

## **Лабораторная работа №2**

**Разработка классов для работы с табулированными  
функциями**

**Цель:** Разработать набор классов для работы с функциями  
одной переменной, заданными в табличной форме.

**Филянов Кирилл Владимирович**  
**6203-010302D**

## **Содержание**

**Задание 1:** Создание пакета functions

**Задание 2:** Класс FunctionPoint

**Задания 3-6:** Класс TabulatedFunction

**Задание 7:** Тестирование

### **Задание 1**

Создание пакета functions

**Ход выполнения:** Создан пакет functions для организации классов программы.

В Java пакеты позволяют логически группировать связанные классы.

**Результат:** Создана структура проекта с пакетом functions, содержащим классы FunctionPoint и TabulatedFunction.

### **Задание 2**

Класс FunctionPoint

**Ход выполнения:** Реализован класс для представления точки функции с координатами (x, y). Созданы три конструктора:

- FunctionPoint(double x, double y) - создание точки с заданными координатами
- FunctionPoint(FunctionPoint point) - копирующий конструктор
- FunctionPoint() - конструктор по умолчанию (0, 0)

**Результат:** Класс инкапсулирует координаты точки и предоставляет геттеры/сеттеры для доступа к ним.

### **Задания 3-6**

Класс TabulatedFunction

**Ход выполнения:** Реализован класс для работы с табулированной функцией

Конструкторы:

- Равномерное распределение точек с нулевыми значениями
- Равномерное распределение с заданными значениями Y

Основные методы:

- getLeftDomainBorder(), getRightDomainBorder() - границы области определения
- getFunctionValue(double x) - вычисление значения с линейной интерполяцией

Методы работы с точками:

- Геттеры для получения информации о точках
- Сеттеры с проверкой сохранения порядка точек по X
- addPoint(), deletePoint() - изменение количества точек

**Результат:** Класс обеспечивает хранение упорядоченных точек, линейную интерполяцию и безопасное управление точками.

## Задание 7

### Тестирование

**Ход выполнения:** Создан класс Main для тестирования функциональности

- Создание функции  $y = x^2$
- Проверка интерполяции в различных точках
- Тестирование операций с точками (добавление, удаление, изменение)

### Результат вывода в консоль:

```
"C:\Users\Кириха Пират\Desktop\ЖДК\bin\java.exe" "-javaagent:C:\Prog  
Тестирование исходной функции:  
f(-3,0) = не определена  
f(-2,0) = 4,00  
f(-1,0) = 1,00  
f(0,0) = 0,00  
f(0,5) = 0,50  
f(1,0) = 1,00  
f(2,0) = 4,00  
f(3,0) = не определена  
  
Точки функции:  
[-2,00; 4,00] [-1,00; 1,00] [0,00; 0,00] [1,00; 1,00] [2,00; 4,00]  
  
После изменения точки:  
f(-3,0) = не определена  
f(-2,0) = 4,00  
f(-1,0) = 1,00  
f(0,0) = 0,00  
f(0,5) = 0,50  
f(1,0) = 1,00  
f(2,0) = 4,00  
f(3,0) = не определена  
  
Точки функции:  
[-2,00; 4,00] [-1,00; 1,00] [0,00; 0,00] [1,00; 1,00] [2,00; 4,00]  
  
После добавления точки:  
f(-3,0) = не определена  
f(-2,0) = 4,00  
f(-1,0) = 1,00  
f(0,0) = 0,00  
f(0,5) = 0,50  
f(1,0) = 1,00  
f(2,0) = 4,00  
f(3,0) = не определена  
  
Точки функции:  
[-2,00; 4,00] [-1,00; 1,00] [0,00; 0,00] [1,00; 1,00] [2,00; 4,00] [2,50; 6,25]  
  
После удаления точки:  
f(-3,0) = не определена  
f(-2,0) = 4,00  
f(-1,0) = 2,00  
f(0,0) = 0,00  
f(0,5) = 0,50  
f(1,0) = 1,00  
f(2,0) = 4,00  
f(3,0) = не определена  
  
Точки функции:  
[-2,00; 4,00] [0,00; 0,00] [1,00; 1,00] [2,00; 4,00] [2,50; 6,25]
```