeg, bmp, tiff, raw и др.

Цифровое изображение обычно описывается следующими параметрами.

1. Глубина цвета I (измеряется в битах) – количество битов, используемых для представления цвета при кодировании одного пикселя изображения. $I = log2\ N$, где N – количество цветов в изображении, а I – глубина цвета.

2. Цветовой диапазон N – максимальное количество цветов (палитра) в изображении, вычисляемое по формуле

$$N=2^i$$

3. Размер изображения – количество пикселей по горизонтали (w) и по вертикали (h). Чтобы получить количество пикселей, необходимо размер в дюймах умножить на оптическое разрешение в точках на дюйм – dpi.

4. Объем памяти, занимаемой изображением II (измеряется в битах, байтах) вычисляется по формуле

$I_I = I \cdot h \cdot w$, где I_i – объем памяти, занимаемой изображением, I – глубина цвета.

ЗАДАЧА 1.

Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 128×128 пикселей при условии, что в изображении могут использоваться 256 различных цветов?

Решение.

По условию w = 128 px, h = 128 px, N = 256 цветов. Один пиксель кодируется 8 битами памяти, так как 2^8= 256. Тогда I = 8 бит.

Тогда минимальный объем памяти для хранения изображения составляет

$$I_I = I \cdot h \cdot w. 15$$

I_I = 128 · 128 · 8 = $2^7 \cdot 2^7 \cdot 2^3$= 2^{17} бит = 2^{14} байт = 2^4 Кбайт = 16 Кбайт.

Ответ: 16 Кбайт.

ЗАДАЧА 2. Автоматическая камера производит растровые изображения размером 600 на 1000 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 250 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Решение.

По условию w = 600 px, h = 1000 px, II < 250 · 2^{13} бит. Объём растрового изображения находится как $I_I = I \cdot h \cdot w$. Составим неравенство

$$600 \cdot 1000 \cdot x < 250 \cdot 2^{13},$$
$$600 \cdot 4 \cdot x < 2^{13},,$$
$$2400 \cdot x < 2^8 \cdot 2^5,,$$
$$75 \cdot 2^5 \cdot x < 256 \cdot 2^5,, ,$$
$$75 \cdot x < 256 ,$$
$$x < 3{,}41.$$

Следовательно, x = 3 бит. Значит, в изображении можно использовать не более 23 = 8 цветов.

Ответ: 8 цветов.

ЗАДАЧА 3.

В процессе преобразования растрового графического файла количество цветов уменьшилось с 256 до 16. Во сколько раз уменьшился информационный объем файла?

Решение.

Объём растрового изображения находится по формуле

$$I_I = I \cdot h \cdot w \text{ , где } I = log_2 N$$

Объем первого изображения будет больше, чем объем второго изображения в $k = \frac{I_{I1}}{I_{I2}} = \frac{I_1 * h_1 * w_1}{I_2 * h_2 * w_2}$ раз. Поскольку при преобразовании изображения его размеры не изменялись, это соотношение примет вид.

$$k = \frac{I_1}{I_2} = \frac{log_2 N_1}{log_2 N_2}$$

$$k = \frac{log_2 256}{log_2 16} = \frac{8}{4} = 2$$

Ответ: информационный объем файла уменьшился в 2 раза.

Лабораторная работа

Вариант 1

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 64×256 пикселей при условии, что в изображении могут использоваться 128 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 512 на 1024 пикселя. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 384 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов увеличилось с 64 до 4096. Во сколько раз увеличился информационный объём файла, если размеры изображения не изменились?

Вариант 2

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 512×128 пикселей при условии, что в изображении могут использоваться 32 различных цвета?

Задача 2. Автоматическая камера производит растровые изображения размером 800 на 600 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 450 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов уменьшилось с 1024 до 32. Во сколько раз уменьшился информационный объём файла, если размеры изображения не изменились?

Вариант 3

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 256×256 пикселей при условии, что в изображении могут использоваться 512 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 1280 на 720 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 1350 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов увеличилось с 16 до 256. Во сколько раз увеличился информационный объём файла, если размеры изображения не изменились?

Вариант 4

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 1024×512 пикселей при условии, что в изображении могут использоваться 16 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 640 на 480 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 300 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов уменьшилось с 65536 до 256. Во сколько раз уменьшился информационный объём файла, если размеры изображения не изменились?

Вариант 5

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 640×320 пикселей при условии, что в изображении могут использоваться 2048 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 1920 на 1080 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 4050 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов увеличилось с 8 до 512. Во сколько раз увеличился информационный объём файла, если размеры изображения не изменились?

Вариант 6

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 800×200 пикселей при условии, что в изображении могут использоваться 64 различных цвета?

Задача 2. Автоматическая камера производит растровые изображения размером 400 на 300 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 75 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов уменьшилось с 256 до 4. Во сколько раз уменьшился информационный объём файла, если размеры изображения не изменились?

Вариант 7

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 192×192 пикселей при условии, что в изображении могут использоваться 1024 различных цвета?

Задача 2. Автоматическая камера производит растровые изображения размером 1600 на 900 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 2250 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов увеличилось с 32 до 16384. Во сколько раз увеличился информационный объём файла, если размеры изображения не изменились?

Вариант 8

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 384×256 пикселей при условии, что в изображении могут использоваться 8 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 128 на 256 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 24 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов уменьшилось с 4096 до 64. Во сколько раз уменьшился информационный объём файла, если размеры изображения не изменились?

Вариант 9

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение

размером 512×1024 пикселей при условии, что в изображении могут использоваться 65536 различных цветов?

Задача 2. Автоматическая камера производит растровые изображения размером 720 на 576 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 324 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов увеличилось с 2 (чёрно-белое) до 256. Во сколько раз увеличился информационный объём файла, если размеры изображения не изменились?

Вариант 10

Задача 1. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 240×320 пикселей при условии, что в изображении могут использоваться 4 различных цвета?

Задача 2. Автоматическая камера производит растровые изображения размером 1024 на 768 пикселей. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Объём файла с изображением не может превышать 768 Кбайт без учёта размера заголовка файла. Какое максимальное количество цветов можно использовать в палитре?

Задача 3. В процессе преобразования растрового графического файла количество цветов уменьшилось с 16777216 (True Color) до 65536. Во сколько раз уменьшился информационный объём файла, если размеры изображения не изменились?

Контрольные вопросы

1. Как определяется глубина цвета (I) для растрового изображения?
2. По какой формуле рассчитывается информационный объём растрового файла?
3. Каков информационный вес пикселя в черно-белом изображении без градаций серого?
4. Как изменится объём файла при увеличении разрешения изображения в 4 раза?
5. Почему при уменьшении количества цветов в палитре объём файла сокращается?
6. Достаточно ли для хранения изображения 256 Кбайт, если оно имеет размер 512×512 пикселей и 256 цветов?

Лабораторная работа №11 «Расчёт объёма звуковой информации»

Цель работы:

Теоретические сведения

Слуховая система человека способна воспринимать упругие волны, имеющие частоту в пределах от 16 Гц до 20 кГц. Упругие волны в любой среде, частоты которых лежат в указанных пределах, называют звуковыми. Регистрация звуковых волн (звука) выполняется с помощью микрофона, который выполняет преобразование звуковой волны в электрический сигнал. Поскольку этот электрический сигнал на выходе микрофона непрерывен, для его ввода в ЭВМ нужно преобразовать его в цифровую форму, т. е. выполнить дискретизацию сигнала.

Дискретизация сигнала во времени – это преобразование непрерывного аналогового сигнала в последовательность его значений в дискретные моменты времени. Эти значения называются отсчетами, или выборками. Данную функцию выполняет аналого-цифровой преобразователь (АЦП).

Аналого-цифровой преобразователь осуществляет следующие функции. 1. Дискретизацию сигнала по времени, т. е. измерение уровня интенсивности звука в определенные фиксированные моменты времени. Частоту, характеризующую периодичность измерения звукового сигнала, принято называть частотой дискретизации. Частота дискретизации выбирается на основе теоремы Котельникова и должна быть как минимум в 2 раза больше 17 максимальной частоты спектра сигнала. Вместе с тем считается, что человек не слышит звук с частотой более 20 000 Гц (20 кГц), поэтому для высококачественного воспроизведения звука верхнюю границу обычно принимают равной 22 кГц. Следовательно, частота дискретизации сигнала должна быть не меньше 44 кГц;

2. Дискретизацию амплитуды звукового сигнала, т. е. представление диапазона интенсивности звука с помощью набора уровней (например, 256 или 65536); текущий уровень измеряемого сигнала округляется до ближайшего из этих уровней. Параметры дискретного звукового сигнала:

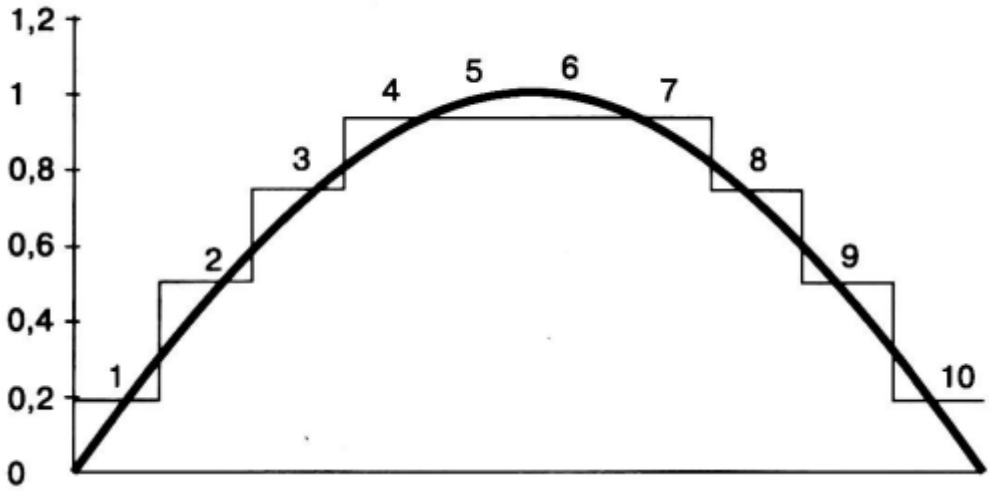
1) длительность сигнала t (измеряется в секундах);

2) глубина кодирования звука I (измеряется в битах) – количество битов, отводимых для хранения одного отсчета дискретного сигнала. Обычно используют 8-битное, 16-битное и 20-битное представление значений амплитуды;

3) частота дискретизации F (измеряется в герцах) – количество измерений амплитуды аналогового сигнала в секунду, котрое показывает, сколько значений сигнала запоминается за одну секунду;

4) количество звуковых каналов записи N (N = 1 – монозапись, N = 2 – стереозапись, N = 4 – запись в формате квадро, N = 8 – формат 7.1);

5) размер файла Is (измеряется в байтах): Is = F · I · t · N. На рисунке 1 показан пример дискретизации аналогового сигнала f(x) ($f(x) \in [0; 1]$]) длительностью в 1 секунду. Частота дискретизации – 10 Гц, т. е. из аналогового сигнала длительностью в 1 секунду формируется дискретный сигнал, состоящий из 10 отсчетов. Глубина кодирования – 4 бита, т. е. всего существует 16 возможных значений амплитуды: 0/16, 1/16, 2/16, ..., 14/16, 15/16. Цифрами на рисунке надписаны номера отсчетов дискретного сигнала.



Аудиофайлы, полученные при оцифровке одной и той же записи, могут отличаться по одному, нескольким или даже всем параметрам.

Соответственно у таких файлов будет также разный размер. Например, если глубина кодирования звука одного файла в m раз больше (меньше), чем у другого файла, то при прочих равных условиях размер первого файла будет в m раз больше (меньше), чем размер второго файла:

$$\frac{I_{s1}}{I_{s2}} = \frac{F \times (m \times I) \times t \times N}{F \times I \times t \times N} = \frac{m}{1}$$

Аналогичным образом определяется объем звукового файла при изменении других (одного, нескольких или всех) его параметров.

ЗАДАЧА 1.

Определите объем памяти для хранения аудиофайла, содержащего стереозапись, время звучания которого составляет 1с при частоте дискретизации 48 кГц и разрешении 8 бит.

Решение.

По условию F = 48 кГц = 48 000 Гц, I = 8 бит = 1 байт, N = 2 (стереозапись), t = 1 с. Объем памяти для хранения аудиофайла вычисляется по формуле

$$I_s = F \cdot I \cdot t \cdot N.$$

I_s = 48000·8·1·2 = 96000 байт, I_s = 96000 / 1024 = 93,75 Кбайт.

Ответ: 93,75 Кбайт.

ЗАДАЧА 2. Рассчитайте время звучания моноаудиофайла, если при 16-битном кодировании и частоте дискретизации 32кГц его объем равен 700 Кбайт.

Решение.

По условию Is = 700 Кбайт = 700·1024 байт, F = 32 кГц = 32000 Гц, I = 16 бит = 2 байта, N = 1 (монозапись).

Время звучания файла вычисляется по формуле

$$I_{s1} = F \cdot I \cdot t \cdot N.$$

Тогда t = I_{s1} / (F·I·N). Получим t = (700·1024) / (32000·2·1) = 11,2 секунды. Ответ: 11,2 секунды.

ЗАДАЧА 3.

Музыкальный фрагмент записан в формате квадро (четырёхканальная запись), оцифрован и сохранён в виде файла без сжатия данных. Размер полученного файла без учёта размера заголовка файла – 12 Мбайт. Затем тот же музыкальный фрагмент записан повторно в формате моно и оцифрован с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер в Мбайт файла, полученного при повторной записи.

Решение.

Продолжительность звучания для 1-го файла $I_{s1} = F_1 \cdot I_1 \cdot t_1 \cdot N_1$, для 2-го файла $I_{s2} = F_2 \cdot I_2 \cdot t_2 \cdot N_2$, .

По условию F_2 = = F_1/ 1,5 = (2/3) · = F_1; $t_2 = t_1$; N_2 = (1/4) · N_1 ; I_1 = 2 · I_1 .

Тогда для 2-го файла: I_{s2} = (2/3)· F_1·2·I_1·t_1·(1/4)· N_1 = 1/3 · I_{s1}, I_{s2} = 12/3 = 4 Мбайт. Ответ: 4 Мбайт.

Лабораторная работа

Вариант 1

Задача 1. Определите объем памяти (в Кбайтах) для хранения стереоаудиофайла, время звучания которого составляет 5 секунд при частоте дискретизации 44.1 кГц и разрешении 16 бит.

Задача 2. Рассчитайте время звучания (в секундах) стереоаудиофайла, если при 8-битном кодировании и частоте дискретизации 48 кГц его объём равен 187.5 Кбайт.

Задача 3. Музыкальный фрагмент был записан в формате стерео, оцифрован и сохранён в виде файла без сжатия данных. Размер полученного файла – 48 Мбайт. Затем тот же фрагмент записали в формате моно, с частотой дискретизации в 2 раза выше, но с разрешением в 4 раза ниже, чем в первый раз. Сжатие данных не производилось. Укажите размер файла (в Мбайтах), полученного при повторной записи.

Вариант 2

Задача 1. Определите объем памяти (в Мбайтах) для хранения квадроаудиофайла (4 канала), время звучания которого составляет 3 минуты при частоте дискретизации 32 кГц и разрешении 24 бита.

Задача 2. Рассчитайте время звучания (в секундах) моноаудиофайла, если при 24-битном кодировании и частоте дискретизации 96 кГц его объём равен 21 Мбайт.

Задача 3. Музыкальный фрагмент был записан в формате моно, оцифрован с частотой 22.05 кГц и глубиной 8 бит. Размер файла – 5 Мбайт. Затем тот же фрагмент записали в формате стерео, с частотой дискретизации 44.1 кГц и глубиной 16 бит. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 3

Задача 1. Определите объем памяти (в байтах) для хранения моноаудиофайла, время звучания которого составляет 0.5 секунды при частоте дискретизации 8 кГц и разрешении 8 бит.

Задача 2. Рассчитайте частоту дискретизации (в кГц) для стереоаудиофайла, если известно, что его объём равен 1.5 Мбайт, время звучания – 10 секунд, а разрешение – 16 бит.

Задача 3. Музыкальный фрагмент был записан в формате стерео (2 канала), оцифрован с частотой 48 кГц и глубиной 16 бит. Размер файла – 30 Мбайт. Затем тот же фрагмент записали в формате квадро (4 канала) с частотой дискретизации в 1.5 раза меньше, но с разрешением в 1.5 раза выше. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 4

Задача 1. Определите объем памяти (в Мбайтах) для хранения стереоаудиофайла, время звучания которого составляет 2 минуты при частоте дискретизации 96 кГц и разрешении 32 бита.

Задача 2. Рассчитайте глубину кодирования (в битах) для моноаудиофайла, если известно, что его объём равен 4 Мбайт, время звучания – 25 секунд, а частота дискретизации – 44.1 кГц.

Задача 3. Музыкальный фрагмент был записан в формате квадро, оцифрован с частотой 96 кГц и глубиной 24 бита. Размер файла – 100 Мбайт. Затем тот же фрагмент записали в формате моно с частотой дискретизации в 4 раза ниже и с разрешением в 2 раза ниже. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 5

Задача 1. Определите объем памяти (в Кбайтах) для хранения моноаудиофайла, время звучания которого составляет 30 секунд при частоте дискретизации 22.05 кГц и разрешении 16 бит.

Задача 2. Рассчитайте время звучания (в минутах) стереоаудиофайла, если при 32-битном кодировании и частоте дискретизации 192 кГц его объём равен 1.125 Гбайт.

Задача 3. Музыкальный фрагмент был записан в формате стерео, оцифрован и сохранён в виде файла без сжатия данных. Размер полученного файла – 60 Мбайт. Затем тот же фрагмент записали в формате моно, с частотой дискретизации в 3 раза ниже, но с разрешением в 3 раза выше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла (в Мбайтах), полученного при повторной записи.

