

Задача заключается в создании программы ввода-вывода с использованием технологии NIO2.

Рекомендации по выполнению этой задачи:

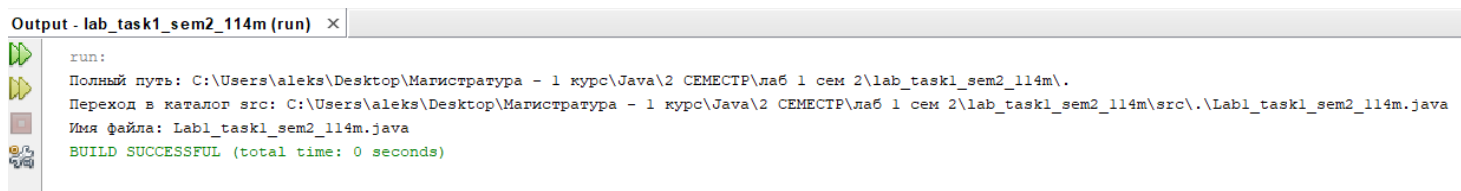
Напишите программу, которая будет выводить свой собственный код (содержимое файла *.java). При разработке обязательно используйте NIO2:

- Определите текущий каталог;
- Укажите имя файла;
- Перейдите в каталог проекта src;
- Сформируйте имя файла с текстом программы (добавьте расширение *.java).
-

Код программы:

```
public static void main(String[] args) throws IOException {
    // текущий каталог;
    Path currentDir = Paths.get(".");
    Path fullPath = currentDir.toAbsolutePath(); //полный
    путь
    System.out.println("Полный путь: "+ fullPath); //вывод
    полного пути
    // каталог src проекта;
    Path srcDir =
    Paths.get("src/./Lab1_task1_sem2_114m.java").toAbsolutePath();
    System.out.println("Переход в каталог src: "+ srcDir);
    //вывод полного пути каталога src
    //имя файла с текстом программы
    Path fileName = srcDir.getFileName();
    System.out.println("Имя файла: "+ fileName); //вывод
    имени файла
}
}
```

Вывод:



```
Output - lab_task1_sem2_114m (run) x
run:
Полный путь: C:\Users\aleks\Desktop\Магистратура - 1 курс\Java\2 СЕМЕСТР\лаб 1 сем 2\lab_task1_sem2_114m\
Переход в каталог src: C:\Users\aleks\Desktop\Магистратура - 1 курс\Java\2 СЕМЕСТР\лаб 1 сем 2\lab_task1_sem2_114m\src\.\Lab1_task1_sem2_114m.java
Имя файла: Lab1_task1_sem2_114m.java
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Разработайте программу, которая записывает в типизированный файл целые числа, затем считывает и рекуррентно рассчитывает среднее.

В файл пишутся целые числа. Затем из этого файла как потока байт считываются числа byte, int и float.

- Произведите расчет считанных элементов при чтении данных разного типа.
- Определите количество чисел типа int.
- Рассчитайте среднее второй половины, третьей четверти.

Код программы:

```
public class Lab1_task2_sem2_114m {

    public static void main(String[] args) throws
FileNotFoundException, IOException {

        Path path = Paths.get("src/test.dat");
        //запись случайных целых чисел в файл
        int[] array = {1, 2, 3, 4, 5}; // Массив целых чисел
        try (FileWriter fw = new FileWriter(path.toFile())) {
            int strs[] = new int[20];
            for (int i = 0; i < strs.length; i++) {
                strs[i] = (int) (Math.random() * 100);
                fw.write(strs[i] + " ");
            }
        }

