



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших
данных

ОТЧЕТ

по лабораторной работе № 7

Название: Строки. Регулярные выражения.

Дисциплина: Языки программирования для работы с большими
данными

Студент ИУ6-22М

Д. Р. Григорян

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2023

Цель работы:

Научиться работать со строками, писать регулярные выражения для обработки текстов.

Выполнение:

Задача 1.1:

В зависимости от признака (0 или 1) в каждой строке текста удалить указанный символ везде, где он встречается, или вставить его после k-го символа.

Листинг 1 программы main:

```
import java.io.*;
import java.util.*;

//7. В зависимости от признака (0 или 1)
//    в каждой строке текста удалить указанный символ везде, где он
//    встречается,
//    или вставить его после k-го символа
public class Main {
    public static void main(String[] args) throws IOException {
        var in = new Scanner(System.in);
        int num = 0;
        while (true) {
            System.out.println("Input a feature: 0 or 1");
            num = in.nextInt();
            if ((num == 0) || (num == 1)) {
                break;
            }
        }
        System.out.println("Input a letter for deletion or insertion");
        var letter = in.next().charAt(0);
        if (num == 0) {
            deletion(letter);
            System.out.println("Deletion is done!");
        } else {
            System.out.println("Input a position for insertion");
            var pos = in.nextInt();
            insertion(pos, letter);
        }
    }

    public static void deletion(char Char) throws IOException {
        var file = new File("text.txt");
        var file2 = new File("Result.txt");
        var fr = new FileReader(file);
        var reader = new BufferedReader(fr);

        var bf = new BufferedWriter(new FileWriter(file2));
        String line = reader.readLine();
        while (line != null) {
            bf.write(line.trim().replaceAll(String.valueOf(Char), ""));
            bf.newLine();
            bf.flush();
            line = reader.readLine();
        }
    }
}
```

```

    }
    public static void insertion(int sign,char Char) throws IOException {
        var file = new File("text.txt");
        var file2 = new File("Result.txt");
        var fr = new FileReader(file);
        var reader = new BufferedReader(fr);

        var bf = new BufferedWriter(new FileWriter(file2));
        String line = reader.readLine();
        while (line != null) {
            var str = line.substring(0, sign) + Char + line.substring(sign +
1);

            bf.write(str);
            bf.newLine();
            bf.flush();
            line = reader.readLine();
        }
    }
}

```

Результат приведен на рисунках 1-2:

