

Лабораторная работа №2

Иванова Виктория Павловна 6203-010302

Задание 1

Создала пакет `functions`, в котором далее создавала классы программы.

Задание 2

В пакете `functions` создала класс `FunctionPoint`, объект которого описывает точку табулированной функции и в классе описала следующие конструкторы:

`FunctionPoint(double x, double y)` – создаёт объект точки с заданными координатами;
`FunctionPoint(FunctionPoint point)` – создаёт объект точки с теми же координатами, что у указанной точки;
`FunctionPoint()` – создаёт точку с координатами (0; 0).

Задание 3

В пакете `functions` создала класс `TabulatedFunction`, объект которого описывает табулированную функцию, для хранения данных о точках использовала массив типа `FunctionPoint`. В классе описала следующие конструкторы:

`TabulatedFunction(double leftX, double rightX, int pointsCount)` – создаёт объект табулированной функции по заданным левой и правой границе области определения и количеству точек для табулирования
`TabulatedFunction(double leftX, double rightX, double[] values)` – аналогичен предыдущему конструктору, но вместо количества точек получает значения функции в виде массива.

Задание 4

В классе `TabulatedFunction` описала методы, необходимые для работы с функцией.

Метод `double getLeftDomainBorder()`, который возвращает значение левой границы области определения функции.

Метод `double getRightDomainBorder()` возвращает значение правой границы области определения функции.

Метод `double getFunctionValue(double x)` возвращает значение функции в точке `x`, если эта точка лежит в области определения функции, иначе метод возвращает значение неопределённости. При расчёте значения функции использовала линейную интерполяцию.

Задание 5

В классе `TabulatedFunction` описала методы, необходимые для работы с точками табулированной функции:

Метод `int getPointsCount()` возвращает количество точек.

Метод `FunctionPoint getPoint(int index)` возвращает копию точки, соответствующей переданному индексу.

Метод `void setPoint(int index, FunctionPoint point)` заменять указанную точку табулированной функции на переданную.

Метод `double getPointX(int index)` возвращает значение абсциссы точки с указанным номером.

Метод `void setPointX(int index, double x)` изменяет значение абсциссы точки с указанным номером. Методы `double getPointY(int index)` и `void setPointY(int index, double y)` аналогичны, только возвращают/изменяют ординату точек.

Задание 6

В классе `TabulatedFunction` описала методы, изменяющие количество точек табулированной функции.

Метод `void deletePoint(int index)` удаляет заданную точку табулированной функции.

