

Лабораторная работа №3

Лебедев Кирилл Дмитриевич 6204-010302D

Задание 1

Были изучены следующие классы, входящие в стандартную библиотеку Java:

`java.lang.Exception` — базовый класс всех исключений;

`java.lang.IndexOutOfBoundsException` — исключение выхода за границы диапазона индексов;

`java.lang.ArrayIndexOutOfBoundsException` — частный случай предыдущего, используется при выходе за границы массива;

`java.lang.IllegalArgumentException` — возникает при передаче недопустимого аргумента в метод;

`java.lang.IllegalStateException` — возникает, когда объект находится в некорректном состоянии для вызова данного метода.

Результат: Ознакомился.

Задание 2

В пакете `functions` были созданы два класса исключений:

```
package functions;
```

```
public class FunctionPointIndexOutOfBoundsException extends  
IndexOutOfBoundsException {  
    public FunctionPointIndexOutOfBoundsException(String message) {  
        super(message);  
    }  
}
```

```
package functions;
```

```
public class InappropriateFunctionPointException extends Exception {  
    public InappropriateFunctionPointException(String message) {  
        super(message);  
    }  
}
```

Результат: Классы созданы.

Задание 3

В ранее разработанный класс `TabulatedFunction` были внесены изменения:

- Оба конструктора класса выбрасывают исключение `IllegalArgumentException`, если левая граница области определения больше или равна правой, а также если предлагаемое количество точек меньше двух.
- 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.

Результат: Класс модифицирован.

Задание 4

Создан новый класс `LinkedListTabulatedFunction`, реализующий хранение точек в двусвязном циклическом списке с выделенной головой и класс `FunctionNode`, содержащий информационное поле для хранения данных типа `FunctionPoint`, а также поля для хранения ссылок на предыдущий и следующий элемент.

```
private static class FunctionNode {  
    FunctionPoint point; // точка  
    FunctionNode next;   // след точка  
    FunctionNode prev;   // пред точка  
  
    FunctionNode(FunctionPoint point) { // конструктор  
        this.point = point;  
    }  
}
```

Результат: список работает, методы для работы с табулируемой функцией и её точками тоже.

Задание 5

Были реализованы в классе `LinkedListTabulatedFunction` конструкторы и методы, аналогичные конструкторам и методам класса `TabulatedFunction`.

Результат: Все методы работают.

Задание 6

Создан интерфейс `TabulatedFunction`, содержащий объявления общих методов для всех реализаций

```
package functions;
```

```
public interface TabulatedFunction {  
    double getLeftDomainBorder();  
    double getRightDomainBorder();  
    double getFunctionValue(double x);  
    int getPointsCount();
```

```
    FunctionPoint getPoint(int index) throws  
    FunctionPointIndexOutOfBoundsException;
```

```
    void setPoint(int index, FunctionPoint point) throws  
    FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException;
```

```
    double getPointX(int index) throws FunctionPointIndexOutOfBoundsException;
```

```
    void setPointX(int index, double x) throws  
    FunctionPointIndexOutOfBoundsException, InappropriateFunctionPointException;
```

```
    double getPointY(int index) throws FunctionPointIndexOutOfBoundsException;
```

```
    void setPointY(int index, double y) throws  
    FunctionPointIndexOutOfBoundsException;
```

```
    void deletePoint(int index) throws FunctionPointIndexOutOfBoundsException,  
    IllegalStateException;
```

```
    void addPoint(FunctionPoint point) throws InappropriateFunctionPointException;
```

```
    void printTabulatedFunction();  
}
```

Теперь суть работы с табулированными функциями заключена в типе интерфейса, а в классах заключена только реализация этой работы.

Результат: интерфейс создан и работает корректно.

Задание 7

Тестирование классов

Ход выполнения:

- Переписан класс Main
- Протестированы все методы на примере функции $y = 2x+1$
- Проверены исключения

Результат работы Main:

Тестирование: ArrayTabulatedFunction

Исходные точки:

1 --- (0.0, 1.0)

2 --- (1.0, 3.0)

3 --- (2.0, 5.0)

4 --- (3.0, 7.0)

5 --- (4.0, 9.0)

6 --- (5.0, 11.0)

Тест интерполяции:

$f(-2,0) = \text{NaN}$

$f(0,0) = 1,0$

$f(0,5) = 2,0$

$f(1,0) = 3,0$

$f(2,5) = 6,0$

$f(3,0) = 7,0$

$f(4,0) = 9,0$

$f(5,0) = 11,0$

$f(6,0) = \text{NaN}$

После замены точки:

1 --- (0.0, 1.0)

2 --- (1.5, 6.0)

3 --- (2.0, 5.0)

4 --- (3.0, 7.0)

5 --- (4.0, 9.0)

6 --- (5.0, 11.0)

После добавления точки:

1 --- (0.0, 1.0)

2 --- (1.5, 6.0)

3 --- (1.7, 4.0)

4 --- (2.0, 5.0)

5 --- (3.0, 7.0)

6 --- (4.0, 9.0)

7 --- (5.0, 11.0)

После удаления точки:

1 --- (0.0, 1.0)

2 --- (1.5, 6.0)

3 --- (1.7, 4.0)

4 --- (3.0, 7.0)

5 --- (4.0, 9.0)

6 --- (5.0, 11.0)

--

FunctionPointIndexOutOfBoundsException: Индекс выходит за границы: 11

InappropriateFunctionPointException: Новый 'x' должен быть между соседними точками

InappropriateFunctionPointException: Точка с таким 'x' уже существует

IllegalStateException: Невозможно удалить точку, так как 'количество точек меньше 3'

IllegalArgumentException: Левая граница области определения больше или равна правой

У LinkedListTabulatedFunction всё тоже самое.