

Отчет по лабораторной работе №3

Студент: Стоколяс Юрий Юрьевич

Группа: 6201-120303D

Выполненные задания

Задание 1

Изучил классы исключений Java:

- `java.lang.Exception`
- `java.lang.IndexOutOfBoundsException`
- `java.lang.ArrayIndexOutOfBoundsException`
- `java.lang.IllegalArgumentException`
- `java.lang.IllegalStateException`

Задание 2

Создал два класса исключений в пакете `functions`:

1. **`FunctionPointIndexOutOfBoundsException`** - наследует от `IndexOutOfBoundsException`

- Выбрасывается при обращении к точкам функции по некорректному индексу

2. **`InappropriateFunctionPointException`** - наследует от `Exception`

- Выбрасывается при попытке добавления или изменения точки функции несоответствующим образом

Задание 3

Модифицировал класс `TabulatedFunction` для выбрасывания исключений:

- **Конструкторы** выбрасывают `IllegalArgumentException` при:
 - Левая граница \geq правой границы
 - Количество точек < 2

- **Методы работы с точками** выбрасывают `FunctionPointIndexOutOfBoundsException` при:
 - Индекс < 0 или \geq количества точек
- **Методы `setPoint()` и `setPointX()`** выбрасывают `InappropriateFunctionPointException` при:
 - Координата X не находится в допустимом диапазоне между соседними точками
- **Метод `addPoint()`** выбрасывает `InappropriateFunctionPointException` при:
 - Попытка добавить точку с уже существующей абсциссой
- **Метод `deletePoint()`** выбрасывает `IllegalStateException` при:
 - Попытка удалить точку когда их остается менее 3

Задание 4

Создал класс `LinkedListTabulatedFunction` с двусвязным циклическим списком:

1. Внутренний класс `FunctionNode`:

- Содержит филд `FunctionPoint point` для данных
- Содержит ссылки `prev` и `next` на соседние элементы
- Инкапсулирован как private static класс

2. Основной класс `LinkedListTabulatedFunction`:

- Содержит филд `head` - выделенная голова списка
- Содержит поля для оптимизации доступа: `lastAccessedNode`, `lastAccessedIndex`
- Реализует методы работы со списком:
 - `getNodeByIndex(int index)` - оптимизированный доступ к элементам
 - `addNodeToTail()` - добавление в конец списка
 - `addNodeByIndex(int index)` - добавление в указанную позицию
 - `deleteNodeByIndex(int index)` - удаление элемента

Задание 5

Реализовал конструкторы и методы класса `LinkedListTabulatedFunction`:

- Конструкторы аналогичны `TabulatedFunction`
- Все методы имеют те же сигнатуры и выбрасывают те же исключения
- Использовал методы работы со связным списком
- Оптимизировал методы за счет прямого доступа к элементам списка

Задание 6

1. **Создание интерфейса** `TabulatedFunction` с объявлениями всех общих методов
2. **Реализация интерфейса** в классах `ArrayTabulatedFunction` и `LinkedListTabulatedFunction`

Задание 7

- Проверка создания объектов обоих типов функций
- Тестирование всех видов исключений
- Демонстрация работы с функциями через интерфейс `TabulatedFunction`

Результаты тестирования

Тестирование `ArrayTabulatedFunction`:

`ArrayTabulatedFunction` создан успешно

Количество точек: 5

Левая граница: 0.0

Правая граница: 10.0

Значение функции при $x=2.5$: 2.0

Тестирование исключений:

Поймано `IllegalArgumentException` для некорректных границ: Левая граница

Поймано `IllegalArgumentException` для недостаточного количества точек

Поймано `FunctionPointIndexOutOfBoundsException` для отрицательного индекса

Поймано `FunctionPointIndexOutOfBoundsException` для индекса вне границ

Поймано `InappropriateFunctionPointException` для некорректного X: Координаты

Поймано `InappropriateFunctionPointException` для дублирующегося X: Точка

Тестирование LinkedListTabulatedFunction

:

LinkedListTabulatedFunction создан успешно

Значение функции при $x=2.5$: 2.0

Тестирование исключений:

Поймано IllegalArgumentException для некорректных границ: Левая граница

Поймано IllegalArgumentException для недостаточного количества точек

Поймано FunctionPointIndexOutOfBoundsException для отрицательного индекса

Поймано FunctionPointIndexOutOfBoundsException для индекса вне границ

Поймано InappropriateFunctionPointException для некорректного X: Координаты

Поймано InappropriateFunctionPointException для дублирующегося X: Точка