Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(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»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Целью данной работы является изучение базовых возможностей языка Java.

2. Персональный вариант

Класс бинарных отношений на множестве целых чисел от 0 до n с двумя операциями: проверка принадлежности пары чисел отношению; вычисление транзитивного замыкания отношения.

3. Решение

3.1. Код

```
import java.util.Random;

public class Test {
    public static void main(String[] args) {
        Random rand = new Random();
        int n = 5 ; //+ rand.nextInt(10);
        BinaryRelation bin = new BinaryRelation(n);
        System.out.printf("Current size: %d\n", n+1);

        for (int i = 0; i < n; i++) {
            int x = rand.nextInt(n+1), y = rand.nextInt(n+1);
            System.out.printf("Adding new pair: (%d, %d)\n", x,
            ↪  y);
            bin.addPair(x, y);
        }

        System.out.println("Relations matrix:");
        bin.printRelations();

        bin.computeTransitiveClosure();
        System.out.println("Transitive closure:");
        System.out.println(bin.toString());

        for (int i = 0; i < n / 2; i++) {
            int x = rand.nextInt(n+1), y = rand.nextInt(n+1);
            System.out.printf("Pair (%d, %d) in relation? %s\n",
            ↪  x, y, bin.isPairInRelations(x, y) ? "yes" :
            ↪  "no");
        }
    }
}
```

Код 1: Test.java

```
public class BinaryRelation {
    private int size;
    private int[][] relations;

    public BinaryRelation(int n) {
        this.size = n;
        this.relations = new int[n+1][n+1];
    }

    public void addPair(int x, int y) {
        if (Math.min(x, y) < 0 || Math.max(x, y) > this.size) {
            System.out.printf("Can't add pair, invalid values
            ↪  (%d, %d).\n", x, y);
        } else {
            relations[x][y] = 1;
        }
    }

    public boolean isPairInRelations(int x, int y) {
        if (Math.min(x, y) < 0 || Math.max(x, y) > this.size) {
            System.out.printf("Invalid pair (%d, %d).\n", x, y);
            return false;
        } else {
            return relations[x][y] == 1;
        }
    }

    //
    ↪  https://neerc.ifmo.ru/wiki/index.php?title=Алгоритм_Флойда
    public void computeTransitiveClosure() {
        for (int k = 0; k <= this.size; k++) {
            for (int i = 0; i <= this.size; i++) {
                for (int j = 0; j <= this.size; j++) {
                    this.relations[i][j] |=
                    ↪  (this.relations[i][k] &
                    ↪  this.relations[k][j]);
                }
            }
        }
    }

    public String toString() {
```

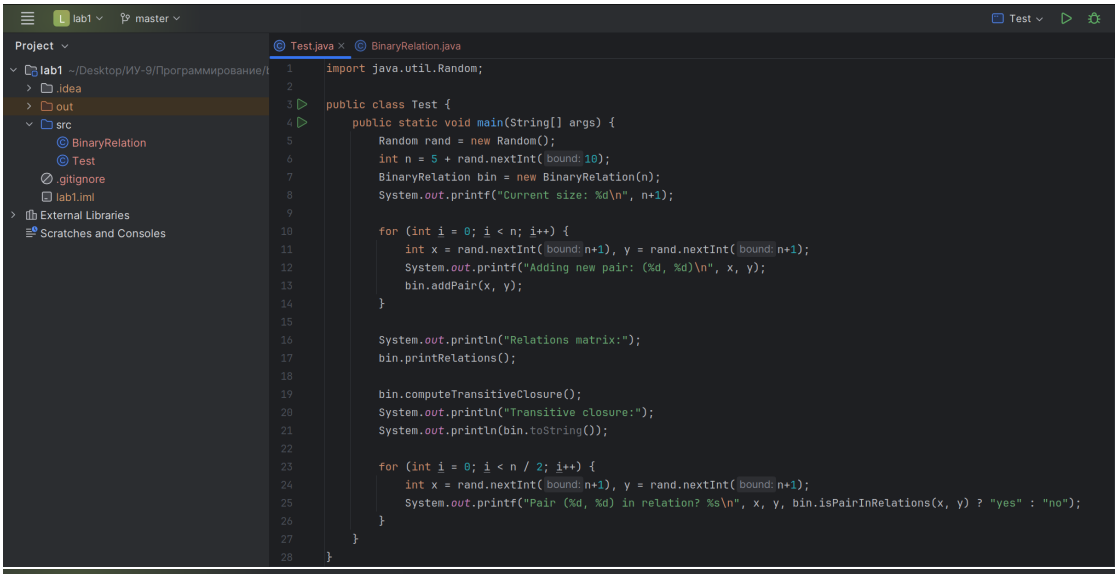
```
        String result = "";
        for (int i = 0; i <= this.size; i++) {
            for (int j = 0; j <= this.size; j++) {
                result += (relations[i][j] == 0) ? "0" : "1";
                result += (j != this.size) ? " " : "";
            }
            result += (i != this.size) ? "\n" : "";
        }
        return result;
    }

    public void printRelations() {
        System.out.println(this.toString());
    }
}
```