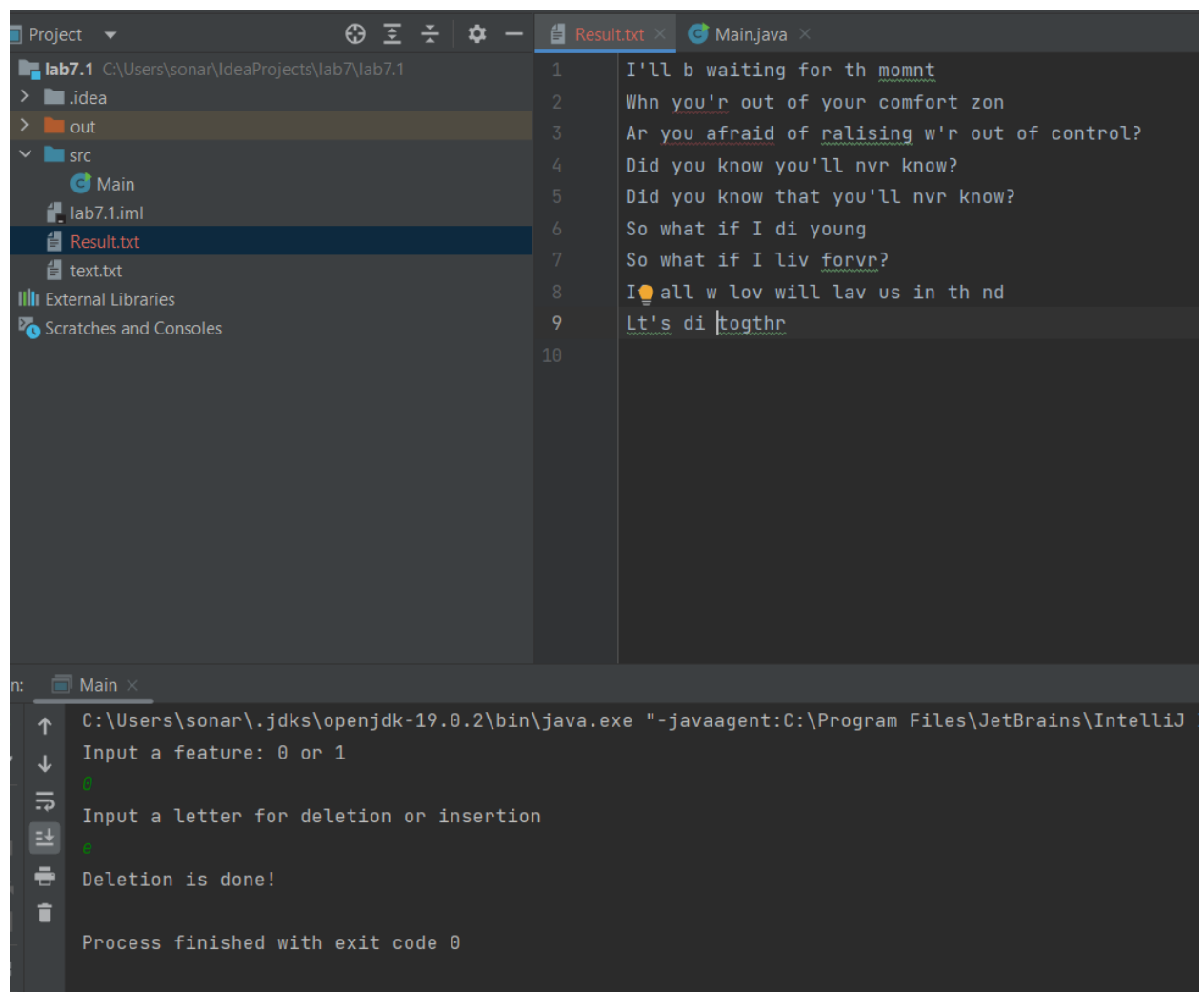


Рисунок 1 – Текст после удаления

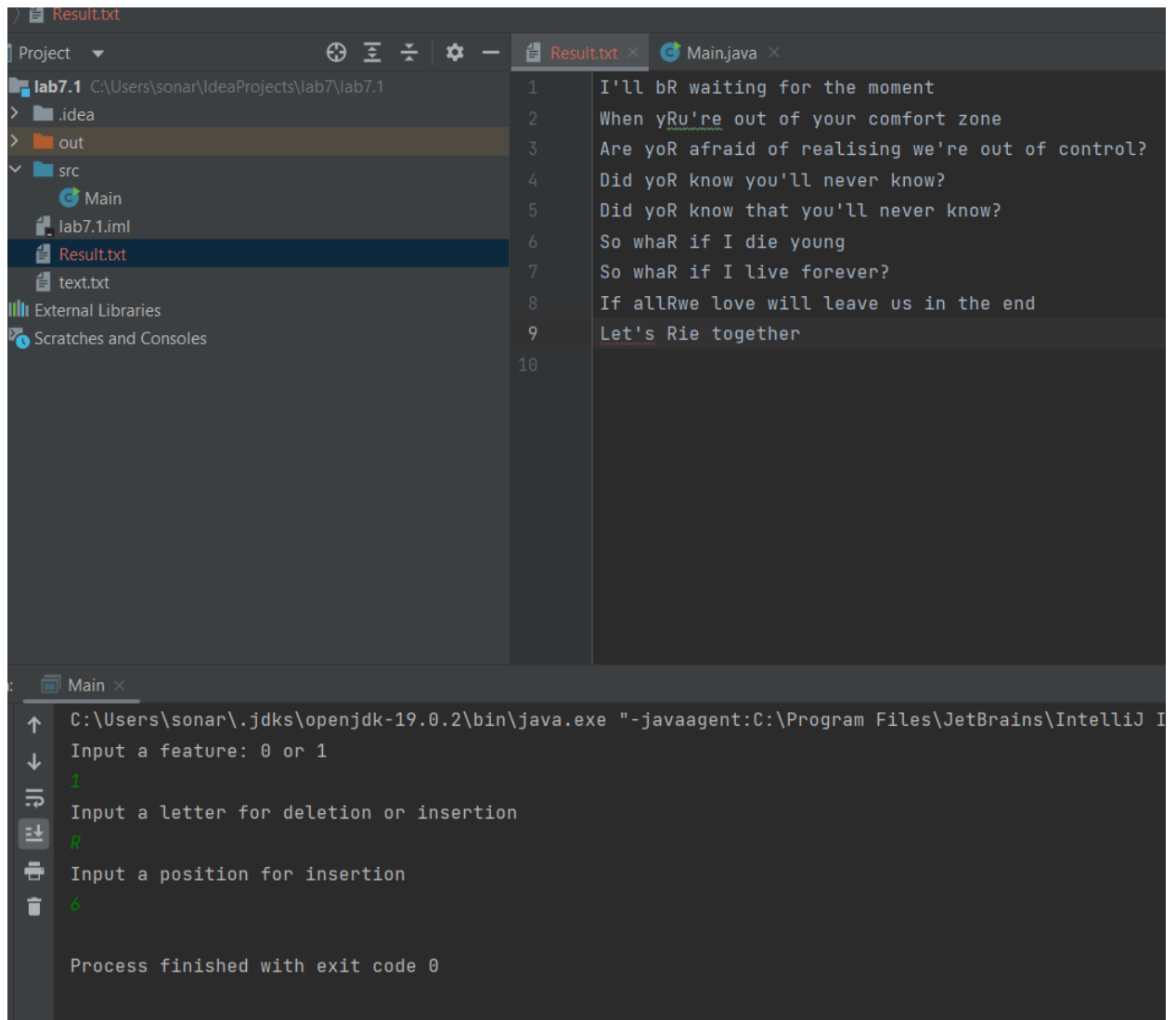


Рисунок 2 – Текст после вставки символа

Задача 1.2:

Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.

