|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)**

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6**

**по дисциплине «Структуры и алгоритмы обработки данных»**

|  |  |  |  |
| --- | --- | --- | --- |
| Отчет представлен к  рассмотрению:  Студент группы ИКБО-36-22 | «06» ноября  2023 г. | (подпись) | Утенков Ю.Ю. |
|  |  |  |  |
| Преподаватель | «06» ноября  2023 г. | (подпись) | Красников С.А. |

Москва 2023 г.

Оглавление

[Цель работы. 3](#_Toc151044301)

[Задание №1. Исследование алгоритмов сжатия на примерах. 3](#_Toc151044302)

[Постановка задачи. 3](#_Toc151044303)

[Алгоритм решения. 3](#_Toc151044304)

[Задание №2. Разработать программы сжатия и восстановления текста методами Хаффмана и Шеннона – Фано. 6](#_Toc151044305)

[Постановка задачи. 6](#_Toc151044306)

[Описание алгоритма (математическая модель решения). 6](#_Toc151044307)

[Способ реализации. 7](#_Toc151044308)

[Код программы с комментариями. 10](#_Toc151044309)

[Результаты тестирования. 20](#_Toc151044310)

[Пример 1: 20](#_Toc151044311)

[Пример 2: 22](#_Toc151044312)

[Пример 3: 24](#_Toc151044313)

[Выводы. 25](#_Toc151044314)

[Список информационных источников. 26](#_Toc151044315)

# Цель работы.

Получить знания алгоритмов кодирования и сжатия данных методами без потерь и навыки их применения.

# Задание №1. Исследование алгоритмов сжатия на примерах.

## Постановка задачи.

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Закодировать фразу методами Шеннона–  Фано | Сжатие данных по методу Лемпеля– Зива LZ77  Используя двухсимвольный алфавит (0, 1) закодировать следующую фразу | Закодировать следующую фразу, используя код LZ78 |
|  | Тише, мыши, кот на крыше,  А котята ещё выше. Кот пошёл за молоком,  А котята кувырком. | 110101011001100001001 | долделдолдилделдил |

*Табл.1. Исходные данные.*

1. Выполнить каждую задачу варианта, представив алгоритм решения в виде таблицы и указав результат сжатия.
2. Описать процесс восстановления сжатого текста.
3. Сформировать отчет, включив задание, вариант задания, результаты выполнения задания варианта.

## Алгоритм решения.

1. **Метод Шеннона–Фано.**

Закодировать фразу «Тише, мыши, кот на крыше, а котята ещё выше. Кот пошёл за молоком, а котята кувырком.», используя метод Шеннона–Фано.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Символ** | **Кол-во** | **1-я**  **цифра** | **2-я**  **цифра** | **3-я**  **цифра** | **4-я**  **цифра** | **5-я**  **цифра** | **6-я**  **цифра** | **Код** | **Количество бит** |
| < > | 15 | 0 | 0 | 0 |  |  |  | 000 | 45 |
| о | 9 | 0 | 0 | 1 |  |  |  | 001 | 27 |
| к | 8 | 0 | 1 | 0 |  |  |  | 010 | 24 |
| т | 7 | 0 | 1 | 1 | 0 |  |  | 0110 | 28 |
| а | 6 | 0 | 1 | 1 | 1 |  |  | 0111 | 24 |
| ш | 5 | 1 | 0 | 0 | 0 |  |  | 1000 | 20 |
| е | 4 | 1 | 0 | 1 | 0 |  |  | 1010 | 16 |
| м | 4 | 1 | 0 | 1 | 1 |  |  | 1011 | 16 |
| ы | 4 | 1 | 0 | 0 | 1 | 0 |  | 10010 | 20 |
| , | 4 | 1 | 0 | 0 | 1 | 1 |  | 10011 | 20 |
| р | 2 | 1 | 1 | 0 | 0 | 0 |  | 11000 | 10 |
| ё | 2 | 1 | 1 | 0 | 1 | 0 |  | 11010 | 10 |
| в | 2 | 1 | 1 | 0 | 1 | 1 |  | 11011 | 10 |
| и | 2 | 1 | 1 | 1 | 0 | 0 |  | 11100 | 10 |
| . | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 110010 | 12 |
| я | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 110011 | 12 |
| л | 2 | 1 | 1 | 1 | 0 | 1 | 0 | 111010 | 12 |
| у | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 111011 | 6 |
| щ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 111100 | 6 |
| з | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 111101 | 6 |
| н | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 111110 | 6 |
| п | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 111111 | 6 |
|  |  |  |  |  |  |  |  |  | 346 |

*Табл. 2. Таблица кодировки по методу Шеннона-Фано.*

Исходная строка: 85 \* 8 = 680 бит.

Закодированная строка: 346 бит.

Беря во внимание, что каждый код уникален, т.к алгоритм Шеннона-Фано строит префиксное дерево, где каждый код уникального символа не является началом кода другого символа, можно сделать вывод, что раскодировать данный код возможно.

1. **Метод Лемпеля– Зива LZ77.**

Сжать двоичный код: 110101011001100001001.

|  |  |
| --- | --- |
| Исходный текст | 1.10.101.01.100.11.00.001.001 |
| LZ-код | 1.10.101.001.0100.0011.0000.1111 |
| R | 2 3 4 |
| Вводимые коды | 10 11 100 101 110 111 1000 |

*Табл. 3. Метод LZ77 для бинарного алфавита.*

|  |  |
| --- | --- |
| Текст | Код |
| 0 | 0000 |
| 1 | 0001 |
| 10 | 0010 |
| 101 | 0011 |
| 01 | 0100 |
| 100 | 0101 |
| 11 | 0110 |
| 00 | 0111 |
| 001 | 1000 |

*Табл. 4. Соответствия комбинаций символов и кодов.*

Где LZ – сжатый текст (в данном примере в связи с небольшим размером исходного текста размер текста не уменьшился).

R отмечает шаги кодирования, после которых происходит переход на представление кодов А увеличенным числом разрядов R. Так, на первом шаге вводится код 10 для комбинации 10, и поэтому на следующих двух шагах R = 2, после третьего шага R = 3, после седьмого шага R = 4, т.е. в общем случае R = K после шага 2K-1 –1.

1. **Метод Лемпеля– Зива LZ78.**

Необходимо закодировать фразу: долделдолдилделдил

|  |  |  |
| --- | --- | --- |
| Словарь | Считываемое содержимое | Код |
|  | д | <0, д> |
| д = 1 | о | <0, о> |
| д = 1, о = 2 | л | <0, л> |
| д = 1, о = 2, л = 3 | де | <1, e> |
| д = 1, о = 2, л = 3, де = 4 | лд | <3, д> |
| д = 1, о = 2, л = 3, де = 4, лд = 5 | ол | <2, л> |
| д = 1, о = 2, л = 3, де = 4, лд = 5, ол = 6 | ди | <1, и> |
| д = 1, о = 2, л = 3, де = 4, лд = 5, ол = 6, ди = 7 | лде | <5, е> |
| д = 1, о = 2, л = 3, де = 4, лд = 5, ол = 6, ди = 7, лде = 8 | лди | <5, и> |
| д = 1, о = 2, л = 3, де = 4, лд = 5, ол = 6, ди = 7, лде = 8, лди = 9 |  | <3, EOF> |

*Табл. 5. Метод LZ78 для данного алфавита.*

Результат сжатия: (<0, д>, <0, о>, <0, л>, <1, e>, <3, д>, <2, л>, <1, и>, <5, е>, <5, и>, <3, EOF>)

Словарь хранится в префиксном дереве, что позволяет легко находить самое длинное продолжение входной строки, уже присутствующее в словаре. При декодировании строится в точности этот же словарь. В сжатом представлении строки словарь не хранится.

