Отчёт
Лабораторная работа №7

Выполнил: Салихов Камран Рустам угли
Группа: 6204-010302D

В ходе выполнения задания 1 в **TabulatedFunction** было добавлено наследование от **Iterable<FunctionPoint>.**

В **ArrayTabulatedFunction** были добавлены:

метод **iterator()**, возвращающий анонимный класс **Iterator<FunctionPoint>**;

метод **next()**, который выбрасывает исключение **NoSuchElementException** если следующего элемента нет;

метод **remove()**, который выбрасывает **UnsupportedOperationException**.
Для **LinkedListTabulatedFunction** реализован аналогичный итератор, но работающий с узлами связного списка.
Итератор начинает с **head.next** и проходит до возвращения к **head**.
В оба класса добавлен **java.util.Iterator**.

Проверка в **Main** при **leftX** = -2, **rightX** = 3, **PointsCount** = 5

```
задание 1:
ArrayTabulatedFunction:
(-2.0; 0.0)
(-0.75; 0.0)
(0.5; 0.0)
(1.75; 0.0)
(3.0; 0.0)

LinkedListTabulatedFunction:
(-2.0; 0.0)
(-0.75; 0.0)
(0.5; 0.0)
(1.75; 0.0)
(3.0; 0.0)
```

В ходе выполнения задания 2 был использован паттерн «Фабричный метод». В пакете **functions** был описан базовый интерфейс фабрик табулированных функций **TabulatedFunctionFactory**. Интерфейс объявляет три перегруженных метода **TabulatedFunction createTabulatedFunction()**, параметры которых соответствуют параметрам конструкторов классов табулированных функций.

В классах **ArrayTabulatedFunction** и **LinkedListTabulatedFunction** были описаны классы фабрик **ArrayTabulatedFunctionFactory** и **LinkedListTabulatedFunctionFactory**, реализующие интерфейс фабрики и порождающие объекты соответствующих классов табулированных функций.

В классе **TabulatedFunctions** было объявлено приватное статическое поле типа **TabulatedFunctionFactory**. Был объявлен метод **setTabulatedFunctionFactory**(), позволяющий заменить объект фабрики.

В классе **TabulatedFunctions** были описаны три перегруженных метода **TabulatedFunction createTabulatedFunction()**, возвращающие объекты табулированных функций, созданные с помощью текущей фабрики. Параметры методов соответствуют параметрам методов фабрики.

В остальных методах класса, где требуется создание объектов табулированных функций, было заменено явное создание объектов с помощью конструкторов на вызов соответствующего метода **createTabulatedFunction()**.

```
задание 2
class functions.ArrayTabulatedFunction
class functions.LinkedListTabulatedFunction
class functions.ArrayTabulatedFunction
```

В ходе выполнения задания 3 в классе **TabulatedFunctions** были добавлены ещё три перегруженных версии метода **createTabulatedFunction()**. Их параметры повторяют параметры трёх аналогичных методов, основанных на использовании фабрики. Эти методы также получают ссылку типа **Class** на описание класса, объект которого требуется создать. В эти методы можно передать только ссылки на классы, реализующие интерфейс **TabulatedFunction** благодаря **Class<? extends TabulatedFunction>.**

Текст работы с консолью:

задание 3

class functions.ArrayTabulatedFunction

{(0.0; 0.0), (5.0; 0.0), (10.0; 0.0)}

class functions.ArrayTabulatedFunction

{(0.0; 0.0), (10.0; 10.0)}

class functions.LinkedListTabulatedFunction

{(0.0; 0.0), (10.0; 10.0)}

class functions.LinkedListTabulatedFunction

{(0.0; 0.0), (0.3141592653589793; 0.3090169943749474), (0.6283185307179586; 0.5877852522924731), (0.9424777960769379; 0.8090169943749475), (1.2566370614359172; 0.9510565162951535), (1.5707963267948966; 1.0), (1.8849555921538759; 0.9510565162951536), (2.199114857512855; 0.8090169943749475), (2.5132741228718345; 0.5877852522924732),

(2.827433388230814; 0.3090169943749475), (3.141592653589793; 1.2246467991473532E-16)}

Process finished with exit code 0