



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

Вариант № 17

Название: Использование Stream API

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

М.О. Усманов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы

Получение навыков работы с Stream API языка программирования Java.

Ход работы

Задание 1.

Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

- Задана коллекция чисел. Вернуть сумму нечетных чисел.
- Задана коллекция чисел. Разделить числа на четные и нечетные.

Листинг 1 – Код программы первого задания

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.function.Predicate;

public class App {
    public static void main(String[] args) throws Exception {
        ArrayList<Integer> numbers = new ArrayList<>()
            Arrays.asList(
                1, 3, 5, 7, 10, 12, 16, 44, 37
            )
        );

        Predicate<Integer> isOdd = arg -> arg % 2 == 1;
        Predicate<Integer> isEven = arg -> arg % 2 == 0;

        ArrayList<Integer> oddNumbers = new
ArrayList<>(numbers.stream().filter(isOdd).toList());

        ArrayList<Integer> evenNumbers = new
ArrayList<>(numbers.stream().filter(isEven).toList());

        System.out.println("Odd:");
        oddNumbers.stream().forEach(System.out::println);

        System.out.println("Even:");
        evenNumbers.stream().forEach(System.out::println);

        System.out.println("Odd Sum:");
        Integer sum = oddNumbers.stream().reduce(0, Integer::sum);
        System.out.println(sum);
    }
}
```

Приведем результаты выполнения данного кода.

```

PS E:\Projects\Java\Repo\lab_9_var_1_3_4> & 'C:\
Projects\Java\Repo\lab_9_var_1_3_4\bin' 'App'
Odd:
1
3
5
7
37
Even:
10
12
16
44
Odd Sum:
53
PS E:\Projects\Java\Repo\lab_9_var_1_3_4>

```

Рисунок 1 – Результат выполнения варианта задания 1

Задание 2.

Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

– Коллекция из 2 задания. Отсортировать по возрасту.

```

Collection<People> peoples = Arrays.asList(
    new People("Ivan", 16),
    new People("Petr", 23),
    new People("Maria", 42)
);

```

– Задана коллекция строк. Преобразовать в MAP, где первый символ ключ, второй – значение.

В коллекцию была добавлена дополнительная запись, поскольку на исходной коллекции проверить успех сортировки представляется затруднительным.

Листинг 2 – Код выполнения задания 2

```

import java.util.Arrays;
import java.util.Collection;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

public class App {
    public static void main(String[] args) throws Exception {

```

```

        Collection<People> peoples = Arrays.asList(
            new People("Ivan", 16),
            new People("Petr", 23),
            new People("Maria", 42),
            new People("Slava", 12)
        );

        List<App.People> sortedList = peoples.stream()
            .sorted(Comparator.comparingInt(People::getAge))
            .collect(Collectors.toList());

        sortedList.forEach(System.out::println);

        Collection<String> strings = Arrays.asList(
            "da", "le", "ko", "ty", "hi"
        );

        Map<Character, Character> map = strings.stream()
            .collect(Collectors.toMap(a -> a.charAt(0), a ->
a.charAt(1)));
        System.out.println("Map:" + map);
    }

    private static class People {
        public String name;
        public int age;

        public People(String name, int age) {
            this.name = name;
            this.age = age;
        }

        public int getAge() {
            return age;
        }

        @Override
        public String toString() {
            return name + " " + age;
        }
    }
}

```

Результаты выполнения задания приведем на рисунке далее.

```
PS E:\Projects\Java\Repo\lab_9_var_2_3_4> & 'C:\Program Fi  
ges' '-cp' 'E:\Projects\Java\Repo\lab_9_var_2_3_4\bin' 'App  
Slava 12  
Ivan 16  
Petr 23  
Maria 42  
Map:{d=a, t=y, h=i, k=o, l=e}  
PS E:\Projects\Java\Repo\lab_9_var_2_3_4>
```

Рисунок 2 – Выполненный код задания 2

Местоположение репозитория с файлами проекта

Файлы проекта расположены в репозитории веб-платформы для совместной разработки Github. Местоположение в репозитории:

https://github.com/s314/big-data-studies/tree/main/lab_9_var_1_3_4

https://github.com/s314/big-data-studies/tree/main/lab_9_var_2_3_4

Вывод

По итогам выполнения лабораторной работы были получены навыки программирования с использованием Stream API языка Java.