Вариант 6

Задача 1. Определите объем памяти (в Мбайтах) для хранения стереоаудиофайла, время звучания которого составляет 1 час при частоте дискретизации 44.1 кГц и разрешении 16 бит.

Задача 2. Рассчитайте глубину кодирования (в битах) для квадроаудиофайла (4 канала), если известно, что его объём равен 75 Мбайт, время звучания – 20 секунд, а частота дискретизации – 48 кГц.

Задача 3. Музыкальный фрагмент был записан в формате моно, оцифрован с частотой 11.025 кГц и глубиной 8 бит. Размер файла – 2 Мбайт. Затем тот же фрагмент записали в формате стерео с частотой дискретизации и разрешением, увеличенными в 2 раза по сравнению с первым файлом. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 7

Задача 1. Определите объем памяти (в байтах) для хранения стереоаудиофайла, время звучания которого составляет 100 миллисекунд (0.1 с) при частоте дискретизации 192 кГц и разрешении 24 бита.

Задача 2. Рассчитайте частоту дискретизации (в Гц) для моноаудиофайла, если известно, что его объём равен 864 Кбайт, время звучания – 12 секунд, а разрешение – 16 бит.

Задача 3. Музыкальный фрагмент был записан в формате стерео, оцифрован с частотой 32 кГц и глубиной 8 бит. Размер файла – 10 Мбайт. Затем тот же фрагмент записали в формате квадро с частотой дискретизации в 2 раза выше, но с разрешением, уменьшенным в 4 раза. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 8

Задача 1. Определите объем памяти (в Кбайтах) для хранения моноаудиофайла, время звучания которого составляет 10 секунд при частоте дискретизации 16 кГц и разрешении 8 бит.

Задача 2. Рассчитайте время звучания (в секундах) квадроаудиофайла, если при 16-битном кодировании и частоте дискретизации 88.2 кГц его объём равен 42.1875 Мбайт.

Задача 3. Музыкальный фрагмент был записан в формате квадро, оцифрован и сохранён в виде файла без сжатия данных. Размер полученного файла – 24 Мбайт. Затем тот же фрагмент записали в формате моно, с частотой дискретизации в 1.5 раза выше, но с разрешением в 1.5 раза ниже, чем в первый раз. Сжатие данных не производилось. Укажите размер файла (в Мбайтах), полученного при повторной записи.

Вариант 9

Задача 1. Определите объем памяти (в Мбайтах) для хранения стереоаудиофайла, время звучания которого составляет 45 секунд при частоте дискретизации 176.4 кГц и разрешении 32 бита.

Задача 2. Рассчитайте количество каналов (моно/стерео/квадро) для аудиофайла, если известно, что его объём равен 50 Мбайт, время звучания – 15 секунд, частота дискретизации – 96 кГц, а разрешение – 24 бита.

Задача 3. Музыкальный фрагмент был записан в формате моно, оцифрован с частотой 8 кГц и глубиной 16 бит. Размер файла – 0.5 Мбайт. Затем тот же фрагмент записали в формате стерео, с частотой дискретизации и разрешением, увеличенными в 4 раза по сравнению с первым файлом. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Вариант 10

Задача 1. Определите объем памяти (в Кбайтах) для хранения квадроаудиофайла (4 канала), время звучания которого составляет 2 секунды при частоте дискретизации 96 кГц и разрешении 8 бит.

Задача 2. Рассчитайте частоту дискретизации (в кГц) для стереоаудиофайла, если известно, что его объём равен 900 Кбайт, время звучания – 5 секунд, а разрешение – 8 бит.

Задача 3. Музыкальный фрагмент был записан в формате стерео, оцифрован с частотой 44.1 кГц и глубиной 16 бит. Размер файла – 40 Мбайт. Затем тот же фрагмент записали в формате квадро с частотой дискретизации и разрешением, уменьшенными в 2 раза по сравнению с первым файлом. Сжатие данных не производилось. Укажите размер нового файла (в Мбайтах).

Контрольные вопросы

1. Что такое частота дискретизации и как она влияет на качество и размер аудиофайла?
2. Что такое глубина кодирования (разрешение) звука? В каких единицах измеряется?
3. По какой формуле рассчитывается информационный объём цифрового аудиофайла?
4. Как количество каналов (моно, стерео, квадро) влияет на объём файла при прочих равных параметрах?
5. Что произойдёт с размером файла, если частоту дискретизации увеличить в 2 раза, а глубину кодирования уменьшить в 4 раза?
6. Как связаны между собой битрейт аудиопотока и параметры его оцифровки?

Лабораторная работа №12 «Шифр Цезаря»

Цель работы:

Криптосистема – алгоритм (или ряд алгоритмов), необходимый для реализации шифрования и расшифрования. Для наших целей слова «зашифровать» и «закодировать» будут использоваться взаимозаменяемо, как и «расшифровать» и «декодировать». Идея шифрования: мы хотим, чтобы какое-то сообщение было доставлено куда-то безопасным способом, не будучи перехваченным и прочитанным «плохими парнями». Открытый текст: сообщение перед шифрованием. Шифрованный текст или криптограмма: сообщение после шифрования.

Криптоанализ – наука о методах расшифрования криптограмм, без знания предназначенного для этого ключа. Криптоаналитик может пытаться провести атаку с целью определения ключа или провести атаку для восстановления исходного текста без знания ключа. Симметричный шифр строится с помощью различных комбинаций двух типов шифров: шифров замены и шифров перестановки. Шифр перестановки – это метод симметричного шифрования, в котором при построении криптограммы переставляют местами элементы открытого текста. Элементами исходного текста, которые переставляются при шифровании, могут быть буквы, пары букв (биграммы), тройки букв (триграммы) и так далее.

В шифрах замены, в отличие от шифра перестановки, во время шифрования буквы открытого текста заменяются на символы из того же или другого алфавита. Шифр Цезаря. Самым старым из известных шифров замены является шифр Цезаря, в котором каждая буква меняется на букву, смещенную на три позиции в алфавите (сдвиг буквы на три позиции в алфавите). Например, последние три буквы в алфавите заменяются на первые три буквы:

А меняется на Г
Б меняется на Д
В меняется на Е
Г меняется на Ж
…
Ю меняется на Б
Я меняется на В:

Выходит следующая таблица замен:

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

Применим шифр Цезаря к сообщению:

АНЯ ОЧЕНЬ УМНАЯ ДЕВУШКА

Убрав пробелы, получим криптограмму:

ГРВСЪЗРЯЦПРГВЖЗЕЦЫНГ

Построим математическую модель шифра Цезаря. Сопоставим каждому символу алфавита его порядковый номер (начиная с 0). Например, для русского алфавита:

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	…	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…	22	23	24	25	26	27	28	29	30	31	32

Тогда шифрование можно выразить формулой: $y = x + 3 \ (mod\ 33)$,

где x – символ открытого текста, y – символ шифрованного текста (то есть при шифровании вместо x записываем y).

Расшифрование можно выразить формулой: $x = y - 3 \ (mod\ 33)$.

Алфавит можно расширить, добавив пробелы, знаки препинания, прописные буквы и так далее. Это изменит операцию по модулю 33 на операцию по модулю мощности нового алфавита.

Шифр Цезаря со сдвигом на произвольное число позиций. Шифр Цезаря можно обобщить, позволив сдвиг на произвольное количество позиций. При таком шифровании число позиций, на которое смещается буква, держится в секрете и называется ключом шифрования/расшифрования. В таком шифровании/расшифровании происходит сдвиг на k позиций в произвольном алфавите из n букв:

$y = x + k \ (mod\ n)$,

$x = y - k \ (mod\ n)$/

x – символ открытого текста, y – символ шифрованного текста, n – мощность алфавита (кол-во символов), k – ключ. Можно заметить, что суперпозиция (то есть последовательное применение) двух шифрований на ключах $k1$ и $k2$ есть просто шифрование на ключе $k1 + k2$.

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке. При шифровании исходного текста каждая буква заменяется другой буквой того же алфавита по следующему правилу. Заменяющая буква определяется путем смещения по алфавиту к концу от исходной буквы на k букв. При достижении конца алфавита выполняется циклический переход к его началу. Например: пусть A – используемый алфавит:

$$A=\{a1, a2,\ldots,am,...,aN\},$$

где a1, a2,…,am,...,a

Алфавит:

Буква	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
Номер	1	2	3	4	5	6	7	8	9	10	11
Буква	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Номер	12	13	14	15	16	17	18	19	20	21	22
Буква	Х	Ц	Ч	Ш	Щ	Ь	Ы	Ъ	Э	Ю	Я
Номер	23	24	25	26	27	28	29	30	31	32	33

Пример:

Сообщение	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я
Номер 1	12	18	10	17	20	16	4	18	1	22	10	33
Номер 1 +5	17	23	15	22	25	21	9	23	6	27	15	5
Шифр	П	Х	Н	Ф	Ч	У	З	Х	Е	Щ	Н	Д

Ответ: «Пхнфчузхещнд»

Практическое занятие

Задание 1:

Зашифровать шифром Цезаря, используя английский алфавит, следующие предложения.

1. ABBI IS INCREDIBLY AWESOME.
2. I LOVE COLLEGE ALGEBRA.
3. WE ARE THE ANOKA FAMILY.
4. WINTER IS SUPER COLD.
5. SENIORS ROCK MY SOCKS.

Расшифровать следующие криптограммы методом Цезаря.

1. FDFLH FDUUROO HQMRBV PDWK
2. VXEZDB VDQGZLFKHV DUH WDVWB
3. PLQQHVRWD JRSKHUV
4. FKRFRODWH PDNHV WKH ZRUOG JR URXQG
5. IXCCB VZHDWHUV NHHS BRX ZDUP

Задание 2:

Зашифровать обобщенным шифром Цезаря, используя английский алфавит (в скобках задан ключ (сдвиг)).

1. HAPPY HOLIDAYS EVERYBODY (‘k’)
2. TIM LIKES TO JUMP ROPE (‘h’)
3. I LOVE CIS DONUT FRIDAYS (‘d’)

4. THE DOG FETCHED A BALL (‘y’)
5. SPONGEBOB IS YELLOW (‘u’)

Расшифровать криптограммы обобщенным методом Цезаря:

1. GNWYMIFD HFPJ NX IJQNHNTZX (‘f’)
2. IQDJQ’I SECYDW JE JEMD (‘q’)
3. XLVP XLYJ XPXZCTPD (‘l’)
4. PCDZP SPCRT ITPB ZXRZH WXVW (‘p’)
5. OZSL’K LZW ESYAU OGJV? (‘s’)

Контрольные вопросы

1. Дайте определение подстановочному шифру и объясните, к какому его типу относится шифр Цезаря. В чём ключевое отличие моноалфавитного шифра от полиалфавитного?

2. Запишите математические формулы для шифрования и дешифрования в шифре Цезаря. Что означает операция взятия по модулю (mod n) и какую роль она играет?

3. Объясните, почему множество преобразований шифра Цезаря образует группу. Каким свойством обладает суперпозиция (последовательное применение) двух шифрований Цезаря?

Лабораторная работа №13 «Шифры замены»

Цель работы:

Теоретические сведения:

Шифр замены, также называемый подстановочным шифром, представляет собой фундаментальный метод криптографического преобразования информации, известный с глубокой древности. Его основная идея заключается в замене каждого элемента исходного открытого текста — будь то отдельная буква, слог или целое слово — на другой элемент в соответствии с заранее установленным фиксированным правилом. Этим правилом, которое и является секретным ключом шифра, задаётся конкретное соответствие между алфавитом открытого текста и алфавитом шифртекста. Важнейшим криптографическим свойством таких шифров является то, что они меняют не положение символов в тексте, как это делают шифры перестановки, а их сущность, при этом статистическая частота появления символов в открытом тексте переносится и отражается в шифртексте. Именно это свойство, являясь основой метода, одновременно составляет и его главную уязвимость, открывающую путь для криптоанализа с помощью частотного анализа.

Исторически и концептуально простейшим представителем шифров замены является шифр Цезаря, названный в честь Гая Юлия Цезаря, использовавшего его для секретной переписки. Этот шифр относится к классу моноалфавитных или одноалфавитных подстановок, где один символ открытого текста всегда заменяется на один и тот же символ шифртекста на протяжении всего сообщения. Алгоритм шифра Цезаря заключается в циклическом сдвиге алфавита на фиксированное число позиций. Например, при сдвиге на 3 позиции в латинском алфавите буква A заменяется на D, B на E, и так далее, а буквы в конце алфавита (X, Y, Z) преобразуются в начала (A, B, C). Математически это можно описать формулой $C = (P + K) \bmod N$, где P — порядковый номер буквы открытого текста, C — номер буквы шифртекста, K — ключ (величина сдвига), а N — мощность используемого алфавита. Несмотря на простоту, шифр Цезаря наглядно демонстрирует принцип замены, однако его криптостойкость ничтожна, так как даже для большого алфавита количество возможных ключей ограничено числом N, и злоумышленник может вскрыть шифр простым перебором всех вариантов (метод brute-force) или, что ещё быстрее, с помощью базового частотного анализа.

Более общим случаем моноалфавитной подстановки является афинный шифр, который можно рассматривать как развитие идеи Цезаря. В нём для шифрования применяется линейная функция вида $C = (a * P + b) \bmod N$. Здесь ключами служат два числа: множитель 'a' и сдвиг 'b'. Для обеспечения возможности однозначного расшифровывания параметр 'a' должен быть взаимно простым с мощностью алфавита N (то есть их наибольший общий

делитель должен быть равен единице). Например, для русского алфавита из 33 букв число 'a' не должно делиться на 3 или на 11. Афинный шифр значительно увеличивает пространство ключей по сравнению с шифром Цезаря, но остаётся в том же классе моноалфавитных шифров и, следовательно, наследует их основную уязвимость перед полноценным частотным анализом.

Особняком среди простых шифров стоит шифр Атбаш, известный ещё как библейский шифр. Его правило замены предельно симметрично: алфавит мысленно записывается в строку, а затем производится замена первой буквы на последнюю, второй — на предпоследнюю и так далее. Для латинского алфавита это означает A→Z, B→Y, C→X и т.д. Для русского алфавита соответствие будет А→Я, Б→Ю, В→Э. Важной особенностью шифра Атбаш является его инволюционность: одна и та же операция применяется как для зашифровывания, так и для расшифровывания. Несмотря на кажущуюся сложность замены, этот шифр также является моноалфавитным и легко вскрывается частотным анализом, хотя его нельзя взломать перебором сдвига.

Наиболее криптостойким в рамках класса одноалфавитных подстановок является шифр произвольной, или случайной, моноалфавитной замены. В нём каждой букве исходного алфавита ставится в соответствие не какая-то соседняя или симметричная, а совершенно произвольная, но уникальная буква алфавита шифртекста. Соответствие задаётся таблицей, представляющей собой случайную перестановку всех символов алфавита. Количество возможных ключей для такого шифра астрономически велико и равно факториалу мощности алфавита (например, 33! для русского). Казалось бы, такой шифр должен быть невскрываемым при отсутствии ключа. Однако его фундаментальная слабость кроется в неизменности подстановки. Поскольку одна и та же буква открытого текста всегда шифруется одним и тем же символом, все статистические закономерности языка — частотность отдельных букв, сочетаний, окончаний — полностью сохраняются в зашифрованном тексте. Это позволяет криптоаналитику, не зная ключа, сопоставить частоту появления зашифрованных символов с известными таблицами частотности языка.

Метод криптоанализа, основанный на этом свойстве, называется частотным анализом. Он был впервые подробно описан арабским учёным Аль-Кинди в IX веке. Аналитик начинает с подсчёта, сколько раз каждый символ встречается в перехваченном шифртексте. Получив ранжированный список, он сравнивает его с эталонным для предполагаемого языка открытого текста. Для русского языка, например, наиболее частотными буквами являются гласные О, Е (вместе с Ё), А, И, а также согласные Н, Т, С, Р. Наименее частотными — буквы Ф, Щ, Ц, Э. Выдвинув гипотезу, что самый частый символ шифртекста соответствует букве «О», а следующий за ним — букве «Е», аналитик делает первую частичную расшифровку. Далее в ход идут дополнительные лингвистические закономерности: поиск однобуквенных слов (в русском языке это предлоги и союзы: В, К, С, У, О, А, И, Я), анализ

частых коротких слов (НЕ, НА, ПО, ЧТО, ЭТОТ), типичных окончаний (-ЫЙ, -ОЙ, -АЯ, -ОЕ, -ИЕ) и устойчивых сочетаний букв (биграмм и триграмм, таких как СТ, НО, ЕН, ТО, СТО, ЕНИ). Работа идёт методом проб, ошибок и логических догадок: угадывается одно слово по контексту, что позволяет определить несколько новых букв замены, затем с этими новыми знаниями угадывается следующее слово, и так до полного восстановления таблицы подстановки и всего открытого текста. Таким образом, частотный анализ превращает взлом шифра из задачи слепого перебора в интеллектуальную лингвистическую головоломку, успех в которой зависит от длины шифртекста (чем он длиннее, тем точнее статистика) и навыков аналитика.

Для преодоления этой фатальной уязвимости были разработаны полиалфавитные шифры, такие как знаменитый шифр Виженера. Их ключевая идея — использовать не одну, а несколько различных алфавитных подстановок, которые применяются к буквам открытого текста циклически или в соответствии с ключевым словом. Это приводит к тому, что одна и та же буква открытого текста в разных позициях может шифроваться разными символами, что эффективно «размывает» частотные характеристики и делает простой частотный анализ бесполезным. Однако и эти шифры были вскрыты более сложными методами криптоанализа, такими как метод Казиски и анализ индекса совпадений, которые позволяют определить длину ключевого слова, а затем применить модифицированный частотный анализ к подпоследовательностям, зашифрованным одним и тем же сдвигом.