Листинг 2 программы main:

```

import java.io.*;
import java.util.*;

//8. Из небольшого текста удалить все символы, кроме пробелов, не
являющиеся буквами.
// Между последовательностями подряд идущих букв оставить хотя бы один
пробел.
public class Main {
    public static void main(String[] args) throws IOException {
        var file = new File("text2.txt");
        var file2 = new File("Result2.txt");
        var fr = new FileReader(file);
        var reader = new BufferedReader(fr);

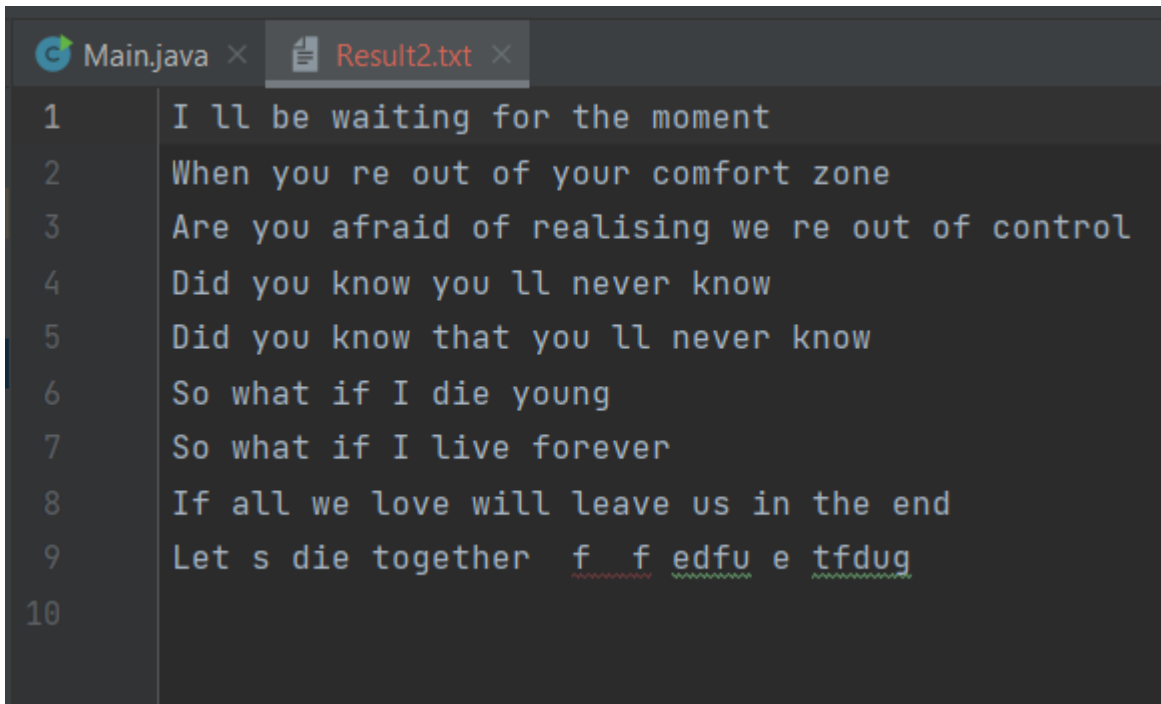
```

```

var bf = new BufferedWriter(new FileWriter(file2));
String line = reader.readLine();
var regex = "(?<=[a-zA-z])[^[a-zA-Z]&&[^\ ]]+(?=[a-zA-z])";
var regex1 = "[^[a-zA-Z]&&[^\ ]]+";
while (line != null) {
    var str = line.trim().replaceAll(regex, " ");
    bf.write(str.replaceAll(regex1, ""));
    bf.newLine();
    bf.flush();
    line = reader.readLine();
}
System.out.println("The Deletion is done!");
}
}

```

Результат приведен на рисунке 3:



```

Main.java x Result2.txt x
1 I ll be waiting for the moment
2 When you re out of your comfort zone
3 Are you afraid of realising we re out of control
4 Did you know you ll never know
5 Did you know that you ll never know
6 So what if I die young
7 So what if I live forever
8 If all we love will leave us in the end
9 Let s die together f f edfu e tfdug
10

```

Рисунок 3 – Результирующий текст после обработки

Задача 2.1:

В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

Листинг 3 программы main:

```

import java.io.*;
import java.util.*;
//В тексте найти и напечатать все слова максимальной и все слова минимальной
длинны.
public class Main {
    public static void main(String[] args) throws IOException {
        var file = new File("text3.txt");
        var fr = new FileReader(file);
        var reader = new BufferedReader(fr);

        ArrayList<String> list = new ArrayList<>();

