

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 25.09.2024 21:30:51

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d79e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники



Кодирование информации

Методические указания
по выполнению лабораторной работы по дисциплине
«Информатика» для студентов направлений подготовки:
09.03.01 Информатика и вычислительная техника,
11.03.02 Инфокоммуникационные технологии и
системы связи, 10.03.01 Информационная безопасность,
11.03.03 Конструирование и технология электронных средств,
специальности 10.05.02 Информационная безопасность
телекоммуникационных систем

Курск 2021 г.

УДК 681.3

Составители: С.В. Дегтярев, Е.Н. Иванова, Т.Н. Конаныхина

Рецензент

Доцент кафедры программной инженерии,
кандидат технических наук

Ю.А. Халин

Кодирование информации: методические указания по выполнению лабораторной работы / Юго-Зап. гос. ун-т; сост.: С.В. Дегтярев, Е.Н. Иванова, Т.Н. Конаныхина. – Курск, 2021. – 22 с.: табл. 5. – Библиограф.: с. 16.

Рассматриваются алгоритмы кодирования информации. Приводятся приемы использования встроенных средств табличного процессора для кодирования/декодирования текстовой информации. Теоретический материал сопровождается примерами.

Методические указания соответствуют требованиям программ, утвержденным учебно-методическим объединением по направлениям Информатика и вычислительная техника, Инфокоммуникационные технологии и системы связи, Информационная безопасность, Конструирование и технология электронных средств, специальности Информационная безопасность телекоммуникационных систем.

Предназначены для студентов очной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ.л. Уч.-изд.л. . Тираж 20 экз. Заказ . Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель работы

Получить знания о методах и алгоритмах кодирования информации; научиться анализировать возможности средств табличного процессора для реализации различных методов кодирования; приобрести навык использования различных инструментов табличного процессора для выполнения кодирования/декодирования информации

Краткая теоретическая информация

Информация

Информация есть информация –
не материя и не энергия.

Норберт Винер

Развитие человеческого общества в целом связано с накоплением и обменом информации. Тысячи лет люди в процессе накопления и распространения информации использовали только свою память и речь. Появление телеграфа, телефона, радио, телевидения позволило передавать огромные потоки информации.

Таким образом информацию можно передавать, воспринимать, использовать, запоминать, принимать, кроме того создавать, копировать, формализовать, распространять, преобразовывать, комбинировать, обрабатывать, делить на части, упрощать, собирать, хранить, искать, измерять, разрушать, и др.

Все эти процессы, связанные с определенными операциями над информацией, называются информационными процессами.

Информация обладает следующими свойствами, характеризующими ее качественные признаки: достоверность, полнота, ценность, своевременность, понятность, доступность, краткость и др.

Для того чтобы человек мог быстро ориентироваться в огромном потоке информации, ему необходимы надёжные и неутомимые помощники, которые должны накапливать её, сортировать и перерабатывать в соответствии с запросами потребителя, выдавать ему нужную информацию в систематизированном и удобном для использования виде. Компьютеризация дает людям быстрый доступ к источникам информации, избавляет их от рутинной работы, обес-

печивает высокий уровень автоматизации обработки информации в производственной и социальной сферах.

Информация – фундаментальное понятие, поэтому определить его исчерпывающим образом через какие-то более простые понятия невозможно. Можно лишь, обращаясь к различным аспектам этого понятия, пояснить его смысл. В широком смысле информация – это отражение реального мира, выражаемое в виде сигналов и знаков. Сигналы отражают физические характеристики различных процессов и объектов.

В интуитивном, житейском смысле под информацией мы понимаем совокупность интересующих нас сведений, знаний и т.д. При этом подразумевается, что существует источник информации и её потребитель. От источника к потребителю информация передается с помощью сообщения. Сообщение может содержать слова, выражения или знаки, смысл и значение которых заранее известны адресату.

