



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**

О Т Ч Е Т

по лабораторной работе № 6

Вариант 10

Название: Коллекции в Java

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

И.С. Коломенский

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков работы с коллекциями в Java.

Выполнение:

Задание 1:

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

Задание 2:

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

Листинг файла Numbers.java

```
package com.company;

import java.util.ArrayList;

// Collections based
public class Numbers{
    private ArrayList<Double> data;

    public Numbers()
    {
        data=new ArrayList<Double>();
    }

    public boolean add(Double t) {
        return data.add(t);
    }
    // Overriding
    public boolean add(Integer i)
    {
        return data.add(i.doubleValue());
    }
}
```

```

public int size() {
    return data.size();
}

public boolean remove(Double o) {
    return data.remove(o);
}

public boolean remove(Integer ind){
    return data.remove(ind.doubleValue());
}

public Double removeByIndex(int ind)
{
    return data.remove(ind);
}
// if number is in a collection
public boolean contains(Double o) {
    return data.contains(o);
}

// Nearest number to an input number
public Double findNear(Double elem)
{
    double nr=Math.abs(elem-data.get(0));
    int ind=0;
    for(int i=0;i<size();i++) {
        if (Math.abs(elem - data.get(i)) < nr)
        {
            nr = Math.abs(elem - data.get(i));
            ind=i;
        }
    }
    return data.get(ind);
}

Double getBy(int ind)
{
    return data.get(ind);
}

@Override
public String toString() {
    String ans="[ ";
    for(int i=0;i<size()-1;i++)
    {
        ans+=String.format("%.2f ; ",getBy(i));
    }
    ans+=String.format("%.2f ; ",getBy(size()-1))+"]";
    return ans;
}
}

```

Листинг файла Main1_1.java

```

package com.company;

import java.util.*;

public class Main1_1 {
    public static void main(String[] args) {
        List<Integer> list1 = new ArrayList<Integer>(Arrays.asList(1, 2, 3, 4, 5, 6));
        List<Integer> list2 = new ArrayList<Integer>(Arrays.asList(7, 8, 9, 1, 2, 3));
    }
}

```

```

        System.out.println("Изначальное множество 1: ");
        System.out.println(list1);
        System.out.println("Изначальное множество 2: ");
        System.out.println(list2);
        System.out.println();

        System.out.println("Пересечение множеств: ");
        System.out.println(new Main1_1().intersection(list1, list2));
        System.out.println("Объединение множеств: ");
        System.out.println(new Main1_1().union(list1, list2));
    }

    public <T> List<T> union(List<T> list1, List<T> list2) {
        Set<T> set = new HashSet<T>();

        set.addAll(list1);
        set.addAll(list2);

        return new ArrayList<T>(set);
    }

    public <T> List<T> intersection(List<T> list1, List<T> list2) {
        List<T> list = new ArrayList<T>();

        for (T t : list1) {
            if(list2.contains(t)) {
                list.add(t);
            }
        }

        return list;
    }
}

```

Листинг файла Main1_2.java

```

package com.company;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class Main1_2 {

    public static double MNKR(Stack<Double>I, Queue<Double>U)
    {
        double n=(double) I.size();
        Stack<Double>revI=new Stack<Double>(); // direct order is needed so stack is
used (because stack is LIFO)
        while(!I.empty())
        {
            revI.push(I.pop()); // add
        }
        // approximation using MNK
        double su2,su,si,siu;
        double a,b,d,da,db;
        su2=su=si=siu=0.0; // initialization
        // sum in MNK
        while(!U.isEmpty())
        {
            su+=U.peek();
            si+=revI.peek();
            su2+=(U.peek()*revI.peek());
            siu+=(U.peek()*revI.peek());
            U.poll();
            revI.pop();
        }
    }
}

```

```

    }
    // basic logic of MNK
    d=n*su2-su*su;
    da=n*siu-su*si;
    db=su2*si-su*siu;
    a=da/d;
    b=db/d; // regression
    return 1/a;
}
public static void main(String[] args)
{
    Stack<Double>I=new Stack<Double>();
    Queue<Double>U=new LinkedList<Double>(); // because Queue is only an interface
    int n;
    Scanner cin=new Scanner(System.in); // quantity
    System.out.print("Введите N:");
    n=cin.nextInt();
    System.out.println("Введите "+n+" напряжение тока:");
    for(int i=0;i<n;i++)
    {
        double cur=cin.nextDouble();
        U.add(cur);
    }
    System.out.println("Введите "+n+" сила тока:");
    for(int i=0;i<n;i++)
    {
        double cur=cin.nextDouble();
        I.push(cur);
    }
    double R=MNKR(I,U);
    System.out.printf("%f",R);
}
}