В заключение можно сказать, что шифры простой замены сыграли колоссальную роль в истории криптографии, заложив базовые принципы преобразования информации. Их изучение важно не только с исторической точки зрения, но и как наглядная демонстрация фундаментальной связи между криптографией и лингвистикой. Они наглядно показывают, что абсолютная секретность не может опираться только на сложность ключа или алгоритма, если сам алгоритм оставляет в шифртексте статистические следы исходного языка. Борьба с этими следами, их маскировка и уничтожение стали главным двигателем развития криптографии на протяжении многих веков, вплоть до появления современных алгоритмов с открытым ключом и блочных шифров, которые оперируют уже не буквами, а битами, полностью разрывая связь с естественноязыковой статистикой.

Практическое задание

Вам предоставлен шифртекст, полученный в результате шифрования **шифром простой замены** (произвольная моноалфавитная подстановка). Текст на русском языке, знаки препинания удалены, заглавные буквы приведены к строчным, **но пробелы между словами сохранены**, что облегчает анализ.

ЬХЧГЁЁЧХЧ ЮЧФТСРЩ МФЩЭ Ц МЩТПЩОЮЭЦПЩБ Ц ЁЭЦХЧЩЭ. Щ МФЩЭЁЩБЩПЩБ ЁЭЩЕФХЦПЩЗ ЧЩЬФХЩПЩБ, ЧЩБСЩЬ ЁЩПЩЗЩЭЩБС ЩЗЩПЩБЩХФТПЩЗ ЁЩ ПЩПФТЩЭЩБ Ц ПЩЭФЩЕЩБ ЩЭЩПЩЗЩПЩХЩБЩО ЩЗЩЭЩХЧЩЗ. ЩЗЩПЩБЩХФТПЩЗ МЩТЭЩЁЩПЩБ ФЩЭ ЩЭЩЬХЦЩЭ Ц ЩПЩЭЩПЩБЩЬЦПЩЗ, ЦЩПЩ ЗЩОЩЭЩПЩБ Ц ХЩЬЩПЩБЩПЩБС ЩЗЩЭЩХЧЩЗ ЁЩПЩЗЩЭЩБС. ЮЩПЩО ЩЭЩХЧЩЗ ФЩЭ ЬЩПЩЭФТПЩЗ Ц ЁЩЬЦХЦЩЗЩПЩБЩО, ЁЩПЩЗЩЭЩБС МЩТПЩОЮЭЦПЩБ Ц ЩПЩХЦЩЭ Ц МФЩЭ Ц МЩТПЩОЮЭЦПЩБ Ц ЁЭЦХЧЩЭ.

Таблица стандартной частотности букв русского язык:

Буква	Частота %	Буква	Частота %	Буква	Частота %	Буква	Частота %
О	11,08	Р	4,45	Ы	1,96	Х	0,89
Е, Ё	8,41	В	4,33	Ь	1,92	Ш	0,81
А	7,92	К	3,36	З	1,75	Ю	0,61
И	6,83	М	3,26	Г	1,74	Э	0,38
Н	6,72	Д	3,05	Б	1,71	Щ	0,37
Т	6,18	П	2,81	Ч	1,47	Ц	0,36
С	5,33	У	2,80	Й	1,12	Ф	0,19
Л	5,00	Я	2,13	Ж	1,05	Ъ	0,02

1. **Вручную** или с помощью программы Exel **подсчитайте, сколько раз встречается каждый символ** в предоставленном шифртексте.

Исключите из подсчёта пробел.

Результаты занесите в таблицу в порядке убывания частоты.

Сопоставьте полученный частотный рейтинг символов шифртекста со стандартным рейтингом русского языка. Сделайте первые гипотезы: какая самая частая шифробуква, вероятно, соответствует букве «о», следующая — «е» или «а» и т.д. Запишите эти гипотезы.

2. **Расшифровка и контекстуальный анализ**

Перепишите шифртекст в тетрадь или создайте текстовый документ.

На основе ваших гипотез начните делать **пробные замены**. Заменяйте в тексте наиболее частые шифробуквы на предполагаемые буквы открытого текста.

Находите закономерности:

Однобуквенные слова. В русском языке это «в», «и», «а», «я», «у», «с», «к», «о». Если вы видите однобуквенное слово в шифртексте, это сильная подсказка.

Двух- и трёхбуквенные предлоги/союзы: «на», «не», «но», «по», «за», «от», «до», «из», «со», «то», «что», «как», «это».

Частые окончания: «-ый», «-ой», «-ая», «-ое», «-ие», «-ть», «-ся».

Частые биграммы/триграммы: «сто», «ени», «ово», «при», «про».

После каждой пробной замены анализируйте получающийся частично открытый текст. Появляются ли узнаваемые фрагменты слов? Корректируйте свои гипотезы.

Ведите таблицу ключа (соответствия) по мере работы. Пример:

Шифробуква	Предполагаемая буква	Уверенность (высокая/средняя/гипотеза)
щ	? (возможно, «о» или «е»)	гипотеза
э	?	
п	?	
...	...	

4. Финальный этап:

Продолжайте итеративный процесс замен и анализа контекста, пока весь текст не станет читаемым.

Полностью восстановите таблицу подстановки (ключ), использованную для шифрования. Она должна содержать 33 строки (для всех букв русского алфавита).

Запишите полученный открытый текст.

Ответьте на контрольные вопросы:

1. Почему сохранение пробелов в шифртексте значительно облегчает криптоанализ?
2. Какой этап анализа (частотный подсчёт или работа с контекстом) оказался для вас наиболее трудоёмким?
3. Какие буквы или короткие слова стали «ключиками» для начала расшифровки в вашем случае?
4. Как можно усложнить шифр простой замены, чтобы он лучше сопротивлялся частотному анализу?

Лабораторная работа №14 «Шифры перестановки»

Шифр, преобразования из которого изменяют только порядок следования символов исходного текста, но не изменяют их самих, называется шифром перестановки (ШП).

Шифрами перестановки называются такие шифры, преобразования из которых приводят к изменению только порядка следования символов исходного сообщения. Примером преобразования, которое может содержаться в шифре перестановки, является следующее правило. Каждая буква исходного сообщения, стоящая в тексте на позиции с четным номером, меняется местами с предшествующей ей буквой. В этом случае ясно, что и исходное, и шифрованное сообщение состоят из одних и тех же букв.

Рассмотрим преобразование из ШП, предназначенное для шифрования сообщения длиной *n* символов. Его можно представить с помощью таблицы

$$\begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$$

где i_1 - номер места шифртекста, на которое попадает первая буква исходного сообщения при выбранном преобразовании, i_2 - номер места для второй буквы и т.д. В верхней строке таблицы выписаны по порядку числа от 1 до *n*, а в нижней - те же числа, но в произвольном порядке. Такая таблица называется подстановкой степени *n*.

Зная подстановку, задающую преобразование, можно осуществить как шифрование, так и дешифрование текста. Например, если для преобразования используется подстановка

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 2 & 3 & 1 & 4 & 6 \end{pmatrix}$$

и в соответствии с ней зашифровывается слово МОСКВА, то получится КОСВМА. Попробуйте расшифровать сообщение НЧЕИУК, полученное в результате преобразования с помощью указанной выше подстановки.

В соответствии с методом математической индукции, можно легко убедиться в том, что существует $(1 \cdot 2 \cdot 3 \cdot \ldots \cdot n)$ вариантов заполнения нижней строки таблицы (1). Таким образом, число различных преобразований шифра перестановки, предназначенного для шифрования сообщений длины *n*, меньше либо равно *n!* (заметим, что в это число входит и вариант преобразования, оставляющий все символы на своих местах!).

С увеличением числа *n* значение *n!* растет очень быстро. Приведем таблицу значений *n!* для первых 10 натуральных чисел:

n	1	2	3	4	5	6	7	8	9	10
n!	1	2	6	24	120	720	5040	40320	362880	3628800

При больших *n* для приближенного вычисления *n!* можно пользоваться известной формулой Стирлинга

$$n! \approx \sqrt{2 \cdot \pi \cdot n}\left(\frac{n}{e}\right)^n$$

где $e = 2{,}718281828\ldots$

Примером ШП, предназначенного для шифрования сообщений длины *n*, является шифр, в котором в качестве множества ключей взято множество всех подстановок степени *n*, а соответствующие им преобразования шифра задаются, как было описано выше. Число ключей такого шифра равно *n!*.

Для использования на практике такой шифр не удобен, так как при больших значениях *n* приходится работать с длинными таблицами.

Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру. Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по ходу одного «маршрута», а затем по ходу другого выписывается с нее. Такой шифр называют маршрутной перестановкой. Например, можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение будем по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Зашифруем, например, указанным способом фразу:

ПРИМЕРМАРШРУТНОЙПЕРЕСТАНОВКИ

используя прямоугольник размера 4 x 7:

П	Р	И	М	Е	Р	М
Н	Т	У	Р	Ш	Р	А
О	Й	П	Е	Р	Е	С
И	К	В	О	Н	А	Т

Зашифрованная фраза выглядит так:

МАСТАЕРРЕШРНОЕРМИУПВКЙТРПНОИ

Теоретически маршруты могут быть значительно более изощренными, однако запутанность маршрутов усложняет использование таких шифров.

1. Математическое представление перестановки

Рассмотрим преобразование из ШП, предназначенное для шифрования сообщения длиной n символов. Его можно представить с помощью таблицы-подстановки:

text

(1 2 3 ... n)

(i1 i2 i3 ... in)

где i1 - номер позиции в шифртексте, на которую попадает первая буква исходного сообщения, i2 - номер позиции для второй буквы и т.д. В верхней строке таблицы выписаны по порядку числа от 1 до n, а в нижней - те же числа, но в произвольном порядке. Такая таблица называется подстановкой степени n.

Пример: Для подстановки (1 2 3 4 5 6) → (3 1 5 2 6 4) и слова МОСКВА получится:

- М(1)→позиция 3 → КОС_ _ _
- О(2)→позиция 1 → О К С _ _ _
- С(3)→позиция 5 → О К С _ В _
- К(4)→позиция 2 → О К К С _ В _
- В(5)→позиция 6 → О К К С _ В А
- А(6)→позиция 4 → О К К С А В А → ОККСАВА (исправляем наложение) → Правильнее: позиции заполняются строго: на 1-ю позицию ставится 2-я буква (О), на 2-ю - 4-я (К) и т.д.

2. Количество возможных ключей

Число различных преобразований ШП для сообщений длины n равно n! (факториал n), что при росте n дает огромное количество теоретически возможных ключей:

- n=6 → 720 ключей
- n=10 → 3 628 800 ключей
- n=20 → ~2.43×10^{18} ключей

Однако на практике используются ШП с более простой структурой ключа для удобства.

3. Шифры маршрутной перестановки

Широкое распространение получили ШП, использующие геометрическую фигуру (чаще всего - прямоугольник). Сообщение вписывается в таблицу по одному маршруту, а выписывается по другому. Основные виды маршрутов:

- По строкам слева направо → выписывание по столбцам сверху вниз
- По спирали от центра к краям или наоборот
- "Шахматный" порядок и др.

Пример:

Исходный текст: ПРИМЕРМАРШРУТНОЙПЕРЕСТАНОВКИ

Таблица 4×7:

text

П Р И М Е Р М

Н Т У Р Ш Р А

О Й П Е Р Е С

И К В О Н А Т

Маршрут записи: по горизонтали слева направо, построчно. Маршрут считывания для шифрования: по вертикали, начиная с правого верхнего угла, сверху вниз и снизу вверх попеременно. Результат: МАСТАЕРРЕШРНОЕРМИУПВКЙТРПНОИ

4. Уязвимости и методы криптоанализа

1. Сохранение частотного спектра - все символы исходного текста присутствуют в шифртексте в тех же количествах.
2. Сохранение биграммных/триграммных сочетаний, но в разорванном виде.
3. Метод анаграммирования - поиск осмысленных вариантов перестановки символов.
4. Метод подбора размеров таблицы для маршрутных перестановок.

Практическое задание

Криптоанализ маршрутной перестановки

Задание 1: Анализ простой подстановочной перестановки

Дана подстановка степени 6:

text

(1 2 3 4 5 6)

(3 1 5 2 6 4)

Задача:

1. Зашифруйте слово КРИПТО с помощью этой подстановки.
2. Расшифруйте сообщение ТОАКЕР, зашифрованное этой же подстановкой.

Решение оформите в виде таблицы:

Позиция в исходном тексте	1	2	3	4	5	6
Буква исходного текста						
Новая позиция (из подстановки)						
Позиция в шифртексте						
Буква шифртекста						

Задание 2: Криптоанализ маршрутной перестановки

Вам перехвачено сообщение, зашифрованное маршрутной перестановкой с использованием прямоугольной таблицы. Известно, что:

- Исходный текст на русском языке
- Пробелы и знаки препинания удалены
- Сообщение содержит 24 символа
- Маршрут записи: по строкам слева направо
- Маршрут считывания при шифровании: по столбцам сверху вниз

Шифртекст:

text

УИРТАКЧЕИСННОАЦИФЯМЕРПООБ

Задача:

1. Определите возможные размеры прямоугольной таблицы (все пары множителей числа 24).
2. Для каждого возможного размера (m×n) попытайтесь восстановить таблицу, записывая шифртекст по столбцам и читая по строкам.
3. Найдите размер таблицы, при котором получается осмысленный русский текст.
4. Запишите найденный открытый текст.

Пример работы для размера 4×6:

1. Записываем шифртекст в таблицу 4×6 по столбцам:

   text

   Колонка 1: У И Р Т

   Колонка 2: А К Ч Е

   ... и т.д.
2. Читаем по строкам → получаем текст
3. Проверяем на осмысленность

Задание 3. Дешифровка текстов по вариантам

1. Изречение немецкого философа Фридриха Ницце: ОЬТСО НЙАЧУ ЛСВТЯ РЕВЕН ИЛЕТИ ДЕБОЛ

2. Изречение немецкого ученого – гуманиста Эразма Роттердамского ЙЫТЫР КСТНА ЛАТЕН ТЕАДЗ ОСИИЦ АТУПЕ РОООО

3. Изречение чешского писателя Карела Чапека: ЕЛЙГС АМОЛТ ЕМИЬР УНСЕО МОООП МОЖОЕ ОЕКШО ШРАОЬ АЙОСЙ ДОДНДР ОЕЕУО

4. Изречение датского ученого-физика Нильса Бора: ТПРРО УСЕБД ОДИН ОБЖВЛ ООЕЕУ ИОЧОЕ НАДТО ЩНЬЕУ ОТДБУ

5. Изречение французского философа Жана-Поля Сартра: ИНККО ОТСОЧ ЯЧПОТ ЕАРЕЯ ОЛНЕА АЕМТК ОНСТШ

6. Высказывания польского писателя-фантаста Станислава Лемма: ТОУМА МЕЖЕЧ ЫАООО ОММГЗ ЕСНМЕ ДЕООО ЧЫАОД НЛОТМ УМООО ТДЕРО ЕОЧОМ МОООО

7. Высказывание американского писателя Джона Стейнбека: АРЕНО ЫЕТМО ЕЖОИБ ЕДДЖЙ ЯПТВС ОДОКМ ПСИОЖ ОЙЛГО ОИЕН

Контрольные вопросы:

1. Почему количество возможных ключей в шифре перестановки для сообщения длины n равно n!, а не больше?

2. Как по шифртексту, полученному перестановкой, можно определить, что использовался именно ШП, а не шифр замены?

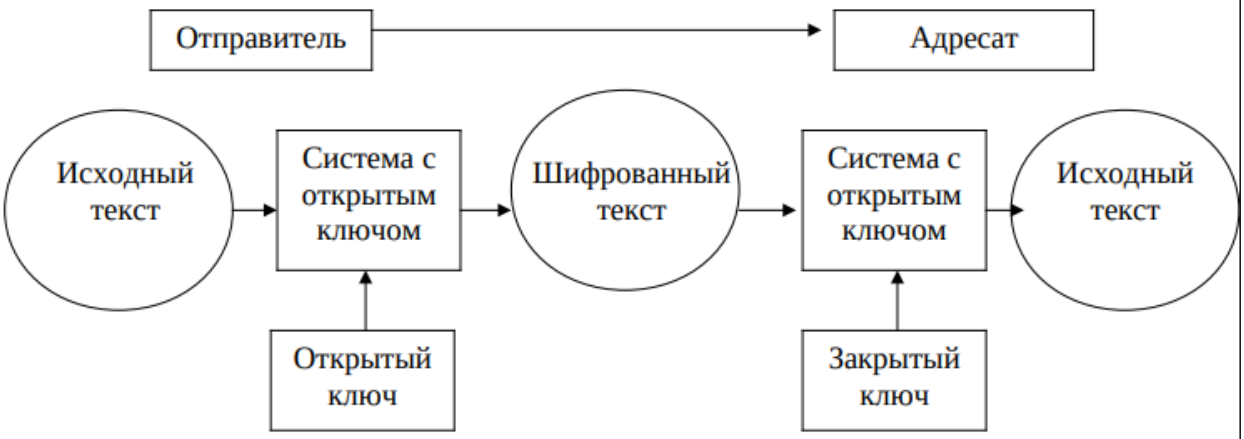
3. Почему маршрутные перестановки с таблицей m×n имеют не m!×n! ключей, а значительно меньше?

4. Какие лингвистические статистики наиболее полезны при криптоанализе ШП?

Лабораторная работа №15 «Сравнение методов шифрования»

Цель работы: изучить принцип работы несимметричных методов шифрования и сравнить их с классическими симметричными шифрами по способу использования ключей и уровню защищённости информации.