```

```

String line = reader.readLine();
while (line != null) {
    for (String element: line.split("\\s")) {
        list.add(element);
    }
    line = reader.readLine();
}
System.out.println("The longest words");
System.out.println(LongestWords(list));
System.out.println("The smallest words");
System.out.println(SmallestWords(list));
}
public static ArrayList<String> LongestWords(ArrayList listofwords) {
    ArrayList<String> listfinalBIG = new ArrayList<>();
    int longest_length = 0;
    for (Object str : listofwords) {
        int length = str.toString().length();
        if (length > longest_length) {
            longest_length = length;
            listfinalBIG.clear();
        }
        if (length == longest_length) {
            listfinalBIG.add(str.toString());
        }
    }
    return listfinalBIG;
}
public static ArrayList<String> SmallestWords(ArrayList listofwords) {
    ArrayList<String> listfinalSMALL = new ArrayList<>();
    int smallest_length = 100;
    for (Object str : listofwords) {
        int length = str.toString().length();
        if (length < smallest_length) {
            smallest_length = length;
            listfinalSMALL.clear();
        }
        if (length == smallest_length) {
            listfinalSMALL.add(str.toString());
        }
    }
    return listfinalSMALL;
}
}

```

Результат приведен на рисунке 4:

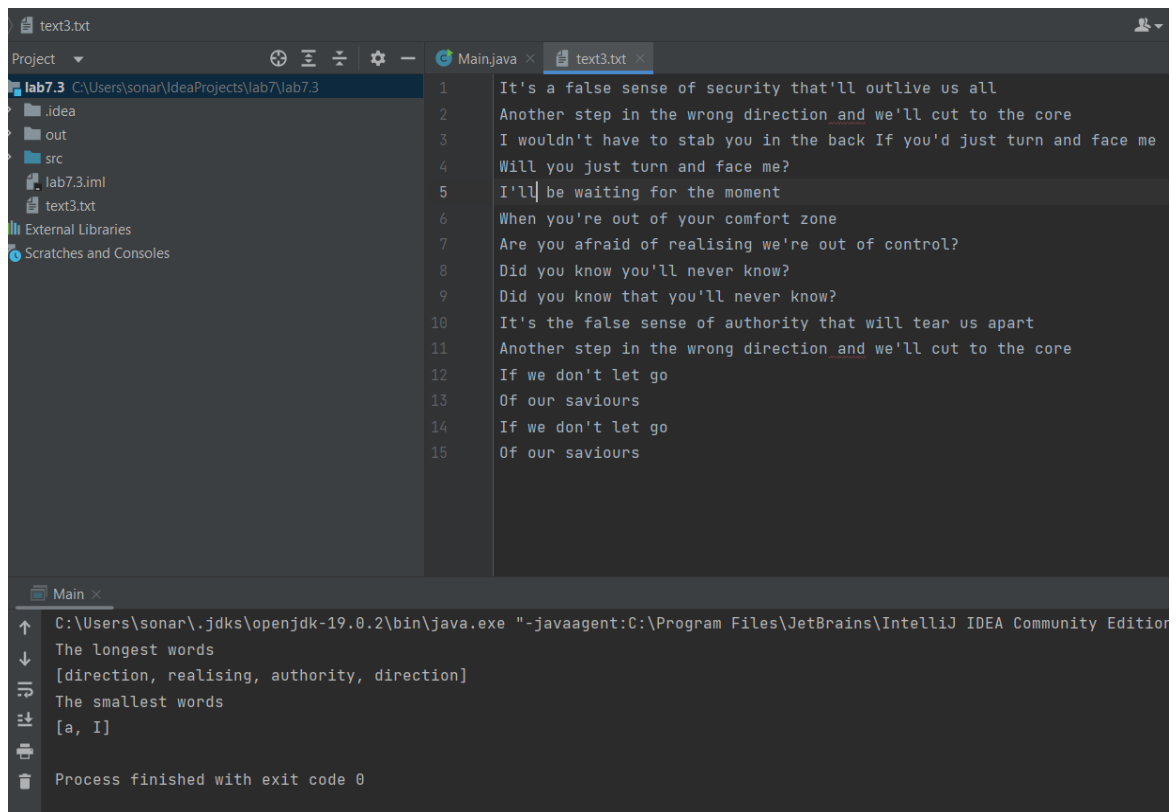


Рисунок 4 – Результат выполнения программы

Задача 2.2:

Напечатать квитанцию об оплате телеграммы, если стоимость одного слова задана.

