



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

«Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 2
по курсу «Языки и методы программирования»
«Разработка простейшего класса на языке Java.»
«Вариант 18»

Студент группы ИУ9-21Б: Пенкин А. Д.

Преподаватель: Посевин Д. П.

Москва 2023

1 Цель

Целью данной работы является изучение базовых возможностей языка Java.

2 Условие

Выполнение лабораторной работы заключается в составлении на языке Java класса, представляющего матрицу расстояний между всеми парами из n городов с операцией вычисления длины пути, заданного последовательностью посещаемых городов. И реализация тестирующего его файла.

3 Код решения

1. Cities.java

```
public class Cities {
    private int n;
    private int [][] roads;
    public Cities(int k){
        roads = new int[k][k];
        for (int i = 0; i < k; i++){
            for (int j = 0; j < k; j++){
                if (i == j)
                    this.roads[i][j] = 0;
                else
                    this.roads[i][j] = -1;
            }
        }
        n = k;
    }
    public void setRoad(int a, int b, int l){
        System.out.println("дорога из *" + a + " * в *" + b + " * равна: " + l);
        this.roads[a][b] = l;
        this.roads[b][a] = l;
    }
    public void outMatrix(){
        for (int i = 0; i < n; i++){
            for (int j = 0; j < n; j++) {
```

```

        System.out.print(roads[i][j] + " ");
    }
    System.out.println();
}
}
public int path(int[] nums){
    int sum = 0;
    System.out.print("пройдём по пути городов под номерами: ");
    for (int i = 0; i < nums.length - 1; i++){
        System.out.print(nums[i] + "->");
    }
    System.out.println(nums[nums.length - 1]);
    for (int i = 0; i < nums.length - 1; i++){
        if (this.roads[nums[i]][nums[i + 1]] < 0) {
            System.out.print("дороги из города *" + nums[i]);
            System.out.println("* в город *" + nums[i + 1] + "*" не существует,");
            System.out.print("пожалуйста, создайте дорогу или ");
            System.out.println("выберите другую последовательность городов.");
            return -1;
        }
        sum += this.roads[nums[i]][nums[i + 1]];
    }
    return sum;
}
public int getRoad(int a, int b){
    return roads[a][b];
}
}

```

2. Test.java

```

public class Test {
    public static void main(String[] args) {
        int n = 5;
        System.out.println("пусть количество городов: 5");
        Cities loc1 = new Cities(n);
        System.out.println();
        System.out.println("начальная матрица:");
        loc1.outMatrix();
        System.out.println();
    }
}

```

```

    System.out.println("запишем некоторые дороги");
    loc1.setRoad(1, 2, 25);
    loc1.setRoad(1, 0, 10);
    loc1.setRoad(1, 4, 65);
    loc1.setRoad(3, 4, 3);
    loc1.setRoad(0, 2, 1);
    loc1.setRoad(2, 3, 12);
    System.out.println();
    System.out.println("новая матрица городов: ");
    loc1.outMatrix();
    System.out.println();
    int [] a = new int[] {0, 3, 2, 3};
    int [] b = new int[] {0, 1, 2, 3};
    int path1 = loc1.path(a);
    if (path1 > 0)
        System.out.println("этот путь равен: " + path1);
    System.out.println();
    int path2 = loc1.path(b);
    if (path2 > 0)
        System.out.println("этот путь равен: " + path2);
}

}

```

4 Результаты работы программы

```
пусть количество городов: 5

начальная матрица:
0 -1 -1 -1 -1
-1 0 -1 -1 -1
-1 -1 0 -1 -1
-1 -1 -1 0 -1
-1 -1 -1 -1 0
```

Рис. 1 — создание объекта матрицы расстояний

```
запишем некоторые дороги
дорога из *1* в *2* равна: 25
дорога из *1* в *0* равна: 10
дорога из *1* в *4* равна: 65
дорога из *3* в *4* равна: 3
дорога из *0* в *2* равна: 1
дорога из *2* в *3* равна: 12

новая матрица городов:
0 10 1 -1 -1
10 0 25 -1 65
1 25 0 12 -1
-1 -1 12 0 3
-1 65 -1 3 0
```

Рис. 2 — создание дорог, изменение матрицы

```
пройдём по пути городов под номерами: 0->3->2->3
дороги из города *0* в город *3* не существует,
пожалуйста, создайте дорогу или выберите другую последовательность городов.

пройдём по пути городов под номерами: 0->1->2->3
этот путь равен: 47

Process finished with exit code 0
```

Рис. 3 — подсчёт пути