|  |  |
| --- | --- |
| 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, 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**

**Отчет**

**по лабораторной работе №9**

**Название:** Stream API

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

Вариант: 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Н.А. Аскерова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

**Вариант 1**

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

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

Листинг 1 – Код программы

import java.util.Arrays;

import java.util.List;

public class Main {

public static void main(String[] args) {

List<String> strings = Arrays.asList("apple", "apbanana", "apricot", "apavocado");

String template = "ap";

String firstElement = strings.stream()

.findFirst()

.orElse(null);

boolean allMatch = strings.stream()

.allMatch(string -> string.startsWith(template));

System.out.println("First element: " + firstElement);

System.out.println("All match with template: " + allMatch);

}

}

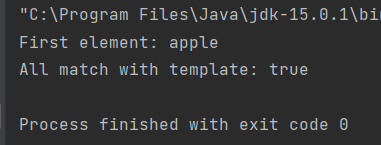


Рисунок 1 – Результат работы программы

3. Задана коллекция чисел. Вернуть сумму нечетных чисел.

Листинг 2 – Код программы

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

Integer[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int sumOfOddNumbers = Arrays.stream(numbers)

.filter(number -> number % 2 != 0)

.mapToInt(Integer::intValue)

.sum();

System.out.println("Sum of odd numbers: " + sumOfOddNumbers);

}

}

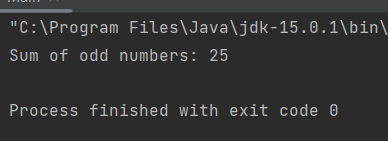


Рисунок 2 – Результат работы программы

**Вариант 2**

2. 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 3 – Код программы

import java.util.Arrays;

import java.util.Comparator;

import java.util.List;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

List<People> peoples = Arrays.asList(

new People("Ivan", 16),

new People("Petr", 23),

new People("Maria", 42)

);

List<People> sortedPeoples = peoples.stream()

.sorted(Comparator.comparing(People::getName).reversed())

.collect(Collectors.toList());

System.out.println("Sorted peoples: " + sortedPeoples);

}

}

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;

}

@Override

public String toString() {

return "People{" +

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

", age=" + age +

'}';

}

}

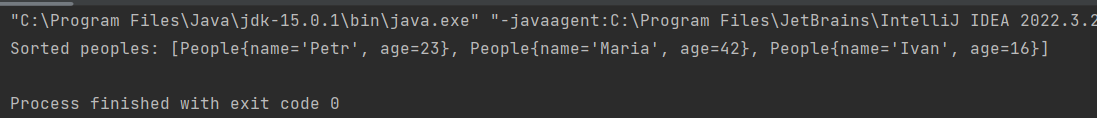


Рисунок 3 – Результат работы программы

3. Коллекция из 2 задания. Отсортировать по возрасту.  
Листинг 4 – Код программы

import java.util.Arrays;

import java.util.Comparator;

import java.util.List;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

List<People> peoples = Arrays.asList(

new People("Ivan", 27),

new People("Petr", 23),

new People("Maria", 42)

);

List<People> sortedPeoples = peoples.stream()

.sorted(Comparator.comparing(People::getAge))

.collect(Collectors.toList());

System.out.println("Sorted peoples: " + sortedPeoples);

}

}

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;

}

@Override

public String toString() {

return "People{" +

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

", age=" + age +

'}';

}

}

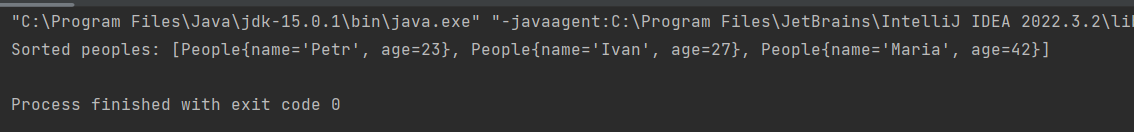


Рисунок 4 – Результат работы программы

**Вывод:** приобретен навык работы с Stream API.

**Ссылка на репозиторий с программами:** [**https://github.com/nargi3/BigData**](https://github.com/nargi3/BigData)