В предыдущих лабораторных работах мы рассматривали классические симметричные методы шифрования, такие как шифр Цезаря, шифр простой замены и шифр перестановки. Все эти методы используют **один и тот же секретный ключ** для шифрования и расшифровки информации. Симметричные шифры достаточно просты для понимания и легко реализуются вручную, однако они обладают рядом ограничений. Основной недостаток заключается в необходимости безопасной передачи ключа между отправителем и получателем. Если ключ будет перехвачен третьей стороной, вся информация становится доступной злоумышленнику. Кроме того, классические симметричные шифры легко взламываются методом частотного анализа или подбора ключа, особенно при небольшом алфавите или коротком тексте.



Современные системы защиты информации используют более сложные методы, называемые **несимметричными (асимметричными) шифрами**, которые решают проблему передачи ключа и значительно повышают криптостойкость. В несимметричных методах применяется **пара ключей**: открытый и закрытый. Открытый ключ предназначен для шифрования и может быть доступен всем, кто хочет отправить сообщение. Закрытый ключ используется для расшифровки и хранится в тайне у получателя. Принцип работы асимметричных шифров основан на том, что по открытому ключу практически невозможно вычислить закрытый ключ, даже зная алгоритм шифрования. Таким образом, открытый ключ можно безопасно распространять через публичные каналы связи без угрозы для конфиденциальности информации.

Одним из наиболее известных и распространённых асимметричных алгоритмов является **RSA**, предложенный Ривестом, Шамиром и Адлеманом. Его безопасность основана на математической задаче разложения больших чисел на простые множители, которая считается вычислительно трудной. В

RSA каждый символ или блок текста сначала преобразуется в число, которое затем возводится в степень, заданную открытым ключом, с последующим взятием остатка от деления на модуль n. Полученный результат является зашифрованным числом. Для расшифровки используется закрытый ключ, который позволяет восстановить исходное число и, соответственно, исходный символ текста.

Асимметричные методы обладают рядом преимуществ по сравнению с классическими симметричными шифрами. Во-первых, отпадает необходимость передачи секретного ключа между участниками обмена информацией, что снижает риск компрометации. Во-вторых, они позволяют создавать цифровые подписи, подтверждающие подлинность отправителя и целостность сообщения. В-третьих, асимметричные алгоритмы применяются в сочетании с симметричными шифрами для повышения скорости работы: симметричный ключ используется для шифрования больших объёмов данных, а несимметричный — для безопасной передачи этого ключа.

Однако у несимметричных методов есть и недостатки. Они более ресурсоёмки, требуют больших вычислительных мощностей и неэффективны для шифрования больших объёмов данных напрямую. Поэтому на практике применяются гибридные системы, сочетающие преимущества симметричного и асимметричного шифрования.

Опишем процесс шифрования. Для каждого абонента вырабатывается своя пара ключей. Для этого генерируются два больших простых числа p и q (мастер-ключ), вычисляется произведение $N = p*q$. Затем вычисляется значение функции Эйлера $\varphi(N)$ от числа N. Поскольку p и q – простые числа, $\varphi(N) = (p-1)*(q-1)$.

Затем вырабатывается случайное число e, такое что: $e<\varphi(N)$, e – взаимно простое со значением функции $\varphi(N)$.

После этого ищется число d из условия $e*d = 1(\text{mod } \varphi(N))$. Так как e и $\varphi(N)$ – взаимно просты, $НОД(e,\varphi(N))=1$, то такое число d существует и оно единственно. Число d можно найти с помощью расширенного алгоритма Евклида.

Пара (N, e) является открытым ключом абонента и помещается в открытый доступ.

Пара (N, d) является личным (секретным) ключом. Для расшифровки достаточно знать секретный ключ.

Числа $p, q, \varphi(N)$ в дальнейшем не нужны, поэтому они уничтожаются.

Пользователь А, отправляющий сообщение X абоненту В, выбирает из открытого каталога пару (N, e) абонента В и вычисляет шифрованное сообщение $Y = X^e \pmod N$.

Чтобы получить исходный текст, абонент В вычисляет $Y^d \pmod N = X$.

Пример: Пусть $p = 7$, $q = 17$. Тогда $N = 7*17 = 119$, $\varphi(N) = 6*16 = 96$.

Выбираем значение e: $e<96$, $(e,96)=1$. Пусть выбрано $e = 5$.

Находим d: $e*d = 1(\text{mod } \varphi(N))$, $d = e^{-1} \pmod{\varphi(N)}$. Получаем $d = 77$, так как $77*5 = 4*96+1$.

Открытый ключ (119,5), личный ключ (119,77). Пусть $\boldsymbol{X}$ = 19. Для зашифрования число 19 возводим в 5-ю степень по модулю 119, тогда имеем 19^5 = 2 476 099 и остаток от деления 2 476 099 на 119 равен 66. Итак, $\boldsymbol{Y}$ = 195 mod 119 = 66, расшифровка: $\boldsymbol{X} = 66^{77} \bmod 119 = 19$.

Асимметричные, или несимметричные, шифры представляют собой принципиально иной подход к защите информации. В этих методах используется пара ключей: открытый и закрытый. Открытый ключ доступен всем и применяется для шифрования сообщений, в то время как закрытый ключ хранится в тайне и используется исключительно для расшифровки. Основная особенность асимметричных алгоритмов заключается в том, что по открытому ключу практически невозможно восстановить закрытый ключ, даже если известен алгоритм шифрования. Благодаря этому отпадает необходимость защищённой передачи ключа, что делает такие шифры идеальными для использования в открытых каналах связи. Одним из наиболее известных примеров асимметричного шифрования является алгоритм RSA, который основан на вычислительной сложности разложения больших чисел на простые множители. В RSA исходное сообщение сначала преобразуется в числа, затем возводится в степень, заданную открытым ключом, с последующим взятием остатка от деления на модуль. Расшифровка производится с использованием закрытого ключа, что позволяет восстановить исходное сообщение.

Основные преимущества асимметричных шифров заключаются в высокой криптостойкости и возможности безопасной передачи ключей, а также в реализации цифровых подписей, которые обеспечивают аутентификацию отправителя и проверку целостности данных. Однако вычислительная сложность асимметричных алгоритмов выше, чем у симметричных, что делает их менее эффективными для шифрования больших объёмов данных напрямую. На практике для повышения эффективности используют гибридные системы: симметричный ключ шифруется асимметрично, а само сообщение — симметричным методом. Таким образом удаётся объединить скорость и безопасность шифрования.

Сравнение симметричных и асимметричных методов показывает, что каждый из них имеет свои сильные и слабые стороны. Симметричные шифры просты и быстры, но требуют защищённого канала передачи ключей и обладают ограниченной криптостойкостью при небольших ключах. Асимметричные шифры обеспечивают высокий уровень безопасности и решают проблему передачи ключей, но более ресурсоёмки и сложны в реализации. В современных информационных системах широко применяются комбинации этих методов, что позволяет эффективно защищать данные при передаче через открытые каналы, обеспечивать целостность и подлинность сообщений.

Контрольные вопросы

1. Почему симметричные шифры требуют защищённого канала передачи ключа?

2. Какова основная криптографическая слабость шифра Цезаря и простых шифров замены?

3. В чём принципиальная разница между симметричными и асимметричными методами шифрования?

4. Как обеспечивается безопасность асимметричных шифров, если открытый ключ доступен всем?

5. Почему современные системы используют гибридное шифрование (симметричное + асимметричное)?

6. Какие задачи решает цифровая подпись и какой тип шифра используется для её реализации?

7. В каких ситуациях симметричный шифр предпочтительнее асимметричного и наоборот?

8. Какие критерии применяются для оценки криптостойкости шифров?

Лабораторная работа №16 «Определение хэш-сумм файлов»

Цель работы: ознакомиться с операцией хэширования, определить хэш-суммы файла с помощью встроенных инструментов Windows.

Теоретические сведения

Хеширование (иногда хэширование, англ. *hashing*) — преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями или функциями свёртки, а их результаты называют хешем, хеш-кодом или дайджестом сообщения (англ. *message digest*).

Хеширование применяется для сравнения данных: если у двух массивов хеш-коды разные, массивы гарантированно различаются; если одинаковые — массивы, скорее всего, одинаковы. В общем случае однозначного соответствия между исходными данными и хеш-кодом нет в силу того, что количество значений хеш-функций меньше, чем вариантов входного массива; существует множество массивов, дающих одинаковые хеш-коды — так называемые коллизии. Вероятность возникновения коллизий играет немаловажную роль в оценке качества хеш-функций.

Существует множество алгоритмов хеширования с различными характеристиками (разрядность, вычислительная сложность, криптостойкость и т. п.). Выбор той или иной хеш-функции определяется спецификой решаемой задачи. Простейшими примерами хеш-функций могут служить контрольная сумма или CRC.

Контрольные суммы

Несложные, крайне быстрые и легко реализуемые аппаратные алгоритмы, используемые для защиты от непреднамеренных искажений, в том числе ошибок аппаратуры.

По скорости вычисления в десятки и сотни раз быстрее, чем криптографические хеш-функции, и значительно проще в аппаратной реализации.

Платой за столь высокую скорость является отсутствие криптостойкости — лёгкая возможность подогнать сообщение под заранее известную сумму. Также обычно разрядность контрольных сумм (типичное число: 32 бита) ниже, чем криптографических хешей (типичные числа: 128, 160 и 256 бит), что означает возможность возникновения непреднамеренных коллизий.

Простейший алгоритм — деление сообщения на 16- или 32-битные слова и их суммирование (используется в TCP/IP). Для более надёжного обнаружения ошибок применяют циклические избыточные коды (CRC), например CRC32 в Ethernet или формате ZIP.

- Криптографические хеш-функции

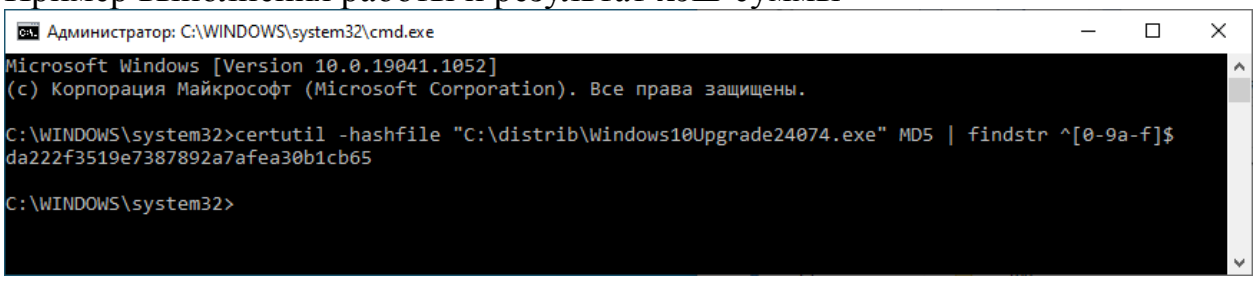
Среди множества существующих хеш-функций принято выделять криптографически стойкие, применяемые в криптографии. Для того чтобы хеш-функция *H* считалась криптографически стойкой, она должна удовлетворять трем основным требованиям, на которых основано большинство применений хеш-функций в криптографии:

- *Необратимость*: для заданного значения хеш-функции *m* должно быть вычислительно неосуществимо найти блок данных X , для которого $H(X) = m$.
- Стойкость к *коллизиям первого рода*: для заданного сообщения *M* должно быть вычислительно неосуществимо подобрать другое сообщение *N*, для которого $H(N) = H(M)$.
- Стойкость к *коллизиям второго рода*: должно быть вычислительно неосуществимо подобрать пару сообщений (M, M') , имеющих одинаковый хеш.

Лабораторная работа

Для выполнения лабораторной работы по определению хеш-сумм файлов на Windows используется встроенная утилита **certutil**.

Пример выполнения работы и результат хэш-суммы



Ход работы:

1. Открыть командную строку (CMD).
2. Перейти в каталог, где находится файл для проверки: `cd "C:\путь\к\папке"`
3. Выполнить команду для вычисления хеш-суммы: certutil -hashfile "имя_файла" [алгоритм]

где [алгоритм] — один из: MD2, MD4, MD5, SHA1, SHA256, SHA384, SHA512.

Пример:

```
certutil -hashfile "example.txt" SHA256
```

4. Скопировать хеш-сумму в таблицу результатов.
5. Для вывода только значения хеш-суммы (без лишнего текста) можно использовать:

```
certutil -hashfile "example.txt" MD5 | findstr ^[0-9a-f]$
```

Файл	Алгоритм	Хеш-сумма
example.txt	MD5	
example.txt	SHA1	
example.txt	SHA256	
example2.docx	MD5	
example2.docx	SHA1	
example2.docx	SHA256	

Контрольные вопросы

1. Что такое хеширование и какова цель использования хеш-функций?
2. Чем отличается контрольная сумма от криптографической хеш-функции?
3. Что такое коллизия в хешировании и почему она может возникать?
4. Назовите три основных требования к криптографически стойкой хеш-функции.
5. Какие алгоритмы хеширования доступны в Windows через утилиту certutil?
6. Как с помощью команды certutil вычислить SHA256-хеш файла example.txt?
7. Как вывести только значение хеш-суммы без дополнительной информации в командной строке?

Лабораторная работа №17 «Проверка целостности данных»

Цель работы: изучить методы проверки целостности данных с использованием хеш-сумм и научиться определять, были ли изменения в файле после передачи или хранения.

Целостность данных — это состояние данных, при котором они не изменены и не повреждены с момента создания или последнего контроля. Проверка целостности данных позволяет обнаружить случайные ошибки или преднамеренные изменения.

Основные методы проверки целостности:

1. **Контрольные суммы** — простые и быстрые алгоритмы, вычисляющие сумму всех элементов файла. Позволяют выявлять случайные ошибки, но не защищают от умышленного изменения данных. Пример: CRC32.
2. **Хеш-функции** — преобразуют файл в строку фиксированной длины. Криптографические хеш-функции, такие как MD5, SHA1, SHA256, позволяют проверить, не было ли изменений в файле, даже если злоумышленник знает алгоритм.
3. **Цифровые подписи** — используются для проверки целостности и подлинности одновременно. Они применяют криптографические методы для защиты от подделки.

Основные свойства хеш-функций для проверки целостности:

- Однозначность: одинаковые файлы дают одинаковый хеш.
- Высокая чувствительность: даже одно изменённое значение файла полностью меняет хеш.
- Необратимость: по хешу нельзя восстановить исходный файл (важно для криптографически стойких алгоритмов).

Применение проверки целостности:

- Передача файлов через интернет (например, ISO-образы, архивы).
- Хранение резервных копий и критических данных.
- Контроль изменений в программных и конфигурационных файлах.

- Выберите один или несколько файлов для проверки целостности.
- Вычислите хеш-сумму выбранного файла до внесения изменений.
- Измените файл (например, откройте и сохраните его с изменением одного символа).
- Снова вычислите хеш-сумму файла.
- Сравните хеш-суммы до и после изменения и сделайте вывод о целостности данных.
- Для выполнения лабораторной работы используйте утилиту Windows **certutil** или аналогичные средства.

Файл	Хеш до изменения	Хеш после изменения	Целостность
example.txt			
example2.docx			

Контрольные вопросы

1. Что такое целостность данных и зачем её проверяют?
2. В чём разница между контрольной суммой и хеш-функцией?
3. Как влияет даже небольшое изменение файла на хеш-сумму?
4. Какие алгоритмы хеширования подходят для проверки целостности данных?
5. Как можно проверить целостность данных после передачи файла по интернету?
6. Почему MD5 не рекомендуется использовать для криптографической проверки целостности?
7. Какие практические задачи решает проверка целостности данных?
8. Что такое цифровая подпись и как она связана с целостностью данных?

Лабораторная работа №18 «Анализ свойств файлов»

Цель работы: изучить методы анализа файлов с точки зрения их размеров, кодировок и других свойств, получить практические навыки работы с файлами для обеспечения информационной безопасности.

Теоретические сведения

Анализ файлов в информационной безопасности позволяет проверять соответствие их содержимого назначению, выявлять аномалии, предотвращать утечки информации и обнаруживать потенциально опасные файлы. Контроль размеров и кодировок, а также изучение метаданных, является частью процедур цифровой криминалистики и общей политики защиты данных, обеспечивая целостность и безопасность информации.

Файл является основной логической единицей хранения данных на компьютере. Для анализа и контроля файлов важно учитывать их размер, кодировку и другие свойства. Размер файла измеряется в байтах, килобайтах или мегабайтах и отражает объём информации, занимаемый файлом на диске. Контроль размера важен для обнаружения аномалий, планирования хранения данных и выявления скрытой информации, которая может быть использована злоумышленниками.

Кодировка текста определяет способ представления символов в виде байтов. Наиболее распространённые кодировки включают ASCII, которая использует один байт на символ и подходит для базовой латиницы; UTF-8, универсальную кодировку для всех языков с переменной длиной символа от одного до четырёх байтов; UTF-16, с фиксированной длиной символа два или четыре байта; а также Windows-1251, применяемую для кириллицы в системах Windows. Анализ кодировки текстовых файлов важен для корректного отображения текста и обнаружения скрытой информации или вредоносного кода, который может быть спрятан в нестандартной кодировке.

Помимо размера и кодировки, к свойствам файла относятся метаданные, такие как расширение, которое указывает на тип файла, даты создания и изменения, а также атрибуты файла, включая системные, скрытые и только для чтения. Свойства могут быть полезны при аудите файлов и расследовании инцидентов, позволяя выявлять подозрительные файлы и отслеживать их изменения.

Практическое задание

1. Выберите 3–5 файлов на компьютере (разные типы: текстовые, исполняемые, документы).
2. Определите для каждого файла:
   - размер;
   - кодировку (для текстовых файлов);
   - расширение;
   - дату создания и изменения.
3. Сделайте вывод о том, соответствуют ли свойства файлов их назначению.
4. Зафиксируйте результаты в таблице.

