

Лабораторная работа № 3

Выполнила: Сингатуллина Алина Марсовна

Группа 6204-010302D

Оглавление

Задание 1	3
Задание 2	3
Задание 3.....	3
Задание 4.....	3
Задание 5.....	4
Задание 6.....	4
Задание 7.....	4
Вывод.....	5

Задание 1

Ознакомилась со следующими классами исключений, входящих в API Java:

• java.lang.Exception • java.lang.IndexOutOfBoundsException • java.lang.ArrayIndexOutOfBoundsException • java.lang.IllegalArgumentException • java.lang.IllegalStateException.

Задание 2

В пакете functions создала два класса исключений:

- 1) FunctionPointIndexOutOfBoundsException – исключение выхода за границы набора точек при обращении к ним по номеру, наследует от класса IndexOutOfBoundsException;
- 2) InappropriateFunctionPointException – исключение, выбрасываемое при попытке добавления или изменения точки функции несоответствующим образом, наследует от класса Exception.

Задание 3

В разработанный ранее класс TabulatedFunction внесла изменения, обеспечивающие выбрасывание исключений методами класса. Оба конструктора класса выбрасывают исключение IllegalArgumentException. Методы getPoint(), setPoint(), getPointX(), setPointX(), getPointY(), setPointY() и deletePoint() выбрасывают исключение FunctionPointIndexOutOfBoundsException, если переданный в метод номер выходит за границы набора точек.

Методы setPoint() и setPointX() выбрасывают исключение InappropriateFunctionPointException в том случае, если координата x задаваемой точки лежит вне интервала, определяемого значениями соседних точек табулированной функции. Метод addPoint() также выбрасывает исключение InappropriateFunctionPointException, если в наборе точек функции есть точка, абсцисса которой совпадает с абсциссой добавляемой точки.

Метод deletePoint() выбрасывает исключение IllegalStateException, если на момент удаления точки количество точек в наборе менее трех.

Задание 4

Реализовала класс LinkedListTabulatedFunction на основе двусвязного циклического списка:

Создала внутренний класс FunctionNode для хранения точек и связей между элементами

Реализовала методы работы со списком: getNodeByIndex(), addNodeToTail(), addNodeByIndex(), deleteNodeByIndex()

Добавила оптимизацию доступа через кэширование последнего используемого узла

Обеспечила полную совместимость с интерфейсом TabulatedFunction

Все методы класса корректно обрабатывают исключительные ситуации

Задание 5

Для обеспечения второй функции класса `LinkedListTabulatedFunction` реализовала в классе конструкторы и методы, аналогичные конструкторам и методам класса `TabulatedFunction`. Конструкторы имеют те же параметры, методы - те же сигнатуры. Выбрасываются те же виды исключений в тех же случаях.

Задание 6

Для обеспечения полной взаимозаменяемости классов `ArrayTabulatedFunction` и `LinkedListTabulatedFunction` создала общий интерфейс `TabulatedFunction`, содержащий объявления всех методов работы с табулированной функцией. Оба класса были модифицированы для реализации этого интерфейса через ключевое слово `implements` в объявлении класса. Это позволяет использовать единую ссылочную переменную типа `TabulatedFunction` для работы с любым типом реализации.

Задание 7

Разработала класс `Main` для проверки работы функций и исключений. Ссылочную переменную для работы с объектом функции объявила типа `TabulatedFunction`, а при создании объекта указала реальный класс.
Результаты запуска `Main.java`

```
=== ТЕСТИРОВАНИЕ ARRAY ===
Функция: ArrayTabulatedFunction
Все точки функции:
x = -3.0 y = 9.0
x = -2.0 y = 4.0
x = -1.0 y = 1.0
x = 0.0 y = 0.0
x = 1.0 y = 1.0
x = 2.0 y = 4.0
x = 3.0 y = 9.0
Границы: -3.0 - 3.0
f(-2) = 4.0
f(-4) = NaN
Количество точек: 7
Точка (-2.5, 6) успешно добавлена
Точка с индексом 1 успешно изменена
Точка с индексом 3 удалена
Функция после операций:
x = -3.0 y = 9.0
x = -2.1 y = 6.2
x = -2.0 y = 4.0
x = 0.0 y = 0.0
x = 1.0 y = 1.0
x = 2.0 y = 4.0
x = 3.0 y = 9.0
-----

=== ТЕСТИРОВАНИЕ LINKEDLIST ===
Функция: LinkedListTabulatedFunction
Все точки функции:
x = -3.0 y = 9.0
x = -2.0 y = 4.0
x = -1.0 y = 1.0
```

x = 0.0 y = 0.0
x = 1.0 y = 1.0
x = 2.0 y = 4.0
x = 3.0 y = 9.0
Границы: -3.0 - 3.0

f(-2) = 4.0

f(-4) = NaN

Количество точек: 7

Точка (-2.5, 6) успешно добавлена

Точка с индексом 1 успешно изменена

Точка с индексом 3 удалена

Функция после операций:

x = -3.0 y = 9.0

x = -2.1 y = 6.2

x = -2.0 y = 4.0

x = 0.0 y = 0.0

x = 1.0 y = 1.0

x = 2.0 y = 4.0

x = 3.0 y = 9.0

=== ТЕСТИРОВАНИЕ ИСКЛЮЧЕНИЙ ===

1. Тестирование некорректных конструкторов:

Поймано исключение: Левая граница области определения должна быть меньше правой

Поймано исключение: Less than 2 points

2. Тестирование выхода за границы индексов:

Поймано исключение: Index out of range: -1

Поймано исключение: Index out of range: 10

3. Тестирование нарушения упорядоченности:

Поймано исключение: Нарушена упорядоченность точек

4. Тестирование добавления точки с существующим x:

Поймано исключение: Точка с таким X уже существует

5. Тестирование удаления при малом количестве точек:

Поймано исключение: Нельзя удалить точку: меньше 3 точек

6. Тестирование методов setPoint:

Поймано исключение: Нарушена упорядоченность точек

7. Тестирование граничных случаев:

Поймано исключение: Index out of range: -5

Поймано исключение: Index out of range: 15

8. Тестирование с LinkedListTabulatedFunction:

Поймано исключение в LinkedList: Point with this X already exists

Вывод

В ходе лабораторной работы успешно разработаны две реализации табулированной функции - на основе массива и связанного списка, обеспечивающие полную функциональную эквивалентность. Реализована система обработки исключений, а также проведено тестирование для проверки корректности работы всех методов.