|  |  |
| --- | --- |
| 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 №** | 6 |

**Название:** Коллекции

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6И-21М |  |  | Ши Жуньфан |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П. В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Вариант 1**

1. Определить множество на основе множества целых чисел. Создать методы для определения пересечения и объединения множеств.

|  |
| --- |
| import java.util.\*; import java.util.stream.Collectors;  public class Lab6\_1\_1 {  public static void main(String[] args) {  int[] arrayA = new int[] {1,2,3,4};  int[] arrayB = new int[] {3,4,5,6};  List<Integer> listA = Arrays.*stream*(arrayA).boxed().collect(Collectors.*toList*());  List<Integer> listB = Arrays.*stream*(arrayB).boxed().collect(Collectors.*toList*());  *getInterval*(listA,listB);  *getSum*(listA,listB);   }  public static void getInterval(List listA,List listB){  Set<Integer> result = new HashSet<>();  result.addAll(listA);  result.retainAll(listB);  System.*out*.println(result);  }  public static void getSum(List listA, List listB){  Set<Integer> result = new HashSet<>();  result.addAll(listA);  result.removeAll(listB);  result.addAll(listB);  System.*out*.println(result);  }  } |

**Вариант 2**

10. На прямой гоночной трассе стоит N автомобилей, для каждого из которых известны начальное положение и скорость. Определить, сколько произойдет обгонов.

|  |
| --- |
| package com.company; import java.util.\*;  public class Lab6\_2\_10 {  static final int *LENGHT* = new Random().nextInt(100) + 109;  static final int *AUTOS\_QUANTITY* = 4;  public static void main(String[] args) {  System.*out*.println("Road lenght = " + *LENGHT*);  ArrayList<Auto> autos = new ArrayList<Auto>();  for (int i = 0; i < *AUTOS\_QUANTITY*; i++) {  autos.add(new Auto("Auto " + String.*valueOf*(i)));  }  autos.sort(new CompareAuto());  int i = 0;  for (Auto auto : autos) {  auto.beMoreFirst = i;  i++;  }  boolean endOfRace = false;  while (!endOfRace) {  endOfRace = true;  for (Auto auto : autos) {  if (auto.currentPosition + auto.speed <= *LENGHT*) {  auto.currentPosition = auto.currentPosition + auto.speed;  endOfRace = false;  } else  auto.currentPosition = *LENGHT*;  }  autos.sort(new CompareAuto());  }  int count = 0;  i = 0;  for (Auto auto : autos) {  if (i > auto.beMoreFirst) {  count = count + (i - auto.beMoreFirst);  }  i++;  }  System.*out*.println("Result = " + count);  } }  class Auto {  String name;  int firstPosition;  int speed;  int currentPosition;  int beMoreFirst;   Auto(String name) {  super();  this.name = name;  this.firstPosition = new Random().nextInt(11);  this.speed = new Random().nextInt(7) + 1;  this.currentPosition = this.firstPosition;  System.*out*.println(this);  }   @Override  public String toString() {  return name + " (sp. " + speed + ") pos. " + currentPosition;  } }  class CompareAuto implements Comparator<Auto> {   @Override  public int compare(Auto o1, Auto o2) {  return Integer.*compare*(o1.currentPosition, o2.currentPosition);  } } |