        //-----int-----
        try {
            //получение массива чисел int
            int[] dataInt = Files.lines(path).flatMap(e ->
Stream.of(e.split(" "))).mapToInt(num ->
Integer.parseInt(num)).toArray();
            int average1 = 0, average2 = 0, average3 = 0;
            //средние
            //расчет среднего арифметического
            if (dataInt.length > 0) { //файл не пустой
                int sum1 = 0, sum2 = 0, sum3 = 0; //суммы всех
чисел
                //среднее арифметическое всех чисел
                for (int j = 0; j < dataInt.length; j++) sum1 +=
dataInt[j];
                average1 = (int) (sum1 / dataInt.length);
                //среднее арифметическое второй половины
                for (int j = dataInt.length / 2; j <
dataInt.length; j++) sum2 += dataInt[j];
                average2 = (int) (sum2 / dataInt.length);
                //среднее арифметическое третьей четверти
                for (int j = (dataInt.length * 3) / 4; j <
dataInt.length; j++) sum3 += dataInt[j];
                average3 = (int) (sum3 / dataInt.length);
            }
            System.out.println("Для int: кол-во чисел int =
"+dataInt.length);
        }
    }
}
```

```

        System.out.println("среднее арифметическое: " +
averagel);
        System.out.println("среднее арифметическое второй
половины: " + average2);
        System.out.println("среднее арифметическое третьей
четверти: " + average3);
        System.out.println(Arrays.toString(dataInt));
    } catch (IOException e) {
        e.printStackTrace();
    }
    //-----float----
    try {
        //получение массива чисел float
        Float[] dataFloat = Files.lines(path).flatMap(e ->
Stream.of(e.split("
")))
.map(Float::valueOf).toArray(Float[]::new);
        float averagel = 0, average2 = 0, average3 = 0;
//средние
        //расчет среднего арифметического
        if (dataFloat.length > 0) { //файл не пустой
            float sum1 = 0, sum2 = 0, sum3 = 0; //суммы всех
чисел
            //среднее арифметическое всех чисел
            for (int j = 0; j < dataFloat.length; j++) sum1
+= dataFloat[j];
            averagel = (float) (sum1 / dataFloat.length);
            //среднее арифметическое второй половины
            for (int j = dataFloat.length / 2; j <
dataFloat.length; j++) sum2 += dataFloat[j];
            average2 = (float) (sum2 / dataFloat.length);
            //среднее арифметическое третьей четверти
            for (int j = (dataFloat.length * 3) / 4; j <
dataFloat.length; j++) sum3 += dataFloat[j];
            average3 = (float) (sum3 / dataFloat.length);
        }
        System.out.println("Для float:");
        System.out.println("среднее арифметическое: " +
averagel);
        System.out.println("среднее арифметическое второй
половины: " + average2);
        System.out.println("среднее арифметическое третьей
четверти: " + average3);
        System.out.println(Arrays.toString(dataFloat));
    } catch (IOException e) {
        e.printStackTrace();
    }
    //-----byte-----

    try {
        int[] dataInt = Files.lines(path).flatMap(e ->
Stream.of(e.split(" ")))
.mapToInt(Integer::parseInt).toArray();

```

```

        ByteArrayOutputStream baos = new
ByteArrayOutputStream();
        DataOutputStream dos = new DataOutputStream(baos);
        for (int i = 0; i < dataInt.length; ++i) {
            dos.writeInt(dataInt[i]);
        }
        byte[] dataByte=baos.toByteArray();
        byte average1 = 0, average2 = 0, average3 = 0;
//средние
        //расчет среднего арифметического
        if (dataByte.length > 0) { //файл не пустой
            byte sum1 = 0, sum2 = 0, sum3 = 0; //суммы всех
чисел
            //среднее арифметическое всех чисел
            for (int j = 0; j < dataByte.length; j++) sum1
+= (byte) dataByte[j];
            average1 = (byte) (sum1 / dataByte.length);
            //среднее арифметическое второй половины
            for (int j = dataByte.length / 2; j <
dataByte.length; j++) sum2 += dataByte[j];
            average2 = (byte) (sum2 / dataByte.length);
            //среднее арифметическое третьей четверти
            for (int j = (dataByte.length * 3) / 4; j <
dataByte.length; j++) sum3 += dataByte[j];
            average3 = (byte) (sum3 / (byte)
dataByte.length);
        }
        System.out.println("Для byte:");
        System.out.println("среднее арифметическое: " +
average1);
        System.out.println("среднее арифметическое второй
половины: " + average2);
        System.out.println("среднее арифметическое третьей
четверти: " + average3);
        System.out.println(Arrays.toString(dataByte));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Вывод:

```

Выход - lab1_task2_sem2_114m (run)
run:
Для int: кол-во чисел int = 20
среднее арифметическое: 55
среднее арифметическое второй половины: 22
среднее арифметическое третьей четверти: 5
[49, 95, 82, 60, 87, 98, 51, 39, 31, 76, 14, 94, 56, 98, 71, 21, 39, 6, 7, 45]
Для float:
среднее арифметическое: 55.95
среднее арифметическое второй половины: 22.55
среднее арифметическое третьей четверти: 5.9
[49.0, 95.0, 82.0, 60.0, 87.0, 98.0, 51.0, 39.0, 31.0, 76.0, 14.0, 94.0, 56.0, 98.0, 71.0, 21.0, 39.0, 6.0, 7.0, 45.0]
Для byte:
среднее арифметическое: 1
среднее арифметическое второй половины: 0

