

## **Лабораторная работа №3**

**Дополнение пакета для работы с функциями одной переменной, заданными в табличной форме.**

**Цель:** Дополнить пакет для работы с функциями одной переменной, заданными в табличной форме, добавив классы исключений, новый класс функций и базовый интерфейс.

**Филянов Кирилл Владимирович  
6203-010302D**

# Содержание

**Задания 1-2: Исключения**

**Задания 3-4: Модификация классов**

**Задания 5-6: Интерфейс и связный список**

**Задание 7: Тестирование**

**Результаты выполнения**

**Задания 1-2: Изучение и создание исключений**

**Ход выполнения:**

**Изучены стандартные классы исключений Java:**

- Exception - базовый класс проверяемых исключений
- IndexOutOfBoundsException - выход за границы коллекции
- ArrayIndexOutOfBoundsException - частный случай для массивов
- IllegalArgumentException - некорректные аргументы метода
- IllegalStateException - некорректное состояние объекта
- Созданы пользовательские исключения:
- FunctionPointIndexOutOfBoundsException - наследует от IndexOutOfBoundsException
- InappropriateFunctionPointException - наследует от Exception

**Результат:**

```
package functions;

public class FunctionPointIndexOutOfBoundsException extends IndexOutOfBoundsException { 14 usages
    public FunctionPointIndexOutOfBoundsException() { 7 usages
        super();
    }

    public FunctionPointIndexOutOfBoundsException(String message) { 3 usages
        super(message);
    }
}

package functions;

public class InappropriateFunctionPointException extends Exception { 21 usages
    public InappropriateFunctionPointException() { no usages
        super();
    }

    public InappropriateFunctionPointException(String message) { 6 usages
        super(message);
    }
}
```

## **Задания 3-4: Модификация TabulatedFunction и создание LinkedListTabulatedFunction**

### **Ход выполнения:**

#### **Модифицирован класс TabulatedFunction для выбрасывания исключений:**

- Конструкторы выбрасывают IllegalArgumentException при некорректных параметрах
- Методы доступа выбрасывают FunctionPointIndexOutOfBoundsException
- Методы изменения выбрасывают InappropriateFunctionPointException
- deletePoint() выбрасывает IllegalStateException при удалении последней точки
- Создан класс LinkedListTabulatedFunction с использованием двусвязного циклического списка:
- Внутренний класс FunctionNode для элементов списка
- Голова списка не содержит данных
- Реализована оптимизация доступа с кэшированием

### **Результат:**

- Класс корректно обрабатывает все исключительные ситуации
- Связный список обеспечивает динамическое изменение размера
- Реализованы все требуемые методы работы со списком

## **Задания 5-6: Создание интерфейса TabulatedFunction**

### **Ход выполнения:**

- Создан интерфейс TabulatedFunction с объявлениями общих методов
- Класс TabulatedFunction переименован в ArrayTabulatedFunction
- Оба класса (ArrayTabulatedFunction и LinkedListTabulatedFunction) реализуют интерфейс TabulatedFunction
- Оптимизированы методы в LinkedListTabulatedFunction:
- Быстрый доступ к граничным элементам
- Кэширование последнего доступа
- Оптимизированная вставка в начало/конец

## Результат:

```
package functions;

public interface TabulatedFunction { 22 usages 2 implementations
    // Основные методы
    double getLeftDomainBorder(); 5 usages 2 implementations
    double getRightDomainBorder(); 5 usages 2 implementations
    double getFunctionValue(double x); 7 usages 2 implementations

    // Методы работы с точками
    int getPointsCount(); 5 usages 2 implementations
    FunctionPoint getPoint(int index); 2 usages 2 implementations
    void setPoint(int index, FunctionPoint point) throws InappropriateFunctionPointException; 3 usages 2 implementations
    double setPointX(int index); 5 usages 2 implementations
    void setPointX(int index, double x) throws InappropriateFunctionPointException; 2 usages 2 implementations
    double setPointY(int index); 2 usages 2 implementations
    void setPointY(int index, double y); 5 usages 2 implementations
    void deletePoint(int index); 4 usages 2 implementations
    void addPoint(FunctionPoint point) throws InappropriateFunctionPointException; 5 usages 2 implementations
}
```