Ход работы

1. Откройте **Проводник Windows** и найдите нужные файлы.
2. Определите **размер файла**:
   - Правой кнопкой мыши → Свойства → Размер.
   - Или через командную строку:
   - `dir "C:\путь\к\файлу"`
3. Определите **кодировку текстового файла**:
   - Через текстовые редакторы (Блокнот, Notepad++, Visual Studio Code).
   - В Notepad++: меню *Кодировка* → определённая кодировка.
   - Через PowerShell: Get-Content "example.txt" | Out-String -Encoding UTF8
4. Определите **даты создания и изменения** файла:
   - Свойства файла → Создан / Изменён.
   - Через командную строку:
     `dir /T:C "example.txt"` (дата создания)
     `dir /T:W "example.txt"` (дата изменения)

Зафиксируйте расширение файла и дополнительные атрибуты (скрытый, системный, только чтение).

Сравните полученные характеристики с ожидаемыми для данного типа файла.

Таблица результатов

Файл	**Размер**	**Кодировка**	**Расширение**	**Дата создания**	**Дата изменения**	**Атрибуты**	**Вывод**
example.txt			.txt				
document.docx			.docx				
program.exe		N/A	.exe				

Контрольные вопросы

1. Что такое целостность данных и зачем её проверяют?
2. В чём разница между контрольной суммой и хеш-функцией?
3. Как влияет даже небольшое изменение файла на хеш-сумму?
4. Какие алгоритмы хеширования подходят для проверки целостности данных?
5. Как можно проверить целостность данных после передачи файла по интернету?

Лабораторная работа №19 «Работа с архивами и защитой паролем»

Цель работы: исследование защиты с применением пароля, а также исследование методов противодействия атакам на пароль.

Теоретические сведения

Архивация данных представляет собой фундаментальный процесс в области информационных технологий, направленный на решение двух ключевых задач: эффективное сжатие информации для экономии пространства хранения и обеспечение её надёжности при передаче или долговременном хранении. В основе работы архиваторов лежат алгоритмы сжатия без потерь, такие как Deflate (используемый в ZIP), LZMA, BZip2 и другие. Эти алгоритмы устраняют избыточность данных на основе статистических закономерностей (как в алгоритмах Хаффмана или арифметического кодирования) или словарных методов (LZ77, LZ78), что позволяет впоследствии восстановить исходную информацию бит в бит. Процесс архивации включает упаковку одного или нескольких файлов и каталогов в единый контейнер — архивный файл, который помимо сжатых данных содержит метаинформацию (оглавление): имена файлов, их структуру, атрибуты и контрольные суммы.

Защита архивов с применением пароля и её исследование

Для обеспечения конфиденциальности архивируемой информации современные архиваторы поддерживают функцию шифрования содержимого с использованием криптографических алгоритмов, чаще всего на основе симметричных ключей, производных от введённого пользователем пароля. Ключевыми форматами, использующими парольную защиту, являются ZIP (с алгоритмами ZipCrypto или AES-256), RAR (с проприетарным алгоритмом на основе AES-256) и 7z (с использованием AES-256 в режиме CBC). Безопасность такой защиты зависит от двух факторов: криптостойкости самого алгоритма шифрования и стойкости пароля к подбору. На практике даже при использовании стойкого алгоритма AES-256 слабый пароль становится ахиллесовой пятой всей системы защиты. Исследование парольной защиты в рамках данной работы включает анализ процесса преобразования пароля в ключ шифрования через функцию формирования ключа (KDF — Key Derivation Function). Современные архиваторы, такие как 7-Zip, применяют многократное хеширование (например, по алгоритму SHA-256) с большим числом итераций (десятки или сотни тысяч), что значительно замедляет попытки перебора. Однако в устаревших реализациях (например, классический ZipCrypto) используются слабые механизмы, уязвимые к атакам по времени и связанным с ними атакам. Практическая часть работы по архивации с паролем позволяет наблюдать, как устанавливается защита, и впоследствии исследовать поведение системы при попытке доступа к данным без знания пароля или с его неправильным вводом.

Исследование методов противодействия атакам на пароль

Атаки на парольную защиту архивов можно классифицировать на несколько основных типов: атака полного перебора (brute-force), атака по словарю (dictionary attack) и гибридная атака. Атака полного перебора предполагает

последовательную проверку всех возможных комбинаций символов из заданного алфавита (например, цифр от 0 до 9 для трёхзначного пароля). Её эффективность прямо зависит от длины и сложности пароля: ключевое пространство для пароля длиной N символов из алфавита мощностью M равно M^N. Для цифрового трёхзначного пароля это всего 1000 комбинаций, которые могут быть перебраны за доли секунды. Атака по словарю использует предварительно составленный список вероятных паролей (на основе утечек, типовых комбинаций, слов естественного языка), что особенно эффективно против паролей, созданных людьми. Гибридная атака комбинирует оба подхода, добавляя к словарным основам числовые и специальные суффиксы/префиксы.

Противодействие этим атакам строится на нескольких уровнях. Первый и главный уровень — создание стойкого пароля. Рекомендуется использовать длинные (12+ символов) пароли, состоящие из случайной комбинации букв верхнего и нижнего регистра, цифр и специальных символов. Второй уровень — применение современных архиваторов с сильными KDF-функциями, которые искусственно увеличивают время, необходимое для проверки каждого пароля, тем самым делая массовый перебор нецелесообразным по времени даже для относительно коротких паролей. Третий уровень — организационные меры: хранение паролей в надёжных менеджерах паролей, использование двухфакторной аутентификации для доступа к системам, где хранятся архивы, и регулярная смена паролей для критически важных данных. Исследование в рамках лабораторной работы включает практическую оценку времени подбора пароля с использованием специализированного ПО (например, John the Ripper, Hashcat, fcrackzip). Этот эксперимент наглядно демонстрирует экспоненциальную зависимость времени взлома от длины и сложности пароля, а также влияние выбранного алгоритма шифрования и KDF на скорость перебора.

Контроль целостности как основа безопасности данных
Важнейшим аспектом, исследуемым в данной работе, является обеспечение и проверка целостности данных. Целостность гарантирует, что информация не была изменена несанкционированно или повреждена в процессе передачи или хранения. Основным инструментом её контроля являются криптографические хеш-функции, такие как CRC32 (используется для быстрой проверки на ошибки в ZIP), MD5 (сейчас считается криптографически нестойким), SHA-1 и SHA-256. Контрольная сумма, рассчитанная с помощью этих функций, представляет собой уникальный короткий цифровой отпечаток (дайджест) исходного набора данных. Даже минимальное изменение в файле (искажение одного бита) приводит к кардинальному изменению его хеш-суммы. Сравнение контрольных сумм исходного и полученного (или распакованного) файла является стандартной процедурой верификации их идентичности. В контексте архивных файлов контрольные суммы могут применяться как к содержимому отдельных файлов внутри архива, так и ко всей структуре архива в целом. Нарушение целостности архива, вызванное сбоями носителя, ошибками передачи или преднамеренной модификацией, приводит к

невозможности его корректной распаковки. Архиваторы, обнаружив несоответствие контрольных сумм или нарушение структуры файла, генерируют диагностические сообщения об ошибках, такие как «Архив повреждён», «Ошибка контрольной суммы» или «Неверный формат файла». Исследование поведения архиватора при работе с повреждёнными данными позволяет оценить его устойчивость к ошибкам и эффективность механизмов проверки целостности.

Лабораторная работа

1. Создание и архивация текстового файла

Создать текстовый файл, содержащий 50 записей формата "Фамилия Имя Отчество".

Провести архивирование этого файла.

Любым текстовым редактором внести изменения в файл согласно заданию.

В отчёте отразить:

- Контрольную сумму исходного файла.
- Контрольную сумму сжатого файла.
- Сообщения об ошибках при попытке разархивирования искажённого файла.

2. Архивирование файла с паролем

Провести архивацию файла с установкой пароля.

Внести искажения в архивный файл.

Попытаться разархивировать искажённый архив.

В отчёте отразить:

- Контрольную сумму исходного файла.
- Контрольную сумму сжатого файла.
- Сообщения об ошибках при разархивировании искажённого файла.

3. Подбор пароля к архиву

Провести архивацию файла с паролем, состоящим из трёх цифр.

Попытаться подобрать пароль с использованием специализированного программного обеспечения.

В отчёте отразить:

- Контрольную сумму исходного файла.
- Контрольную сумму сжатого файла.
- Выдаваемые программой сообщения.
- Время, затраченное на подбор пароля.

4. Анализ эффективности алгоритмов сжатия

Создать набор тестовых файлов разных типов (текст .txt, документ .docx, растровое изображение .bmp и .jpg, исполняемый файл .exe). Архивировать каждый файл с помощью 3-5 различных архиваторов (или алгоритмов одного архиватора: ZIP/Deflate, 7z/LZMA2, 7z/PPMd) с максимальной и стандартной степенью сжатия. Сравнить итоговые размеры архивов, время упаковки и распаковки.

Отразить таблицу с результатами (исходный размер, размер после сжатия разными методами, коэффициент сжатия, время).

Графики или диаграммы, наглядно показывающие эффективность для разных типов данных.

Вывод о том, какой алгоритм оптимален для каждого типа файлов и почему (объяснить, исходя из природы данных: тексты хорошо сжимаются словарными методами, уже сжатые JPEG – плохо и т.д.).

Контрольные вопросы

1. Дайте определение архиватору. В чём заключается основное назначение архиваторов?
2. Объясните разницу между архивацией (упаковкой) и сжатием данных. Всегда ли эти процессы совпадают?
3. Назовите и кратко охарактеризуйте два основных класса алгоритмов сжатия без потерь (например, словарные и энтропийные). Приведите примеры каждого.
4. Что такое коэффициент сжатия? От чего зависит его величина для разных типов файлов (текст, изображение BMP, архив ZIP, исполняемый файл EXE)?
5. Почему файлы, уже сжатые другими алгоритмами (например, JPEG, MP3, MPEG), плохо поддаются дальнейшему сжатию архиваторами?

Лабораторная работа №20 «Настройка параметров безопасности»

Цель работы: ознакомление с встроенными в операционную систему Windows возможностями по оценке текущего состояния подсистемы безопасности и контролю целостности настроек безопасности.

Теоретические сведения

Групповая политика – это компонент серверных и клиентских операционных систем Windows, начиная с Windows 2000, позволяющий централизовано управлять конфигурацией пользователей и компьютеров. Групповые политики основываются на многих параметрах политик, которые в свою очередь указывают на применение определенной настройки для выбранного компьютера или пользователя. Все параметры политик располагаются в **объекте групповых политик GPO (Group Policy Object)**.

Объекты групповых политик делятся на две категории:

- **Доменные объекты групповых политик**, которые используются для централизованного управления конфигурацией компьютеров и пользователей, входящих в состав домена Active Directory. Эти объекты хранятся только на контроллере домена;
- **Локальные объекты групповых политик**, которые позволяют настраивать конфигурацию локального компьютера, а также всех пользователей, созданных на этом компьютере. Эти объекты хранятся только в локальной системе. Локальные объекты групповых политик могут применяться, даже если компьютер входит в состав домена.

В этой статье речь пойдет об управлении локальными объектами групповых политик.

Управление локальными объектами групповых политик

Для управления локальными объектами групповых политик в операционных системах Windows используется оснастка консоли управления **«Редактор локальной групповой политики»**. При помощи данной оснастки вы можете настраивать большинство системных компонентов и приложений. Рассмотрим подробно методы управления компьютером и пользователями при помощи данной оснастки:

Открытие оснастки «Редактор локальных групповых политик»

Вы можете открыть данную оснастку несколькими способами:

1. Откройте **«Консоль управления ММС»**. Для этого нажмите на кнопку **«Пуск»**, в поле поиска введите *mmc*, а затем нажмите на кнопку **«Enter».** Откроется пустая консоль ММС. В меню **«Консоль»** выберите команду **«Добавить или удалить оснастку»** или воспользуйтесь комбинацией клавиш **Ctrl+M**. В диалоге **«Добавление и удаление оснасток»** выберите оснастку **«Редактор объектов групповой политики»** и нажмите на кнопку **«Добавить»**. В появившемся диалоге **«Выбор объекта групповой политики»** нажмите на кнопку **«Обзор»** для выбора компьютера или нажмите на кнопку **«Готово»** (по умолчанию установлен объект **«Локальный компьютер»**). В диалоге **«Добавление или удаление оснасток»** нажмите на кнопку **«ОК»;**

Узлы оснастки

В оснастке редактора локальных объектов групповой политики присутствуют два основных узла:

- **Конфигурация компьютера**, который предназначен для настройки параметров компьютера. В этом узле расположены параметры, которые применяются к компьютеру, невзирая на то, под какой учетной записью пользователь вошел в систему. Эти параметры применяются при запуске операционной системы и обновляются в фоновом режиме каждые 90-120 минут.
- **Конфигурация пользователя**, который предназначен для настроек параметров пользователей. Параметры, которые находятся в этом узле, применяются при входе конкретного пользователя в систему. Так же, как и параметры, расположенные в узле конфигурации компьютера, параметры, расположенные в узле конфигурации пользователя обновляются в фоновом режиме каждые 90-120 минут.

Политика разрешения имен

Этот узел предоставляет возможность управлением расширением таблицы политик разрешения имен (NRTP), которая хранит параметры конфигурации для безопасности DNS (DNSSEC). Политика разрешения имен – это объект групповой политики, в котором указаны сведения о политике, которые отображаются в NRTP. Это расширение стоит настраивать только в том случае, если ваш компьютер входит в состав домена Active Directory. Эта политика расположена только в узле **«Конфигурация компьютера».**

При создании правила вам следует обратить внимание на следующие моменты:

Правила можно создавать для следующих частей пространства DNS:

- **Суффикс** – это зона пространства имен DNS, к которой применяется данное правило. Суффикс является частью полного доменного имени;
- **Префикс** – это первая часть полного доменного имени, к которому применяется правило;
- **Полное доменное имя** – состоит из узла и имени домена, включая домен верхнего уровня;
- **Подсеть (IPv4)** – это адрес подсети в формате IPv4 для зоны, к которой в случае обратного просмотра будет применено правило;
- Подсеть (IPv6) – это адрес подсети в формате IPv6 для зоны, к которой в случае обратного просмотра будет применено правило;
- **Любой** – применяется для всех найденных пространств имен DNS.

В поле **«Центр сертификации»** вы можете указать ЦС, который используется для создания цифровых подписей. Можно вводить ЦС вручную или выбрать нужный, воспользовавшись кнопкой **«Обзор».**

На вкладке **«DNSSEC»** вы можете включить DNSSEC для создаваемого правила и указать принудительную проверку конфигурации для безопасности DNS, а также выбрать тип шифрования, который будет использоваться для данного правила. Доступные значения: **«Без шифрования (только целостность)», «Тройной DES (3DES)», «Advanced Encryption Standard**

(AES)» с длиной ключа 128, 192 или 256 бит, или AES с длиной ключа 192 и 256 бит.

На вкладке **«Параметры DNS для прямого доступа»** вы можете указать серверы, которые клиент DNS будет использовать при соответствии указанного имени; прокси-сервер, который будет использоваться для подключения к Интернету; и можете указать тип шифрования для использования IPSec при взаимодействии между клиентом и сервером DNS.

Если вы нажмете на кнопку **«Дополнительные параметры глобальной политики»**, то сможете настроить параметры роуминга, параметры ошибки запроса и разрешения запроса.

После того как все параметры будут указаны, нажмите на кнопку **«Создать»**. После чего созданное правило отобразится в таблице политик разрешения имен. Изменения политики не будут сохранены, пока вы не нажмете на кнопку **«Применить».**

Политика безопасности

Этот узел позволяет настраивать безопасность средствами GPO. В этом узле для конфигурации компьютера доступны следующие настройки:

Брандмауэр Windows в режиме повышенной безопасности, при помощи которых вы можете создавать правила входящих и исходящих подключений, а также правила безопасности подключений так же, как и в одноименной оснастке. Разница лишь в том, что после создания правила, его настройки нельзя будет изменить, а также в оснастке **«Брандмауэр Windows в режиме повышенной безопасности»** у вас не будет прав для удаления текущего правила.

Настройка параметров туннелирования IPsec

Это диалоговое окно позволяет настроить правило безопасности подключения для использования туннельного, а не транспортного режима.

Чтобы открыть это диалоговое окно:

В оснастке консоли MMC **«Брандмауэр Windows в режиме повышенной безопасности»** в области навигации щелкните пункт Правила безопасности подключения.

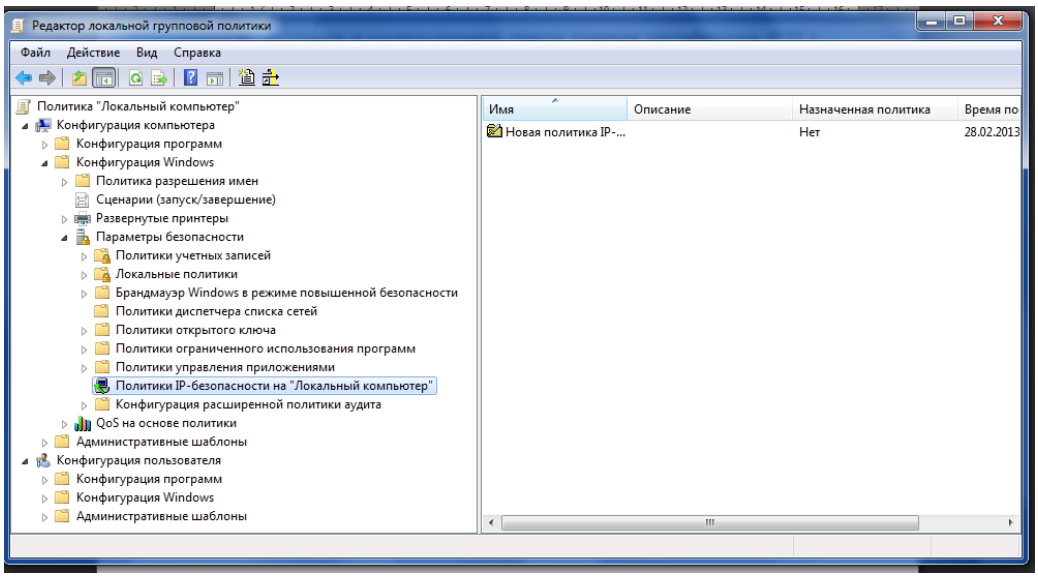
Дважды щелкните правило туннеля, которое требуется изменить.

