

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

ОТЧЁТ  
по лабораторной работе №2

Выполнил:  
студент 3 курса  
группы ПО-8  
Дорошков А.Д.

Проверил:  
Крощенко А.А.

**Цель:** приобрести базовые навыки работы с файловой системой в Java

## Задание 1

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

Входные данные:

```
1  Один из самых простых способов — использовать сочетание BufferedReader и FileReader.
2  FileReader открывает поток для чтения из файла, а BufferedReader считывает текст из этого потока.
3  Это позволяет читать данные построчно, что удобно для большинства текстовых файлов.
```

Один из самых простых способов — использовать сочетание BufferedReader и FileReader.

FileReader открывает поток для чтения из файла, а BufferedReader считывает текст из этого потока.

Это позволяет читать данные построчно, что удобно для большинства текстовых файлов.

Выходные данные:

```
Длина слова | Количество
-----
1           | 2
2           | 3
3           | 4
4           | 1
5           | 5
6           | 6
7           | 1
8           | 1
9           | 6
10          | 2
11          | 1
12          | 1
14          | 2
```

Краткий текст программы:

```
String buff;
StringBuilder lineBuilder = new StringBuilder();
String punctuationMarks = "[!—?::',./,.]";
while ((buff = reader.readLine()) != null) {
    System.out.println(buff);
    lineBuilder.append(buff);
}
String line = lineBuilder.toString().replaceAll(punctuationMarks, " ");
System.out.println(line);
```

```
String[] words = line.split("\\s+");
int countMarks;
Map<Integer, Integer> map = new HashMap<>();
for (String word : words){
    countMarks = word.length();
    if (map.containsKey(countMarks)) {
        map.put(countMarks, map.get(countMarks) + 1);
    } else {
        map.put(countMarks, 1);
    }
}
```

## Задание 2

Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска.

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`.

Результат работы:

 text.txt – Блокнот

Файл Правка Формат Вид Справка

1 из самых простых способов – использовать сочетание `BufferedReader` и `FileReader`. `FileReader` открывает поток для чтения из файла, а `BufferedReader` считывает текст из этого потока. Это позволяет читать данные построчно, что удобно для большинства текстовых файлов.

 text1.txt – Блокнот

Файл Правка Формат Вид Справка

sadfheanvuocaесмосеwnuvbasfnklasdf[anj aeverad a  
sadfdasdd fas

fa sfasd fa  
f asdf  
af sadf|

## Выходные данные:

```
cp -i text.txt text1.txt
File already exists. Overwrite? (y/n): y
File copied successfully.
```

text.txt – Блокнот

Файл Правка Формат Вид Справка

1 из самых простых способов – использовать сочетание `BufferedReader` и `FileReader`.  
`FileReader` открывает поток для чтения из файла, а `BufferedReader` считывает текст из этого потока.  
Это позволяет читать данные построчно, что удобно для большинства текстовых файлов.

text1.txt – Блокнот

Файл Правка Формат Вид Справка

1 из самых простых способов – использовать сочетание `BufferedReader` и `FileReader`.  
`FileReader` открывает поток для чтения из файла, а `BufferedReader` считывает текст из этого потока.  
Это позволяет читать данные построчно, что удобно для большинства текстовых файлов.

## Текст программы:

```
boolean force = false;
boolean interactive = false;
String sourceFile = null;
String targetFile = null;

for (int i = 1; i < params.length; i++) {
    String param = params[i];
    if (param.contains("-")) {
        if (param.contains("i")){
            interactive = true;
        }if (param.contains("f")){
            force = true;
        }if (param.contains("n")){
            interactive = false;
        }
    } else if (sourceFile == null) {
        sourceFile = param;
    } else if (targetFile == null) {
        targetFile = param;
    }
}

if (sourceFile == null || targetFile == null) {
    System.out.println("Error: Please provide both source and target file paths.");
    return;
}
File source = new File(sourceFile);
File target = new File(targetFile);

if (target.exists()) {
    if(interactive){
        System.out.print("File already exists. Overwrite? (y/n): ");
    }
}
```

```

        reader = new BufferedReader(new InputStreamReader(System.in));
        input = reader.readLine();
        if (!input.equals("y")) {
            System.out.println("Copying aborted.");
            return;
        }
    } else if (!force) {
        System.out.println("Target File already exists.");
        return;
    }
}

FileInputStream inputStream = new FileInputStream(source);
FileOutputStream outputStream = new FileOutputStream(target);
byte[] buffer = new byte[4096];
int bytesRead;

while ((bytesRead = inputStream.read(buffer)) != -1) {
    outputStream.write(buffer, 0, bytesRead);
}

inputStream.close();
outputStream.close();

```

**Вывод:** приобрел базовые навыки работы с файловой системой в Java.