# Задание №2. Разработать программы сжатия и восстановления текста методами Хаффмана и Шеннона – Фано.

## Постановка задачи.

1. Реализовать и отладить программы.
2. Сформировать отчет по разработке каждой программы в соответствии с требованиями.

- По методу Шеннона-Фано привести: постановку задачи, описать алгоритм формирования префиксного дерева и алгоритм кодирования, декодирования, код и результаты тестирования. Рассчитать коэффициент сжатия. Сравнить результат сжатия моего алгоритма с результатом любого другого архиватора.

- по методу Хаффмана выполнить и отобразить результаты выполнения всех требований, предъявленных в задании и оформить разработку программы: постановка, подход к решению, код, результаты тестирования.

## Описание алгоритма (математическая модель решения).

Для метода Шеннона-Фано:

Кодирование методом Шеннона-Фано - это метод сжатия данных, который базируется на разбиении исходной последовательности символов на подгруппы и назначении каждой из этих подгрупп битовых кодов различной длины. Основная идея состоит в том, чтобы более вероятным символам присваивать более короткие коды, в то время как менее вероятным символам назначаются более длинные коды. Процесс кодирования методом Шеннона-Фано начинается с исходной последовательности символов. Затем эта последовательность разделяется на две подгруппы таким образом, чтобы вероятности символов в этих подгруппах были приблизительно равны. Это делается путем анализа статистики появления символов в последовательности. После разделения каждая подгруппа снова разделяется, и процесс продолжается до тех пор, пока каждая подгруппа не содержит только один символ. Каждый символ в подгруппе затем кодируется битовой последовательностью, где более вероятным символам присваиваются короткие коды, а менее вероятным - длинные коды. Важно, чтобы коды были префиксными, то есть ни один код не должен быть началом другого кода. Этот метод позволяет сжимать данные, представляя их в виде последовательности кодов переменной длины, что позволяет уменьшить общий объем информации.

Для метода Хаффмана:

Кодирование методом Хаффмана - это метод сжатия данных, который также основан на присвоении переменной длины кодов символам. Основное отличие от метода Шеннона-Фано заключается в процессе построения дерева кодирования. Процесс кодирования методом Хаффмана начинается с анализа статистики появления символов в исходной последовательности данных. Затем символы сортируются по вероятности и объединяются в бинарное дерево, где каждый узел имеет два потомка. Вероятные символы находятся на более низких уровнях дерева, а менее вероятные символы - на более высоких уровнях. Это означает, что более вероятным символам назначаются более короткие коды, а менее вероятным - более длинные коды. В результате построения дерева получается таблица кодирования, где каждому символу соответствует его код. Эти коды также являются префиксными, что означает, что ни один код не начинается с другого кода. Метод Хаффмана также позволяет сжимать данные, представляя их в виде последовательности кодов переменной длины. Он широко применяется в информационных технологиях, и до сих пор является хорошим способом сжатия данных.

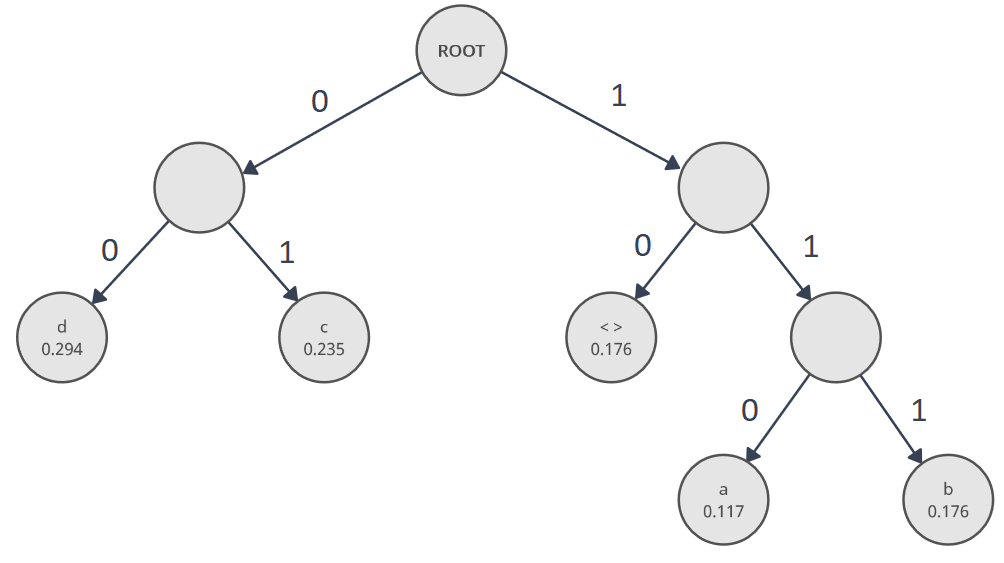
## Способ реализации.

**Метод Шеннона-Фано:**

Сжать строку: “aa bbb cccc ddddd”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алфавит | a | b | c | d | < > |
| Вероятность | 0.11764706 | 0.11764706 | 0.23529412 | 0.29411766 | 0.1764706 |

*Табл. 6. Вероятности каждого символа из строки.*



*Рис. 1. Дерево Шеннона-Фано для исходной строки.*

|  |  |  |
| --- | --- | --- |
|  | Символы из алфавита | Длина, биты |
| До сжатия | 01100001 01100010 01100011 01100100 00100000 | 40 |
| После сжатия | 111 110 01 00 10 | 12 |

*Табл. 7. Уникальные символы из строки.*

Размер исходной строки (LASCII): 17 \* 8 = 136 бита.

