|  |  |
| --- | --- |
| 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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

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

**Название:**

Stream

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

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

Москва, 2022

Вариант 1:

1. Коллекция из 10 задания. Вернуть средний балл.
2. Задана коллекция строк. Вернуть первые два элемента.
3. Задана коллекция:

(Класс People: имя и возраст, пол (enum))  
Collection<People> peoples = Arrays.*asList*(  
 **new** People(**"Ivan"**, 16, Sex.MAN),  
 **new** People(**"Petr"**, 23, Sex.MAN),  
 **new** People(**"Maria"**, 42, Sex.WOMAN)  
);

Найти самого старшего человека мужского пола.

1. Коллекция из 11 задания. Найти самый минимальны возраст человека, у которого есть буква “e” в имени.

Решение:

|  |
| --- |
| package com.company; import java.util.ArrayList; import java.util.Arrays; import java.util.Collection; import java.util.List; import java.util.stream.IntStream; import java.util.stream.Stream;   public class Main {   public static void main(String[] args) {  /\*  Вариант 1  11. Коллекция из 10 задания. Вернуть средний балл.  (Класс Student: имя и рейтинг) Collection<Student> students = Arrays.asList(  new Student("Ivan", 40),  new Student("Petr", 60),  new Student("Olga", 70) );  12. Задана коллекция строк. Вернуть первые два элемента.  \*/  System.out.println("------------------------------");  System.out.println("№ 1.11");  Collection<Student> students = Arrays.asList(  new Student("Ivan", 40),  new Student("Petr", 60),  new Student("Olga", 70)  );  System.out.println("Средний балл - "+students.stream().mapToInt((x) -> Integer.parseInt(String.valueOf(x.getCount()))).average().getAsDouble());  System.out.println("------------------------------");  System.out.println("№ 1.12");  List<String> list12 = new ArrayList<String>();  list12.add("Первая строка");  list12.add("Вторая строка");  list12.add("Третья строка");  list12.add("Четвертая строка");  Stream stream12 = list12.stream();  stream12.filter(x-> list12.indexOf(x.toString()) < 2).forEach(System.out::println);  /\*  Вариант 2  1. Задана коллекция: (Класс People: имя и возраст, пол (enum)) Collection<People> peoples = Arrays.asList(  new People("Ivan", 16, Sex.MAN),  new People("Petr", 23, Sex.MAN),  new People("Maria", 42, Sex.WOMAN) ); Найти самого старшего человека мужского пола. 2. Коллекция из 11 задания. Найти самый минимальны возраст человека, у которого есть буква “e” в имени.  \*/  System.out.println("------------------------------");  System.out.println("№ 2.11");  Collection<People> peoples = Arrays.asList(  new People("Ivan", 16, Sex.MAN),  new People("Petr", 23, Sex.MAN),  new People("Maria", 42, Sex.WOMAN)  );  System.out.println("Max age - "+peoples.stream().filter(x -> x.getPol() == Sex.MAN).mapToInt((x) -> Integer.parseInt(String.valueOf(x.getAge()))).max().getAsInt());  System.out.println("------------------------------");  System.out.println("№ 2.12");  System.out.println("Max age - "+peoples.stream().filter(x -> x.getName().contains("e")).mapToInt((x) -> Integer.parseInt(String.valueOf(x.getAge()))).min().getAsInt());         } }  package com.company;  public class People {  String name;  Integer age;  Sex pol;   public People(String name, Integer age, Sex pol) {  this.name = name;  this.age = age;  this.pol = pol;  }   public String getName() {  return name;  }   public Integer getAge() {  return age;  }   public Sex getPol() {  return pol;  } }  package com.company;  public enum Sex {  MAN,  WOMAN }  package com.company;  public class Student {  String name;  Integer count;   public Student(String name, Integer count) {  this.name = name;  this.count = count;  }   public Integer getCount() {  return count;  } } |

Ссылка на репозиторий:

https://github.com/Carpediem2026/BigData