

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 №5-2  
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

Множество четырёхугольников на плоскости с операциями:

1. порождение потока площадей выпуклых четырёхугольников множества;
2. поиск четырёхугольника, имеющего максимальную сумму длин диагоналей.

Проверить работу первой операции нужно путём подсчёта количества четырёхугольников, площади которых принадлежат интервалам:

$[0, 10), [10, 20), \dots, [90, 100)$

## 3. Решение

### 3.1. Код

```
public class Test {
    public static void main(String[] args) {
        Polynom p = new Polynom();
        p.addCoeff(1);
        p.addCoeff(0);
        p.addCoeff(-1);

        double a = -7, b = 7;
        int n = 5;

        p.getPointsOnRange(a, b, n).forEach(x ->
            System.out.println(x.get()));
        System.out.println(p.getDefiniteIntegral(a,
            b).getAsDouble());
    }
}
```

Код 1: Test.java

```
public class Pair<T, U> {
    private T first;
    private U second;

    public Pair(T first, U second) {
```

```

        this.first = first;
        this.second = second;
    }

    public T getFirst() {
        return first;
    }

    public U getSecond() {
        return second;
    }
}

```

Код 2: Point.java

```

import java.util.ArrayList;
import java.util.Optional;
import java.util.OptionalDouble;
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Polynom {
    ArrayList<Pair<Double, Integer>> coeffs;
    int size;

    Polynom() {
        coeffs = new ArrayList<>();
        size = 0;
    }

    public void addCoeff(double ak) {
        coeffs.add(new Pair<>(ak, size));
        ++size;
    }

    Optional<Double> calculateValueInPoint(double x) {
        return coeffs.stream()
            .map(y -> (Math.pow(x, y.getSecond()) *
                         y.getFirst()))
            .reduce(Double::sum);
    }

    public Stream<Optional<Double>> getPointsOnRange(double a,
        double b, int n) {

```

```
    double s = Math.min(a, b), e = Math.max(a, b);
    double step = (e - s) / (double)n;

    return IntStream.range(0, n + 1)
        .mapToDouble(x -> (s + step * x))
        .mapToObj(this::calculateValueInPoint);
}

public OptionalDouble getDefiniteIntegral(double a, double
→ b) {
    double eps = 1e-5;
    int n = (int)Math.ceil(Math.abs(b - a) / eps);

    double sign = (a <= b ? 1 : -1);

    return getPointsOnRange(a+eps, b-eps, n)
        .mapToDouble(x -> sign * (x.orElse(0.0) * eps))
        .reduce(Double::sum);
}
}
```

Код 3: Quad.java

## 3.2. Скриншоты

```
-48.0
-16.64
-0.9600000000000011
-0.9599999999999996
-16.639999999999993
-48.0
-214.66649333263436
```

```
Process finished with exit code 0
```

Рис. 1: Пример работы программы