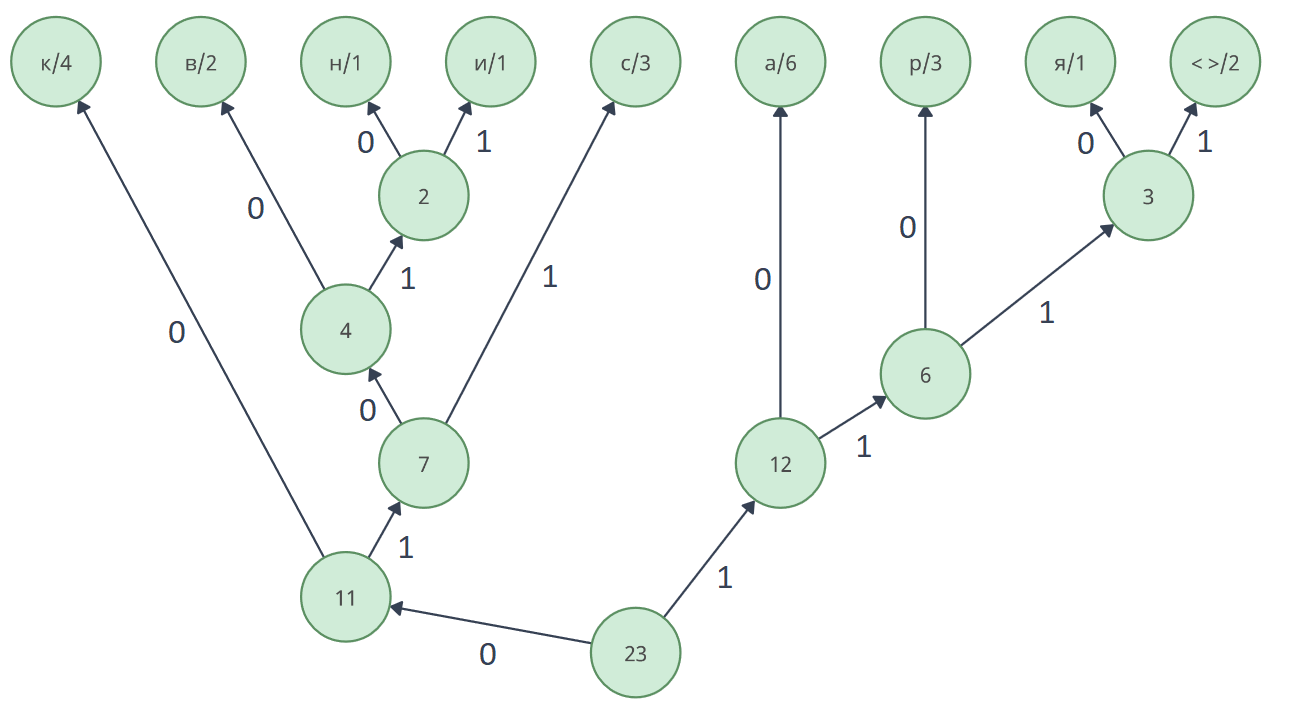
Размер сжатой строки (LHuff): 2\*3+3\*3+4\*2+5\*2+3\*2= 39 бит.

Рассчитаем коэффициент сжатия относительно использования кодировки ASCII (8 бит/символ).

**Метод Хаффмана:**

Сжать строку: “красивая красная краска”.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Алфавит | а | к | р | с | я | < > | в | и | н |
| Количество | 6 | 4 | 3 | 3 | 2 | 2 | 1 | 1 | 1 |

*Табл. 8. Частотность символов в исходной строке.*  


*Рис. 2. Дерево Хаффмана для кодирования исходной строки.*

|  |  |  |
| --- | --- | --- |
|  | Символы из алфавита | Длина, биты |
| До сжатия | 11100000 11101010 11110000 11110001 11111111 00000000 11100010 11101000 11101101 | 72 |
| После сжатия | 10 00 110 011 1110 1111 0100 01011 01010 | 32 |

*Табл. 9. Уникальные символы из строки.*

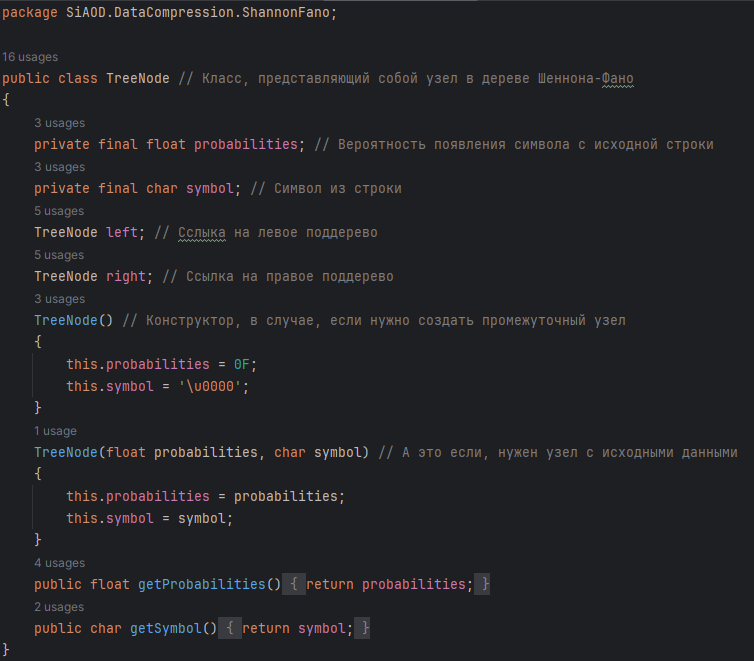
Размер исходной строки (LASCII): 23 \* 8 = 184 бита.

Размер сжатой строки (LHuff): 6\*2+4\*3+3\*2\*3+2\*3+2\*4+3\*4 = 68 бит.

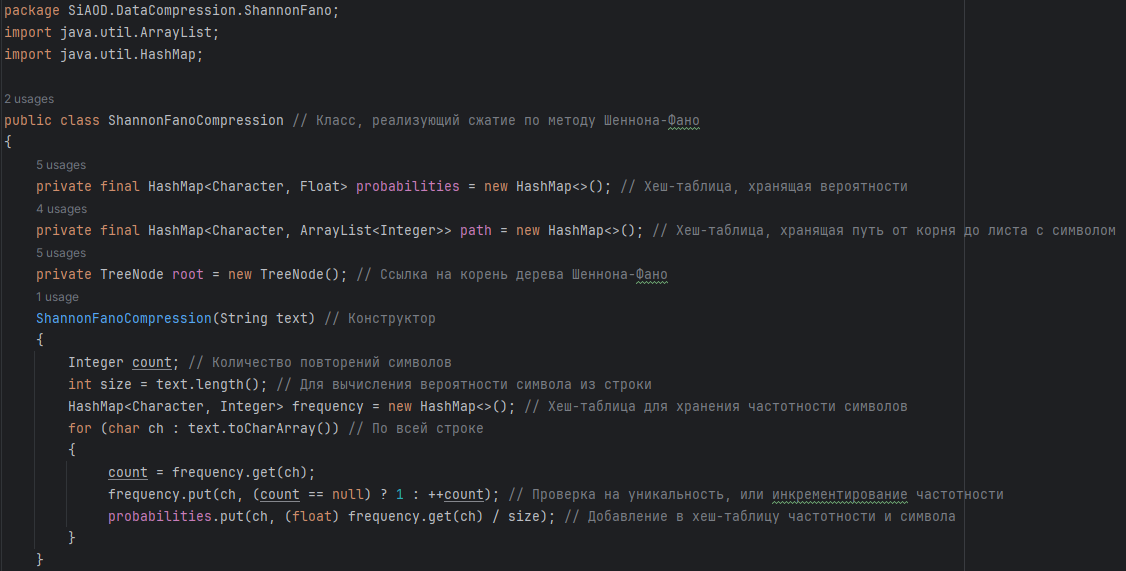
Рассчитаем коэффициент сжатия относительно использования кодировки ASCII (8 бит/символ).

