|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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.**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 9** |
|  |
| **Вариант 16** |

**Название:**

Использование Stream API

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-23М | |  |  | В.И. Тарасов |
|  | | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  | |  |  |  |  |
| Преподаватель | |  |  |  | П.В. Степанов |
|  | |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы** –получение навыков работы с Stream API в языке Java.

**Ход работы**

**Задание 1:**

1. Задана коллекция строк. Вернуть первый элемент коллекции, а также существуют ли все совпадения с шаблоном. Шаблон можно выбрать произвольно.
2. Задана коллекция чисел. Вернуть сумму нечетных чисел.

Листинг 1 – Код задания 1

package src.lab9\_1;

import java.util.Arrays;

import java.util.List;

public class Program {

public static void main(String[] args) {

// 2. Задана коллекция строк.

// Вернуть первый элемент коллекции, а также существуют ли все совпадения с шаблоном.

// Шаблон можно выбрать произвольно.

List<String> strings = Arrays.asList("String 1", "String 2", "Str");

String str = strings.stream().findFirst().orElse(" ");

boolean isAll = strings.stream().allMatch((s) -> s.contains("Str"));

System.out.println(str + " " + isAll);

//3. Задана коллекция чисел. Вернуть сумму нечетных чисел.

List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7);

Integer sumOdd = list.stream().filter(o -> o % 2 != 0).reduce(Integer::sum).orElse(0);

System.out.println(sumOdd);

}

}

Результаты выполнения задачи представлены на рисунке 1.

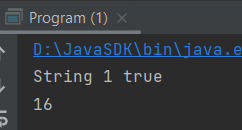


Рисунок 1 – Результат выполнения задания 1

**Задание 2:**

1. 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.

1. Коллекция из 2 задания. Отсортировать по возрасту.

Листинг 2 – Код класса People

package src.lab9\_2;

public class People {

String name;

Integer age;

public People(String name, Integer age){

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public Integer getAge() {

return age;

}

@Override

public String toString() {

return "People{" +

"name='" + name + '\'' +

", age=" + age +

'}';

}

}

Листинг 3 – Основная программа задания 2

package src.lab9\_2;

import java.util.Arrays;

import java.util.Collection;

import java.util.Comparator;

import java.util.List;

import java.util.stream.Collectors;

public class Program {

public static void main(String[] args) {

// 2. Задана коллекция

// Отсортировать по имени в обратном алфавитном порядке.

Collection<People> peoples = Arrays.asList(

new People("Ivan", 16),

new People("Petr", 23),

new People("Maria", 42)

);

List<People> list = peoples.stream().sorted(Comparator.comparing(People::getName, Comparator.reverseOrder())).toList();

System.out.println(list);

// 3. Коллекция из 2 задания. Отсортировать по возрасту.

List<People> listAge = peoples.stream().sorted(Comparator.comparingInt(People::getAge)).toList();

System.out.println(listAge);

}

}

Результаты выполнения представлены на рисунке 2.

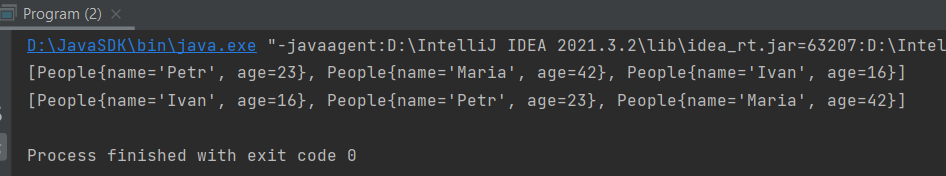


Рисунок 2 – Результат выполнения задания 2

**Местоположение проектных файлов** –файлы проекта расположены в репозитории GitHub по следующим адресам:

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab9_1>

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab9_2>

**Вывод** – в результате выполнения лабораторной работы были получены навыки работы с Stream API в языке Java.