



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

О Т Ч Е Т

по лабораторной работе № 9

Название: Stream API. Вариант 9

Дисциплина: Языки программирования для работы с большими
данными

Студент ИУ6-22М
(Группа)

К.Ю. Каташинский
(Подпись, дата) (И.О. Фамилия)

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Изучить Stream API в языке программирования Java.

Ход работы

Задание 1. Задана коллекция строк. Вернуть количество вхождений строки.

Код программы представлен в листинге 1, результат – на рисунке 1.

Листинг 1 – Код программы задания 1

```
package Lab9;

import java.util.Arrays;

public class Variant19 {

    public static void main(String[] args) {

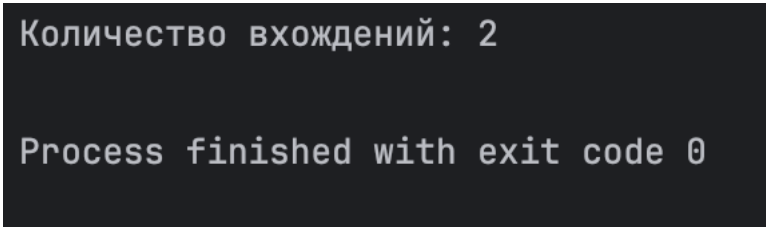
        String[] arr = { "big house", "small house", "big
home", "small home" };

        String word = "house";

        System.out.println("Количество вхождений: " +
Arrays.stream(arr).filter(x->x.contains(word)).count());

    }

}
```



```
Количество вхождений: 2

Process finished with exit code 0
```

Рисунок 1 – Результат задания 1

Задание 2. Задана коллекция:

(Класс Student: имя и рейтинг)

```
Collection<Student> students = Arrays.asList(
    new Student("Ivan", 40),
    new Student("Petr", 60),
    new Student("Olga", 70)
);
```

Вернуть список студентов имя которых начинается на Р и рейтинг находится в интервале 40–60.

Код программы представлен в листинге 2, результат – на рисунке 2.

Листинг 2 – Код программы задания 2

```
package Lab9;

import java.util.Arrays;
import java.util.Collection;

class Student {
    private final String name;
    private final int rating;
    public Student(String name, int rating) {
        this.name = name;
        this.rating = rating;
    }
    public String getName() {
        return this.name;
    }
    public int getRating() {
        return this.rating;
    }
    @Override
    public String toString() {
        return this.name;
    }
}

public class Variant110 {
    public static void main(String[] args) {
        Collection<Student> students = Arrays.asList(
            new Student("Ivan", 40),
            new Student("Petr", 60),
            new Student("Olga", 70)
        );
    }
}
```

```

        var newCollection = students.stream().filter(x->
x.getName().startsWith("P") && x.getRating() > 40 &&
x.getRating() < 60).toArray();
        for (var student : newCollection) {
            System.out.println(student);
        }
    }
}

```

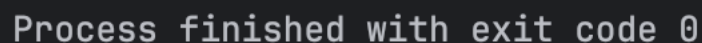


Рисунок 2 – Результат задания 2

Задание 3. Задана коллекция чисел. С помощью метода `reduce` вернуть максимум и минимум.

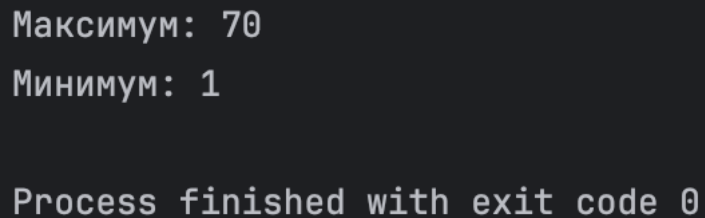
Код программы представлен в листинге 3, результат – на рисунке 3.

Листинг 3 – Код программы задания 3

```

package Lab9;
import java.util.Arrays;
import java.util.stream.Collectors;
public class Variant29 {
    public static void main(String[] args) {
        Integer[] arr = { 1, 10, 2, 3, 4, 40, 5, 60, 70 };
        var stats =
Arrays.stream(arr).collect(Collectors.summarizingInt(Integer
r::intValue));
        System.out.println("Максимум: " + stats.getMax());
        System.out.println("Минимум: " + stats.getMin());
    }
}

```



```
Максимум: 70
Минимум: 1

Process finished with exit code 0
```

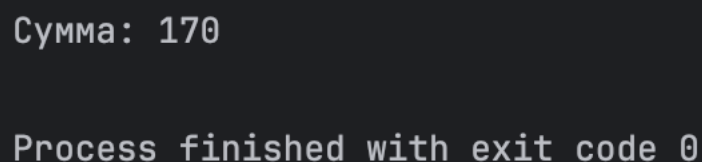
Рисунок 3 – Результат задания 3

Задание 4. Задана коллекция чисел. С помощью метода `reduce` вернуть сумму чисел, которые > 10 .

Код программы представлен в листинге 4, результат – на рисунке 4.

Листинг 4 – Код программы задания 4

```
package Lab9;
import java.util.Arrays;
public class Variant210 {
    public static void main(String[] args) {
        Integer[] arr = { 1, 10, 2, 3, 4, 40, 5, 60, 70 };
        System.out.println("Сумма: " +
Arrays.stream(arr).filter(x->x > 10).reduce(0,
Integer::sum));
    }
}
```



```
Сумма: 170

Process finished with exit code 0
```

Рисунок 4 – Результат задания 4

Вывод

В ходе выполнения лабораторной работы был изучен Stream API в языке программирования Java.