## Код программы с комментариями.

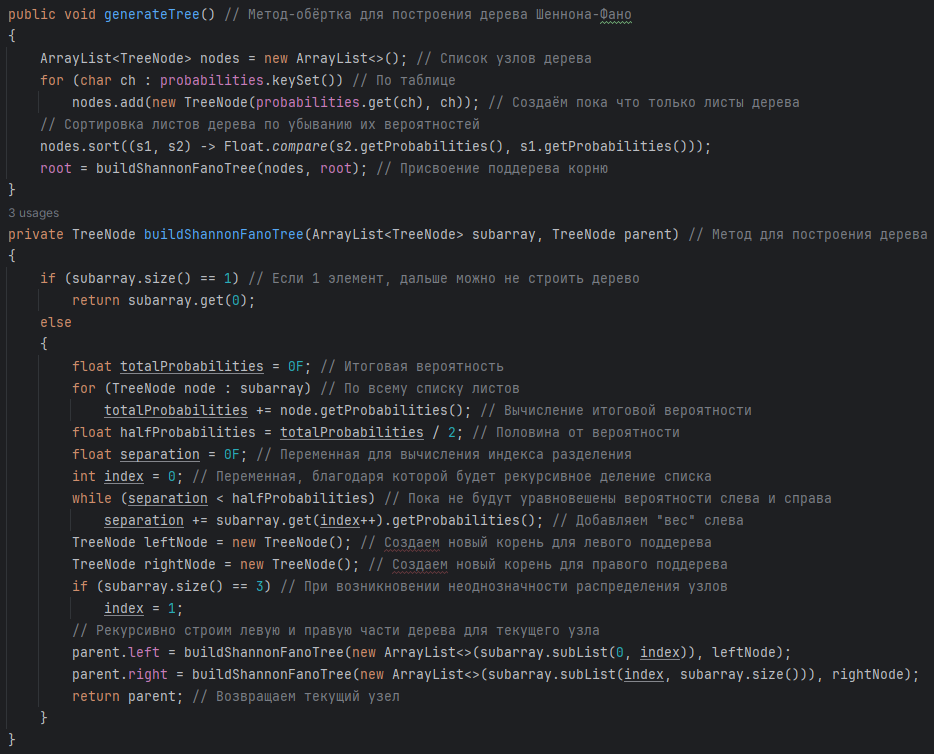
**Метод Шеннона-Фано:**



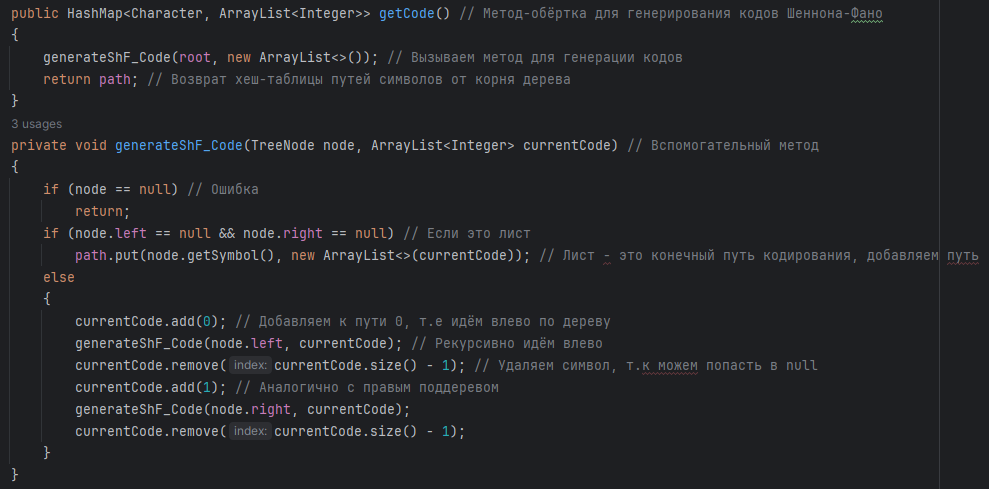
*Листинг 1. Структура узла в дереве Шеннона-Фано.*

**

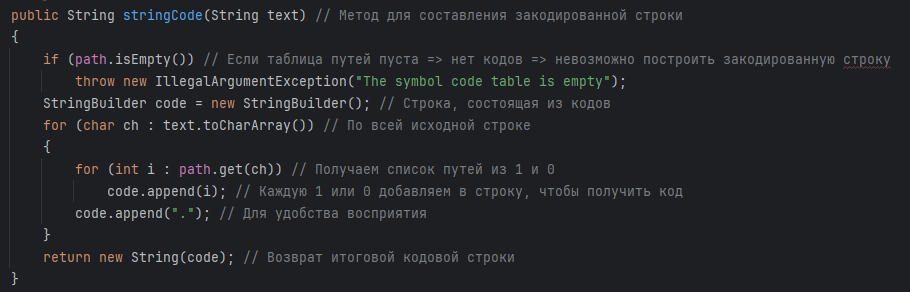
*Листинг 2. Поля и конструктор класса, реализующего сжатие методом Шеннона-Фано.*

**

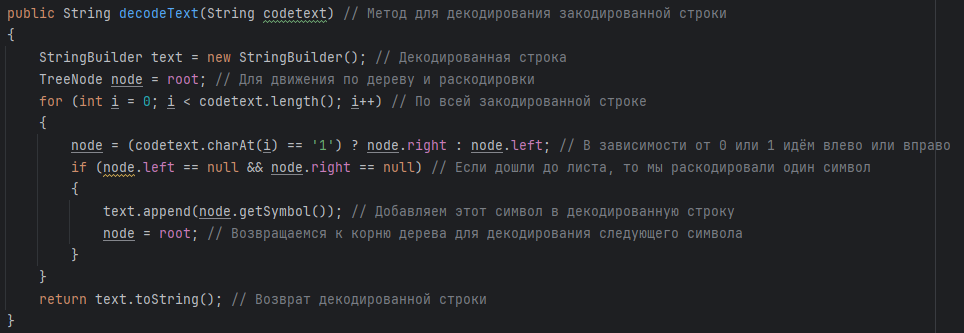
*Листинг 3. Методы для построения дерева Шеннона-Фано по вероятностям.*

**

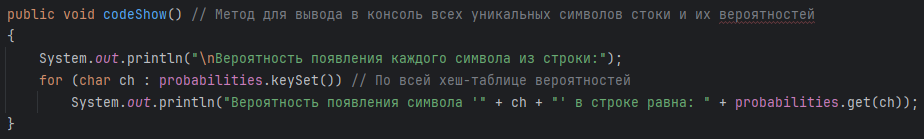
*Листинг 4. Методы для вычисления кодов Шеннона-Фано.*

