

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

КАФЕДРА «ИИТ»

Лабораторная работа №2

Выполнил:

Студент 4 курса

Группа АС-50

Годухин Р.А

Проверил:

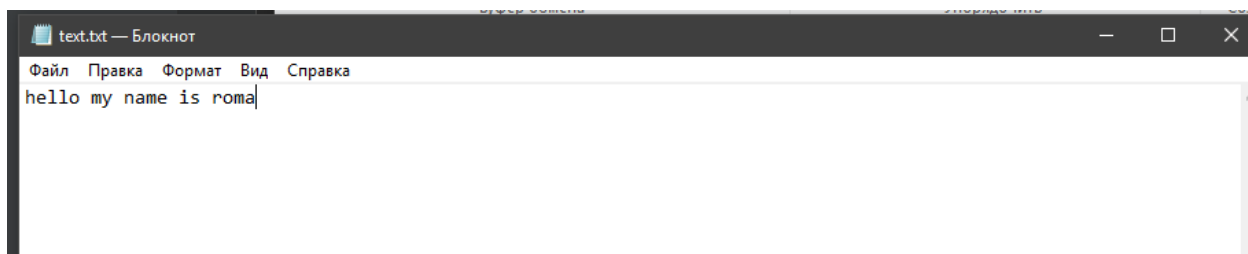
Крощенко.А.А

Брест 2020

Вариант 2

Задание 1 Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку: 2) Напишите программу, которая читает текст построчно, а затем разбивает каждую строку на лексемы и выводит их в обратном порядке.

```
package com.company;
import java.io.File;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) throws Exception{
        Scanner scanner = new Scanner(System.in);
        System.out.println("Please enter file path:");
        String name = scanner.nextLine();
        File file = new File(name);
        scanner = new Scanner(file, "utf-8");
        while(scanner.hasNext())
        {
            String words = scanner.nextLine();
            String[] mas = words.split(" ");
            words = "";
            for(int i=mas.length-1; i>=0; i--)
                words+=mas[i]+" ";
            System.out.println(words);
        }
    }
}
```



```
C:\Users\Роман\.jdk\openjdk-15.0.1\bin\java.exe
Please enter file path:
text.txt
roma is name my hello

Process finished with exit code 0
```

Задание 2 Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска. 2) Утилита tail выводит несколько (по умолчанию 10) последних строк из файла. Формат использования: tail [-n] file Ключ - n (или просто) позволяет изменить количество выводимых строк. Пример использования: tail -n 20 app.log 2 Крощенко А.А., Современные системы программирования, ЛР2 tail 20 app.log Выводит 20 последних строк из файла app.log. Для решения задачи подойдет класс java.io.RandomAccessFile, реализующий произвольный доступ к файлу (чтение и запись с любой позиции в файле).

```
package com.company;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;
public class Main {
    public static void main(String[] args) {
        int length = 10;
        if (args.length == 0 || args.length > 3) {
            System.out.println("неверное число аргументов");
            return;
        }
        else {
            if (args.length == 2) {
                try {
                    length = Integer.parseInt(args[0]);
                } catch (NumberFormatException e) {
                    System.out.println("неверные аргументы");
                    return;
                }
            } else if (args.length == 3) {
                try {
                    length = Integer.parseInt(args[1]);
                } catch (NumberFormatException e) {
                    System.out.println("неверные аргументы");
                    return;
                }
                if (!args[0].equals("-n")) {
                    System.out.println("неверные аргументы");
                    return;
                }
            }
        }
        try {
            RandomAccessFile file = new RandomAccessFile(args[args.length - 1], "r");
            int line = 1;
            long pos = file.length() - 1;
            if (pos < 0)
                return;
```

```

do {
    file.seek(pos);
    char c = (char) file.readByte();
    if (c == '\n' && pos != file.length() - 1)
        line++;
    if (line > length)
        break;
    if (pos != 0)
        pos--;
}
while (pos != 0);
file.seek(pos);
while (true) {
    try {
        String str = new String(file.readLine().getBytes("ISO-8859-1"), "utf-8");
        System.out.println(str);
    } catch (NullPointerException e) {
        break;
    }
}
} catch (FileNotFoundException e) {
    System.out.println("файл не найден");
} catch (IOException e) {
    System.out.println("ошибка файла");
    e.printStackTrace();
}
}
}

```

```

E:\Java\labs\lab2_god>lab2.bat -n 20 app.log

E:\Java\labs\lab2_god>java -jar lab2.jar -n 20 app.log

3535345345

35453534534

34345
34534
5345
345
345
35
345
345
345
345
34
534
534
534
53
53

```

app.log – Блокнот

Файл Правка Формат Вид Справка

```

4
4
45
5
43
34534534534535
345345345345
3535345345
35453534534
34345
34534
5345
345
345
35
345
345
345
345
34
534
534
534
53
53

```