Перейдите на вкладку Дополнительно и в разделе Туннелирование IPsec нажмите кнопку Настроить.

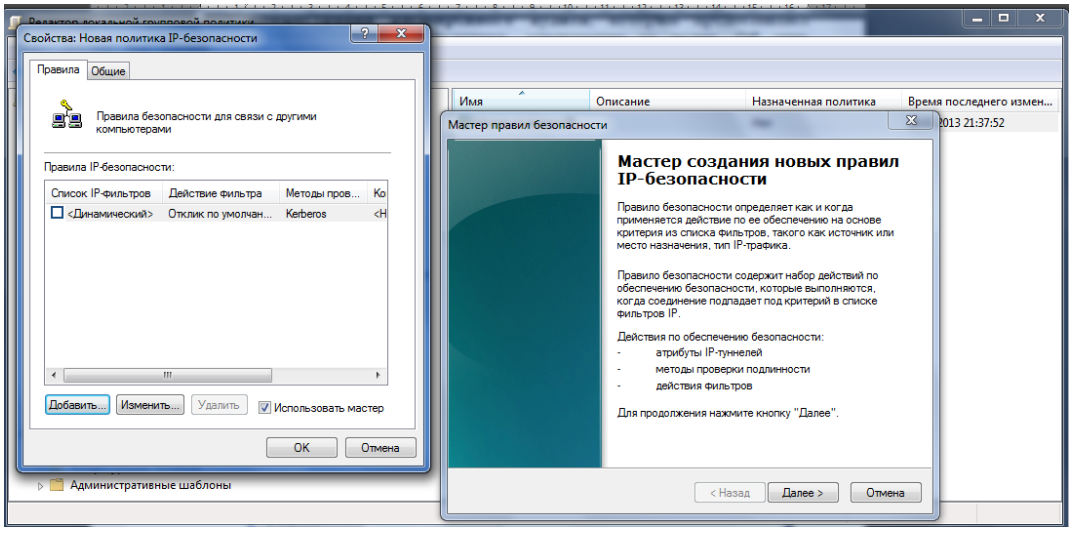
Использовать туннелирование IPsec

Выберите этот параметр, чтобы указать, что сетевой трафик, соответствующий этому правилу, должен проходить из конечной точки 1 до конечной точки 2 через туннель IPsec. При выборе этого параметра разрешаются остальные элементы управления этого диалогового окна.

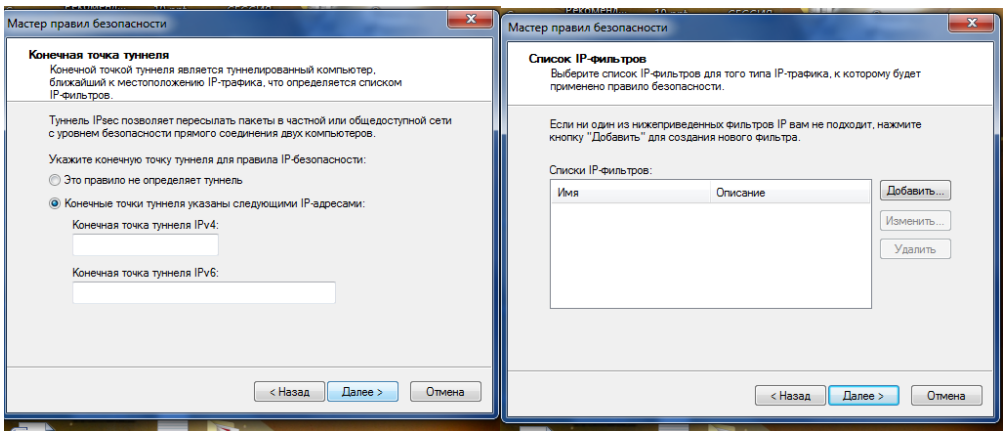
В оснастке **Политика IP-безопасности на «Локальный компьютер»** создайте новую политику IP-безопасности.



После создания новой политики создайте правило для IP-безопасности.



С помощью мастера правил безопасности последовательно задайте конечную точку туннеля, список IP-фильтров



Туннельный режим IPSec используется в первую очередь для обеспечения взаимодействия с другими маршрутизаторами, шлюзами или

конечными системами, которые не поддерживают туннелирование L2TP/IPSec или PPTP VPN. Режим туннеля IPsec поддерживается только при туннелировании от шлюза к шлюзу и для определенных конфигураций сервер-сервер или сервер-шлюз. Туннельный режим IPsec не поддерживается для сценариев удаленного доступа к VPN. Для удаленного доступа к виртуальной частной сети следует использовать подключения L2TP/IPSec или PPTP.

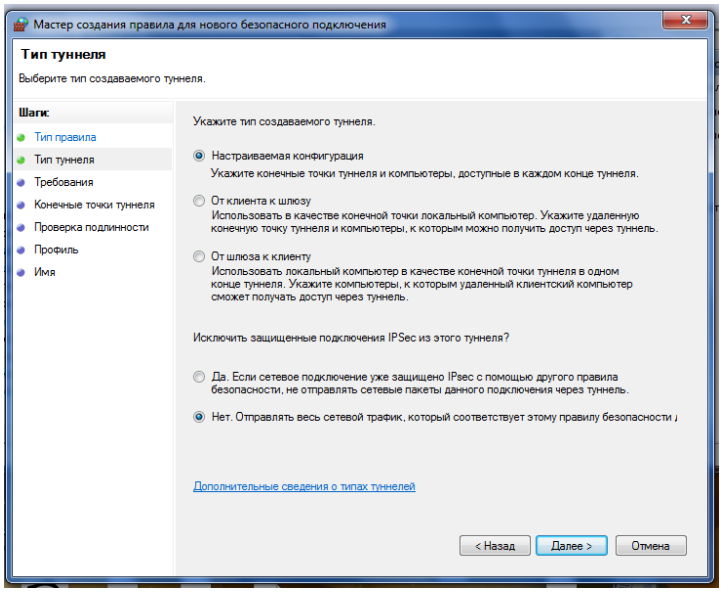
IPsec-туннель должен быть определен на обоих концах соединения. На каждом конце необходимо произвести обмен записями для локального и удаленного компьютеров в туннеле (поскольку локальный компьютер на одном конце туннеля является удаленным для другого конца и наоборот).

Используйте **Брандмауэр Windows в режиме повышенной безопасности,** чтобы выполнить туннелирование уровня 3 для сценариев, в которых нельзя использовать протокол L2TP. Если для удаленных подключений используется протокол L2TP, то настройка туннеля IPsec не требуется, поскольку клиентские и серверные компоненты VPN этой версии Windows автоматически создают соответствующие правила для защиты трафика L2TP.

Страница мастера позволяет настроить тип туннеля IPsec, который необходимо создать. Туннель IPsec обычно используется, чтобы подключить частную сеть за шлюзом либо к удаленному клиенту, либо к удаленному шлюзу к другой частной сети. Режим туннеля IPsec защищает данные, инкапсулируя весь пакет в защищенный IPsec-пакет, и затем пересылает защищенный IPsec-пакет между конечными точками туннеля. По достижении конечной точки пакет данных извлекается и пересылается в место назначения.

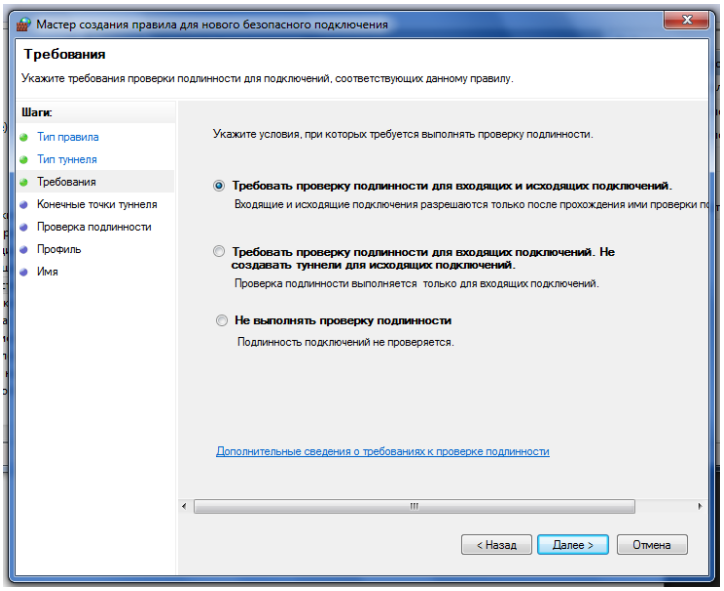
В оснастке консоли MMC «**Брандмауэр Windows в режиме повышенной безопасности**» щелкните правой кнопкой мыши пункт Правила безопасности подключения и выберите команду **Новое правило.**

На странице Тип правила установите переключатель Туннель.



В Шаги выберите Тип туннеля.

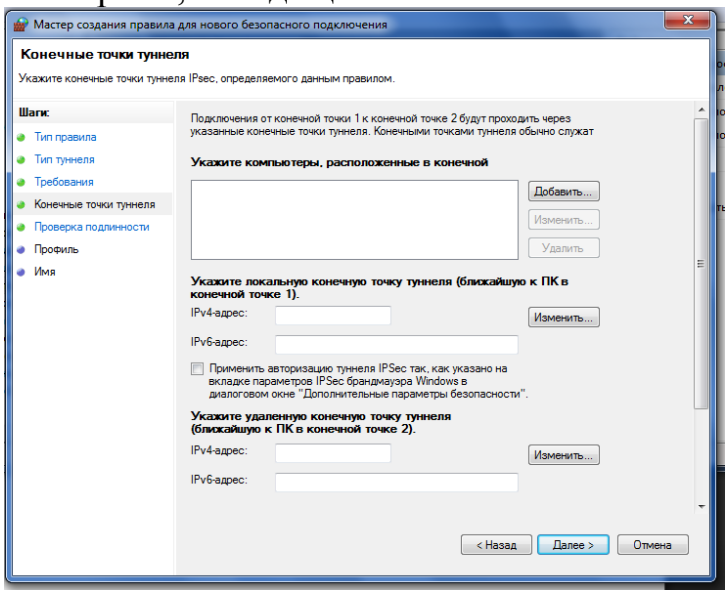
Настраиваемая конфигурация



При выборе этого параметра разрешаются все параметры для настройки конечных точек на странице Конечные точки туннеля - Настраиваемая конфигурация. Можно указать IP-адреса компьютеров, которые служат в качестве конечных точек туннеля, и компьютеров, которые находятся в частных сетях за пределами каждой из конечных точек туннеля. Дополнительные сведения см. в разделе Мастер создания правила безопасности подключения: страница «Конечные точки туннеля» - настраиваемая конфигурация.

От клиента к шлюзу

Выберите этот параметр, если необходимо создать правило для клиентского компьютера, который должен соединяться с удаленным шлюзом и компьютерами, находящимися за шлюзом в частной сети.



Когда клиент посылает сетевой пакет компьютеру в удаленной частной сети, протокол IPsec внедряет пакет данных в IPsec-пакет и адресует его

удаленному шлюзу. Шлюз извлекает пакет и адресует его в частную сеть целевому компьютеру.

При выборе этого параметра можно настраивать только публичный IP-адрес компьютера-шлюза и IP-адреса компьютеров в частной сети. Дополнительные сведения см. в разделе Мастер создания правила безопасности подключения: страница «Конечные точки туннеля» - клиент-шлюз.

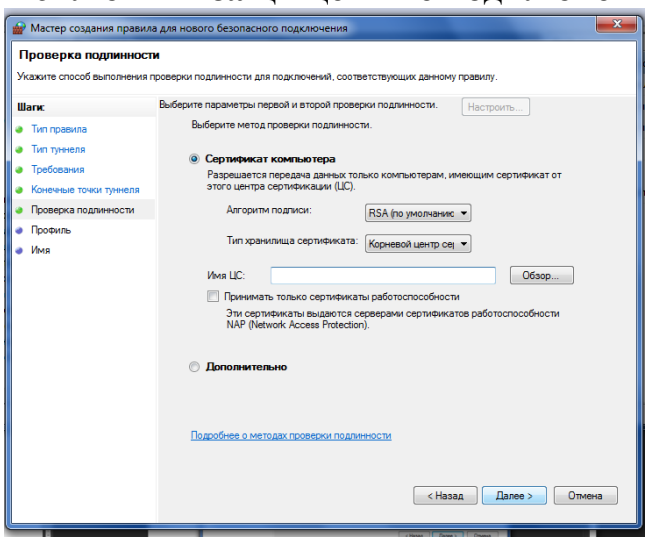
Выберите параметр **От шлюза к клиенту,** если необходимо создать правило для компьютера-шлюза, который подключен и к частной сети, и к публичной сети, из которой он принимает трафик от удаленных клиентов.

Когда клиент посылает сетевой пакет компьютеру в частной сети, протокол IPsec внедряет пакет данных в IPsec-пакет и адресует его на общий IP-адрес этого компьютера-шлюза. Когда компьютер-шлюз получает пакет, он извлекает исходный пакет и адресует его в частную сеть целевому компьютеру.

Когда компьютеру в удаленной частной сети требуется ответить клиентскому компьютеру, пакет данных адресуется компьютеру-шлюзу. Компьютер-шлюз внедряет пакет данных в IPsec-пакет, который адресуется удаленному клиентскому компьютеру, и посылает этот IPsec-пакет по публичной сети удаленному клиентскому компьютеру.

При выборе этого параметра можно настраивать только адреса компьютеров в частной сети и общий IP-адрес компьютера-шлюза. Дополнительные сведения см. в разделе Мастер создания правила безопасности подключения: страница «Конечные точки туннеля» - шлюз-клиент.

Исключить защищенные подключения IPSec



Иногда сетевой пакет может удовлетворять нескольким правилам безопасности подключения. Если одно из правил устанавливает туннель IPsec, можно выбрать, использовать ли этот туннель или передать пакет, защищенный другим правилом, не по туннелю.

Выберите вариант Да, если подключение уже защищено другим правилом безопасности подключения и не требуется, чтобы сетевые пакеты проходили через туннель IPsec. Запрещается передача по туннелю всего сетевого трафика, защищенного протоколом ESP, включая ESP NULL.

Выберите вариант Нет, если необходимо, чтобы все сетевые пакеты, удовлетворяющие правилу туннеля, проходили через туннель, даже если они защищены другим правилом подключения безопасности.

Политики диспетчера списка сетей, позволяют управлять всеми вашими сетевыми профилями.

Политики открытого ключа позволяют:

- настраивать компьютеры на автоматическую отправку запросов в центр сертификации предприятия и установку выдаваемых сертификатов;
- создавать и распространять список доверия сертификатов (CTL);
- добавлять агенты восстановления шифрованных данных и изменение параметров политики восстановления шифрованных данных;
- добавлять агенты восстановления данных шифрования диска BitLocker.

Политики ограниченного использования программ позволяют осуществлять идентификацию программ и управлять возможностью их выполнения на локальном компьютере, в подразделении, домене и узле.

Политики управления приложениями отвечают за создание и управления правилами и свойствами функционала AppLocker, который позволяет управлять установкой приложений и сценариев.

Политики IP-безопасности на «Локальный компьютер» позволяют создавать политику IP-безопасности локального компьютера и управлять списками IP-фильтров.

Конфигурация расширенного аудита предоставляет дополняющие локальные политики, отвечающие за аудит

Применение IPSec

В домене Windows служба IPSec обычно активируется с помощью групповой политики. Тем не менее, существует возможность применения IPSec и на отдельных системах.

1. Кликните правой кнопкой на значке Мое сетевое окружение и выберите в контекстном меню команду Свойства.

2. Кликните правой кнопкой мыши на значке Подключение по локальной сети и выберите в контекстном меню команду Свойства

3. Выберите раздел Протокол Internet (TCP/IP) и кликните на кнопке Свойства.

4. Кликните на кнопке Дополнительно.

5. Перейдите на вкладку Параметры.

6. Выберите раздел IP-безопасность и кликните на кнопке Свойства.

7. Выберите параметр Использовать следующую политику IP-безопасности и применяемую политику безопасности:

· Клиент (Только ответ) — IPSec будет использоваться только при запросе со стороны другой системы;

· Безопасность сервера (Требовать безопасность) — вся передача данных с использованием протокола IP должна осуществляться на основе системы безопасности Kerberos;

· Сервер (Запрос безопасности) — использование IPSec по мере возможности.

8. Кликните на кнопке OK во всех диалоговых окнах.

Для установки компонентов IPSec на компьютере необходимо входить в локальную группу Администраторы. Политику IPSec, настроенную для домена, нельзя переопределить с помощью локальной политики. При этом все описанные выше параметры будут недоступны.

Порядок выполнения работы

1 Ознакомьтесь с теоретическим материалом.

2 В соответствии с указанным порядком создайте новую политику безопасности, правило для безопасного подключения в брандмауэре Windows, правила для входящих и исходящих подключений.

3 Зафиксируйте полученные результаты в отчете.