## Задание 7: Тестирование работы классов

### Ход выполнения:

- Создан класс Main для комплексного тестирования
- Реализовано тестирование через интерфейс TabulatedFunction
- Протестированы все исключительные ситуации:
- Некорректные параметры конструкторов
- Выход за границы индексов
- Нарушение порядка точек
- Некорректные операции удаления
- Проведено сравнительное тестирование обеих реализаций

### Результаты выполнения работы:

```
1. ТЕСТИРОВАНИЕ ArrayTabulatedFunction:
Создан ArrayTabulatedFunction [0, 4] с 5 точками
Установлены значения у = x^2

Демонстрация интерполяции:
f(-0,5) = не определена
f(0,0) = 0,0000
f(0,5) = 0,5000
f(1,0) = 1,0000
f(1,5) = 2,5000
f(2,0) = 4,0000
f(2,5) = 6,5000
f(3,0) = 9,0000
f(3,5) = 12,5000
f(4,0) = 16,0000
f(4,5) = не определена

Тест операций с точками:
Исходные точки:
Точки: [0,00; 0,000] [1,00; 1,000] [2,00; 4,000] [3,00; 9,000] [4,00; 16,000]
Добавляем точку (-0,5, 0,25) в начало:
Точки: [-0,50; 0,250] [0,00; 0,000] [1,00; 1,000] [2,00; 4,000] [3,00; 9,000] [4,00; 16,000]
Изменяем точку с индексом 0 на (-0,3, 0,09):
Точки: [-0,30; 0,090] [0,00; 0,000] [1,00; 1,000] [2,00; 4,000] [3,00; 9,000] [4,00; 16,000]
Удаляем точку с индексом 0:
Точки: [0,00; 0,000] [1,00; 1,000] [2,00; 4,000] [3,00; 9,000] [4,00; 16,000]
Добавляем точку (1,5, 2,25):
Точки: [0,00; 0,000] [1,00; 1,000] [1,50; 2,250] [2,00; 4,000] [3,00; 9,000] [4,00; 16,000]
Все операции выполнены успешно
```

```
2. ТЕСТИРОВАНИЕ LinkedListTabulatedFunction:  
Создан LinkedListTabulatedFunction [0, 4] с 5 точками  
Установлены значения y = sin(x)
```

Демонстрация интерполяции:

```
f(-0,5) = не определена  
f(0,0) = 0,0000  
f(0,5) = 0,4207  
f(1,0) = 0,8415  
f(1,5) = 0,8754  
f(2,0) = 0,9093  
f(2,5) = 0,5252  
f(3,0) = 0,1411  
f(3,5) = -0,3078  
f(4,0) = -0,7568  
f(4,5) = не определена
```

Тест операций с точками:

Исходные точки:

```
Точки: [0,00; 0,000] [1,00; 0,841] [2,00; 0,909] [3,00; 0,141] [4,00; -0,757]
```

Добавляем точку (-0,5, -0,479) в начало:

```
Точки: [-0,50; -0,479] [0,00; 0,000] [1,00; 0,841] [2,00; 0,909] [3,00; 0,141] [4,00; -0,757]
```

Изменяем точку с индексом 0 на (-0,3, -0,295):

```
Точки: [-0,30; -0,295] [0,00; 0,000] [1,00; 0,841] [2,00; 0,909] [3,00; 0,141] [4,00; -0,757]
```

Удаляем точку с индексом 0:

```
Точки: [0,00; 0,000] [1,00; 0,841] [2,00; 0,909] [3,00; 0,141] [4,00; -0,757]
```

Добавляем точку (1,5, 0,997):

```
Точки: [0,00; 0,000] [1,00; 0,841] [1,50; 0,997] [2,00; 0,909] [3,00; 0,141] [4,00; -0,757]
```