**

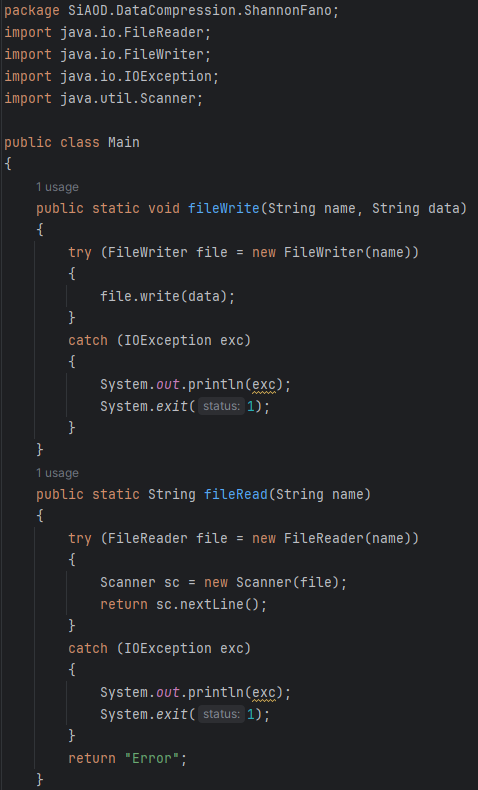
*Листинг 5. Метод, для составления закодированной строки.*

**

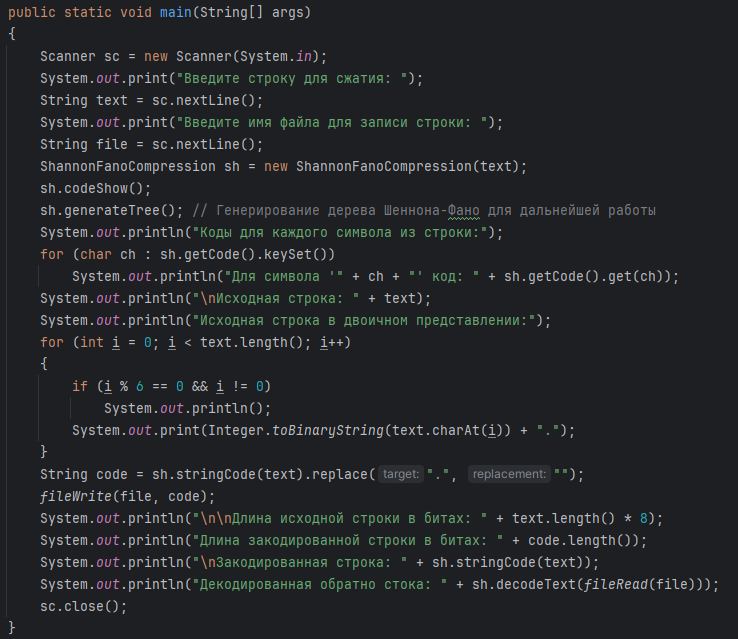
*Листинг 6. Метод для декодирования закодированной последовательности из 0 и 1.*

**

*Листинг 7. Метод для вывода в консоль символов и их вероятностей.*

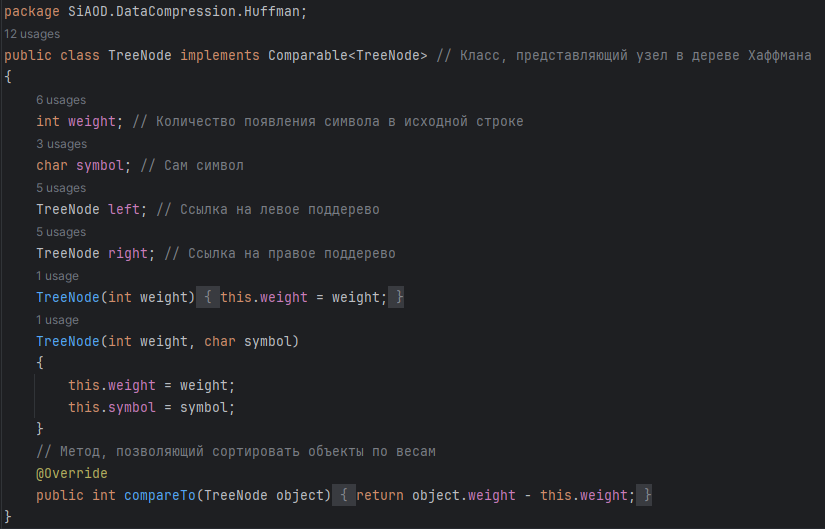
**

*Листинг 8. Методы для записи и чтения из файла строки.*

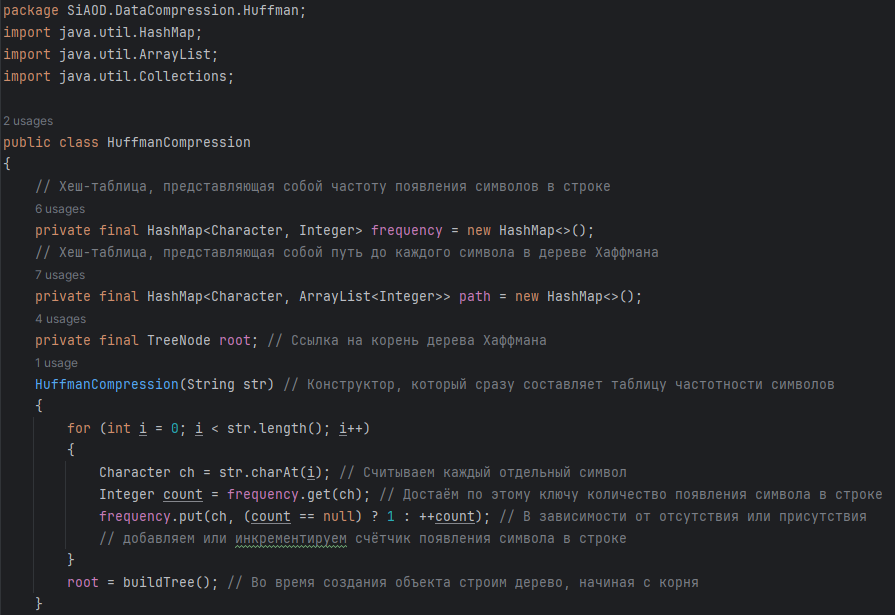
**

*Листинг 9. Основная программа main().*

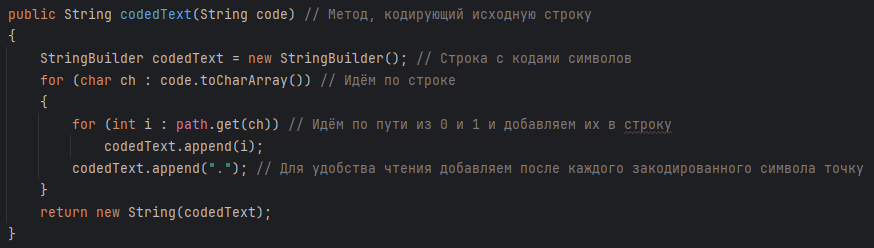
**Метод Хаффмана:**



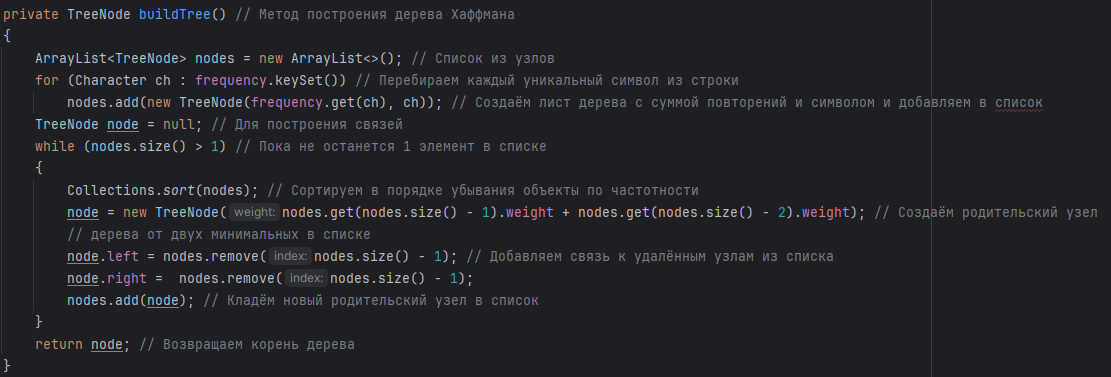
*Листинг 10. Структура узла в дереве Хаффмана.*