Листинг 4 программы main:

```

import java.io.*;
import java.util.*;

//Напечатать квитанцию об оплате телеграммы, если стоимость одного слова
задана.

public class Main {
    public static void main(String[] args) throws IOException {
        var file = new File("text4.txt");
        var fr = new FileReader(file);
        var reader = new BufferedReader(fr);

        ArrayList<String> list = new ArrayList<>();
        String line = reader.readLine();
        while (line != null) {
            for (String element: line.split("\\s")) {
                list.add(element);
            }
            line = reader.readLine();
        }
        int price = 12;
        System.out.println("The price of one word is 12 dollars");
        System.out.print("The final price of your telegram is: ");
        System.out.println(PriceTelegram(list, price));
    }
}

```

```

    public static int PriceTelegram(ArrayList listofwords,int cost) {
        int price = listofwords.size()*cost;
        return price;
    }
}

```

Результат приведен на рисунке 5:

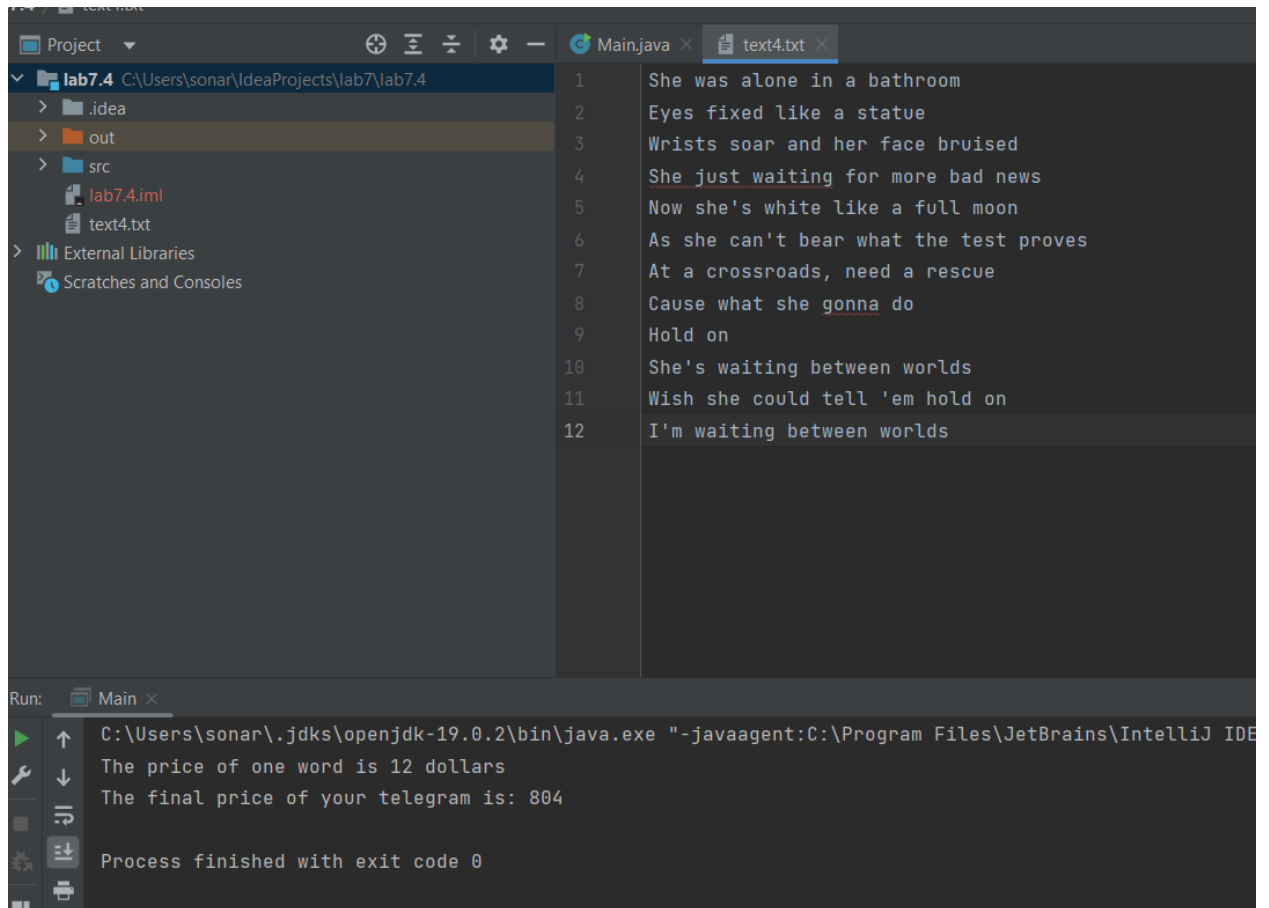


Рисунок 5 – Результат выполнения программы

Задача 3.1-3.2:

Текст шифруется по следующему правилу: из исходного текста выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы, затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д. Зашифровать заданный текст.

На основании правила кодирования, описанного в предыдущем примере, расшифровать заданный набор символов.