Контрольные вопросы

1. Что такое локальная и доменная группа объектов политики (GPO)? Чем они отличаются?
2. Какие основные узлы присутствуют в оснастке «Редактор локальной групповой политики» и что настраивается в каждом из них?
3. Что такое политика разрешения имен и когда её следует настраивать?
4. Какие параметры можно указать в правилах DNSSEC и для чего они нужны?
5. Как настраиваются правила брандмауэра Windows для входящих и исходящих подключений?
6. В чем заключается настройка туннельного режима IPSec и какие сценарии его использования существуют?
7. Чем отличается режим туннеля IPSec от режима L2TP/IPSec и PPTP VPN?
8. Какие действия нужно выполнить для создания политики IP-безопасности на локальном компьютере?
9. Как определить, когда использовать правила «От клиента к шлюзу» и «От шлюза к клиенту»?
10. Что происходит с пакетами данных при прохождении через IPSec-туннель?

Лабораторная работа №21 «Антивирусная защита»

Цель работы: изучить технологию тестирования компьютера на наличие вируса и профилактические меры. Познакомиться со способами лечения зараженных объектов.

Компьютерный вирус – это специально написанная, небольшая по размерам программа (т.е. некоторая совокупность выполняемого кода), которая может “ приписывать” себя к другим программам (“заражать” их), создавать свои копии и внедрять их в файлы, системные области компьютера и т.д., а также выполнять различные нежелательные действия на компьютере.

Программа, внутри которой находится вирус, называется “зараженной” Когда такая программа начинает работу, то сначала управление получает вирус. Вирус находит и заражает другие программы, а также выполняет какие-нибудь вредные действия (портит файлы или таблицу размещения файлов на диске, “засоряет” оперативную память и т.д.).

Классификация вирусов.

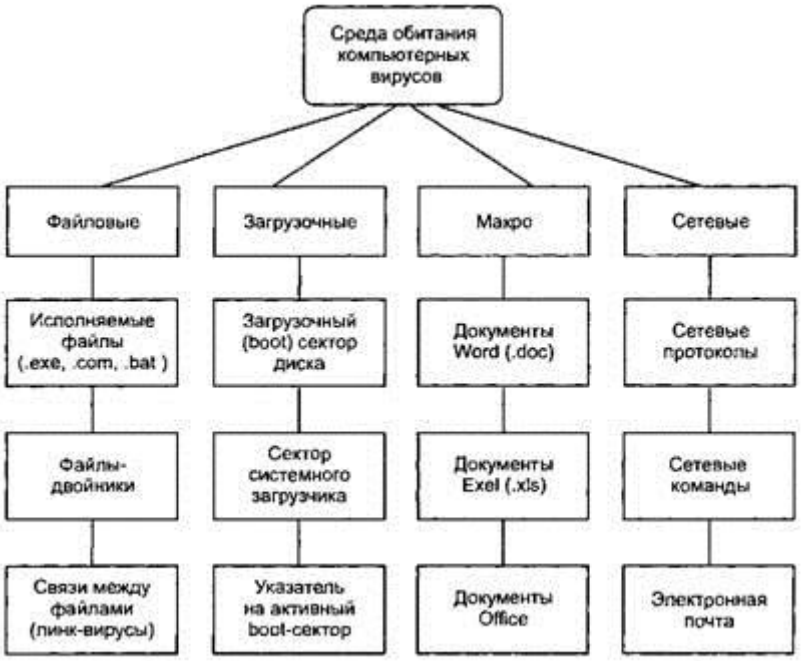


Рис. 15.1. Классификация компьютерных вирусов по среде обитания

Основными путями проникновения вирусов в компьютер являются **съёмные диски** (гибкие и лазерные), а также **компьютерные сети**. Заражение жесткого диска вирусами может произойти при загрузке программы с дискеты, содержащей вирус. Такое заражение может быть и случайным, например, если дискету не вынули из дисковода A: и перезагрузили компьютер, при этом дискета может быть и не системной. Заражение дискеты происходит, даже если её просто вставили в дисковод зараженного компьютера или, например, прочитали её оглавление.

Признаки заражения

- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- неожиданное открытие и закрытие лотка CD-ROM-устройства;

- произвольный, без вашего участия, запуск на компьютере каких-либо программ;

Есть также **косвенные признаки заражения** вашего **компьютера**:

- частые зависания и сбои в работе компьютера;
- прекращение работы или неправильная работа ранее успешно работавших программ;
- медленная работа компьютера при запуске программ;
- невозможность загрузки операционной системы;
- исчезновение файлов и каталогов или искажение их содержимого;
- изменение размеров файлов;
- неожиданное значительное увеличение количества файлов на диске;
- существенное уменьшение размеров свободной оперативной памяти;
- частое обращение к жесткому диску (часто мигает лампочка на системном блоке);
- Microsoft Internet Explorer "зависает" или ведет себя неожиданным образом.

В 90% случаев наличие косвенных симптомов вызвано сбоем в аппаратном или программном обеспечении. Несмотря на то, что подобные симптомы с малой вероятностью свидетельствуют о заражении, при их появлении рекомендуем вам провести **полную проверку вашего компьютера.**

Антивирусные программы.

Для обнаружения, удаления и защиты от компьютерных вирусов разработаны специальные антивирусные программы. Различают следующие **виды антивирусных программ:**

Программы-детекторы осуществляют поиск характерной для конкретного вируса сигнатуры в оперативной памяти и в файлах и при обнаружении выдают соответствующее сообщение. Недостатки: могут находить только те вирусы, которые известны разработчикам этой программы, поэтому быстро устаревают и требуют регулярного обновления.

Программы-доктора или **фаги** не только находят зараженные вирусами файлы, но и «лечат» их, т.е. удаляют из файла тело программы-вируса, возвращая файл в исходное состояние. **Полифаги** – программы-доктора, предназначенные для поиска и уничтожения большого количества вирусов. Недостатки те же, что и у программ-детекторов.

Программы-ревизоры относятся к самым надежным средствам защиты. Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным. Обнаруженные изменения выводятся на экран монитора.

Программы-фильтры или **«сторожа»** представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов (попытки коррекции файлов с расширением EXE или COM, изменение атрибутов файла,

запись в загрузочные сектора и т.п.). При попытке какой-либо программы произвести указанные действия «сторож» посылает пользователю сообщение и предлагает запретить или разрешить соответствующее действие. Эти программы способны обнаружить вирус на самой ранней стадии его существования до размножения. Однако они не лечат файла и диски. Для уничтожения вируса требуется применить другие программы.

Вакцины или **иммунизаторы** это резидентные программы, предотвращающие заражение файлов. Вакцины применяют, если отсутствуют программы-доктора, лечащие этот вирус. Вакцинация возможна только от известных вирусов. Вакцина модифицирует программу или диск таким образом, чтобы это не отражалось на их работе, а вирус будет воспринимать их зараженными и поэтому не внедрится. Имеют ограниченное применение.

Для изучения антивирусных средств необходимо провести установку одного из антивирусных пакетов:

1. Dr.Web

Преимущества:

- возможность установки и работы на уже инфицированном компьютере и вирусоустойчивость;

- возможность нейтрализации активного заражения с внешнего носителя без установки в систему и уже в ходе установки лечение активных угроз;

- возможность противостоять активным угрозам даже если компьютер заражен сложными вредоносными программами.

-реализована возможность обнаруживать и нейтрализовать вирусы, существующие в оперативной памяти и никогда не встречающиеся в виде отдельных файлов;

- способность выявлять с высочайшей степенью точности упакованные вредоносные объекты и детально анализировать с целью обнаружения скрытых угроз.

Независимо от того, что вы и ваши близкие делаете в сети, ваши личные данные, финансовая информация и ценные файлы находятся в полной безопасности.

2. Kaspersky CRYSTAL

Преимущества:

- гибридная защита от интернет-угроз мгновенно устраняет вредоносные программы, спам и другие угрозы, экономя ресурсы компьютера за счет комбинации облачных и антивирусных технологий;

- резервное копирование данных. Для хранения резервных копий можно использовать как жесткий диск, так и съемный носитель;

- менеджер паролей будет генерировать для вас надежные, устойчивые к взлому пароли и автоматически заполнять регистрационные формы на вебсайтах и в приложениях;

- родительский контроль может ограничивать время, которое дети проводят за компьютером и в интернете, блокировать их доступ к «взрослым» сайтам и контролировать переписку по электронной почте, в социальных сетях и через сервисы мгновенного обмена сообщениями.

- шифрование данных позволяет хранить ценные данные в защищен-ных паролем файлах-контейнерах. Без знания пароля получить доступ к содержимому зашифрованного файла невозможно.

- необратимое удаление данных не позволит восстановить стертые файлы и получить доступ к конфиденциальной информации;

- централизованное управление позволяет управлять защитой домашней сети с одного компьютера: выполнять проверки и устанавливать обновления, делать резервные копии данных, настраивать правила Родительского контроля и др.

Порядок выполнения работы

1 Ознакомьтесь с теоретическим материалом.

2 Установите виртуальную машину.

3 Установите антивирусный пакет.

4 Проведите настройку антивирусного пакета и проверку на наличие вирусов.

5 Зафиксируйте полученные результаты в отчете.

Варианты заданий

Вариант 1

1. Dr.Web

Вариант 2

2. Kaspersky CRYSTAL

Контрольные вопросы

1. Что такое компьютерный вирус и каким образом он распространяется?
2. Чем отличается заражённая программа от обычной?
3. Какие основные пути проникновения вирусов в компьютер вы знаете?
4. Назовите прямые и косвенные признаки заражения компьютера вирусами.
5. Чем опасно наличие вирусов для файловой системы и работы компьютера?
6. Какие виды антивирусных программ существуют и каковы их функции?
7. В чем заключается работа программы-детектора и какие у нее недостатки?
8. Что такое программа-доктор (или полифаг) и чем она отличается от детектора?
9. Какие задачи выполняют программы-ревизоры и чем они полезны для профилактики заражений?
10. Что такое программы-фильтры («сторож») и на какой стадии заражения они действуют?

11. Что такое вакцины или иммунизаторы и какие ограничения существуют при их использовании?
12. Какие преимущества у антивирусного пакета Dr.Web и как он защищает компьютер?
13. Какие возможности защиты предоставляет Kaspersky CRYSTAL?
14. Почему важно регулярно обновлять антивирусные базы данных и программы?
15. Какие профилактические меры помогут снизить риск заражения компьютера вирусами?
16. Каковы основные шаги при тестировании компьютера на наличие вирусов в рамках данной лабораторной работы?
17. Чем отличается проверка на вирусы в виртуальной машине от проверки на основном компьютере?
18. Почему для антивирусной защиты важно проводить не только обнаружение, но и лечение зараженных объектов?
19. Как правильно фиксировать результаты проверки в отчете по лабораторной работе?
20. Какие современные методы защиты и мониторинга (например, родительский контроль, шифрование данных, централизованное управление) реализованы в антивирусных пакетах?

2 ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Самостоятельная работа - целенаправленная, планируемая в рамках учебного плана деятельность студентов, которая осуществляется по заданию, при методическом руководстве и контроле преподавателя, но без его непосредственного участия. Самостоятельная работа студентов является одной из важнейших составляющих образовательного процесса.

В учебном процессе учебного заведения выделяют два вида самостоятельной работы: аудиторная и внеаудиторная.

Аудиторная самостоятельная работа выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная — планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Целью самостоятельной работы студентов является:

- систематизация и закрепление полученных теоретических знаний и практических умений;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов, творческой инициативы, самостоятельности, ответственности, организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации;
- формирование общих и профессиональных компетенций.

Самостоятельная работа студентов должна быть хорошо спланирована и организована. При планировании такой работы необходимо учитывать условия, обеспечивающие её успешное выполнение:

- чёткое определение преподавателем объёма и содержания самостоятельной работы;
- определение видов консультативной помощи;
- постановка цели самостоятельной работы и критерии её оценки;
- виды и формы контроля её выполнения.

Выполняя самостоятельную работу под контролем преподавателя, студент должен:

- освоить минимум знаний;
- планировать свою самостоятельную работу в соответствии разработанным графиком;
- выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

Таким образом, самостоятельная работа студентов оказывает важное влияние на формирование личности будущего специалиста.

Самостоятельная работа студентов является обязательной для каждого студента, объем ее определяется учебным планом в соответствии с требованиями Государственных образовательных стандартов.

При изучении тем дисциплины студенты выполняют следующие виды самостоятельной работы:

- проработка конспектов занятий, учебных изданий и специальной технической литературы;
- составление конспекта, тематических схем, таблиц;
- подготовка к лабораторным работам и практическим занятиям с использованием методических рекомендаций преподавателя;
- оформление отчетов по лабораторным работам и практическим занятиям, подготовка к их защите;
- моделирование и решение производственных процессов и ситуационных задач;
- подготовка презентаций;
- работа с электронными ресурсами в сети Интернет;
- подготовка к семинару;
- подготовка к зачетам, экзаменам.

Технология организации самостоятельной работы студентов включает использование информационных и материально-технических ресурсов образовательного учреждения. Материально-техническое и информационно - техническое обеспечение самостоятельной работы студентов включает в себя:

- библиотеку с читальным залом, укомплектованную в соответствии с существующими нормами;
- учебно-методическую базу учебных кабинетов, лабораторий и методического центра;
- компьютерные классы с возможностью работы в Интернет;
- базы практики в соответствии с заключенными договорами;
- аудитории для консультационной деятельности;
- учебную и учебно-методическую литературу, разработанную с учетом увеличения доли самостоятельной работы студентов, и иные методические материалы.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж по выполнению задания, в котором указывает цель задания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения студентами внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить консультации. Самостоятельная работа может осуществляться индивидуально или группами

студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений обучающихся.

3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Общие методические рекомендации студенту при изучении тем дисциплины.

Большая часть самостоятельной работы выполняется студентом вне учебных занятий при подготовке домашних заданий. Общие требования к выполнению этого вида самостоятельной работы заключаются в следующем:

- активно работать на уроке, усваиваю основную часть нового материала;
- если что-то непонятно, не стесняться задавать вопросы преподавателю;
- большое задание необходимо разбивать на части и работать над каждой из них в отдельности;
- выполняя домашнее задание, надо не просто думать, что надо сделать, а еще и решать, с помощью каких средств и приемов этого можно добиться;
- в процессе приготовления домашнего задания необходимо делать перерывы;
- готовиться к докладам, рефератам, защите курсовых работ и проектов, практических и лабораторных занятий надо заранее, равномерно распределяя нагрузку, а не оставлять такую ответственную работу на последний день;
- изучая заданный материал, сначала надо его понять, а уже потом запомнить;
- научиться находить интересующую нужную информацию с помощью компьютера;
- не стесняться обращаться за помощью к взрослым и однокурсникам;
- надо составлять план устного ответа и проверять себя;
- на письменном столе должно лежать только то, что необходимо для выполнения одного задания. После его завершения со стола убираются уже использованные материалы, и кладутся те учебные принадлежности, которые необходимы для выполнения следующего задания;
- нужно решить, в какой последовательности лучше выполнять задания и сколько времени понадобится на каждое из них;
- трудный материал урока лучше повторить в тот же день, чтобы сразу закрепить его и запомнить;
- читая учебник, надо задавать самому себе вопросы по тексту.

Подготовка тематических сообщений, докладов, рефератов

Реферат доклад, сообщение (от латинского refero - передаю, сообщаю) - краткое письменное изложение материала по определенной теме с целью привития студентам навыков самостоятельного поиска и анализа информации, формирования умения подбора и изучения литературных источников,

используя при этом дополнительную научную, методическую и периодическую литературу.

Тема реферата выбирается по желанию студента из списка, предлагаемого преподавателем. Тема может быть сформулирована студентом самостоятельно.

Выбранная тема согласовывается с преподавателем.

После выбора темы требуется:

- составить план реферата;
- подобрать необходимую информацию;
- изучить подобранную информацию;
- составить текст реферата.

План реферата должен включать в себя введение, основной текст и заключение. Во введении аргументируется актуальность выбранной темы, указываются цели и задачи исследования. В нем также отражается методика исследования и структура работы. Основная часть работы предполагает освещение материала в соответствии с планом. В заключении излагаются основные выводы и рекомендации по теме исследования.

Реферат оформляется согласно требованиям, установленным в учебном заведении. Он должен содержать: титульный лист, оглавление и список использованной литературы. На титульном листе указываются: название учебного заведения, название профессионального модуля, междисциплинарного курса, тема работы, курс, группа, фамилии, имена, отчества студента и руководителя работы, название города, в котором находится учебное заведение, год написания данной работы. Реферат может содержать приложения в форме схем, образцов документов и другие изображения в соответствии с темой исследования. Все страницы работы, включая оглавление и список литературы, нумеруются по порядку с титульного листа (на нем цифра не ставится) до последней страницы без пропусков и повторений. Введение, заключение, новые главы, список использованных источников и литературы должны начинаться с нового листа. Подбор литературы производится студентом из предложенного преподавателем списка литературы. Текст реферата необходимо набирать на компьютере на одной стороне листа. Размер левого поля 30 мм, правого - 15 мм, верхнего - 20 мм, нижнего - 20 мм. Шрифт - Times New Roman, размер - 14, межстрочный интервал - 1,5. Фразы, начинающиеся с новой строки, печатаются с абзацным отступом от начала строки (1,25 см). Реферат, выполненный небрежно, неразборчиво, без соблюдения требований по оформлению, возвращается студенту без проверки с указанием причин возврата на титульном листе.

Критерии оценки:

- знание и понимание проблемы;
- умение систематизировать и анализировать материал, четко и обоснованно формулировать выводы;
- «трудозатратность» (объем изученной литературы, добросовестное отношение к анализу проблемы);

- самостоятельность, способность к определению собственной позиции по проблеме и к практической адаптации материала, недопустимость плагиата;
- выполнение необходимых формальностей (точность в цитировании и указании источника текстового фрагмента, аккуратность оформления).