```

3. Создайте текстовый файл, в него скопируйте не менее одной страницы документации по java. Считывайте символьный поток до тех пор, пока в третий раз не встретится слово «java».

Код программы:

```

public static void main(String[] args) throws IOException {
    String str = "java"; //слово триггер
    Path path = Paths.get("src/doc.txt"); //получение файла
    //вызов метода и вывод прочитанного файла до 3го
    упоминания java
    for (String st : readfileCountStr(path, str)) {
        System.out.println(st);
    }
}
//чтение файла до 3го вхождения слова
private static List<String> readfileCountStr(Path path,
String str) {
    List<String> listStr = new LinkedList<>(); //здесь
    хранится прочитанный текст
    try (BufferedReader br = Files.newBufferedReader((path),
StandardCharsets.UTF_8)) {
        int count = 0; //счетчик слова триггера
        String readStr; //считываемая строка
        //считывать пока не закончится файл и слово
        встречалось меньше 3х раз
        while (((readStr = br.readLine()) != null) && count
!= 3) {
            int index = readStr.indexOf(str); //индекс
            вхождения слова в строку
            while (index != -1) {
                count++; //подсчет вхождения слова в текст
                if (count == 3) { //строка содержит
                    readStr = readStr.substring(0, index +
str.length()); //до 3го вхождения слова
                }
                index = readStr.indexOf(str, index + 1);
            }
            listStr.add(readStr); //считанная строка
        }
    }
}

```

```

    } catch (IOException e) {
        e.printStackTrace();
    }
    return listStr;
}

```

Вывод:

```

Вывод - lab1_task3_sem2_114m (run)
reset()
See the general contract of the reset method of InputStream.
long
skip(long n)
See the general contract of the skip method of InputStream.
Methods declared in class java.io.FilterInputStream
read
Methods declared in class java
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

4. Напишите пример, который подсчитывает, сколько раз конкретный символ, например е, появляется в файле. Символ можно указать в командной строке.

Код программы:

```

public class Lab1_task4_sem2_114m {

    public static void main(String[] args) throws IOException {
        System.out.print("Введите символ: ");
        char ch = (char) System.in.read ();
        Path path = Paths.get("src/doc.txt"); //получение файла
        long countChar = counterChar(path, ch);
        System.out.println("Символ '" + ch + "' появляется в
        файле " + countChar + " раз");
    }

    //метод считает сколько в файле переданного символа
    public static long counterChar(Path path, char ch) throws
    IOException {
        try (Stream<String> stream = Files.lines(path))
        { //чтения всех строк из файла в виде потока
            return stream.flatMapToInt(String::chars).filter(c -
            > c == (int) ch).count(); //подсчет нужного символа
        }
    }
}

```

Вывод:

Вывод - lab1_task4_sem2_114m (run)

run:
Введите символ: е
Символ 'е' появляется в файле 357 раз
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)
|