|  |  |
| --- | --- |
|  | **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 | ИУ6-22М |  |  | И. Д. Капкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

1. ****Введение и задание****

**Целью лабораторной работы является формирование базовых навыков работы с языком программирования java, изучение основных элементов – коллекций.**

**Номер в списке – 10.**

**Вариант 1.**

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

**Задание 2.** Списки (стеки, очереди) I(1..n) и U(1..n) содержат результаты n измерений тока и напряжения на неизвестном сопротивлении R. Найти приближенное число R методом наименьших квадратов.

**Вариант 2.**

Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести:

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

**Задание 1.** На базе коллекций реализовать структуру хранения чисел с поддержкой следующих операций: добавление/удаление числа; поиск числа, наиболее близкого к заданному (т.е. модуль разницы минимален).

1. ****Ход работы****

**Выполнение задания №1 первого варианта лабораторной работы:**

**import java.util.\*;**

**public class Main {**

**public static void main(String[] args) {**

**IntegerSet set1 = new IntegerSet(Arrays.asList(1,2,3,4,5));**

**IntegerSet set2 = new IntegerSet(Arrays.asList(2,3,5,7,11));**

**System.out.println(IntegerSet.intersection(set1, set2));**

**System.out.println(IntegerSet.union(set1, set2));**

**}**

**}**

**class IntegerSet extends TreeSet<Integer> {**

**public IntegerSet () {**

**super();**

**}**

**public IntegerSet (Collection<Integer> set) {**

**super(set);**

**}**

**public static IntegerSet intersection (IntegerSet a, IntegerSet b) {**

**IntegerSet intersect = new IntegerSet(a);**

**intersect.retainAll(b);**

**return intersect;**

**}**

**public static IntegerSet union(IntegerSet a, IntegerSet b) {**

**IntegerSet unify = new IntegerSet(a);**

**unify.addAll(b);**

**return unify;**

**}**

**}**

**Выполнение задания №2 первого варианта лабораторной работы:**

**import java.util.\*;**

**import java.util.stream.IntStream;**

**public class Main {**

**public static final int MEASUREMENTS\_COUNT = 42;**

**public static <T, U> List<Pair<T, U>> zip(List<T> list1, List<U> list2) {**

**List<Pair<T, U>> result = new ArrayList<>();**

**Iterator<T> iter1 = list1.iterator();**

**Iterator<U> iter2 = list2.iterator();**

**while (iter1.hasNext() && iter2.hasNext()) {**

**result.add(new Pair<>(iter1.next(), iter2.next()));**

**}**

**return result;**

**}**

**public static void main(String[] args) {**

**Random rnd = new Random(Calendar.getInstance().getTimeInMillis());**

**List<Double> current = new LinkedList<>();**

**List<Double> voltage = new LinkedList<>();**

**IntStream.range(0, MEASUREMENTS\_COUNT).forEach(i -> {**

**current.add(rnd.nextDouble());**

**voltage.add(rnd.nextDouble());**

**});**

**Collections.sort(current);**

**Collections.sort(voltage);**

**double ibar = current.stream().mapToDouble(Double::doubleValue).average().orElse(0.0);**

**double vbar = voltage.stream().mapToDouble(Double::doubleValue).average().orElse(0.0);**

**double r = zip(current, voltage).stream().mapToDouble(pair -> (pair.first()-ibar) \* (pair.second()-vbar)).sum()/**

**current.stream().mapToDouble(i -> Math.pow((i-ibar), 2)).sum();**

**System.out.println(r);**

**}**

**}**

**record Pair<T, U>(T first, U second) {**

**}**

**Выполнение задания №10 второго варианта лабораторной работы:**

**import java.util.\*;**

**import java.util.stream.IntStream;**

**public class Main {**

**public static final Integer AUTOMOBILE\_COUNT = 42;**

**public static final Random RNG = new Random(Calendar.getInstance().getTimeInMillis());**

**public static void main(String[] args) {**

**LinkedList<Automobile> automobiles = new LinkedList<>();**

**IntStream.range(0, AUTOMOBILE\_COUNT).forEach((unused) -> automobiles.add(new Automobile(RNG.nextDouble(), RNG.nextDouble())));**

**automobiles.sort(Comparator.comparing(Automobile::startingPosition));**

**long count = IntStream.range(0, automobiles.size())**

**.flatMap(i -> IntStream.range(i + 1, automobiles.size())**

**.filter(j -> automobiles.get(j).velocity() < automobiles.get(i).velocity()))**

**.count();**

**System.out.println(count);**

**}**

**}**

**record Automobile(double startingPosition, double velocity) {}**

**Выполнение задания №11 второго варианта лабораторной работы:**

**import java.util.Calendar;**

**import java.util.Random;**

**import java.util.TreeSet;**

**import java.util.stream.IntStream;**

**public class Main {**

**public static final int GIVEN\_NUMBER = 256;**

**public static final int SET\_INSERTION\_ATTEMPTS = 100;**

**public static final Random RNG = new Random(Calendar.getInstance().getTimeInMillis());**

**public static void main(String[] args) {**

**Numbers numbers = new Numbers();**

**IntStream.range(0, SET\_INSERTION\_ATTEMPTS).forEach((unused) -> numbers.add(RNG.nextInt(SET\_INSERTION\_ATTEMPTS\*SET\_INSERTION\_ATTEMPTS)));**

**int firstClosest = numbers.closestToGiven(GIVEN\_NUMBER);**

**System.out.println(firstClosest);**

**numbers.remove(firstClosest);**

**System.out.println(numbers.closestToGiven(GIVEN\_NUMBER));**

**}**

**}**

**class Numbers extends TreeSet<Integer> {**

**public int closestToGiven (int given) {**

**return this.lower(given) == null ? this.first() : this.higher(given) == null ? this.last() :**

**Math.abs(given - this.higher(given)) < Math.abs(given - this.lower(given)) ?**

**this.higher(given) : this.lower(given);**

**}**

**}**

1. ****Заключение****

**В процессе выполнения лабораторной работы были получены базовые навыки работы с языком программирования java, были изучены основные элементы – коллекции.**