Обычно информация хранится, передается, обрабатывается в символической (знаковой) форме. Алфавитный подход к измерению информации позволяет определить количество информации в тексте и является объективным, т.е. не зависит от субъекта (человека), воспринимающего текст.

Одна и та же информация может быть представлена в разной форме, с помощью различных знаковых систем. Язык – это определенная знаковая система представления информации. Существуют естественные (разговорные) языки и формальные языки (нотная грамота, язык математики, язык мимики и жестов, дорожные знаки и т.д.).

Кодирование информации

Конечный упорядоченный набор знаков, используемых для передачи информации, называется алфавитом. Последовательность символов алфавита – словом. Сообщение, как правило, представляет собой последовательность слов. Довольно часто передаваемая информация кодируется. Кодирование информации – переход от одной формы представления информации к другой, более удобной для хранения, обработки или передачи.

Способ кодирования зависит от цели, ради которой оно осу-

ществляется: сокращение записи, засекречивание (шифровка) информации, удобство обработки и т.д.

В связи с развитием информационных технологий кодирование является центральным вопросом при решении самых разных задач программирования, таких как:

- представление данных произвольной структуры (числа, текст, графика) в памяти компьютера;
- обеспечение помехоустойчивости при передаче данных по каналам связи;
- сжатие информации в базах данных.

Правило отображения одного алфавита A на другой B называется кодом, а сама процедура – кодированием сообщения. Обратная процедура сопоставления кодовым словам алфавита B символов алфавита A называется декодированием.

Например, при передаче сообщения по телеграфному каналу используется азбука Морзе (таблица 1).

Таблица 1

Азбука Морзе

Русский символ	Код Морзе	Русский символ	Код Морзе
А	· –	Р	· – ·
Б	– · · ·	С	· · ·
В	· – –	Т	–
Г	– – ·	У	· · –
Д	– · ·	Ф	· · – ·
Е (также и Ё)	·	Х	· · · ·
Ж	· · · –	Ц	– · · ·
З	– – · ·	Ч	– – – ·
И	· ·	Ш	– – – –
Й	· – – –	Щ	– – · –
К	– · –	Ъ	– – · – –
Л	· – · ·	Ы	– · – –
М	– –	Ь	– · · –
Н	– ·	Э	· · – · ·
О	– – –	Ю	· · – –
П	· – – ·	Я	· – · –

Азбука Морзе является общеизвестным кодом из символов те-

леграфного алфавита, в котором буквам русского языка соответствуют кодовые слова (последовательности) из «точек» и «тире».

В технических устройствах хранения, передачи и обработки информации для её кодирования часто используют алфавиты, содержащие лишь два различных символа. Например, «плюс», «минус» («+», «-») или «1» и «0». Наличие всего двух символов значительно упрощает электрические схемы с электронными переключателями, которые принимают только два состояния – они либо проводят ток, либо нет.

Как известно, в качестве единицы измерения количества информации принят 1 бит (*англ. bit – binary digit* – двоичная цифра). Бит – один символ двоичного алфавита: 0 или 1. Наряду с битом получила распространение укрупнённая единица – байт, равный 8 битам. Из битов складывается все многообразие данных, которые обрабатывает компьютер. Комбинируя восемь нулей и единиц различными способами, можно получить 256 различных комбинаций. Этого количества достаточно, чтобы каждому символу поставить в соответствие свою неповторимую комбинацию из восьми нулей и единиц. Эти комбинации определяются кодовой таблицей ASCII (*American Standard Code for Information Interchange* – американский стандартный код для обмена информацией).

Одним байтом кодируется любой печатный знак (буква, цифра, любой другой символ). Например, число 7 выражается байтом 00000111; буква L – 01001100; знак «+» имеет двоичный код 00101011. Байт – один символ, который представляет комбинацию из 8 бит.

Порядковый номер символа в таблице ASCII называют десятичным кодом этого символа. Чтобы его определить необходимо сложить номер строки с номером столбца, которые соответствуют выбранному символу. Например, десятичный код цифры 7 равен 55 (7+48), а символа «%» – 37 (5+32). Таким образом, каждый символ имеет десятичный и двоичный код. Первые 32 символа являются управляющими и предназначены, в основном, для передачи различных команд.

