

Лабораторная работа № 3

Выполнила: Рапп Ксения

Александровна

Группа: 6204-010302D

Оглавление

Задание 1 - Изучение классов исключений	3
Задание 2 - Создание классов исключений	3
Задание 3 - Модификация ArrayTabulatedFunction	4
Задание 4 - Реализация LinkedListTabulatedFunction	4
Задание 5 - Методы для табулированной функции	5
Задание 6 - Создание интерфейса	5
Задание 7 - Тестирование	5-6
Результаты работы	6-7

Задание 1

Изучила классы исключений Java: Exception, IndexOutOfBoundsException, ArrayIndexOutOfBoundsException, IllegalArgumentException, IllegalStateException.

Поняла иерархию и назначение каждого класса исключений.

Задание 2

Создала два класса исключений в пакете functions:

1. FunctionPointIndexOutOfBoundsException - наследует от IndexOutOfBoundsException, используется для выхода за границы набора точек.

```
package functions;
public class FunctionPointIndexOutOfBoundsException extends
IndexOutOfBoundsException {
    public FunctionPointIndexOutOfBoundsException() {
        super("Выход за границы набора точек");
    }
    public FunctionPointIndexOutOfBoundsException(String message) {
        super(message);
    }
}
```

2. InappropriateFunctionPointException - наследует от Exception, используется для некорректных операций с точками.

```
package functions;
public class InappropriateFunctionPointException extends Exception {
    public InappropriateFunctionPointException() {
        super("Некорректная точка функции");
    }
    public InappropriateFunctionPointException(String message) {
        super(message); }}
```

Задание 3

Модифицировала класс ArrayTabulatedFunction для выбрасывания исключений:

- Конструкторы выбрасывают `IllegalArgumentException` при неверных параметрах
- Методы доступа к точкам выбрасывают `FunctionPointIndexOutOfBoundsException` при неверных индексах
- Методы `setPoint()` и `setPointX()` выбрасывают `InappropriateFunctionPointException` при нарушении упорядоченности
- Метод `addPoint()` выбрасывает `InappropriateFunctionPointException` при дублировании X
- Метод `deletePoint()` выбрасывает `IllegalStateException` при попытке удалить точку, когда их останется меньше 3

Задание 4

Реализовала класс `LinkedListTabulatedFunction` с двусвязным циклическим списком:

- Создала внутренний класс `FunctionNode` для элементов списка
- Реализовала оптимизированный метод `getNodeByIndex()` с выбором оптимальной точки старта
- Реализовала методы для работы со списком: `addNodeToTail()`, `addNodeByIndex()`, `deleteNodeByIndex()`

```
private static class FunctionNode {  
    private FunctionPoint point;  
    private FunctionNode prev;  
    private FunctionNode next;  
  
    // Конструкторы и методы доступа  
}
```

Задание 5

Реализовала в `LinkedListTabulatedFunction` все методы, аналогичные `ArrayTabulatedFunction`:

- Конструкторы с теми же параметрами
- Методы `getPointsCount()`, `getPoint()`, `setPoint()`, `getPointX()`, `setPointX()`,
`getPointY()`, `setPointY()`
- Методы `deletePoint()`, `addPoint()`, `getLeftDomainBorder()`,
`getRightDomainBorder()`, `getFunctionValue()`
- Все методы выбрасывают соответствующие исключения

Особенность реализации - использование оптимизированного доступа к элементам через `lastAccessedNode`.

Задание 6

- Переименовала `TabulatedFunction` в `ArrayTabulatedFunction`
- Создала интерфейс `TabulatedFunction` с объявлением общих методов
- Сделала так, чтобы оба класса реализовывали этот интерфейс

```
public interface TabulatedFunction {  
    int getPointsCount();  
    FunctionPoint getPoint(int index);  
    void setPoint(int index, FunctionPoint point);  
    // ... другие методы  
}
```

Задание 7

Протестировала работу классов в методе `main()`:

- Создала тесты для обеих реализаций функций
- Проверила обработку исключительных ситуаций
- Убедилась, что обе реализации работают корректно

```
// Тестирование обеих реализаций
testFunction(new ArrayTabulatedFunction(0, 4, 5), "Array");
testFunction(new LinkedListTabulatedFunction(0, 4, 5), "Linked List");
```

Результаты работы

Все задания выполнены успешно. Создан пакет functions, содержащий:

- ArrayTabulatedFunction.java
- LinkedListTabulatedFunction.java
- FunctionPoint.java
- FunctionPointIndexOutOfBoundsException.java
- InappropriateFunctionPointException.java
- TabulatedFunction.java

Обе реализации табулированных функций работают корректно, обрабатывают исключения и проходят тестирование.

Тестирование табулированной функции

Array tabulated function:

1. Создание Array функции $f(x) = x^2$ на интервале $[0, 4]$:

Область определения: $[0.0, 4.0]$

Количество точек: 5

Точки функции:

- 0: (0.00, 0.0000)
- 1: (1.00, 1.0000)
- 2: (2.00, 4.0000)
- 3: (3.00, 9.0000)
- 4: (4.00, 16.0000)

Linked list tabulated function:

1. Создание Linked List функции $f(x) = x^2$ на интервале $[0, 4]$:

Область определения: $[0.0, 4.0]$

Количество точек: 5

Точки функции:

0: (0.00, 0.0000)

1: (1.00, 1.0000)

2: (2.00, 4.0000)

3: (3.00, 9.0000)

4: (4.00, 16.0000)