Лабораторная работа №2

Болясников Вадим Андреевич 6204-010302D

Задание на лабораторную работу:  
Разработать набор классов для работы с функциями одной переменной, заданными в табличной форме.

Задание 1:  
В этом задание я создал пакет functions, в котором будут храниться наши классы.

Задание 2:  
Я создал класс FunctionPoint внутри пакета functions, данный класс описывает одну точку табулированной функции.

В нем я реализовал следующие контрукции:  
• 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).

Также чтобы учесть особенности инкапсуляции, я сделал геттеры и сеттеры.

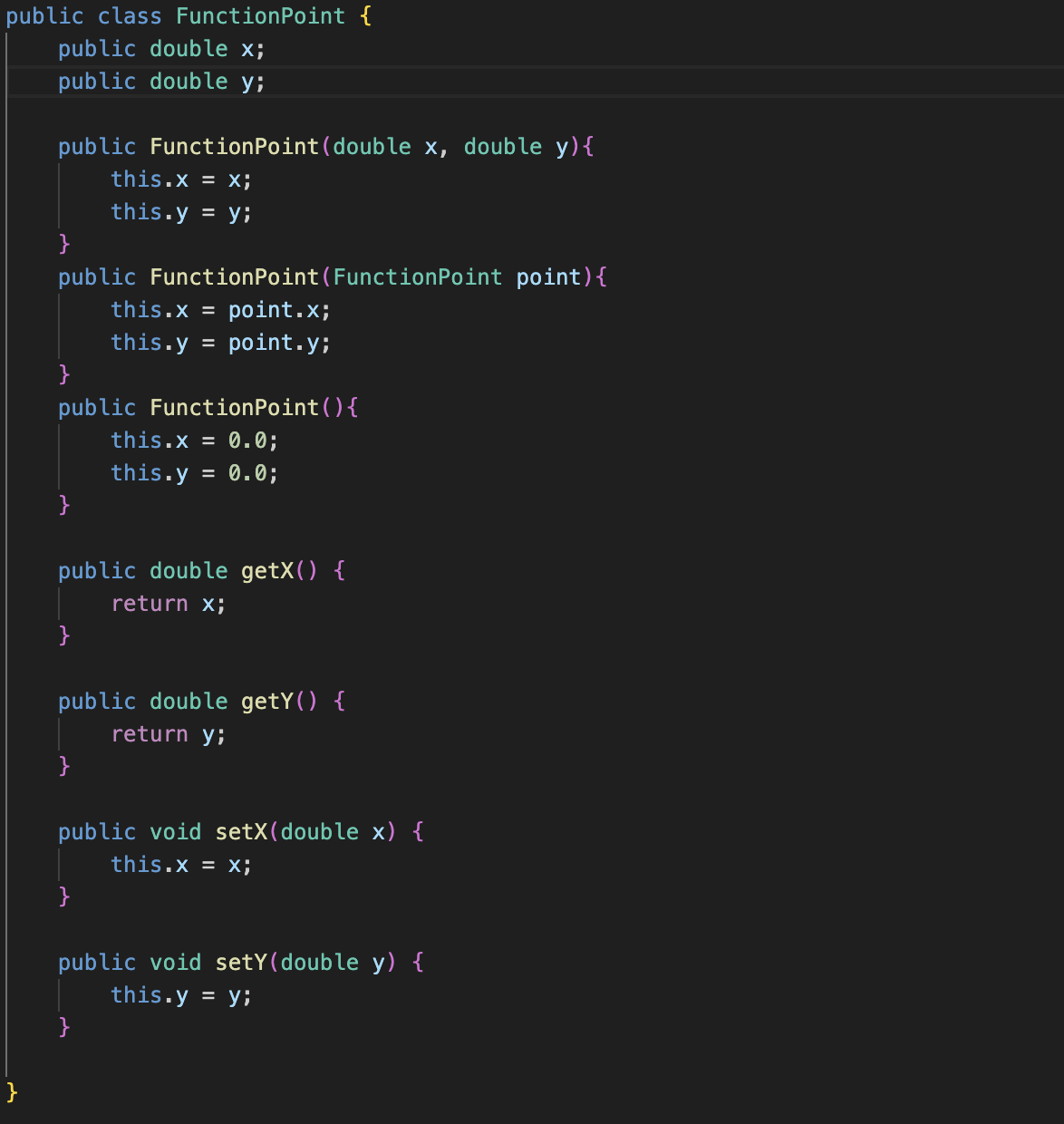


Рисунок 1 – Класс FunctionPoint

Задание 3:  
В этом же пакете function я реализовал класс TabulatedFunction, который описывал табулированную функцию, для хранения данных о точке использовался массив типа FunctionPoint. Также было сделано упорядочение по значение x.  