Листинг 5 программы main:

```

import java.io.*;
import java.util.*;

//7. Текст шифруется по следующему правилу: из исходного текста выбирается

```



```

1, 4, 7, 10-й и т.д. (до конца текста) символы,
// затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й
и т.д.
// Зашифровать заданный текст

public class Main {
    public static void main(String[] args) throws IOException {
        var file = new File("Coder.txt");
        var reader = new BufferedReader(new FileReader(file));

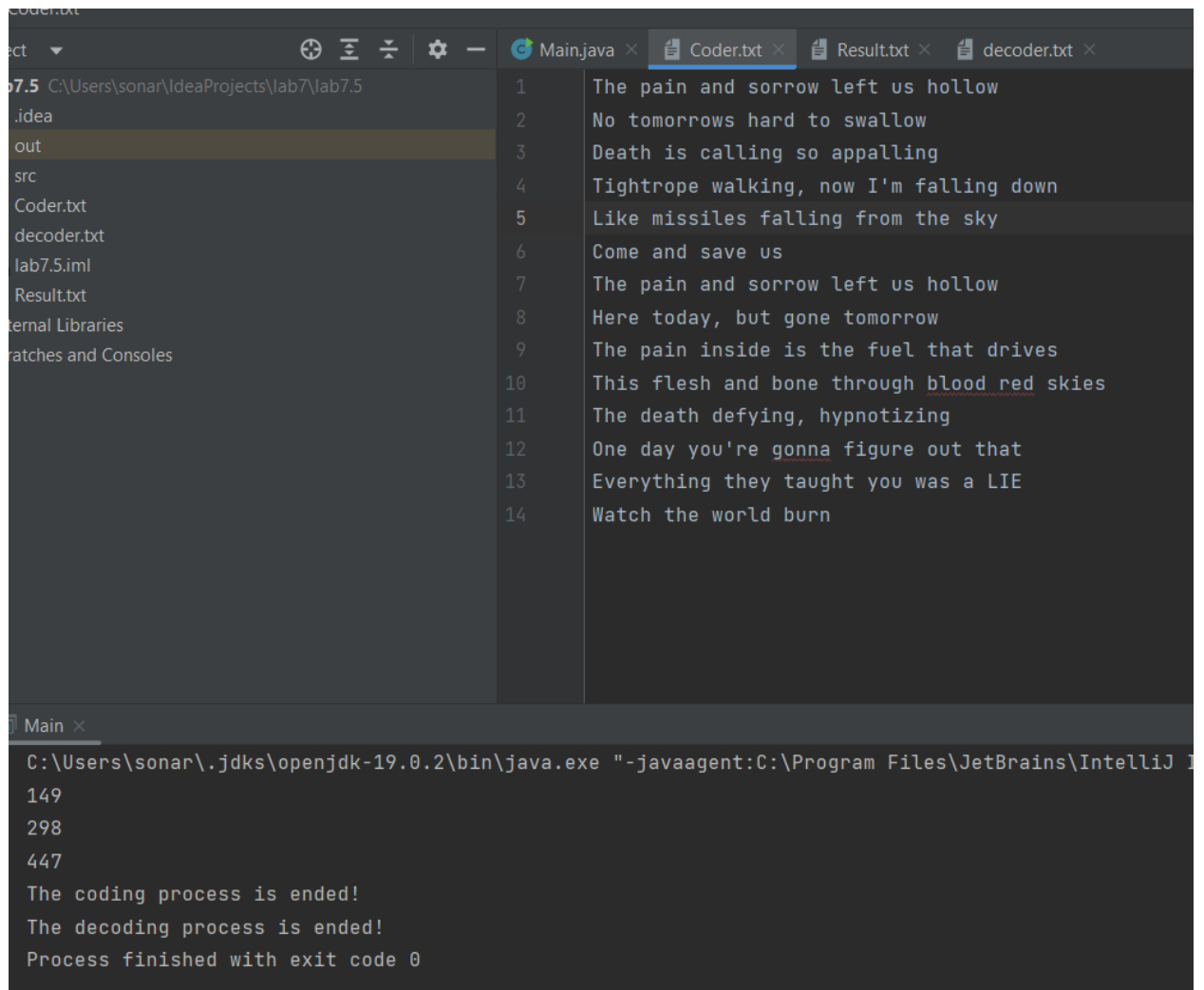
        String line = reader.readLine();
        String Finall="";
        while (line != null) {
            Finall = Finall.concat(line);
            // Finall = Finall.concat(" ");
            line = reader.readLine();
        }
        int dif = coding(Finall);
        System.out.println("The coding process is ended!");
        reader.close();
        decoding(dif);
        System.out.print("The decoding process is ended!");
    }
    public static int coding(String listoflines) throws IOException{
        var file2 = new File("Result.txt");
        var bf = new BufferedWriter(new FileWriter(file2));
        var array = listoflines.toCharArray();
        int l=0;
        for (int i=0;i<3;i++) {
            for (int k=0;k<array.length;k+=3) {
                bf.write(array[k]);
                bf.flush();
                l++;
            }
            System.out.println(l);
            array = Arrays.copyOfRange(array, 1, array.length);
        }
        bf.close();
        return l/3;
    }
    public static void decoding(int step) throws IOException{
        var file = new File("Result.txt");
        var reader = new BufferedReader(new FileReader(file));

        var file2 = new File("decoder.txt");
        var bf = new BufferedWriter(new FileWriter(file2));

        String line = reader.readLine();
        var array1 = line.toCharArray();
        for (int k=0;k<step;k++) {
            for (int i=0;i<3;i++) {
                if (step*i+k>= array1.length) {
                    break;
                }
                bf.write(array1[step*i+k]);
                bf.flush();
            }
        }
        bf.close();
        reader.close();
        return;
    }
}