Проработка занятый, учебных изданий и специальной технической литературы

Работа с конспектом лекций по темам междисциплинарных курсов заключается в том, что студент после рассмотрения темы на учебных занятиях в период между очередными лекциями изучает материал конспекта. При этом непонятные положения конспекта необходимо выяснять у преподавателя на консультациях или при чтении основной и дополнительной литературы.

При работе с книгой необходимо научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги. Правильный подбор учебников рекомендуется преподавателем. Необходимая литература может быть также указана в методических разработках. Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и определения (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода). Полезно составлять опорные конспекты. При изучении материала по учебнику, полезно в тетради (на специально отведенных полях) дополнять конспект лекций, написанный на учебных занятиях. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем. Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при пропитывании записей лучше запоминались. Различают два вида чтения; первичное и вторичное. Первичное - эти внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятного слова. Содержание не всегда может быть понятно после первичного чтения. Задача вторичного чтения - полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Чтение научного текста является частью познавательной деятельности. Ее цель - извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия. Выделяют четыре основные установки в чтении научного текста:

- информационно-поисковая, задача которой - найти, выделить искомую информацию;

- усваивающая, при которой усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения, излагаемые автором, так и всю логику его рассуждений;
- аналитико-критическая - читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему;
- творческая, создающая у читателя готовность в том или ином виде использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке.

Самостоятельная работа при чтении учебной литературы начинается с изучения конспекта материала, полученного при слушании лекций преподавателя. Полученную информацию необходимо осмыслить. При необходимости, в конспект лекций могут быть внесены схемы, эскизы рисунков, другая дополнительная информация.

Составление конспекта, тематических схем, таблиц

При изучении нового материала, как правило, составляется конспект. Конспект - изложение текста, которому присущи краткость, связность и последовательность. При этом максимально точно записываются формулы, определения, схемы, трудные для запоминания места.

При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре текста. Для уточнения и дополнения необходимо оставлять поля. Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

Классификация конспектов:

- плановый конспект, для чего сначала нужно написать план текста, а затем на пункты плана делаются комментарии: свободно изложенный текст либо цитаты;
- обзорный конспект - краткое изложение данной темы с использованием нескольких источников;
- текстуальный конспект состоит из цитат одного текста;
- свободный конспект предполагает цитаты текста и собственные формулировки прочитанного текста;
- сложный - конспект, в котором отражается определенная тема или вопрос;
- хронологический конспект отражает последовательность событий;
- опорный конспект, в котором излагается информация в виде опорных знаков, слов, сигналов.

Методические рекомендации по составлению конспекта:

- определить цель написания конспекта;

- внимательно прочитать текст, уточнить в справочной литературе непонятные слова;
- выделить основные смысловые части текста;
- определить главное, составить план;
- кратко сформулировать основные положения текста, отметить аргументацию автора;
- составить текст конспекта, изложив информацию кратко и своими словами, четко следуя пунктам плана, записи следует вести четко, ясно;
- грамотно записывать цитаты, учитывая лаконичность, значимость мысли;
- в тексте конспекта желательно приводить не только тезисные положения, но и их доказательства.

При составлении тематических схем, таблиц необходимо внимательно прочитать текст соответствующий параграф учебника. Продумать «конструкцию» таблицы или схемы, расположение порядковых номеров, терминов, примеров и пояснений (и прочего). Начертить схему или таблицу и заполнить ее графы необходимым содержимым.

Подготовка к лабораторным работам и практическим занятиям, оформление отчетов по лабораторным работам и практическим занятиям, подготовка к их защите

Программы профессиональных модулей предусматривают выполнение практических и лабораторных занятий.

Лабораторное занятие - форма учебного занятия, ведущей дидактической целью которого является экспериментальное подтверждение и проверка существующих теоретических положений (законов, зависимостей), формирование учебных и профессиональных практических умений и навыков.

Практическое занятие - это одна из форм учебной работы, которая ориентирована на закрепление изученного теоретического материала, его более глубокое усвоение и формирование умения применять теоретические знания в практических целях. Особое внимание на практических занятиях уделяется выработке учебных или профессиональных навыков. Такие навыки формируются в процессе выполнения конкретных заданий - упражнений, задач - под руководством и контролем преподавателя.

Подготовка к практическим и лабораторным занятиям заключается в работе с конспектом лекций по данной теме, в изучении соответствующего раздела учебника или учебного пособия, в просмотре дополнительной литературы. Этапы подготовки к практическому или лабораторному занятию заключаются в следующем: освежить в памяти теоретические сведения, полученные на лекциях и в процессе самостоятельной работы, подобрать необходимую учебную и справочную литературу. Отобрать те материалы, которые позволят в полной мере реализовать цели и задачи предстоящей работы. Еще раз проверить соответствие отобранного материала. Студент должен прийти на лабораторное или практическое занятие подготовленным по данной теме.

При выполнении заданий практического или лабораторного занятия студент должен быть ознакомлен преподавателем с целью и ходом выполнения задания и, по необходимости, с правилами техники безопасности. Если у студентов во время выполнения заданий возникают вопросы, то преподаватель консультирует студентов. Порядок выполнения того или иного задания излагается в инструкционных картах или рабочих тетрадях.

После проведения занятия студент представляет письменный отчет, который оформляется в соответствие с принятыми в образовательном учреждении правилами. Отчеты оформляются на листах писчей бумаги формата А4 или в специальных рабочих тетрадях, разработанных преподавателем. Содержание отчета указано в инструкционных картах или рабочих тетрадях.

При подготовке к защите практических и лабораторных занятий студент должен ответить на контрольные вопросы, указанные также в инструкционных картах или рабочих тетрадях, проштудировав при этом конспект лекций, учебную литературу.

Моделирование и решение производственных процессов и ситуационных задач

При изучении дисциплины очень часто студенту приходится сталкиваться с профессиональными задачами и ситуациями, которые необходимо решить самостоятельно, как во время аудиторной работы, так и во время внеаудиторной. При решении таких задач необходимо:

- провести анализ ситуации для определения проблемы в целом; представить ситуацию и себя в качестве действующего в ней лица; проанализировать ошибочные или правильные действия всех участников ситуации;
- определить проблемные узлы - возможные причины и прогнозируемые последствия развития данной ситуации;
- рассмотреть условное прогнозирование развития ситуации: определить окончательную гипотезу, представить обоснованный и доказательный прогноз вероятностного развития ситуации; предложить варианты действий, обоснованные теоретически и, по возможности, подкрепленные практическим личным опытом, опираясь на принципы профессиональной этики; определить способы и методы воздействия на предлагаемую ситуацию;
- сформулировать итоговые выводы, используя профессиональные термины, доказательства правильности своего решения.

Подготовка презентаций

Подготовка презентации позволит студенту логически выстроить изучаемый материал, систематизировать его, сформировать коммуникативные компетенции. Материал презентации представляется в виде текста, схем, диаграмм, таблиц, которые призваны дополнить текстовую информацию или передать ее в более наглядном виде. Желательно избегать в презентации

изображений, не несущих смысловой нагрузки, если они не являются частью стилевого оформления. Цвет графических изображений не должен резко контрастировать с общим стилевым оформлением слайдов, иллюстрации рекомендуется сопровождать пояснительным текстом.

Анимационные эффекты используются для привлечения внимания слушателей или для демонстрации динамики развития какого - либо процесса. В этих случаях использование анимации оправдано, но не стоит чрезмерно насыщать презентацию такими эффектами, иначе это вызовет негативную реакцию аудитории.

Звуковое сопровождение должно отражать суть или подчеркивать особенность темы слайда, презентации. Фоновая музыка не должна отвлекать внимание слушателей и заглушать слова докладчика.

Оптимальное количество слайдов, как правило, десять - пятнадцать. Для оформления слайдов презентации рекомендуется использовать несложные шаблоны, соблюдать единый стиль. Не рекомендуется на одном слайде использовать более трех цветов. Смену слайдов для управления презентацией докладчиком желательно устанавливать по щелчку без времени. Шрифт, выбираемый для презентации, должен обеспечивать читаемость информации на экране и соответствовать выбранному шаблону оформления. Не желательно использовать разные шрифты в одной презентации.

Алгоритм выстраивания презентации должен соответствовать логической структуре работы и отражать последовательность ее этапов. Независимо от алгоритма выстраивания презентации на первом слайде рекомендуется выносить следующие данные: полное наименование образовательной организации; тема презентации; фамилия, имя, отчество студента; специальность обучения; фамилия, имя, отчество руководителя. Последний слайд должен содержать фразу «Спасибо за внимание».

Работа с электронными ресурсами в сети Интернет

Для повышения эффективности самостоятельной работы студент должен учиться работать в поисковой системе сети Интернет, в электронно-библиотечной системе и использовать найденную информацию при подготовке к занятиям.

Интернет сегодня - правомерный источник научных статей, статистической и аналитической информации, и использование его наряду с книгами давно уже стало нормой. Однако, несмотря на то, что ресурсы Интернета позволяют достаточно быстро и эффективно осуществлять поиск необходимой информации, следует помнить о том, что эта информация может быть неточной или вовсе не соответствовать действительности. В связи с этим при поиске материала по заданной тематике следует обращать внимание на научные труды признанных авторов, которые посоветовали вам преподаватели.

Поиск информации можно вести по автору, заглавию, виду издания, году издания или издательству. Также в сети Интернет доступна услуга по

скачиванию методических указаний и учебных пособий, подбору необходимой учебной и научно - технической литературы.

Подготовка к семинару

Семинар — это особая форма учебно-теоретических занятий, которая, как правило, служит дополнением к лекционному курсу. Семинар обычно посвящен детальному изучению отдельной темы.

Этапы подготовки к семинару:

- проанализировать тему семинара, подумать о цели и основных проблемах, вынесенных на обсуждение;
- внимательно прочитать материал, данный преподавателем по этой теме на лекции;
- изучить рекомендованную литературу, делая при этом конспекты прочитанного или выписки, которые понадобятся при обсуждении на семинаре;
- постараться сформулировать свое мнение по каждому вопросу и аргументированно его обосновать;
- записать возникшие во время самостоятельной работы с учебниками и научной литературой вопросы, чтобы затем на семинаре получить на них ответы.

При подготовке к семинарским занятиям следует руководствоваться указаниями и рекомендациями преподавателя, использовать основную и дополнительную литературу из представленного им списка.

При подготовке доклада на семинарское занятие желательно заранее обсудить с преподавателем перечень используемой литературы, за день до семинарского занятия предупредить его о необходимых для представления материала технических средствах. Напечатанный текст доклада представить преподавателю на рецензию.

Подготовка к зачетам, экзаменам

Изучение выше перечисленных тем дисциплины завершается зачетами или экзаменами.

Подготовка к зачету или экзамену способствует закреплению, углублению и обобщению знаний, получаемых в процессе обучения, а также применению их к решению практических задач. Готовясь к зачету или экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На зачете или экзамене студент демонстрирует то, что он приобрел в процессе обучения конкретным темам междисциплинарных курсов или модулям в целом.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами, согласно графику их проведения, дается интервал времени в несколько дней. Не следует думать, что их достаточно для успешной подготовки к экзаменам. В эти дни нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными

требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки студента к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня: сон не менее 8 часов в сутки, занятия должны заканчиваться не позднее, чем за 2-3 часа до сна.

Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая- либо лекция, необходимо во время ее восстановить, обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к зачету или экзамену у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных конспектов. Вначале следует просмотреть весь материал по сдаваемой теме, отметить для себя трудные вопросы, обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения. Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Правила подготовки к экзамену:

- сориентироваться во всем материале и обязательно расположить его согласно экзаменационным вопросам или вопросам, обсуждаемым на семинарах, учебных занятиях. Эта работа может занять много времени, но все остальное - уже технические детали, главное - это ориентировка в материале;
- постараться максимально запомнить материал, переосмыслить его, рассмотреть альтернативные идеи;
- подготовить «шпаргалки», главный смысл которых систематизация и оптимизация знаний, однако пользоваться таким подспорьем не рекомендуется. Это очень сложная и важная для студента работа, более сложная и важная, чем простое поглощение массы учебной информации. Если студент самостоятельно подготовил такие «шпаргалки», то, скорее всего, он и экзамены сдавать будет более уверенно, так как у него уже сформирована общая ориентировка в сложном материале. Как это ни парадоксально, но использование «шпаргалок» часто позволяет отвечающему студенту лучше демонстрировать свои познания, точнее - ориентировку в знаниях, что намного важнее знания «запомненного» и «тут же забытого» после сдачи экзамена.

При ответе на экзамене студент сначала должен продемонстрировать преподавателю усвоенный по программе обучения материал, и лишь после этого высказать иную, желательно аргументированную точку зрения.

4 МЕТОДИКА ВЫПОЛНЕНИЯ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Получить у преподавателя задание и необходимую литературу.
2. Найти предложенную литературу на образовательном портале или в библиотеке.
3. Изучить имеющуюся литературу в электронном или печатном виде, прочитать материалы лекций, практических и (или) семинарских занятий по теме.
4. Изучить методические рекомендации.
5. Оформить работу в тетради или на компьютере в соответствии с требованиями преподавателя.
6. Сдать самостоятельную работу преподавателю, предварительно ответив на вопросы для самоконтроля.

5 МЕТОДЫ КОНТРОЛЯ И ОЦЕНКА ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Контроль результатов самостоятельной работы проводится преподавателем одновременно с текущим и промежуточным контролем знаний обучающихся. Для контроля самостоятельной работы обучающегося используются разнообразные формы и методы: фронтальный, индивидуальный, выборочный, самоконтроль, защита презентации, участие в семинарском занятии, ответы на контрольные вопросы и т. д. При контроле результатов самостоятельной работы используются следующие критерии:

- уровень освоения обучающимся учебного материала;
- умение обучающегося использовать теоретические знания при выполнении заданий;
- обоснованность и чёткость изложения ответа;
- оформления материала в соответствии с требованиями.

Критерии оценки выполненной обучающимися работы:

- оценка «5» - работа выполнена без ошибок; чисто, без исправлений; тема раскрыта полностью;
- оценка «4» - работа выполнена с незначительными ошибками; тема раскрыта не полностью;
- оценка «3» - работа выполнена со значительными ошибками; тема практически не раскрыта;
- оценка «2» - работа не выполнена.

СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ

№ п/п	Библиографическое описание	Ресурс
1	Щербак, А.В.Информационная безопасность: учебник для среднего профессионального образования/ А.В. Щербак.— 2-е изд., перераб. и доп.— Москва: Издательство Юрайт, 2025.— 252 с.— (Профессиональное образование). — ISBN 978-5-534-20154-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL:https://www.biblio-online.ru/bcode/567521.	ЭБС ЮРАЙТ
2	Козырь, Н.С. Анализ и оценка рисков информационной безопасности: учебник для среднего профессионального образования/ Н. С.Козырь, В. Н. Хализев.— Москва: Издательство Юрайт, 2025.— 157с.— (Профессиональное образование).— ISBN 978-5-534-20645-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL:https://www.biblio-online.ru/bcode/581503.	ЭБС ЮРАЙТ
3	Казарин, О.В. Программно-аппаратные средства защиты информации. Защита программного обеспечения: учебник и практикум для среднего профессионального образования / О. В. Казарин, А.С. Забабурин.— Москва: Издательство Юрайт, 2025. — 312с.— (Профессиональное образование).— ISBN 978-5-534-13221-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://www.biblio-online.ru/bcode/567283.	ЭБС ЮРАЙТ
4	Волк, В. К. Информатика: учебное пособие для среднего профессионального образования/ В.К.Волк. — 2-е изд.— Москва: Издательство Юрайт, 2024.— 226 с.— (Профессиональное образование).— ISBN 978-5-534-18452-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://www.biblio-online.ru/bcode/535033	ЭБС ЮРАЙТ

Перечень учебно-методического обеспечения

№ п/п	Библиографическое описание	Ресурс
1	Соколова О.И. Информационная безопасность и защита информации [Электронный ресурс]: учебно-методическое пособие для выполнения лабораторных работ. / О.И. Соколова; ФГБОУ ВО РГУПС. – Ростов н/Д, 2025 - 199 с.	ЭБС РГУПС

№ п/п	Библиографическое описание	Ресурс
2	Чернов, А.В. Информационная безопасность и защита данных: учеб.-метод. пособие для выполнения курс. и расчет.-граф. работ / А. В. Чернов, О. В. Дейнеко; ФГБОУ ВО РГУПС. - Ростов н/Д: [б. и.], 2017. - 19 с. - Библиогр.: 4 назв..- Текст : электронный ЭБС РГУПС.	ЭБС РГУПС
3	Доманский, В.В. Информационная безопасность и защита информации: практикум / В. В. Доманский, А. В. Чернов; ФГБОУ ВО РГУПС. - Ростов н/Д: [б. и.], 2016. - 43 с.: ил., табл. - Библиогр.: 32 назв..- Текст : электронный	ЭБС РГУПС

Электронные образовательные ресурсы в сети "Интернет"

№ п/п	Адрес в Интернете, наименование
1	http://rgups.ru/. Официальный сайт РГУПС
2	http://www.iprbookshop.ru/. Электронно-библиотечная система "IPR SMART"
3	https://urait.ru/. Электронно-библиотечная система "Юрайт"
4	http://cmko.rgups.ru/. Центр мониторинга качества образования РГУПС
5	https://portal.rgups.ru/. Система личных кабинетов НПР и обучающихся в ЭИОС
6	http://www.umczdt.ru/. Электронная библиотека "УМЦ ЖДТ"
7	https://webirbis.rgups.ru/. Электронно-библиотечная система РГУПС
8	https://eivis.ru/. Универсальная база данных "ИВИС"