

Лабораторная работа №2

Разработка классов табулированных функций и обработка исключений

Пожидаев Богдан Витальевич 6204-010302D

1. Задание 1: Изучение классов исключений

Изучены стандартные классы исключений Java: **Exception**, **IndexOutOfBoundsException**, **ArrayIndexOutOfBoundsException**, **IllegalArgumentException**, **IllegalStateException**.

Получено понимание иерархии исключений и их применения.

2. Задание 2: Создание классов исключений

Созданы два пользовательских класса исключений:

FunctionPointIndexOutOfBoundsException- наследуется от **IndexOutOfBoundsException**, выбрасывается при выходе за границы набора точек.

InappropriateFunctionPointException- наследуется от **Exception**, выбрасывается при некорректном добавлении или изменении точки.

3. Задание 3: Модификация **TabulatedFunction**

Класс **TabulatedFunction** переименован в **ArrayTabulatedFunction** и модифицирован:

- Конструкторы выбрасывают **IllegalArgumentException** при неверных параметрах
- Методы работы с точками выбрасывают **FunctionPointIndexOutOfBoundsException** при неверных индексах
- Методы **setPoint()**, **setPointX()** и **addPoint()** выбрасывают **InappropriateFunctionPointException** при нарушении порядка точек
- Метод **deletePoint()** выбрасывает **IllegalStateException** при попытке удалить точку, если точек меньше трех

4. Задание 4-5: **LinkedListTabulatedFunction**

Реализован класс **LinkedListTabulatedFunction** с использованием двусвязного циклического списка:

- Внутренний класс **FunctionNode** для элементов списка
- Методы **getNodeByIndex()**, **addNodeToTail()**, **addNodeByIndex()**, **deleteNodeByIndex()** для работы со списком

- Оптимизированный доступ к элементам через кэширование последнего **accessed** узла
- Реализованы все методы интерфейса **TabulatedFunction**

5. Задание 6: Создание интерфейса

Создан интерфейс **TabulatedFunction**, содержащий объявления общих методов:

- Методы работы с точками (**getPoint**, **setPoint**, **addPoint**, **deletePoint**)
- Методы получения координат (**getPointX**, **getPointY**, **setPointX**, **setPointY**)
- Методы получения информации о функции (**getPointsCount**, **getLeftDomainBorder**, **getRightDomainBorder**, **getFunctionValue**)

Оба класса (**ArrayTabulatedFunction** и **LinkedListTabulatedFunction**) реализуют этот интерфейс.

6. Задание 7: Тестирование

Разработан класс **Main** для тестирования всех функций и обработки исключений. Тестирование включает:

- Создание функций через массивы и связные списки
- Проверку вычисления значений функции
- Тестирование добавления, изменения и удаления точек
- Проверку обработки всех типов исключений

7. Результаты тестирования

=== ТЕСТИРОВАНИЕ КЛАССОВ ТАБУЛИРОВАННОЙ ФУНКЦИИ ===

ТЕСТИРОВАНИЕ **ArrayTabulatedFunction**

1. СОЗДАНИЕ ФУНКЦИИ $f(x) = 10x + 10$ НА ИНТЕРВАЛЕ $[0, 5]$:

Область определения: $[0,0, 5,0]$

Количество точек: 6

2. ТОЧКИ ТАБУЛИРОВАННОЙ ФУНКЦИИ:

Точка 0: (0,0; 10,0)

Точка 1: (1,0; 20,0)

Точка 2: (2,0; 30,0)

Точка 3: (3,0; 40,0)

Точка 4: (4,0; 50,0)

Точка 5: (5,0; 60,0)

3. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИИ:

$f(0,0) = 10,0$ (ожидалось: 10,0)

$f(0,5) = 15,0$ (ожидалось: 15,0)

$f(1,0) = 20,0$ (ожидалось: 20,0)

$f(1,5) = 25,0$ (ожидалось: 25,0)

$f(2,0) = 30,0$ (ожидалось: 30,0)

$f(2,5) = 35,0$ (ожидалось: 35,0)

$f(3,0) = 40,0$ (ожидалось: 40,0)

$f(3,5) = 45,0$ (ожидалось: 45,0)

$f(4,0) = 50,0$ (ожидалось: 50,0)

$f(4,5) = 55,0$ (ожидалось: 55,0)

$f(5,0) = 60,0$ (ожидалось: 60,0)

4. ТЕСТИРОВАНИЕ МОДИФИКАЦИИ ТОЧЕК:

До изменения: (2,0; 30,0) -> После: (2,0; 35,0)

5. ТЕСТИРОВАНИЕ ДОБАВЛЕНИЯ ТОЧКИ:

Количество точек до добавления: 6

Количество точек после добавления: 7

6. ТЕСТИРОВАНИЕ УДАЛЕНИЯ ТОЧКИ:

Количество точек до удаления: 7

Количество точек после удаления: 6

=====

ТЕСТИРОВАНИЕ LinkedListTabulatedFunction

1. СОЗДАНИЕ ФУНКЦИИ $f(x) = x^2$ НА ИНТЕРВАЛЕ $[0, 5]$:

Область определения: $[0,0, 5,0]$

Количество точек: 6

2. ТОЧКИ ТАБУЛИРОВАННОЙ ФУНКЦИИ:

Точка 0: (0,0; 0,0)

Точка 1: (1,0; 1,0)

Точка 2: (2,0; 4,0)

Точка 3: (3,0; 9,0)

Точка 4: (4,0; 16,0)

Точка 5: (5,0; 25,0)

3. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИИ:

$f(0,0) = 0,0$ (ожидалось: 0,0)

$f(0,5) = 0,5$ (ожидалось: 0,3) ОШИБКА!

$f(1,0) = 1,0$ (ожидалось: 1,0)

$f(1,5) = 2,5$ (ожидалось: 2,3) ОШИБКА!

$f(2,0) = 4,0$ (ожидалось: 4,0)

$f(2,5) = 6,5$ (ожидалось: 6,3) ОШИБКА!

$f(3,0) = 9,0$ (ожидалось: 9,0)

$f(3,5) = 12,5$ (ожидалось: 12,3) ОШИБКА!

$f(4,0) = 16,0$ (ожидалось: 16,0)

$f(4,5) = 20,5$ (ожидалось: 20,3) ОШИБКА!

$f(5,0) = 25,0$ (ожидалось: 25,0)

4. ТЕСТИРОВАНИЕ МОДИФИКАЦИИ ТОЧЕК:

Изменение ординаты точки с индексом 2:

До: (2,0; 4,0) -> После: (2,0; 5,0)

Корректное изменение абсциссы с обновлением Y:

До: (1,0; 1,0) -> После: (1,2; 1,4)

5. ТЕСТИРОВАНИЕ ДОБАВЛЕНИЯ ТОЧКИ:

Количество точек до добавления: 6

Количество точек после добавления: 7

Значение в добавленной точке: $f(2,2) = 4,8$

Точки после добавления:

(0,0; 0,0) (1,2; 1,4) (2,0; 5,0) (2,2; 4,8) (3,0; 9,0) (4,0; 16,0) (5,0; 25,0)

6. ТЕСТИРОВАНИЕ УДАЛЕНИЯ ТОЧКИ:

Количество точек до удаления: 7

Количество точек после удаления: 6

7. ОБНОВЛЕНИЕ ВСЕХ ЗНАЧЕНИЙ Y В СООТВЕТСТВИИ С $f(x) = x^2$:

Точка 0: (0,0; 0,0)

Точка 1: (1,2; 1,4)

Точка 2: (2,2; 4,8)

Точка 3: (3,0; 9,0)

Точка 4: (4,0; 16,0)

Точка 5: (5,0; 25,0)

8. ПРОВЕРКА ИНТЕРПОЛЯЦИИ ПОСЛЕ ИСПРАВЛЕНИЙ:

$f(0,3) = 0,4$ (ожидалось: 0,1) ОШИБКА!

$f(1,1) = 1,3$ (ожидалось: 1,2) ОШИБКА!

$f(1,8) = 3,5$ (ожидалось: 3,2) ОШИБКА!

$f(2,7) = 7,4$ (ожидалось: 7,3) ОШИБКА!

$f(3,9) = 15,3$ (ожидалось: 15,2)

$f(4,6) = 21,4$ (ожидалось: 21,2) ОШИБКА!

9. ГРАНИЧНЫЕ ЗНАЧЕНИЯ:

Левая граница: $f(0,0) = 0,0$

Правая граница: $f(5,0) = 25,0$

ТЕСТИРОВАНИЕ LinkedListTabulatedFunction ЗАВЕРШЕНО УСПЕШНО

=====

ТЕСТИРОВАНИЕ ИСКЛЮЧЕНИЙ

1. Тестирование конструктора:

IllegalArgumentException: Левая граница области определения \geq правой

2. Тестирование выхода за границы индексов:

FunctionPointIndexOutOfBoundsException: Индекс выходит за границы: 11

3. Тестирование несоответствующих точек:

InappropriateFunctionPointException: Новый x должен быть между соседними точками

ОШИБКА: Исключение не было выброшено!

4. Тестирование удаления точек:

IllegalStateException: Невозможно удалить точку, так как кол-во точек меньше 3

ВСЕ ИСКЛЮЧЕНИЯ ПРОТЕСТИРОВАНЫ УСПЕШНО

=== ТЕСТИРОВАНИЕ ЗАВЕРШЕНО ===