Также были реализованы конструкторы:  
• TabulatedFunction(double leftX, double rightX, int pointsCount) – создаёт объект табулированной функции по заданным левой и правой границе области определения, а также количеству точек для табулирования (значения функции в точках при этом следует считать равными 0);

• TabulatedFunction(double leftX, double rightX, double[] values) – аналогичен предыдущему конструктору, но вместо количества точек получает значения функции в виде массива.

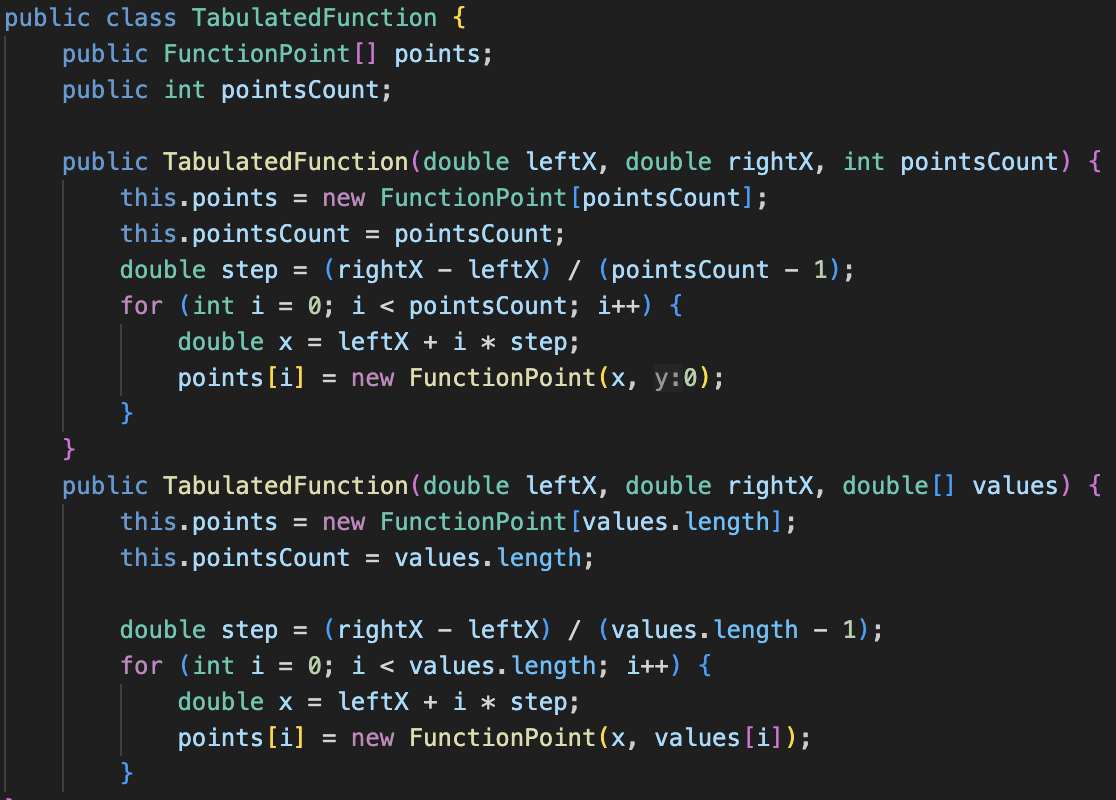


Рисунок 2 – Реализация класса TabulatedFunction

Задание 4:

В этом задание я дополнил класс TabulatedFunction, дописал следующие функции:  
• Метод double getLeftDomainBorder() возвращает значение левой границы области определения табулированной функции.