В таблице 2 представлены ASCII коды для букв русского алфавита (пробел кодируется кодом 32₁₀).

ASCII – коды

	192	208	224	240
0	А	Р	а	р
1	Б	С	б	с
2	В	Т	в	т
3	Г	У	г	у
4	Д	Ф	д	ф
5	Е	Х	е	х
6	Ж	Ц	ж	ц
7	З	Ч	з	ч
8	И	Ш	и	ш
9	Й	Щ	й	щ
10	К	Ъ	к	ъ
11	Л	Ы	л	ы
12	М	Ь	м	ь
13	Н	Э	н	э
14	О	Ю	о	ю
15	П	Я	п	я

Известны два класса методов кодирования: равномерное и неравномерное кодирование. Под равномерным кодированием понимается использование кодов со словами постоянной длины. Для того чтобы декодирование равномерного кода было возможным, разным символам алфавита источника должны соответствовать разные кодовые слова. При этом длина кодового слова должна быть не меньше $\lceil \log_n m \rceil$ символов, где m – размер исходного алфавита, n – размер кодового алфавита, знак $\lceil \rceil$ означает округление до большего целого.

Для кодирования информации на русском языке, алфавит которого включает 33 буквы, равномерным двоичным кодом требуется построить кодовые слова длиной не меньше $\lceil \log_2 33 \rceil = 6$.

При неравномерном кодировании используются кодовые слова разной длины. Причем кодовые слова обычно строятся так, чтобы часто встречающиеся символы кодировались более короткими

кодowymi словами, а редкие символы – более длинными (за счет этого и достигается «сжатие» данных).

Под сжатием данных понимается компактное представление данных, достигаемое за счет избыточности информации, содержащейся в сообщениях. Большое значение для практического использования имеет неискажающее сжатие, позволяющее полностью восстановить исходное сообщение. При неискажающем сжатии происходит кодирование сообщения перед началом передачи или хранения, а после окончания процесса сообщение однозначно декодируется (это соответствует модели канала без шума (помех)).

Методы сжатия данных можно разделить на две группы: статические методы и адаптивные методы. Статические методы сжатия данных предназначены для кодирования конкретных источников информации с известной статистической структурой, порождающих определенное множество сообщений. Эти методы базируются на знании статистической структуры исходных данных. К наиболее известным статическим методам сжатия относятся коды Хаффмана, Шеннона, Фано, Гилберта-Мура, арифметический код и другие методы, которые используют известные сведения о вероятностях порождения источником различных символов или их сочетаний.

Если статистика источника информации неизвестна или изменяется с течением времени, то для кодирования сообщений такого источника применяются адаптивные методы сжатия. В адаптивных методах при кодировании очередного символа текста используются сведения о ранее закодированной части сообщения для оценки вероятности появления очередного символа. В процессе кодирования адаптивные методы «настраиваются» на статистическую структуру кодируемых сообщений, т.е. коды символов меняются в зависимости от накопленной статистики данных. Это позволяет адаптивным методам эффективно и быстро кодировать сообщение за один просмотр.

Код Хаффмана

Метод оптимального побуквенного кодирования был разработан в 1952 г. Д. Хаффманом. Оптимальный двоичный код Хаффмана обладает минимальной средней длиной кодового слова среди

всех побуквенных кодов для источника с алфавитом $A = \{a_1, a_2, \dots, a_n\}$ и вероятностями $p_i = P(a_i), i = \overline{1, n}, \sum_{i=1}^n p_i = 1$.

Алгоритм построения оптимального кода Хаффмана заключается в следующих шагах.

1. Упорядочим символы исходного алфавита $A = \{a_1, a_2, \dots, a_n\}$ по убыванию их вероятностей $p_1 \geq p_2 \geq \dots \geq p_n$.