Все операции выполнены успешно

### 3. СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ:

Создаем одинаковые функции разными реализациями:

Функции созданы ( $y = e^x$ )

Сравнение значений в точках:

x = 0,0: Массив =	1,000000, Список =	1,000000 ✓
x = 0,3: Массив =	1,515485, Список =	1,515485 ✓
x = 0,7: Массив =	2,202797, Список =	2,202797 ✓
x = 1,0: Массив =	2,718282, Список =	2,718282 ✓
x = 1,3: Массив =	4,119514, Список =	4,119514 ✓
x = 1,7: Массив =	5,987824, Список =	5,987824 ✓
x = 2,0: Массив =	7,389056, Список =	7,389056 ✓
x = 2,5: Массив =	NaN, Список =	NaN ✓

Результаты полностью совпадают!

#### 4. ТЕСТИРОВАНИЕ ИСКЛЮЧЕНИЙ:

Тестируем обработку исключительных ситуаций:

##### 1. Тест конструкторов (IllegalArgumentException):

Пытаемся создать функцию с `leftX >= rightX`... `IllegalArgumentException`: Левая граница должна быть меньше правой  
Пытаемся создать функцию с `pointsCount < 2`... `IllegalArgumentException`: Количество точек должно быть не менее 2  
Пытаемся создать функцию с отрицательным `pointsCount`... `IllegalArgumentException`: Количество точек должно быть не менее 2

##### 2. Тест выхода за границы (FunctionPointIndexOutOfBoundsException):

Пытаемся получить точку с индексом -1... `FunctionPointIndexOutOfBoundsException`  
Пытаемся получить точку с индексом 10... `FunctionPointIndexOutOfBoundsException`  
Пытаемся изменить точку с индексом 5... `FunctionPointIndexOutOfBoundsException`  
Пытаемся `setPoint` с индексом -2... `FunctionPointIndexOutOfBoundsException`  
Пытаемся `deletePoint` с индексом 10... `FunctionPointIndexOutOfBoundsException`

##### 3. Тест нарушения порядка точек (InappropriateFunctionPointException):

Тест 1: Устанавливаем точку (3,y) на позицию 0... `InappropriateFunctionPointException`: Нарушение порядка точек по X  
Тест 2: Меняем X точки 1 (было 1) на 0 (равно предыдущей)... `InappropriateFunctionPointException`: Нарушение порядка точек по X  
Тест 3: Меняем X точки 1 (было 1) на 2 (равно следующей)... `InappropriateFunctionPointException`: Нарушение порядка точек по X  
Тест 4: Добавляем точку с `X=1` (точно существует)... `InappropriateFunctionPointException`: Точка с таким X уже существует  
Тест 5: Добавляем точку с `X=1.00000000001`... `InappropriateFunctionPointException`: Точка с таким X уже существует  
Тест 6: Добавляем точку с `X=1.0001` (должно быть OK)... Успешно добавлено

##### 4. Тест некорректного удаления (IllegalStateException):

Пытаемся удалить точку из функции с 2 точками... `IllegalStateException`: Нельзя удалить точку: останется меньше 2 точек  
Пытаемся удалить точку с индексом -1... `FunctionPointIndexOutOfBoundsException`

#### 5. ТЕСТИРОВАНИЕ ИНДЕКСА 0:

Тестирование операций с индексом 0:

##### 1. Установка точки с индексом 0:

Устанавливаем корректную точку (0.5, 1) на позицию 0... Успешно!  
Устанавливаем некорректную точку (2, 4) на позицию 0... `InappropriateFunctionPointException`: Нарушение порядка точек по X

##### 2. Удаление точки с индексом 0:

Удаляем точку с индексом 0 из 5 точек... Успешно! Осталось 4 точек  
Удаляем точку с индексом 0 из 5 точек... Успешно! Осталось 4 точек

##### 3. Изменение X/Y точки с индексом 0:

Изменяем X точки 0 с 1 на 1.5... Успешно! Новое X = 1.5  
Изменяем Y точки 0 на 100... Успешно! Новое Y = 100.0