

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 №5-3
по курсу: «Языки и методы программирования»
«Монады в языке Java»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Приобретение навыков использования монад Optional и Stream в программах на языке Java.

2. Персональный вариант

Полином с вещественными коэффициентами, для которого определены операции:

1. порождение потока значений полинома в точках, полученных делением отрезка $[a, b]$ на n равных частей;
2. вычисление определённого интеграла полинома на отрезке $[a, b]$ (если $b < a$, то будем считать, что интеграл не определён).

Проверить работу первой операции нужно путём подсчёта сумм отрицательных и неотрицательных значений полинома в потоке.

3. Решение

3.1. Код

```
import java.util.Random;
import java.util.stream.IntStream;

public class Test {

    public static Quad newQuad(Random rnd) {
        return new Quad(
            new Point(rnd.nextInt(-10, 10), rnd.nextInt(-10,
                → 10)),
            new Point(rnd.nextInt(-10, 10), rnd.nextInt(-10,
                → 10)),
            new Point(rnd.nextInt(-10, 10), rnd.nextInt(-10,
                → 10)),
            new Point(rnd.nextInt(-10, 10), rnd.nextInt(-10,
                → 10)))
    };

    public static void main(String[] args) {
        QuadSet s = new QuadSet();

        Random r = new Random();
        int n = r.nextInt(5, 10);

        IntStream.range(0, n)
```

```

        .mapToObj(x -> newQuad(r))
        .forEach(s::addQuad);

    IntStream.range(0, 20)
        .forEach(x -> s.getAllQuadsArea()
            .filter(y -> (10 * x <= y && y
                < 10 * (x+1)))
        .forEach(System.out::println));

    System.out.println(s.getMaximumDiagonalsSum().get());
}
}

```

Код 1: Test.java

```

public class Point {
    private Integer x, y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    double distanceTo(Point other) {
        return Math.sqrt(Math.pow(x - other.x, 2) + Math.pow(y -
            other.y, 2));
    }

    Point vectorTo(Point other) {
        return new Point(x - other.x, y - other.y);
    }

    Integer getX() { return x; }
    Integer getY() { return y; }
}

```

Код 2: Point.java

```

public class Quad {
    private Point a, b, c, d;

    public Quad(Point a, Point b, Point c, Point d) {

```

```

        this.a = a;
        this.b = b;
        this.c = c;
        this.d = d;
    }

    public Double getSurfaceArea() {
        double A = a.distanceTo(b), B = b.distanceTo(c),
               C = c.distanceTo(d), D = d.distanceTo(a);

        double p = (A + B + C + D) / 2;
        return Math.sqrt((p - A) * (p - B) * (p - C) * (p - D));
    }

    public Double getSumOfDiagonals() {
        return a.distanceTo(c) + b.distanceTo(d);
    }

    public String toString() {
        return "Quad: area=" + getSurfaceArea().toString() + ", "
            + "sum of diagonals=" + getSumOfDiagonals().toString();
    }
}

```

Код 3: Quad.java

```

import java.util.ArrayList;
import java.util.Comparator;
import java.util.Optional;
import java.util.stream.Stream;

public class QuadSet {
    private ArrayList<Quad> quads;

    public QuadSet() {
        quads = new ArrayList<>();
    }

    public void addQuad(Quad q) {
        quads.add(q);
    }

    public Stream<Double> getAllQuadsArea() {
        return quads.stream().map(Quad::getSurfaceArea);
    }
}

```

```
}

    public Optional<Quad> getMaximumDiagonalsSum() {
        return quads.stream().max(new DiagonalsSumComparator());
    }

    class DiagonalsSumComparator implements Comparator<Quad> {
        public int compare(Quad a, Quad b) {
            return (int)(a.getSumOfDiagonals() -
                    b.getSumOfDiagonals());
        }
    }
}
```

Код 4: QuadSet.java

3.2. Скриншоты

```
46.939221753585
54.346933210384485
82.61491492791738
98.65441547260562
91.31799828160275
109.60331255636034
114.0788331515063
122.82389274268895
135.73713410141
Quad: area=54.346933210384485, sum of diagonals=33.77782151846333

Process finished with exit code 0
```

Рис. 1: Пример работы программы