

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
Лабораторная работа №2

Выполнила:
Студентка 4 курса
Группы АС-50
Клиническая Р.П.
Проверил:
Крощенко А.А.

Брест 2020

Цель работы:

приобрести базовые навыки работы с файловой системой в Java.

Задание. Вариант 5.

Напишите программу, которая считывает текст и печатает таблицу, показывающую, сколько раз в этом тексте встречаются однобуквенные слова, двухбуквенные слова, трехбуквенные слова и т.д.

Код программы:

```
import java.io.*;
import java.util.*;
public class Task1 {
    public static void main(String[] args) {

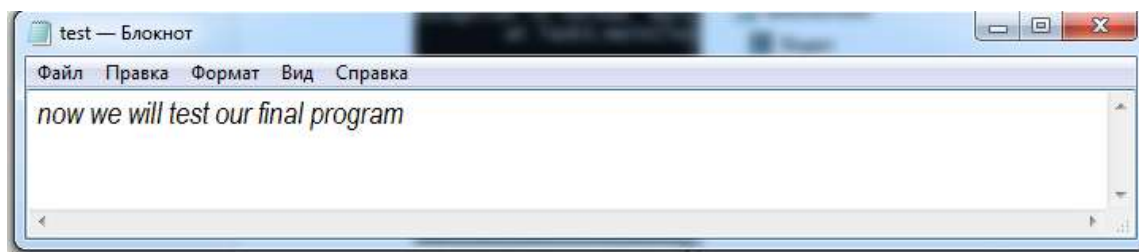
        //пытаемся считать файл, имя которого введено с вызовом программы
        try(FileReader file = new FileReader(args[0])) {

            //если удалось найти файл, то определяем объект класса reader для
            чтения из файла
            BufferedReader reader = new BufferedReader(file);
            //считывать из файла будем посимвольно
            char probel = ' ';
            int ch, length = 0;
            int count_of_words[] = new int[255];
            //пока не добрались до конца файла считываем по символу
            while((ch=reader.read())!=-1){
                System.out.print((char)ch);
                length++;
                //словили новое слово
                if((char)ch == probel){
                    length--;
                    count_of_words[length]++;
                    length = 0;
                }
            }
            System.out.println("");
            for(int i = 0; i < count_of_words.length; i++){
                if(count_of_words[i] != 0){
                    System.out.println("Count words witch have " + i +
                    "letters = " + count_of_words[i]);
                }
            }

            //если считать файл не удалось выводим сообщение ошибки
            catch(IOException ex){
                System.out.println(ex.getMessage());
            }
        }
    }
}
```

Скриншоты:

```
G:\SSP\var5>cd Lab2_var5_task1
G:\SSP\var5\Lab2_var5_task1>java Task1
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
    at Task1.main(Task1.java:8)
G:\SSP\var5\Lab2_var5_task1>java Task1 test.txt
now we will test our final program
Count words witch have 2letters = 1
Count words witch have 3letters = 2
Count words witch have 4letters = 2
Count words witch have 5letters = 1
Count words witch have 7letters = 1
G:\SSP\var5\Lab2_var5_task1>
```



Задание 2.

Утилита `ср` осуществляет копирование файла из одного каталога в другой.
Исходный файл

остаётся неизменным, имя созданного файла может быть таким же, как у исходного, или измениться. Формат использования: `ср [-f][-i][-n] исходный_файл целевой_файл`

- `-f` Разрешает удаление целевого файла, в который производится копирование, если он не может быть открыт для записи.
- `-i` Утилита будет запрашивать, следует ли перезаписывать конечный файл, имя которого

совпадает с именем исходного. Для того, чтобы перезаписать файл, следует ввести `y` или его эквивалент. Ввод любого другого символа приведёт к отмене перезаписи данного файла.

- `-n` Не перезаписывать существующий файл (отменяет предыдущий параметр `-i`).

Пример использования: `ср -fn src.txt dest.txt` Копирует содержимое из `src.txt` в `dest.txt` с ключами `-f` и `-n`.