```

Листинг файла Main2_1.java

```

package com.company;

import java.util.Scanner;

public class Main2_1 {
    public static void main(String[] args) {
        Numbers nm=new Numbers();
        boolean quit=false;
        Scanner cin=new Scanner(System.in);
        while (! quit)
        {
            System.out.println("1 -Добавить число: ");
            System.out.println("2 -Удалить число: ");
            System.out.println("3 -Найти ближайшее число к заданному: ");
            System.out.println("4 -Выход: ");
            byte cn= cin.nextByte();
            switch (cn)
            {
                case 1:
                {
                    System.out.println("Введите число");
                    double num=cin.nextDouble();
                    nm.add(num);
                    System.out.println("=====После
переобразования=====");
                    System.out.println(nm);
                    break;
                }
                case 2:
                {
                    System.out.println("Введите число для удаления");

```

```

        double num=cin.nextDouble();
        nm.remove(num);
        System.out.println("=====После
переобразования=====");
        System.out.println(nm);
        break;
    }
    case 3:
    {
        System.out.println("Введите число для нахождения ближайшего");
        double m= cin.nextDouble();
        System.out.println("Самое близкое: "+nm.findNear(m));
        break;
    }
    case 4:
    {
        quit=true;
        break;
    }
}
}
}
}

```

Листинг файла Main2_10.java

```

package com.company;

import java.util.*;

public class Main2_10 {

    public static void main(String[] args) {
        int n = 5;
        int road_lenght = (int) (Math.random()*2000+1000);
        System.out.println("Длина трассы = "+ road_lenght);
        ArrayList<Car> car_mass = new ArrayList<>();
        for (int i=0; i!=n; i++){
            car_mass.add(new
Car((int) (Math.random()*16+100), (int) (Math.random()*2+10), (int) (Math.random()*2+20),
road_lenght));
        }
        System.out.println(car_mass);
        int count =0;
        for (int i=0; i!=n; i++){
            for (int j=0; j!=n; j++){
                if (car_mass.get(i).time_for_road > car_mass.get(j).time_for_road){
                    count++;
                }
            }
        }
        System.out.println("Кол-во обгонов = "+ count);
    }

    public static class Car{
        int speed;
        int length;
        int leng_from_start;
        double time_for_road;

        public Car(int speed, int length, int leng_from_start, int road_lenght) {
            this.speed = speed;
            this.length = length;
            this.leng_from_start = length + leng_from_start;
            this.time_for_road = time_count(this.speed, road_lenght);
        }
    }
}

```

```

    }

    public double time_count(int speed, int road_lenght){
        return (double) road_lenght/ (double)speed;
    }

    @Override
    public String toString() {
        return "Car{" +
            "speed=" + speed +
            ", length=" + length +
            ", leng_from_start=" + leng_from_start +
            ", time_for_road=" + time_for_road +
            '}';
    }
}

```

```

Изначальное множество 1:
[1, 2, 3, 4, 5, 6]
Изначальное множество 2:
[7, 8, 9, 1, 2, 3]

Пересечение множеств:
[1, 2, 3]
Объединение множеств:
[1, 2, 3, 4, 5, 6, 7, 8, 9]

```

Рисунок 1 - Результат выполнения кода решения задачи 1, подзадачи 1

```

Введите N:3
Введите 3 напряжение тока:
10
2
4
Введите 3 сила тока:
6
32
63
-1,786885

```

Рисунок 2 - Результат выполнения кода решения задачи 1, подзадачи 2

```

1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
1
Введите число
32
=====После преобразования=====
[ 32,00 ; ]
1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
1
Введите число
10
=====После преобразования=====
[ 32,00 ; 10,00 ; ]
1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
2
Введите число для удаления
10
=====После преобразования=====
[ 32,00 ; ]
1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
1
Введите число
45
=====После преобразования=====
[ 32,00 ; 45,00 ; ]
1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
3
Введите число для нахождения ближайшего
38
Самое близкое: 32.0
1 -Добавить число:
2 -Удалить число:
3 -Найти ближайшее число к заданному:
4 -Выход:
4
Process finished with exit code 0

```

Рисунок 3 - Результат выполнения кода решения задачи 2, подзадачи 1

```

Длина трассы = 1855
[Car{speed=105, length=10, leng_from_start=30, time_for_road=17.666666666666668}, Car{speed=108,
Кол-во обгонов = 10

```

Рисунок 4 - Результат выполнения кода решения задачи 2, подзадачи 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Wingo11/BigDataLanguages/tree/Lab6/src>

Вывод:

При выполнении лабораторной работы были получены навыки работы с коллекциями в Java.