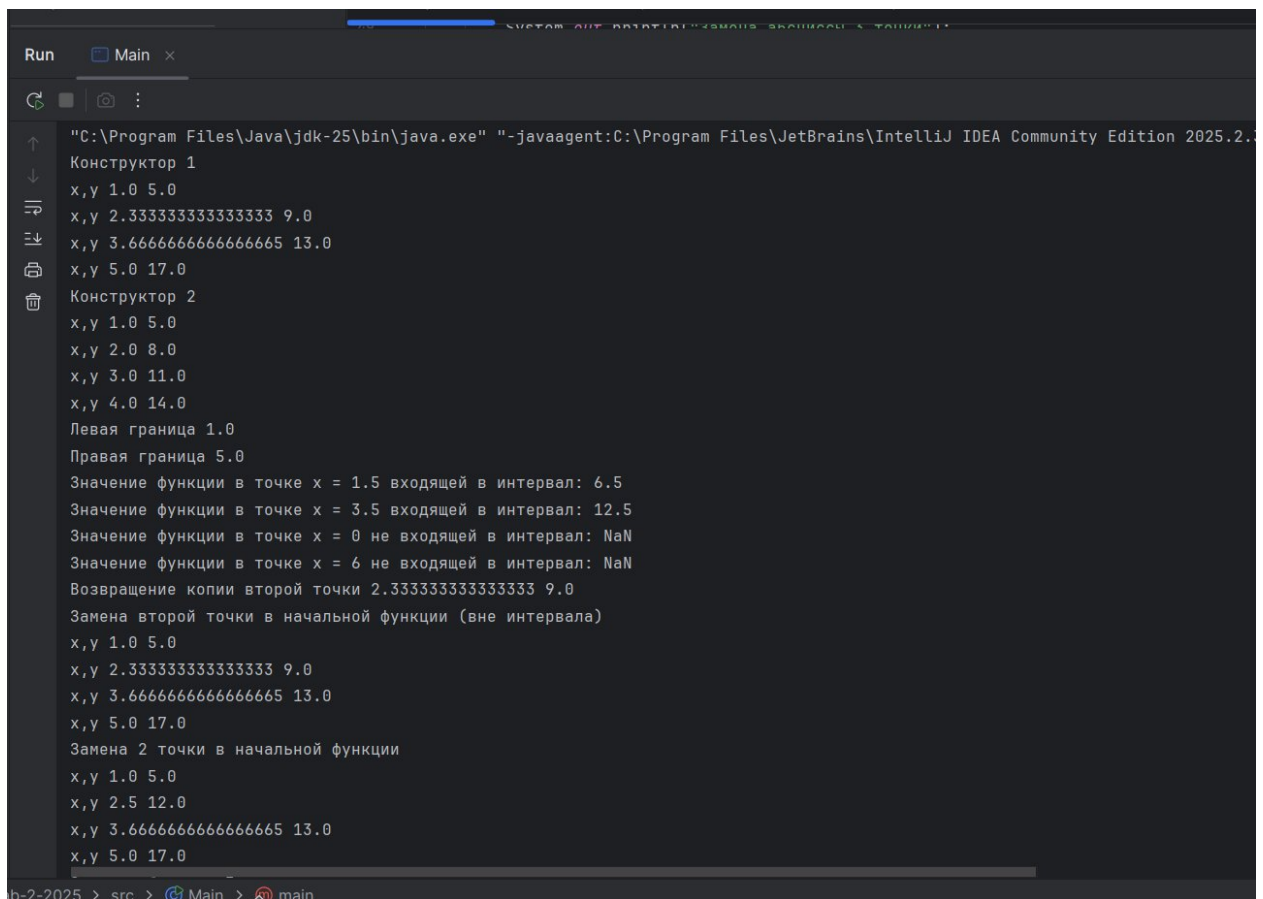
Метод `void addPoint(FunctionPoint point)` добавляет новую точку табулированной функции.

Для копирования участков массивов воспользовалась методом `arraycopy()` класса `System`.

Задание 7

Проверила работу написанных классов: в пакете по умолчанию создала класс `Main`, содержащий точку входа программы, в нем создала экземпляр класса `TabulatedFunction` и задала для него табулированные значения линейной функции $y=3x+2$

Вывела в консоль значения функции на ряде точек, проверила, как изменяется результат работы программы после изменения точек, добавления и удаления точек.



```
Run Main x
C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.
Конструктор 1
x,y 1.0 5.0
x,y 2.333333333333333 9.0
x,y 3.6666666666666665 13.0
x,y 5.0 17.0
Конструктор 2
x,y 1.0 5.0
x,y 2.0 8.0
x,y 3.0 11.0
x,y 4.0 14.0
Левая граница 1.0
Правая граница 5.0
Значение функции в точке x = 1.5 входящей в интервал: 6.5
Значение функции в точке x = 3.5 входящей в интервал: 12.5
Значение функции в точке x = 0 не входящей в интервал: NaN
Значение функции в точке x = 6 не входящей в интервал: NaN
Возвращение копии второй точки 2.333333333333333 9.0
Замена второй точки в начальной функции (вне интервала)
x,y 1.0 5.0
x,y 2.333333333333333 9.0
x,y 3.6666666666666665 13.0
x,y 5.0 17.0
Замена 2 точки в начальной функции
x,y 1.0 5.0
x,y 2.5 12.0
x,y 3.6666666666666665 13.0
x,y 5.0 17.0
b-2-2025 > src > Main > main
```

```
Run Main x
x,y 5.0 17.0
Замена 2 точки в начальной функции
x,y 1.0 5.0
x,y 2.5 12.0
x,y 3.6666666666666665 13.0
x,y 5.0 17.0
Замена абсциссы 3 точки
x,y 1.0 5.0
x,y 2.5 12.0
x,y 3.8 13.0
x,y 5.0 17.0
Замена ординаты 2 точки
x,y 1.0 5.0
x,y 2.5 15.0
x,y 3.8 13.0
x,y 5.0 17.0
Удаление 3 точки
x,y 1.0 5.0
x,y 2.5 15.0
x,y 5.0 17.0
Добавление 2 новых точек
x,y 1.0 5.0
x,y 2.2 9.0
x,y 2.5 15.0
x,y 3.5 13.0
x,y 5.0 17.0

Process finished with exit code 0
```