



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 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений.

О Т Ч Е Т
по лабораторной работе № 6
Вариант № 11

Название: Коллекции

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

С.В.Мельников

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2023

Цель работы

Изучить коллекции в языке программирования Java и рассмотреть их применение.

Задание 1 (Вариант 1, 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

Листинг программы:

Код класса ListIUR:

```
public class ListIUR {
    public static void main(String[] args) {
        List<Double> I = new ArrayList<>();
        List<Double> U = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите количество измерений: ");
        int n = scanner.nextInt();
        System.out.println("Введите измерения тока (I):");
        for (int i = 0; i < n; i++) {
            double value = scanner.nextDouble();
            I.add(value);
        }
        System.out.println("Введите измерения напряжения (U):");
        for (int i = 0; i < n; i++) {
            double value = scanner.nextDouble();
            U.add(value);
        }
        double num = 0;
        double den = 0;
        for (int i = 0; i < n; i++) {
            num += I.get(i) * U.get(i);
            den += I.get(i) * I.get(i);
        }
        System.out.println("Сопротивление (R): " + num / den);
    }
}
```

Работа программы представлена на рисунке 1.

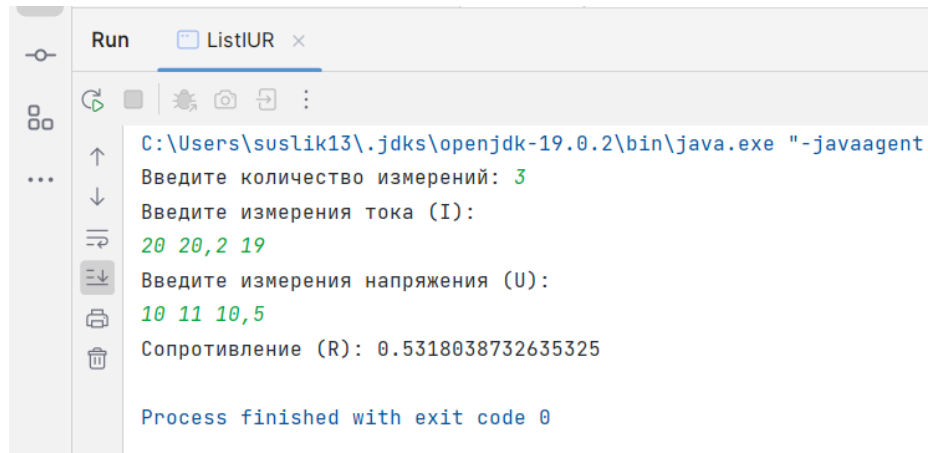


Рисунок 1 – Работа программы ListIUR.java

Задание 2 (Вариант 1, Задание 3):

С использованием множества выполнить попарное суммирование произвольного конечного ряда чисел по следующим правилам: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число

Листинг программы:

Код класса SetSum:

```
public class SetSum {
    public static void main(String[] args) {
        Set<Integer> nums_set = new TreeSet<>();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите количество элементов: ");
        int n = scanner.nextInt();
        for (int i = 0; i < n; i++) {
            nums_set.add(i + 1);
            System.out.print(i + 1 + " ");
        }
        System.out.println();
        Set<Integer> nextSet = new TreeSet<>();
        while (nums_set.size() > 1) {
            for (Iterator<Integer> it = nums_set.iterator(); it.hasNext(); ) {
                int next_a = it.next();
                it.remove();
                int next_b;
                if (it.hasNext()) {
                    next_b = it.next();
                    it.remove();
                } else {
                    next_b = 0;
                }
                nextSet.add(next_a + next_b);
                System.out.print(next_a + next_b + " ");
            }
        }
        System.out.println();
    }
}
```

```

        for (Iterator<Integer> it = nextSet.iterator(); it.hasNext(); ) {
            int next_a = it.next();
            it.remove();
            int next_b;
            if (it.hasNext()) {
                next_b = it.next();
                it.remove();
            } else {
                next_b = 0;
            }
            nums_set.add(next_a + next_b);
            System.out.print(next_a + next_b + " ");
        }
        System.out.println();
    }
    System.out.println("Сумма: " + nums_set.iterator().next());
}
}

```

Работа программы представлена на рисунке 2.



Рисунок 2 – Работа программы SetSum.java

Задание 3 (Вариант 2, 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).

