|  |  |
| --- | --- |
| 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**

**Отчет**

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

**Название:** Stream API

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21М |  |  | Ю.А. Вишневская |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П. В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Вариант 1**

3. Задана коллекция чисел. Вернуть сумму нечетных чисел.

|  |
| --- |
| package com.company;  import java.util.Arrays;  import java.util.Collection;  public class LR9\_V1 {  public static void main(String[] args) {  Collection<Integer> collection = Arrays.asList(1, 2, 3, 4, 2, 10 ,11, 23, 42, 24);  System.out.println("Сумма нечетных чисел: ");  Integer sumOdd = collection.stream().filter(o -> o % 2 != 0).reduce((s1, s2) -> s1 + s2).orElse(0);  System.out.println(sumOdd);  }  } |

**Вариант 2**

11. 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.

People

|  |
| --- |
| import java.util.Objects;  public class People {  enum Sex {  MAN,  WOMEN  }  String name;  Integer age;  Sex sex;  public People(String name, Integer age, Sex sex) {  this.name = name;  this.age = age;  this.sex = sex;  }  public String getName() {  return name;  }  public Integer getAge() {  return age;  }  public Sex getSex() {  return sex;  }  public void setName(String name) {  this.name = name;  }  public void setAge(Integer age) {  this.age = age;  }  public void setSex(Sex sex) {  this.sex = sex;  }  public void getNameOfAge(Integer age){  if (age==this.age){  System.out.println(this.name);  }  } |

LR9\_V2

|  |
| --- |
| import java.util.Arrays;  import java.util.Collection;  public class LR9\_V2 {  public static void main(String[] args) {  Collection<People> peoples = Arrays.asList(  new People("Ivan", 16, People.Sex.MAN),  new People("Petr", 23, People.Sex.MAN),  new People("Maria", 42, People.Sex.WOMEN)  );  int oldAge = peoples.stream().filter((p) -> p.getSex() == People.Sex.MAN).map(People::getAge)  .reduce((s1, s2) -> s1 > s2 ? s1 : s2).get();  System.out.println("Возраст самого старого мужчины: " + oldAge);  }  } |