2. Если $A = \{a_1, a_2\}$, то $a_1 \rightarrow 0, a_2 \rightarrow 1$.

3. Если $A = \{a_1, a_2, \dots, a_j, \dots, a_n\}$ и известны коды $\langle a_j \rightarrow b_j \rangle, j = \overline{1, n}$, то для алфавита $A' = \{a_1, a_2, \dots, a'_j, a''_j, \dots, a_n\}$ с новыми символами a'_j, a''_j вместо a_j , и вероятностями $p_j = p'_j + p''_j$ код символа a_j заменяется на коды $a'_j \rightarrow b_j0$ и $a''_j \rightarrow b_j1$.

Например, пусть имеется алфавит $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ с вероятностями $p_1 = 0,4, p_2 = 0,2, p_3 = 0,2, p_4 = 0,1, p_5 = 0,05, p_6 = 0,05$

В рассматриваемом случае символы алфавита уже упорядочены в соответствии с вероятностями их появления в сообщении. Будем складывать две наименьшие вероятности и включать суммарную вероятность на соответствующее место в упорядоченном списке вероятностей до тех пор, пока в списке не останется два символа. Тогда закодируем эти два символа 0 и 1. Далее кодовые слова достраиваются, как показано на рисунке 1.

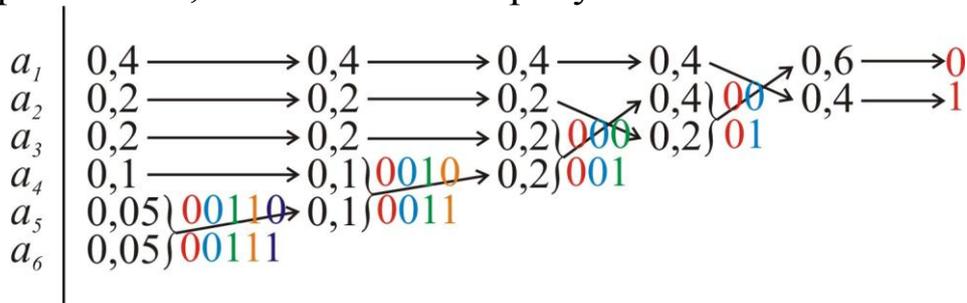


Рисунок 1 – Процесс построения кода Хаффмана

Результат кодирования представлен в таблице 3.

Результат кодирования кодом Хаффмана

Символ	a_1	a_2	a_3	a_4	a_5	a_6
Вероятность	0,4	0,2	0,2	01	0,05	0,05
Код	1	01	000	0010	00110	00111

Код Шеннона

Пусть имеется алфавит $A = \{a_1, a_2, \dots, a_n\}$ с вероятностями символов $p_i = P(a_i)$, $i = \overline{1, n}$, $\sum_{i=1}^n p_i = 1$.

Код Шеннона строится следующим образом:

1. Упорядочим символы исходного алфавита $A = \{a_1, a_2, \dots, a_n\}$ по убыванию их вероятностей $p_1 \geq p_2 \geq \dots \geq p_n$.

2. Вычислим величины Q_i , $i = \overline{1, n}$, которые называются кумулятивные вероятности:

$$Q_0 = 0$$

$$Q_1 = p_1$$

$$Q_2 = p_1 + p_2$$

...

$$Q_i = p_1 + p_2 + \dots + p_i$$

...

$$Q_n = p_1 + p_2 + \dots + p_i + \dots + p_n = 1.$$

3. Представим Q_i , $i = \overline{1, n}$ в двоичной системе счисления и возьмем в качестве кодового слова первые $\lceil \log_2 p_i \rceil$ двоичных знаков после запятой.

Например, пусть имеется алфавит $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ с вероятностями $p_1 = 0,4$, $p_2 = 0,2$, $p_3 = 0,2$, $p_4 = 0,1$, $p_5 = 0,05$, $p_6 = 0,05$. Построенный код приведен в таблице 4.

