

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Лабораторная работа №4

«Ввод-вывод в Java»

Выполнил:

Швалов Даниил Андреевич

Группа: К33211

Проверил:

Иванов Сергей Евгеньевич

Санкт-Петербург

2024

Введение

Цель работы: разработка приложений, использующие потоки ввода вывода.

Ход работы

Упражнение 1. Простая работа с файлом

В данном упражнении необходимо добавить метод *readList*, который будет считывать целочисленные значения из файла. На рисунке 1 представлен исходный код получившегося метода.

```
public void readList() {
    File file = new File("infile.txt");

    Scanner scanner;
    try {
        scanner = new Scanner(file);
    } catch (FileNotFoundException e) {
        System.err.println("Cannot open input file: " + e.getMessage());
        return;
    }

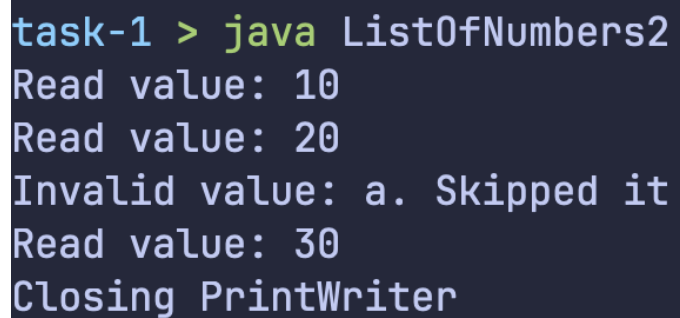
    while (scanner.hasNext()) {
        if (!scanner.hasNextInt()) {
            System.err.printf(
                "Invalid value: %. Skipped it\n",
                scanner.next()
            );
            continue;
        }
        int value = scanner.nextInt();
        System.out.println("Read value: " + value);
        arr.add(new Integer(value));
    }
}
```

Рисунок 1 – Исходный код функции *readList*

В нем происходит считывание данных из файла с помощью класса *Scanner*

из стандартной библиотеки Java. Метод считывает все значения до тех пор, пока файл не закончится. В случае, если считываемое значение является некорректным, то выводится сообщение об ошибке, и считывание продолжается дальше.

На рисунке 2 представлен пример работы программы. Во входной файл были записаны значения 10, 20, а, 30. Как видно на рисунке 2, все целочисленные значения были успешно считаны. Также они появились и в выходном файле.



```
task-1 > java ListOfNumbers2
Read value: 10
Read value: 20
Invalid value: a. Skipped it
Read value: 30
Closing PrintWriter
```

Рисунок 2 – Пример работы программы

Упражнение 2. Знакомство с классом `BufferedReader`

В данном упражнении необходимо реализовать функцию *main*, в которой, с помощью уже существующего метода *countTokens*, будет происходить вывод количества вхождений слова, которое вводит пользователь.

На рисунке 3 представлен исходный код получившейся функции *main*. Сначала проверяется количество переданных аргументов. Если аргументов передано не было, пользователю сообщается об ошибке. Затем первый аргумент используется в качестве имени файла.

Для считывания пользовательского ввода используется *BufferedReader* из стандартной библиотеки Java. Считывание происходит построчно, и при считывании проверяется, не равна ли входная строка значению *q*. Если равна, то работа программы прекращается. При вводе любого другого слова выводится количество вхождений данного слова в переданный файл.

```

public static void main(String[] args) {
    if (args.length < 1) {
        System.out.println("Expected filename");
        System.exit(-1);
    }

    String file = args[0];
    FileScanInteractive scanner = new FileScanInteractive();

    try (
        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in)
        )
    ) {
        while (true) {
            System.out.print("Enter the search string or q to exit: ");
            String line = br.readLine().trim();
            if (line.equalsIgnoreCase("q")) {
                break;
            }

            System.out.printf(
                "Count of %s in file: %d\n",
                line,
                scanner.countTokens(file, line)
            );
        }
    } catch (IOException e) {
        System.err.println("Error: " + e.getMessage());
    }
}

```

Рисунок 3 – Исходный код функции *main*

На рисунке 4 приведен пример работы программы. Как видно, программа выводит верное количество вхождений. Это можно проверить, пересчитав количество вхождений с помощью других инструментов.

```
task-2 > java FileScanInteractive ObamaInaguralAddress.txt
Enter the search string or q to exit: when
Count of when in file: 7
Enter the search string or q to exit: rights
Count of rights in file: 1
Enter the search string or q to exit: free
Count of free in file: 2
Enter the search string or q to exit: q
```

Рисунок 4 – Пример работы программы

Заключение

В данной лабораторной работе были разработаны приложения, использующие потоки ввода и вывода для считывания информации от пользователя и вывода информации. Цель, поставленная в начале работы, достигнута, задачи выполнены.