Листинг программы:

Код класса NumList:

```
public class NumList implements Collection {
    private List<Integer> store;

    public NumList() {
        store = new ArrayList<>();
    }

    @Override
    public boolean add(Object o) {
        return store.add((int) o);
    }

    @Override
    public boolean remove(Object o) {
        return store.remove(o);
    }

    @Override
    public boolean addAll(Collection collection) {
        return store.addAll(collection);
    }

    @Override
    public boolean removeIf(Predicate filter) {
        return store.removeIf(filter);
    }

    @Override
    public boolean removeAll(Collection collection) {
        return store.retainAll(collection);
    }

    public int closest(int value) {
        Iterator iterator = this.iterator();
        int minimal = (int)iterator.next();
        while (iterator.hasNext()) {
            int element = (int)iterator.next();
            if (Math.abs(element - value) < Math.abs(minimal - value)) {
                minimal = element;
            }
        }
        return minimal;
    }
}
```

Работа программы представлена на рисунке 3.

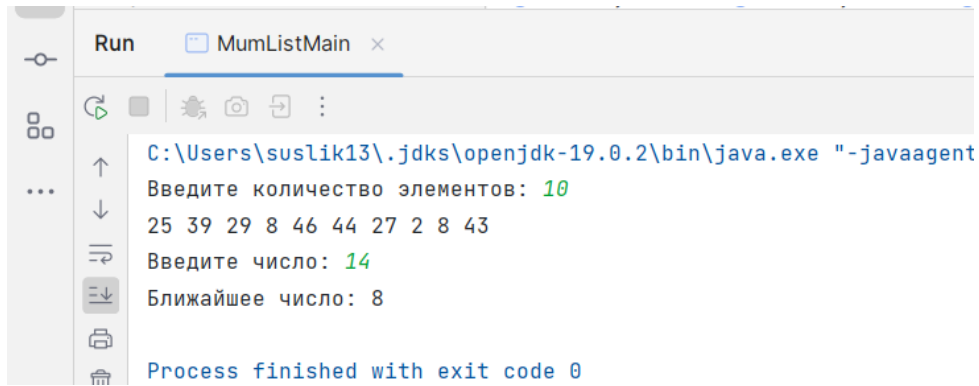


Рисунок 3 – Работа программы NumsListMain.java

Задание 4 (Вариант 2, Задание 2):

Реализовать класс, моделирующий работу N-местной автостоянки. Машина подъезжает к определенному месту и едет вправо, пока не встретится свободное место. Класс должен поддерживать методы, обслуживающие приезд и отъезд машины.

Листинг программы:

Код класса Parking:

```
public class Parking {
    private int[] parking;
    private int count;
    private int hash;

    public Parking(int n, int hash) {
        parking = new int[n];
        this.hash = hash;
        count = 0;
    }

    public void addCar(int car_num) {
        if (count == parking.length) {
            System.out.println("Нет свободного места!");
            return;
        }
        int index = car_num % parking.length % this.hash;
        while (parking[index] != 0) {
            index = (index + 1) % parking.length;
        }
        parking[index] = car_num;
        count++;
        System.out.println("Машина " + car_num + " заехала на место " + index + ".
Осталось мест: " + (parking.length - count));
    }

    public void removeCar(int car_num) {
        if (count == 0) {
            System.out.println("Парковка свободна!");
        }
    }
}
```

```

        return;
    }
    int index = car_num % parking.length % this.hash;
    int init = index;
    while (parking[index] != car_num) {
        index = (index + 1) % parking.length;
        if (index == init) {
            System.out.println("Машины " + car_num + " нет на парковке");
            return;
        }
    }
    parking[index] = 0;
    count--;
    System.out.println("Машина " + car_num + " уехала с места " + index + ".  
Осталось мест: " + (parking.length - count));
}
}

```

Работа программы представлена на рисунке 4.

```

C:\Users\suslik13\jdk\openjdk-19.0.2\bin\java.exe "-javaagent
Введите количество машин: 10
Машина 9 заехала на место 9. Осталось мест: 19
Машина 50 заехала на место 10. Осталось мест: 18
Машина 34 заехала на место 14. Осталось мест: 17
Машина 41 заехала на место 1. Осталось мест: 16
Машина 2 заехала на место 2. Осталось мест: 15
Машина 44 заехала на место 4. Осталось мест: 14
Машина 9 заехала на место 11. Осталось мест: 13
Машина 50 заехала на место 12. Осталось мест: 12
Машина 47 заехала на место 7. Осталось мест: 11
Машина 10 заехала на место 13. Осталось мест: 10
Введите номер машины: 4
Машины 4 нет на парковке
Введите номер машины: 9
Машина 9 уехала с места 9. Осталось мест: 11
Введите номер машины: 1
Машины 1 нет на парковке
Введите номер машины: 9
Машина 9 уехала с места 11. Осталось мест: 12
Введите номер машины: 41
Машина 41 уехала с места 1. Осталось мест: 13
Process finished with exit code 0

```

Рисунок 4 – Работа программы ParkingMain.java

Вывод

В ходе выполнения лабораторной работы были изучены коллекции в языке программирования Java.