Таблица 4

Результат кодирования кодом Шеннона

Символ	a_1	a_2	a_3	a_4	a_5	a_6
Вероятность	0,4	0,2	0,2	0,1	0,05	0,05
Величины $Q_i, i = \overline{1, n}$	0	0,4	0,6	0,8	0,9	0,95
Код	00	011	100	1100	11100	11110

Код Фано

Метод Фано, заключается в следующем.

1. Упорядоченный по убыванию вероятностей список символов алфавита источника делится на две части так, чтобы суммы вероятностей букв, входящих в эти части, как можно меньше отличались друг от друга.

2. Символам первой части приписывается 0, а символам из второй части – 1. Далее также поступают с каждой из полученных частей.

3. Процесс продолжается до тех пор, пока весь список не разобьется на части, содержащие по одной букве.

Например, пусть имеется алфавит $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ с вероятностями $p_1 = 0,4$, $p_2 = 0,2$, $p_3 = 0,2$, $p_4 = 0,1$, $p_5 = 0,05$, $p_6 = 0,05$. Построенный код приведен в таблице 5.

Таблица 5

Результат кодирования кодом Фано

Символ	Вероятность	Формирование кодового слова				Код
a_1	0,4	0	0			00
a_2	0,2		1			01
a_3	0,2	1	0			10
a_4	0,1			0		110
a_5	0,05		1		0	1110
a_6	0,05			1	1	1111

Шифр Цезаря

Шифр Цезаря, также известный как шифр сдвига, – один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря – это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 3 «А» была бы заменена на «Г», «Б» станет «Д», и так далее.



Рисунок 2 – Шифр Цезаря со сдвигом 3

Используемое преобразование обычно обозначают как ROTN, где N – сдвиг, ROT – сокращение от слова ROTATE, в данном случае «циклический сдвиг». Число разных преобразований конечно и зависит от длины алфавита. Для русского языка возможно 32 разных преобразования (преобразования ROT0 и ROT33 сохраняют исходный текст, а дальше начинаются уже повторения).

Задание

Вариант 1

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью азбуки Морзе.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода фразы;
- на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;
- ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 2

Используя стандартные средства табличного процессора (таб-

лица 1), разработать инструмент декодирования фразы, закодированной с помощью азбуки Морзе.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода закодированной фразы;
- на листе табличного процессора должна быть отведена ячейка для вывода декодированной фразы;
- ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 3

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью кода ASCII.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода фразы;
- на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;
- ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 4

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент декодирования фразы, закодированной с помощью кода ASCII.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода закодированной фразы;
- на листе табличного процессора должна быть отведена ячейка для вывода декодированной фразы;
- ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 5

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью кода Хаффмана.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка

ка для ввода фразы;

– на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;

– ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 6

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью кода Шеннона.

Требования к выполнению задания.

– на листе табличного процессора должна быть отведена ячейка для ввода фразы;

– на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;

– ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 7

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью кода Фано.

Требования к выполнению задания.

– на листе табличного процессора должна быть отведена ячейка для ввода фразы;

– на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;

– ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 8

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент кодирования фразы на русском языке с помощью шифра Цезаря.

Требования к выполнению задания.

– на листе табличного процессора должна быть отведена ячейка для ввода фразы;

– на листе табличного процессора должна быть отведена ячейка для вывода закодированной фразы;

– на листе табличного процессора должна быть отведена ячейка

ка для ввода величины сдвига;

- ячейки должны содержать комментарии, поясняющие их назначение.

Вариант 9

Используя стандартные средства табличного процессора (таблица 1), разработать инструмент декодирования фразы, закодированной с помощью шифра Цезаря.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода закодированной фразы;

- на листе табличного процессора должна быть отведена ячейка для вывода декодированной фразы;

- на листе табличного процессора должна быть отведена ячейка для ввода величины сдвига;

- ячейки должны содержать комментарии, поясняющие их назначение.

