



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ 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

Дисциплина: Языки программирования для работы с большими
данными

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

А.М. Панфилкин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022 г.

Весь приведенный ниже код также доступен в следующем репозитории:

<https://github.com/SilkSlime/iu6plfbd>

Задание 1: Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Вернуть последний элемент и третий элемент коллекции..

Листинг 1 – Задание 1

```
package l9;

import java.util.ArrayList;
import java.util.List;

public class e1 {
    /**
     * Вариант 1. Задача 1.
     * Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.
     * Задана коллекция строк. Вернуть последний элемент и третий элемент коллекции.
     */
    public static void main(String[] args) {
        // Объявляем коллекцию строк
        List<String> list = new ArrayList<String>();
        list.add("lorem");
        list.add("ipsum");
        list.add("dolor");
        list.add("sit");
        list.add("amet");

        // Выводим последний элемент коллекции через Stream API
        System.out.println(list.stream().skip(list.size() - 1).findFirst().get());

        // Выводим третий элемент коллекции через Stream API
        System.out.println(list.stream().skip(2).findFirst().get());
    }
}
```

Задание 2: Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Вернуть первый элемент коллекции, а также существуют ли все совпадения с шаблоном. Шаблон можно выбрать произвольно.

Листинг 2 – Задание 2

```
package l9;

import java.util.ArrayList;
import java.util.List;

public class e2 {
    /**
     * Вариант 1. Задача 2.
     * Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.
     */
}
```

```

* Задана коллекция строк. Вернуть первый элемент коллекции, а также
* существуют ли все совпадения с шаблоном. Шаблон можно выбрать произвольно.
*/
public static void main(String[] args) {
    // Объявляем коллекцию строк
    List<String> list = new ArrayList<String>();
    list.add("loremor");
    list.add("ipsumor");
    list.add("dolor");
    list.add("molor");
    list.add("color");
    list.add("sitor");
    list.add("ametor");

    // Выводим первый элемент коллекции через Stream API
    System.out.println(list.stream().findFirst().get());

    // Шаблон для слов, которые оканчиваются на "olor"
    String pattern = ".*or";

    // Проверяем, существуют ли все совпадения с шаблоном без использования циклов и
    условий
    System.out.println(list.stream().allMatch(s -> s.matches(pattern)));
}
}

```

Задание 3: Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Отсортировать значения по алфавиту и убрать повторы.

Листинг 3 – Задание 3

```

package l9;

import java.util.ArrayList;
import java.util.List;

public class e3 {
    /**
     * Вариант 2. Задача 1.
     * Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.
     * Задана коллекция строк. Отсортировать значения по алфавиту и убрать повторы.
     */
    public static void main(String[] args) {
        // Объявляем коллекцию строк
        List<String> list = new ArrayList<String>();
        list.add("lorem");
        list.add("ipsum");
        list.add("dolor");
        list.add("sit");
        list.add("amet");
        list.add("sit");
        list.add("amet");
    }
}

```

```

        // Сортируем значения по алфавиту и убираем повторы
        list.stream().sorted().distinct().forEach(System.out::println);
    }
}

```

Задание 4: Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция: (Класс People: имя и возраст) `Collection<People> peoples = Arrays.asList(new People("Ivan", 16), new People("Petr", 23), new People("Maria", 42));` Отсортировать по имени в обратном алфавитном порядке.

Листинг 4 – Задание 4

```

package l9;

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

public class e4 {
    /**
     * Вариант 2. Задача 2.
     * Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.
     * Задана коллекция:
     * (Класс People: имя и возраст)
     * Collection<People> peoples = Arrays.asList(
     * new People("Ivan", 16),
     * new People("Petr", 23),
     * new People("Maria", 42)
     * );
     * Отсортировать по имени в обратном алфавитном порядке.
     */
    public static void main(String[] args) {
        // Объявляем коллекцию пиплов
        Collection<People> peoples = Arrays.asList(
            new People("Ivan", 16),
            new People("Petr", 23),
            new People("Maria", 42));

        // Сортируем по имени в обратном алфавитном порядке через Stream API
        peoples.stream().sorted((p1, p2) ->
            p2.getName().compareTo(p1.getName())) .forEach(System.out::println);
    }
}

// Класс People
class People {
    private String name;
    private int age;
}

```

```
public People(String name, int age) {  
    this.name = name;  
    this.age = age;  
}  
  
public String getName() {  
    return name;  
}  
  
public int getAge() {  
    return age;  
}  
  
// Оверрайд для строкового представления объекта  
@Override  
public String toString() {  
    return "People{" +  
        "name='" + name + '\'' +  
        ", age=" + age +  
        '}';  
}  
}
```

Вывод: В ходе лабораторной работы мы познакомились с использованием Stream API в Java и научились решать различные задачи с помощью его методов. Мы решили четыре задания, в которых использовали только методы Stream API без использования циклов и условий. В первом задании мы вернули последний и третий элементы коллекции строк, во втором задании мы вернули первый элемент и проверили, совпадают ли все элементы коллекции со шаблоном. В третьем задании мы отсортировали значения коллекции строк в алфавитном порядке и убрали повторы, а в четвертом задании мы отсортировали коллекцию People по имени в обратном алфавитном порядке. Все задания были решены с помощью методов Stream API, что демонстрирует преимущества использования данного инструмента при работе с коллекциями данных в Java.