Код 2: BinaryRelation.java

3.2. Скриншоты

```
[thecakeisfalse@desktop-sc src]$ java Test
Current size: 6
Adding new pair: (2, 2)
Adding new pair: (0, 4)
Adding new pair: (2, 4)
Adding new pair: (5, 3)
Adding new pair: (3, 1)
Relations matrix:
0 0 0 0 1 0
0 0 0 0 0 0
0 0 1 0 1 0
0 1 0 0 0 0
0 0 0 0 0 0
0 0 0 1 0 0
Transitive closure:
0 0 0 0 1 0
0 0 0 0 0 0
0 0 1 0 1 0
0 1 0 0 0 0
0 0 0 0 0 0
0 1 0 1 0 0
Pair (0, 3) in relation? no
Pair (0, 2) in relation? no
[thecakeisfalse@desktop-sc src]$
```

Рис. 1: Пример работы



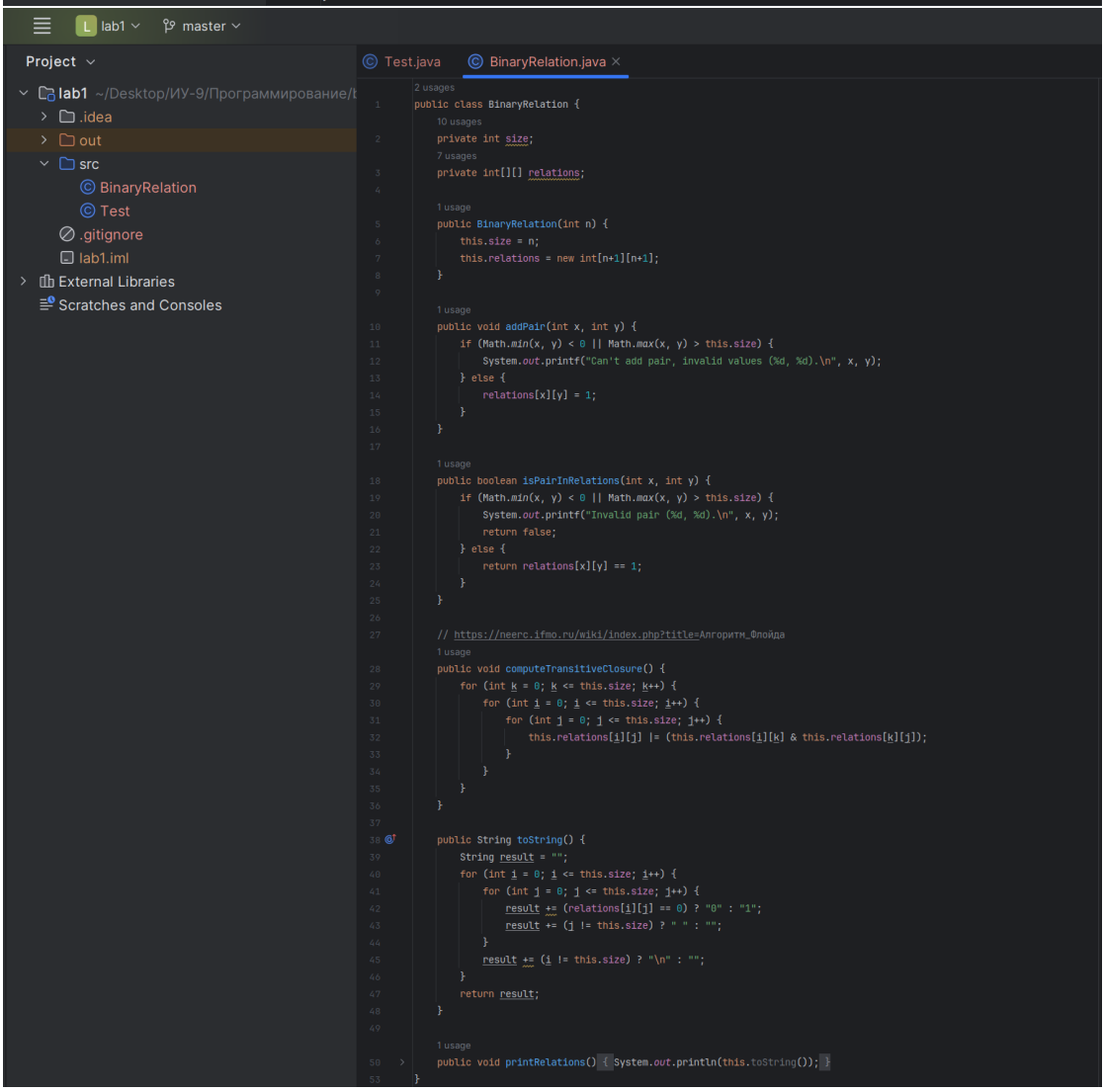


Рис. 2: Исходный код