Общие требования

При оформлении работы в Excel необходимо использовать шрифт Times New Roman, размер 14, начертание обычное. Размеры ячеек выбираются по контексту, каждая ячейка должна иметь пояснения, какая информация в ней находится.

Составить отчет по результатам выполнения лабораторной работы. Отчет должен содержать:

- титульный лист (Приложение А);

- Содержание;

- Цель работы;

- Задание;

- Словесный подробный алгоритм выполнения задания с указанием функций процессора, используемых для выполнения конкретного действия;

- Скриншот интерфейса инструмента преобразования чисел;

- Скриншот листа выполнения задания с отображением формул и с отображением результатов вычислений;

- Выводы по работе.

Номера страниц проставляются внизу страницы по правому краю (шрифт – Times New Roman 12 пт). Первой страницей является титульный лист, нумерация проставляется, начиная с Цели рабо-

ты отчета.

Макет Оглавления приведен в Приложении Б.

Для оформления заголовков разделов следует использовать стиль «Заголовок 1». Параметры стиля:

- название шрифта – Times New Roman;
- размер шрифта – 14 пт;
- междустрочный интервал – полуторный;
- интервалы перед и после абзаца – 12 пт;
- выравнивание – по ширине;
- отступы слева и справа - 0 см;
- отступ первой строки – 1,25 см.

Для оформления текста отчета необходимо использовать стиль «Обычный». Параметры стиля:

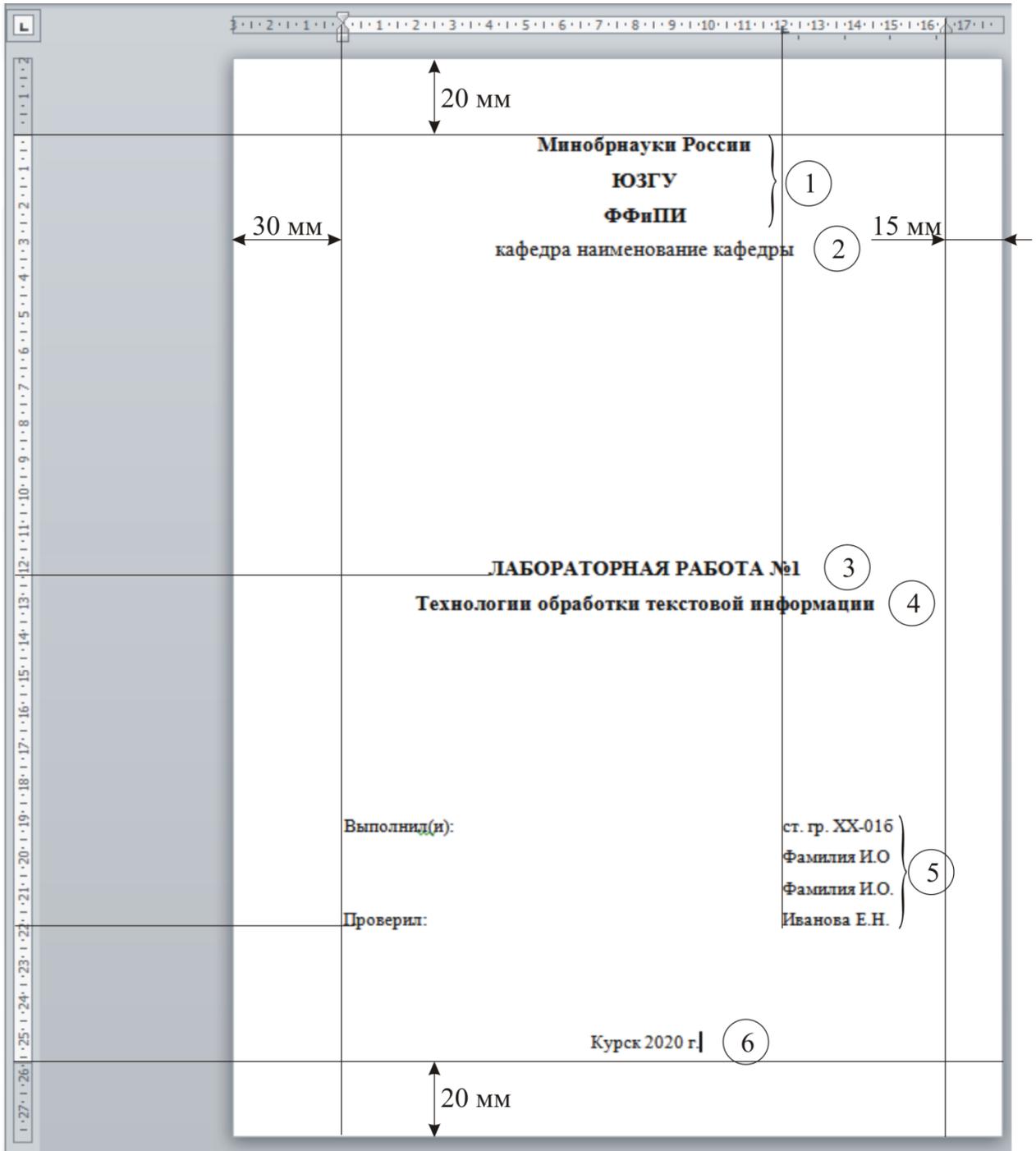
- название шрифта – Times New Roman;
- размер шрифта – 14 пт;
- междустрочный интервал – полуторный;
- интервалы перед и после абзаца – 0 пт;
- выравнивание – по ширине;
- отступы слева и справа - 0 см;
- отступ первой строки – 1,25 см.