```

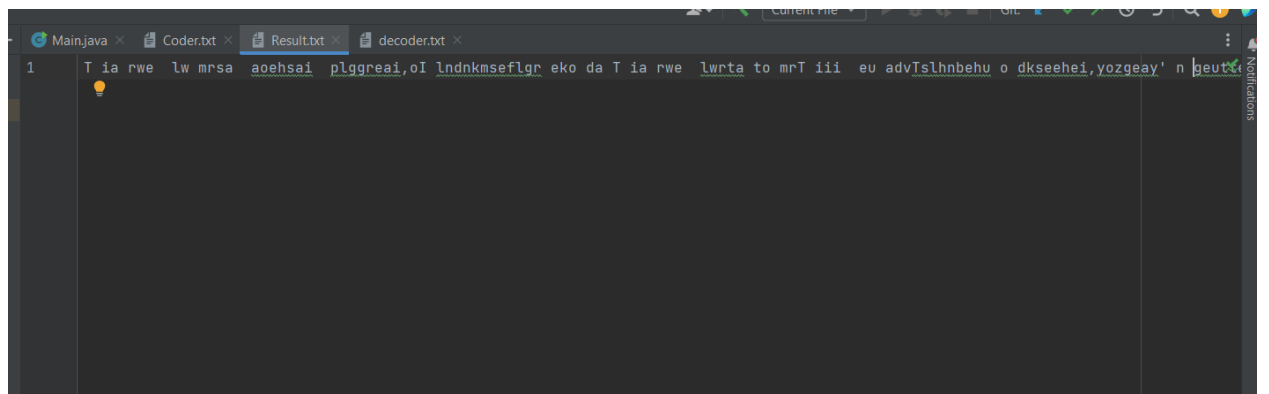
Результат приведен на рисунках 6-8:



```
1 The pain and sorrow left us hollow
2 No tomorrows hard to swallow
3 Death is calling so appalling
4 Tightrope walking, now I'm falling down
5 Like missiles falling from the sky
6 Come and save us
7 The pain and sorrow left us hollow
8 Here today, but gone tomorrow
9 The pain inside is the fuel that drives
10 This flesh and bone through blood red skies
11 The death defying, hypnotizing
12 One day you're gonna figure out that
13 Everything they taught you was a LIE
14 Watch the world burn
```

```
C:\Users\sonar\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ I
149
298
447
The coding process is ended!
The decoding process is ended!
Process finished with exit code 0
```

Рисунок 6 – Текст введенный на кодировку



```
1 T ia rwe lw mrsa aoehsai plggreai,oI lndnkmseflgr eko da T ia rwe lwrta to mrT iiii eu advIslhnbhu o dkseehei,yozgeay' n beut
```

Рисунок 7 – Кодировка

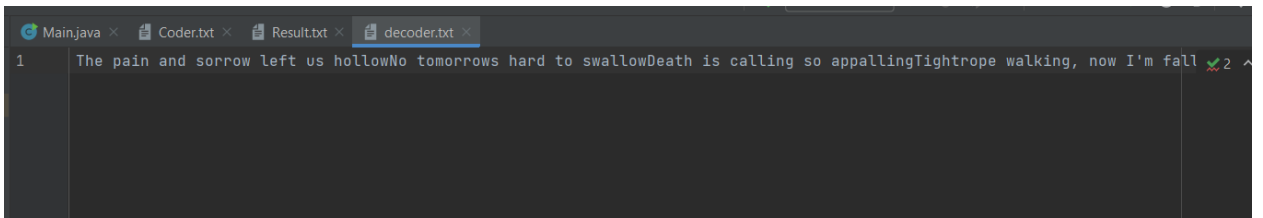


Рисунок 8 – Декодировка

Задача 4.1-4.2:

Подсчитать, сколько слов в заданном тексте начинается с прописной буквы

Подсчитать, сколько раз заданное слово входит в текст.

