|  |  |
| --- | --- |
| 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 № 12**

**Название:** stream API

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

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

Москва, 2024

**Цель:**изучить stream API в java.

**Задание 1:** Задана коллекция строк. Вернуть первые два элемента. Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Код класса Main:

import java.util.List;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

List<String> strings = List.of("Первый", "Второй", "Третий", "Четвертый");

List<String> result = strings.stream()

.limit(2) // Ограничиваем поток двумя элементами

.collect(Collectors.toList()); // Собираем результат в список

System.out.println(result);

}

}}Работа программы показана на рисунке 1.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 1 – Работа программы

**Задание 2:** Задана коллекция строк. Вернуть два элемента начиная со второго. Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Код класса Main:

import java.util.List;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

List<String> strings = List.of("Первый", "Второй", "Третий", "Четвертый");

List<String> result = strings.stream()

.skip(1)

.limit(2) // Ограничиваем поток двумя элементами

.collect(Collectors.toList()); // Собираем результат в список

System.out.println(result);

}

}

Работа программы показана на рисунке 2.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2 – Работа программы

**Задание 3:** Найти самый минимальны возраст человека, у которого есть буква “e” в имени. Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Код класса Main:

import java.util.Arrays;

import java.util.Collection;

class People{

public String name;

public int age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public People(String name, int age) {

this.name = name;

this.age = age;

}

}

public class Main {

public static void main(String[] args) {

Collection<People> people = Arrays.asList(

new People("Ivan", 16),

new People("Petr", 23),

new People("Maria", 42)

);

int minAge = people.stream()

.filter(p -> p.name.contains("e")) // Фильтруем людей с буквой 'e' в имени

.mapToInt(People::getAge) // Преобразуем поток People в поток возрастов

.min() // Ищем минимальное значение

.orElse(-1); // В случае, если таких людей нет, возвращаем -1

System.out.println("Minimal age of people with 'e' in their name: " + minAge);

}

}

Код класса People:

import java.util.Arrays;

import java.util.Collection;

class People{

public String name;

public int age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public People(String name, int age) {

this.name = name;

this.age = age;

}

}

public class Main {

public static void main(String[] args) {

Collection<People> people = Arrays.asList(

new People("Ivan", 16),

new People("Petr", 23),

new People("Maria", 42)

);

int minAge = people.stream()

.filter(p -> p.name.contains("e")) // Фильтруем людей с буквой 'e' в имени

.mapToInt(People::getAge) // Преобразуем поток People в поток возрастов

.min() // Ищем минимальное значение

.orElse(-1); // В случае, если таких людей нет, возвращаем -1

System.out.println("Minimal age of people with 'e' in their name: " + minAge);

}

}

Работа программы показана на рисунке 3.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 3 – Работа программы

**Задание 4:** Задана коллекция строк: Collection<String> collection Arrays.asList("a1", "a2", "a3", "a1"). Убрать первый символ и вернуть числа. Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Код класса Main:

import java.util.Arrays;

import java.util.Collection;

import java.util.List;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

Collection<String> collection = Arrays.asList("a1", "a2", "a3", "a1");

List<Integer> numbers = collection.stream()

.map(s -> s.substring(1)) // Удаляем первый символ каждой строки

.map(Integer::parseInt) // Преобразуем оставшуюся строку в число

.collect(Collectors.toList()); // Собираем результаты в список

System.out.println(numbers); // Вывод: [1, 2, 3, 1]

}

}

Работа программы показана на рисунке 4.

Изображение выглядит как снимок экрана

Автоматически созданное описание со средним доверительным уровнем

Рисунок 4 – Работа программы

**Вывод:** во время выполнения лабораторной работы были изучена stream API в java.