

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7**  
**дисциплины «Алгоритмизация»**  
**Вариант \_\_\_\_**

Выполнил:  
Репкин Александр Павлович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., канд. техн. наук,  
доцент кафедры инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

## Порядок выполнения работы:

1. Создана программа на основе поставленного задания – необходимо написать программу кодирования и декодирования текста при помощи кодировки Хаффмана (Распределение бинарных кодов между символами так, чтобы длина зашифрованного слова была кратчайшей). Потребовалось добавление отдельного класса node, представляющего из себя узел в бинарном дереве. Элемент типа node хранит в себе значения: переданный ему символ, частота, с которой этот символ встречается и дети данного узла (Тоже тип node, по умолчанию null). Время работы данной программы занимает  $O(n^2)$ .

```
System.out.println("Добрый день! Введите, пожалуйста, требуемый для зашифровки текст:");
String words = input.nextLine(); // next работал неправильно, до пробела, пришлось использовать nextLine.
Map<Character, Integer> characters = new HashMap<>(); // Словарь всех символов и их количества в тексте.
int length = words.length();
// Счёт, сколько раз символ встречается в тексте. merge увеличивает на value (=1), благодаря операции sum.
// Integer::sum - лямбда-выражение.
for (int i = 0; i < length; i++) characters.merge(words.charAt(i), value: 1, Integer::sum);
List<node> nodes = transformation(characters); // Распределение двоичных значений между символами.
for (Map.Entry<Character, String> entry : huffmanCodes.entrySet()) {
    System.out.println("Символ: " + entry.getKey() + ", его двоичный код: " + entry.getValue());
}
System.out.println("Зашифрованный текст:");
StringBuilder binary_text = new StringBuilder();
for (int i = 0; i < words.length(); i++) {
    binary_text.append(huffmanCodes.get(words.charAt(i)));
}
System.out.println(binary_text);
System.out.println("Расшифрованный текст:");
StringBuilder back_to_normal_text = new StringBuilder();
node current = nodes.get(0);
for (int i = 0; i < binary_text.length(); i++) {
    if (binary_text.charAt(i) == '0') { // 0 - Справа, 1 - Слева.
        current = current.first;
    } else {
        current = current.second;
    }
    if (current.first == null && current.second == null) {
        // Если дошло до последнего узла ветки, то добавляем полученный символ в конец строки.
        back_to_normal_text.append(current.data);
        current = nodes.get(0); // Вернулись в начало дерева.
    }
}
```

Рисунок 1. Полученный код программы.

Добрый день! Введите, пожалуйста, требуемый для зашифровки текст:

*Something, somewhere!*

Символ: , его двоичный код: 0000

Символ: !, его двоичный код: 11001

Символ: e, его двоичный код: 111

Символ: g, его двоичный код: 0011

Символ: h, его двоичный код: 100

Символ: i, его двоичный код: 10110

Символ: ,, его двоичный код: 1010

Символ: m, его двоичный код: 1101

Символ: n, его двоичный код: 0100

Символ: o, его двоичный код: 011

Символ: r, его двоичный код: 0101

Символ: S, его двоичный код: 0010

Символ: s, его двоичный код: 10111

Символ: t, его двоичный код: 11000

Символ: w, его двоичный код: 0001

Зашифрованный текст:

00100111101111110001001011001000011101000001011101111011110001100111010111111001

Расшифрованный текст:

*Something, somewhere!*

Рисунок 2. Полученный результат программы.

**Вывод:** в ходе выполнения практической работы был рассмотрен метод Хаффмана, благодаря которому есть возможность распределения бинарных кодов между требуемых символов.