Листинг 6 программы main:

```
import java.io.*;
import java.util.*;

//7. Подсчитать, сколько слов в заданном тексте начинается с прописной
буквы.
//8. Подсчитать, сколько раз заданное слово входит в текст.
public class Main {
    public static void main(String[] args) throws IOException {
        var file = new File("text7.txt");
        var reader = new BufferedReader(new FileReader(file));

        ArrayList<String> list = new ArrayList<>();
        String line = reader.readLine();
        while (line != null) {
            for (String element: line.split("\\s")) {
                list.add(element);
            }
            line = reader.readLine();
        }
        reader.close();
        System.out.println("The count of the lowercase words");
        System.out.println(lowercase(list));
        System.out.println("Input the main word");
        Scanner in = new Scanner(System.in);
        String str1 = in.nextLine();
        System.out.printf("The count of the main word in the text: ",str1);
        System.out.println(entryword(list,str1));
    }
    public static int lowercase(ArrayList listofwords) {
        int count = 0;
        for (Object str : listofwords) {
            char[] array=str.toString().toCharArray();
            if(Character.isLowerCase(array[0])) {
                count++;
            }
        }
        return count;
    }
    public static int entryword(ArrayList listofwords,String word) {
        int count = 0;
        for (Object str : listofwords) {
            if (str.toString().equals(word)) {
                count++;
            }
        }
    }
}
```

```

    }
    }
    return count;
}
}

```

Результат приведен на рисунке 9:

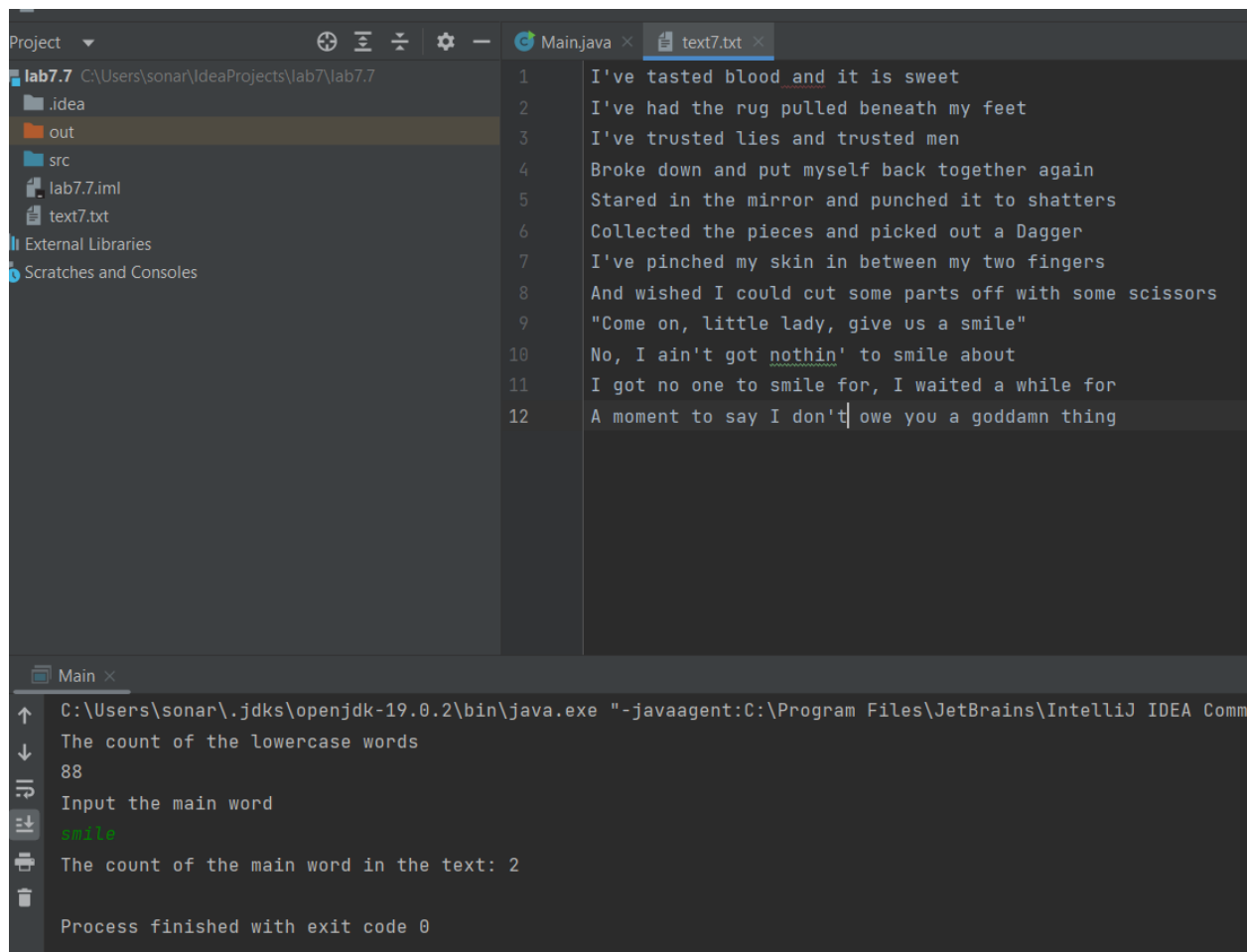


Рисунок 9 – Результат выполнения программы

Вывод: в ходе выполнения лабораторной работы были написаны программы согласно выданному заданию, выполнены функции:

- подсчета количества слов в заданном тексте;
- подсчета повторений определенного слова в заданный текст;
- шифрования текста со смещением на 3 символа вправо и дешифрования;
- поиска слов максимальной и минимальной длины;
- удаления всех символов, кроме пробелов.