Список использованных источников

1. Котенко, В.В. Теория информации [Текст] : учебное пособие / В.В. Котенко. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2018. – 240 с. – Режим доступа: URL: <https://biblioclub.ru/>

2. Майстренко, Н.В. Основы теории информации и криптографии: учебное электронное издание [Текст] / Н.В. Майстренко, А.В. Майстренко. – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2018. – 81 с. – Режим доступа: URL: <https://biblioclub.ru/index.php?page=book&id=494706>.

Оформление титульного листа отчета по лабораторной работе



Примечание: в кружках обозначены номера наборов параметров форматирования текста.

1: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

2: Шрифт Times New Roman, размер 16 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

3: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, все прописные. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

4: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

5: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по ширине, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки, позиция табуляции 12 см.

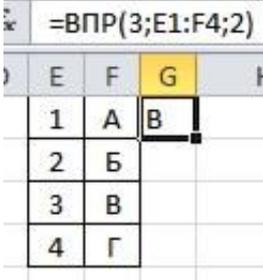
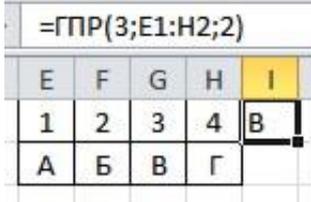
6: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

1: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, все прописные. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

2: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по левому краю, отступы слева 0, справа 0, интервалы перед 0, после 0, междустрочный 1,5 строки.

