Лабораторная №3

Студент: Гонеев Роман Олегович
Группа: 6201-120303D

Задания:

1. -
2. В пакете functions создал два класса исключений:

   cd ~/lab3_rep/functions

- FunctionPointIndexOutOfBoundsException:

  nano FunctionPointIndexOutOfBoundsException.java

  Наследует от класса IndexOutOfBoundsException:

  ```
  class FunctionPointIndexOutOfBoundsException extends IndexOutOfBoundsException
  ```

- InappropriateFunctionPointException:

  nano InappropriateFunctionPointException.java

  Наследует от класса Exception:

  ```
  public class InappropriateFunctionPointException extends RuntimeException
  ```

3. В классе TabulatedFunction добавил исключения методами класса.
   - IllegalArgumentException для проверки границ
   - FunctionPointIndexOutOfBoundsException индекс вне границ набора точек
   - InappropriateFunctionPointreportxception Абсцисса нарушает упорядочность или абсцисса добавляемой совпадает с существующей
   - IllegalStateException Количество точек меньше трех
4. В пакете functions создал класс LinkedListTabulatedFunction, использующий двувсязный список с выделнной головой для хранения точек.
   Для реализации списка определил два класса:
   - FunctionNode — описывает элемент списка. Для обеспечения инкапсуляции класс описан как вложеный приватный.
     В классе LinkedListTabulatedFunction реализовал вспомогательные методы для работы со структурой списка:
     1. FunctionNode getNodeByIndex(int index) — возвращает ссылку ну зел по его номеру
     2. addPoint(FunctionPoint point) — вставляет точку, соблюдая упорядоченность.
   - LinkedListTabulatedFunction — внешний класс. Описывает список целиком, храня ссылку на голову и размер.
     Конструкторы и методы класса LinkedListTabulatedFunction реализованы аналогично ArrayTabulatedFunction, с теми же видами исключений. Логика методов getPoint, setPointX, deletePoint использует реализованные методы работы со связным списком.
5. Реализация функционала TabulatedFunction в LinkedListTabulatedFunction
   В классе реализовал два конструктора, аналогичные конструкторам ArrayTabulatedFunction, и обеспечивающие те же проверки на корректность входных данных:

   - LinkedListTabulatedFunction(double leftX, double rightX, int pointsCount)

- LinkedListTabulatedFunction(double leftX, double rightX, double[] values)

  оба выбрасывают исключения IllegalArgumentException в случае некорректных параметров. Инициализация через addNodeTail.

  Реализация методов и оптимизация:

  - getPointX(int index) и getPointY(int index) — использует getNodeByIndex(index) для доступа к узлу, потом напрямую возвращает x или y из поля data
  - setPointX(int index, double x) — получает узел через getNodeByIndex(index); проверяет новое значение x, обращаясь напрямую к полям соседних узлов (prev.data.getX() и next.data.getX()) - это является оптимизацией для связного списка.
  - deletePoint(int index) - использует deleteNodeByIndex(index) для удаления узла путем пересвязывания ссылок prev и next у соседей.
  - addPoint(FunctionPoint point) - осуществляет проход по списку для проверки на дубликат x и нахождения места для вставки. Вставка точки производится путем переназначения ссылок у соседних узлов.

6. Провел реструктуризацию классов для внедрения концепции полиморфизма.
   Класс TabulatedFunction был переименован в ArrayTabulatedFunction. Был создан интерфейс TabulatedFunction, который определяет шаблон(«контракт») для всех реализаций (массив и список):
   public interface TabulatedFunction {}
   Оба класса, ArrayTabulatedFunction и LinkedListTabulatedFunction, были модифицированы для реализации этого интерфейса с помощью ключевого слова
   - public class ArrayTabulatedFunction implements TabulatedFunction {}
   - public class LinkedListTabulatedFunction implements TabulatedFunction {}
7. Проверка работы написанных классов
   В созданном классе Main.java выполнил проверку работы классов, реализующих интерфейс TabulatedFunction.
   - Проверка полиморфизма и корректности:
     Создал две функции с одинаковыми параметрами и изменена одна точка (setPointY). Сравнение результатов интерполяции (getFunctionValue x=2.5) показало совпадение значений, подтверждая корректность логики и реализацию единого интерфейса:
   - 1. Проверка Array и List: Интерполяция (x=2.5): Array=21.25, List=21.25 Сравнение результатов: Ок Проверка значения: Ок

   - Проверка обработки исключений:
     Провел проверка для всех классов исключений, которые были реализованы в рамках Заданий 2 и 3, включая проверку для реализации на массиве (ArrayTabulatedFunction) и на связном списке:
     2. Проверка исключений: Array Ок: Индекс вне границ пойман. Array Ок: Пойман IllegalStateException. Конструктор Ок: Пойман IllegalArgumentException. List Ок: Пойман дубликат X. List Ок: Пойман InappropriateFunctionPointException.

Исключения:

FunctionPointIndexOutOfBoundsException - Вызов getPoint(100) на ArrayTabulatedFunction.
IllegalStateException - Попытка удалить точку, когда их остается менее 3.
IllegalArgumentException - Вызов конструктора с leftX≥rightX.
InappropriateFunctionPointException (Дубликат x) - Вызов addPoint() с уже существующей абсциссой.
InappropriateFunctionPointException (Порядок) - Вызов setPointX() с нарушением упорядоченности абсцисс.