Код программы:

```
//package com.company;
import java.io.*;
import java.util.ArrayList;
import java.util.List;
public class task2 {
    public static void main(String[] args) {
```

```

try(FileReader file = new FileReader(args[2])) {
    BufferedReader reader = new BufferedReader(file);
    //проверяем есть ли вообще аргументы и является ли первый аргумент - ср
    if (args.length == 0 || !args[0].equals("ср")) {
        System.out.println("Incorrect input");
        return;
    }

    //проверим ввел ли пользователь ключи
    char[] flag_check = args[1].toCharArray();
    boolean f_f = false, f_i = false, f_n = false;
    if (flag_check[0] == '-') {
        //если ввел, то узнаем какие именно
        for (int i = 1; i < flag_check.length; i++) {
            //при нахождении соответствующего ключа его значение
устанавливается как истина
            switch (flag_check[i]) {
                case 'f': {
                    f_f = true;
                    break;
                }
                case 'i': {
                    f_i = true;
                    break;
                }
                case 'n': {
                    f_n = true;
                    break;
                }
            }
        }
    }

    //теперь у нас есть булевы значения на каждый ключ и мы можем начать
работу с файлами
    File fileout = new File(args[3]);
    //если файла не было, то создаем
    if(!fileout.exists()){
        fileout.createNewFile();
    }
    //если нельзя записать и удаление разрешено, то удаляем файл
    if(f_f && !fileout.canWrite()){
        fileout.delete();
    }
    //если пользователь готов перезаписать файл и поставил ключ n, то
запрашиваем разрешение
    if(f_i != false && f_n != true){
        System.out.println("If you want to rewrite the file enter 1");
        int user_answer = System.in.read();
    }
}

```

```

//если он согласен, то ставить флаг append значение false
if(user_answer == 1){
    try(FileWriter writeInFile = new FileWriter(fileout, false)){
        //теперь можно начать запись в файл из исходного файла
        int ch;
        while((ch = reader.read()) != -1){
            System.out.print((char)ch);
            writeInFile.append((char)ch);
        }
    }
    catch(IOException ex){
        System.out.println(ex.getMessage());
    }
}
//иначе ставим флаг true и просто дописываем текс в конец исходного файла
else
    try(FileWriter writeInFile = new FileWriter(fileout, true)) {
        //теперь можно начать запись в файл из исходного файла
        int ch;
        while ((ch = reader.read()) != -1) {
            System.out.print((char) ch);
            writeInFile.append((char) ch);
        }
    }
    catch(IOException ex){
        System.out.println(ex.getMessage());
    }
}
catch(IOException ex){
    System.out.println(ex.getMessage());
}
}
}

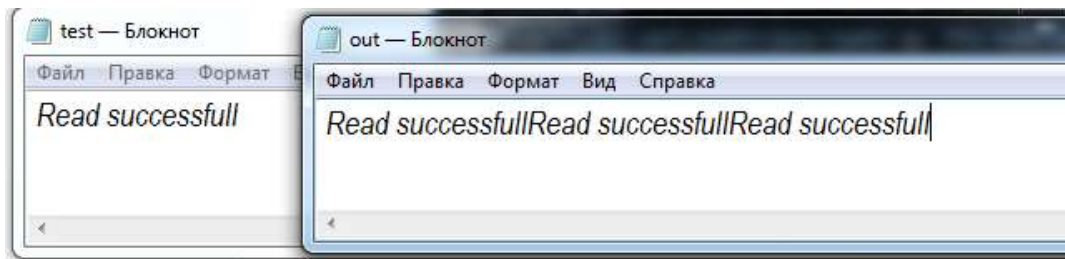
```

Скриншоты:

```

G:\SSP\var5\Lab2_var5_task2>java task2 cp -fin test.txt out.txt
Read successful
G:\SSP\var5\Lab2_var5_task2>java task2 cp -fin test.txt out.txt
Read successful
G:\SSP\var5\Lab2_var5_task2>java task2 cp -fin test.txt out.txt
Read successful
G:\SSP\var5\Lab2_var5_task2>

```



Вывод: в ходе лабораторной работы ознакомилась со способами работы с файлами: записью в файл и чтением из файла в java.