**

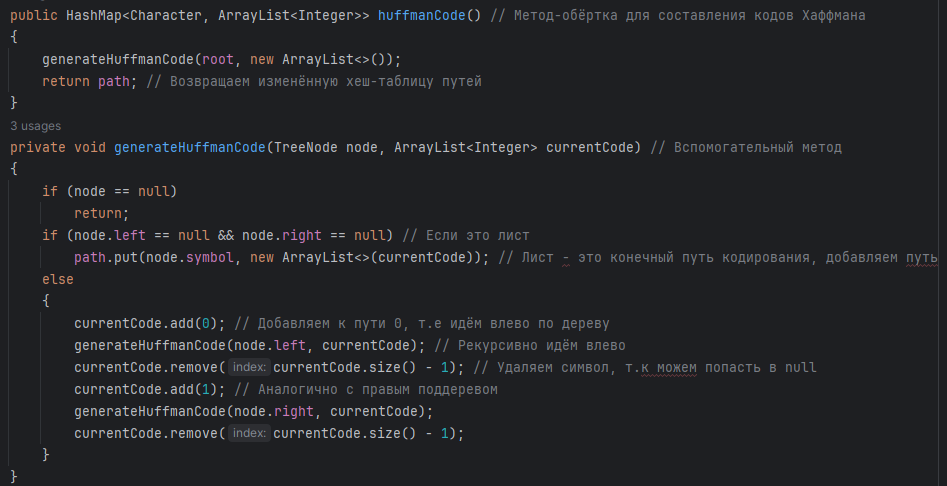
*Листинг 11. Поля и конструктор класса, реализующего сжатие методом Хаффмана.*



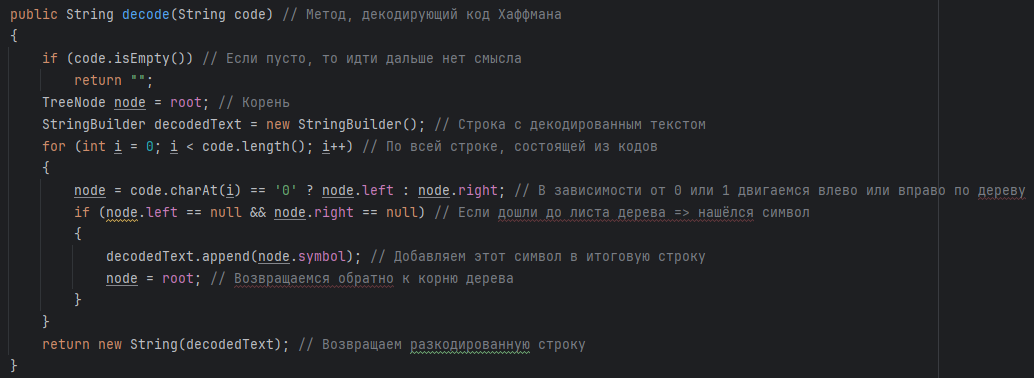
*Листинг 12. Метод, кодирующий введённую строку.*

**

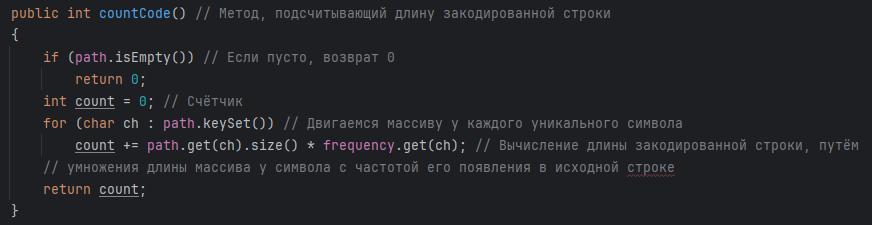
*Листинг 13. Метод построения дерева от листов до корня.*



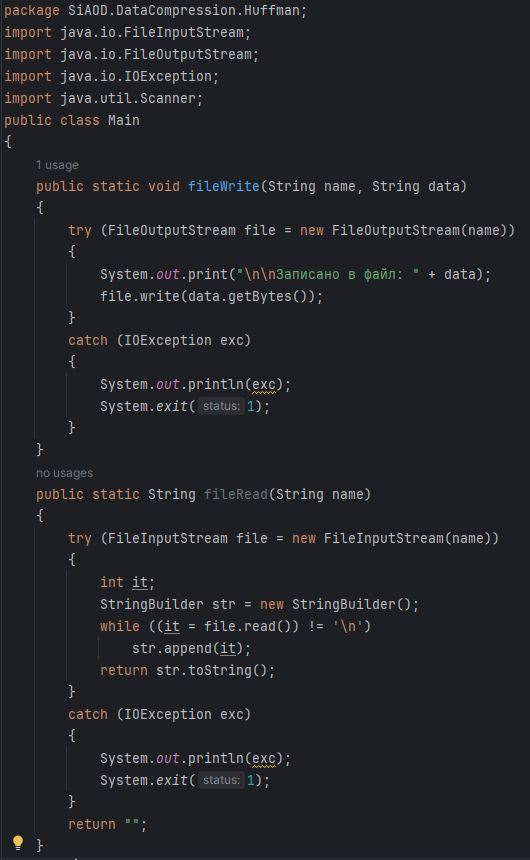
*Листинг 14. Метод, составляющий коды Хаффмана для символов.*