Справка по функциям табличного процессора

функция	назначение	пример использования										
1	2	3										
ДЛСТР(текст)	возвращает количество знаков в текстовой строке	<p>fx =ДЛСТР("Информатика")</p> <table border="1"> <tr> <td>D</td> <td>E</td> <td>F</td> </tr> <tr> <td></td> <td>11</td> <td></td> </tr> </table>	D	E	F		11					
D	E	F										
	11											
ЗНАЧЕН(текст)	преобразует текстовый аргумент в число	<p>fx =ЗНАЧЕН(D1)</p> <table border="1"> <tr> <td>D</td> <td>E</td> </tr> <tr> <td>001,1110</td> <td>1,111</td> </tr> </table>	D	E	001,1110	1,111						
D	E											
001,1110	1,111											
ЛЕВСИМВ(текст; количество_знаков)	возвращает указанное количество знаков с начала строки текста	<p>fx =ЛЕВСИМВ(D1;3)</p> <table border="1"> <tr> <td>D</td> <td>E</td> </tr> <tr> <td>1110,011</td> <td>111</td> </tr> </table>	D	E	1110,011	111						
D	E											
1110,011	111											
НАЙТИ(искомый_текст; просматриваемый_текст; начальная позиция)	Возвращает позицию начала искомой строки текста в содержащей ее строке текста. Прописные и строчные буквы различаются	<p>fx =НАЙТИ(", ";D1;1)</p> <table border="1"> <tr> <td>D</td> <td>E</td> </tr> <tr> <td>1110,011</td> <td>5</td> </tr> </table>	D	E	1110,011	5						
D	E											
1110,011	5											
ПРАВСИМВ(текст; число_знаков)	возвращает указанное число знаков с конца строки текста	<p>fx =ПРАВСИМВ(D1;3)</p> <table border="1"> <tr> <td>D</td> <td>E</td> </tr> <tr> <td>1110,011</td> <td>011</td> </tr> </table>	D	E	1110,011	011						
D	E											
1110,011	011											
ПСТР(текст; начальная_позиция; количество_знаков)	возвращает заданное число знаков из строки текста, начиная с указанной позиции	<p>fx =ПСТР(D1;3;2)</p> <table border="1"> <tr> <td>D</td> <td>E</td> </tr> <tr> <td>1110,011</td> <td>10</td> </tr> </table>	D	E	1110,011	10						
D	E											
1110,011	10											
СЦЕПИТЬ(текст1; текст2;...)	объединяет несколько текстовых строк в одну	<p>fx =СЦЕПИТЬ(D1;E1;F1;G1)</p> <table border="1"> <tr> <td>D</td> <td>E</td> <td>F</td> <td>G</td> <td>H</td> </tr> <tr> <td>Ю</td> <td>З</td> <td>Г</td> <td>У</td> <td>ЮЗГУ</td> </tr> </table>	D	E	F	G	H	Ю	З	Г	У	ЮЗГУ
D	E	F	G	H								
Ю	З	Г	У	ЮЗГУ								

1	2	3																				
<p>ВПР(искомое_значение; таблица; номер_столбца; ...)</p>	<p>ищет значение в крайнем левом столбце таблицы и возвращает значение ячейки, находящейся в указанном столбце той же строки</p>	 <p>=ВПР(3;E1:F4;2)</p> <table border="1" data-bbox="1093 380 1356 604"> <thead> <tr> <th></th> <th>E</th> <th>F</th> <th>G</th> </tr> </thead> <tbody> <tr> <th>1</th> <td>А</td> <td></td> <td>В</td> </tr> <tr> <th>2</th> <td></td> <td>Б</td> <td></td> </tr> <tr> <th>3</th> <td></td> <td>В</td> <td></td> </tr> <tr> <th>4</th> <td></td> <td>Г</td> <td></td> </tr> </tbody> </table>		E	F	G	1	А		В	2		Б		3		В		4		Г	
	E	F	G																			
1	А		В																			
2		Б																				
3		В																				
4		Г																				
<p>ГПР(искомое_значение; таблица; номер_строки; ...)</p>	<p>ищет значение в верхней строке таблицы и возвращает значение ячейки, находящейся в указанной строке того же столбца</p>	 <p>=ГПР(3;E1:H2;2)</p> <table border="1" data-bbox="1069 772 1380 918"> <thead> <tr> <th></th> <th>E</th> <th>F</th> <th>G</th> <th>H</th> <th>I</th> </tr> </thead> <tbody> <tr> <th>1</th> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>В</td> </tr> <tr> <th></th> <td>А</td> <td>Б</td> <td>В</td> <td>Г</td> <td></td> </tr> </tbody> </table>		E	F	G	H	I	1	1	2	3	4	В		А	Б	В	Г			
	E	F	G	H	I																	
1	1	2	3	4	В																	
	А	Б	В	Г																		