**

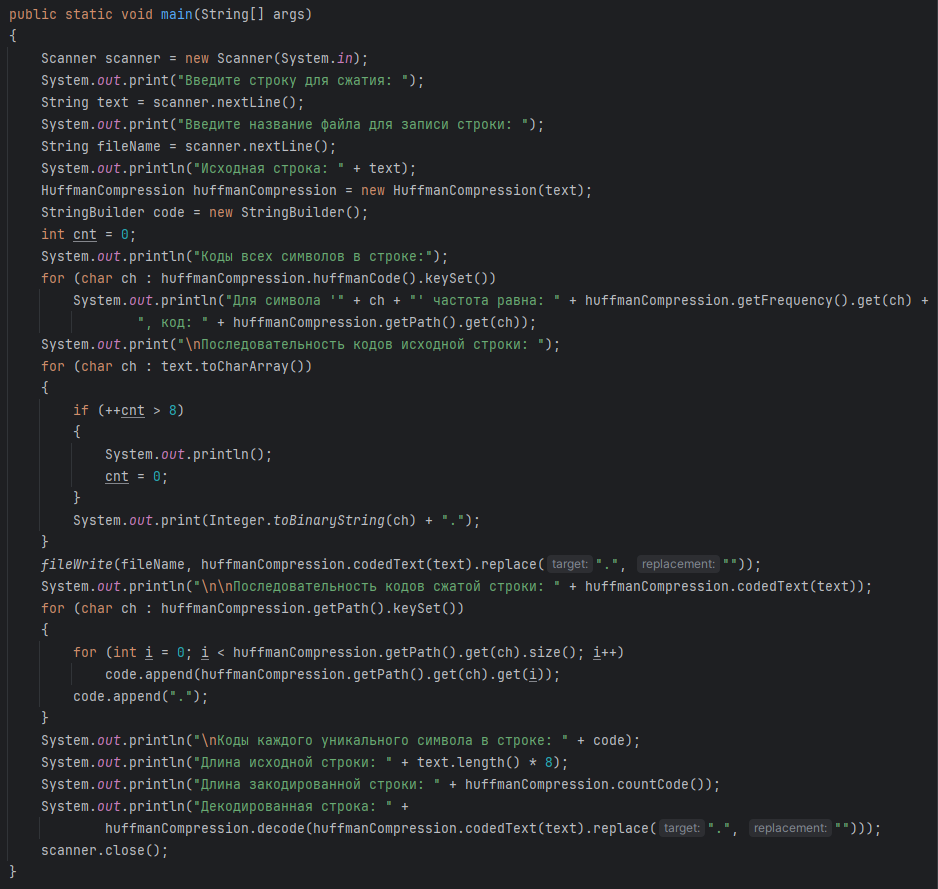
*Листинг 15. Метод, расшифровывающий код Хаффмана.*

**

*Листинг 16. Метод, вычисляющий длину закодированной строки.*



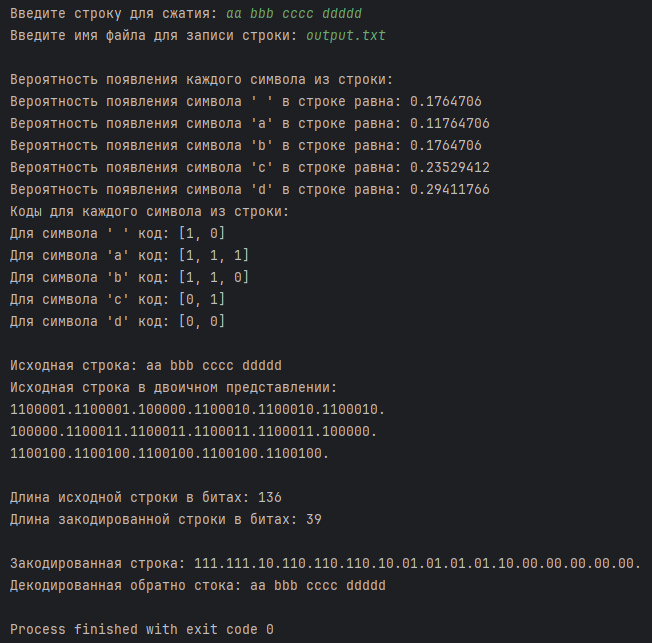
*Листинг 17. Методы для записи и чтения из файла строки.*



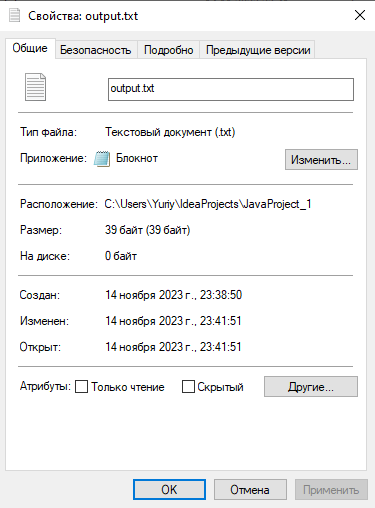
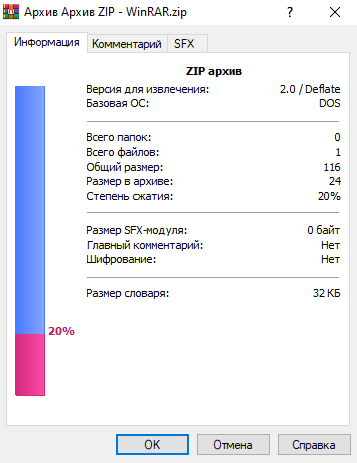
*Листинг 18. Основная программа main().*

## Результаты тестирования.

### **Пример 1:**



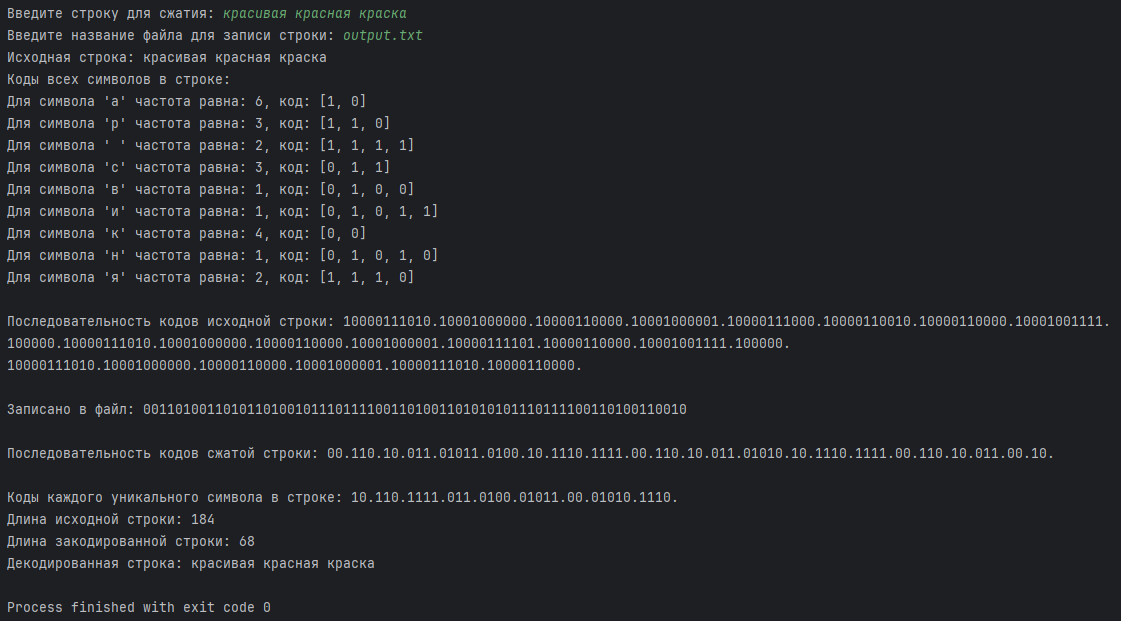
*Рис.3. Результат компрессии и декомпрессии строки методом Шеннона-Фано.*

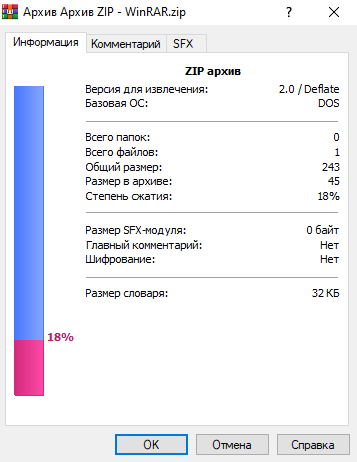
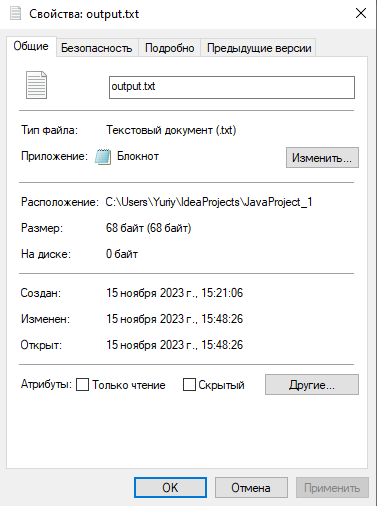
*Рис. 4. Сравнение результата кодирования Шеннона-Фано с результатом архиватора по алгоритму Deflate.*

Можно заметить, что алгоритм Deflate сжал исходный файл почти в два раза сильнее, чем алгоритм Шеннона-Фано (24 байта против 39).

### **Пример 2:**



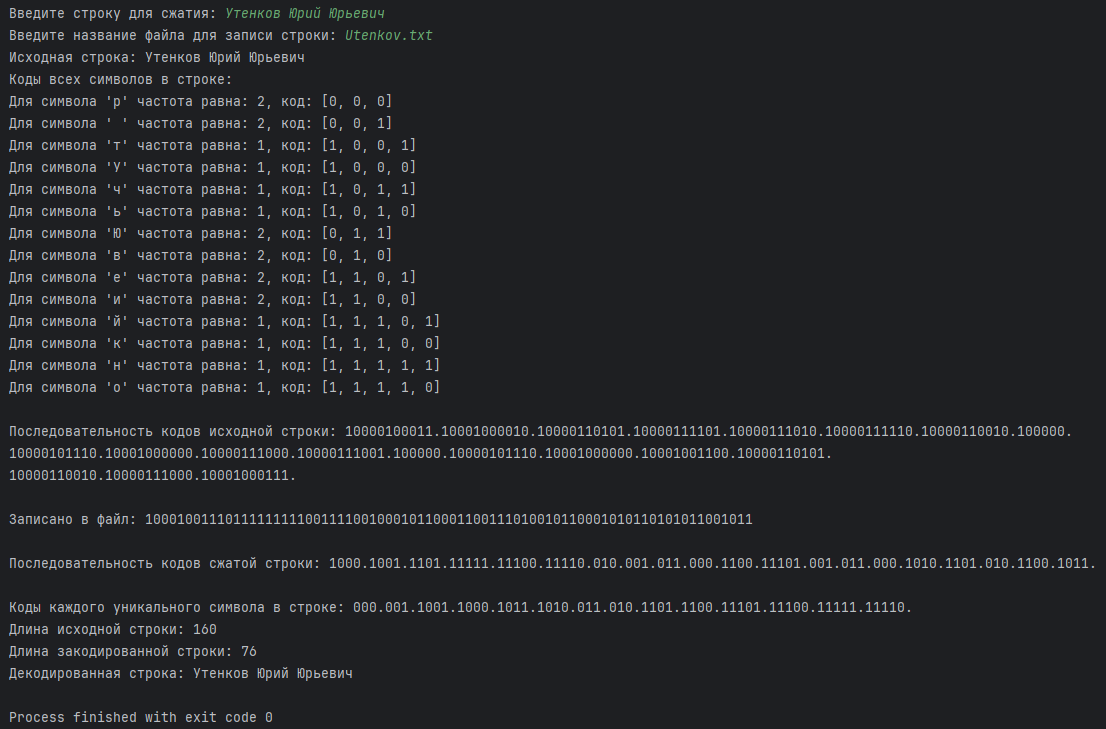
*Рис.5. Результат компрессии и декомпрессии строки методом Хаффмана.*

**

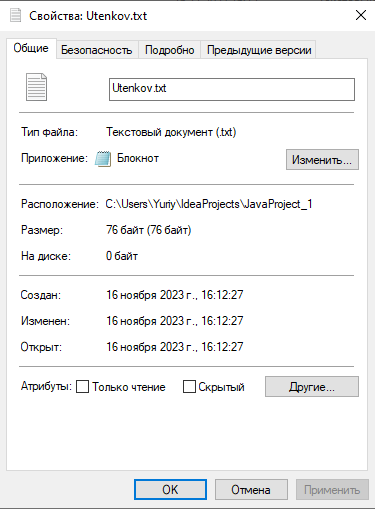
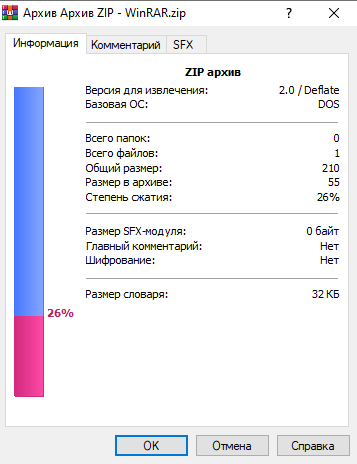
*Рис. 6. Сравнение результата кодирования Хаффмана с результатом архиватора по алгоритму Deflate.*

Можно заметить, что алгоритм Deflate сжал исходный файл примерно в полтора раза сильнее, чем алгоритм Хаффмана (45 байт против 68).

### **Пример 3:**



*Рис.7. Результат компрессии и декомпрессии ФИО методом Хаффмана.*

** **

*Рис. 8. Сравнение результата кодирования Хаффмана с результатом архиватора по алгоритму Deflate.*

Можно заметить, что алгоритм Deflate сжал исходный файл примерно в 1,38 раза сильнее, чем алгоритм Хаффмана (55 байт против 76).

# Выводы.

В ходе выполнения данной работы я получил важные навыки работы с алгоритмами сжатия данных, таких как алгоритм Шеннона-Фано и алгоритм Хаффмана, которые основаны на построении бинарного дерева кодов, и которые повсеместно используются в информационных технологиях. Сжатие данных позволяет существенно сократить расход памяти для хранения информации, поэтому данные алгоритмы очень важны в информационных технологиях.

# Список информационных источников.

1. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих, 2017. – С. 100-126
2. Кораблин Ю.П. Структуры и алгоритмы обработки данных. Часть 1 [Электронный ресурс]: учебно методическое пособие / Ю.П.Кораблин, В.П.Сыромятников, Л.А. Скворцова – М.: РТУ МИРЭА, 2020. – 1 электрон. опт. диск (CD-ROM)
3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. URL:https://online-edu.mirea.ru/course/view.php?id=4020 (дата обращения 20.09.2023).