• Аналогично, метод double getRightDomainBorder() возвращает значение правой границы области определения табулированной функции.

• Метод double getFunctionValue(double x) возвращает значение функции в точке x, если эта точка лежит в области определения функции. В противном случае метод возвращает значение неопределённости. Для которой была реализована дополнительная функция linearInterpolation.

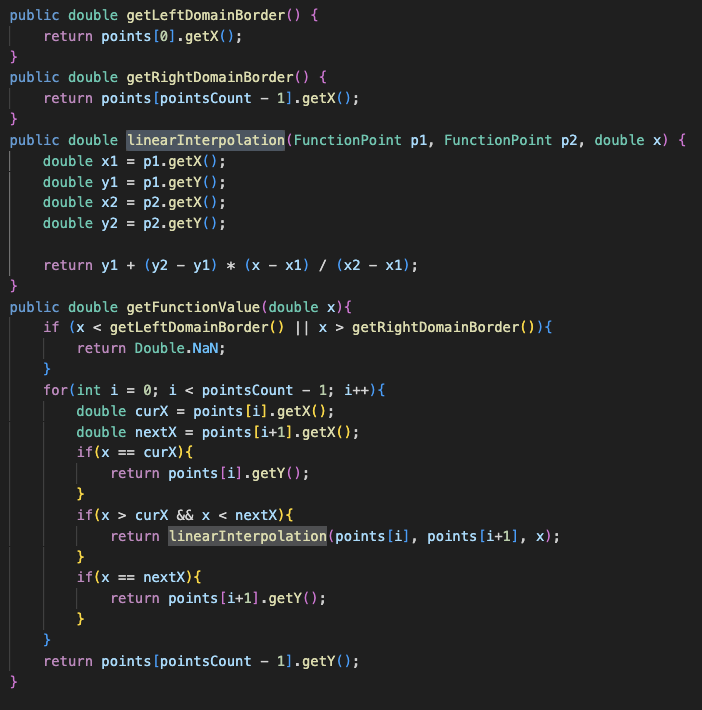


Рисунок 3 – Доп. методы в классе TabulatedFunction  
  
Задание 5-6.

Я ходе этих двух заданий я дописал еще следующие методы:  
• Метод int getPointsCount() который возвращает текущее количество точек в табулированной функции.

• Метод FunctionPoint getPoint(int index) который возвращает копию точки по указанному индексу. Создание копии обеспечивает инкапсуляцию, предотвращая модификацию исходной точки извне класса.

• Метод void setPoint(int index, FunctionPoint point) который заменяет точку по указанному индексу на переданную точку. Метод включает проверки для сохранения упорядоченности точек по координате X.

• Метод double getPointX(int index) который возвращает координату X точки с указанным индексом.

• Метод void setPointX(int index, double x) который изменяет координату X точки с указанным индексом. Как и в setPoint(), метод проверяет, что новое значение X не нарушает упорядоченность точек.

• Метод double getPointY(int index) который возвращает координату Y точки с указанным индексом.

• Метод void setPointY(int index, double y) который изменяет координату Y точки с указанным индексом.

• Метод void deletePoint(int index) удаляет точку с указанным индексом. Для эффективного удаления используется System.arraycopy() для сдвига элементов массива, после удаления последний элемент обнуляется для помощи сборщику мусора.

• Метод void addPoint(FunctionPoint point) Добавляет новую точку в табулированную функцию с сохранением упорядоченности по X

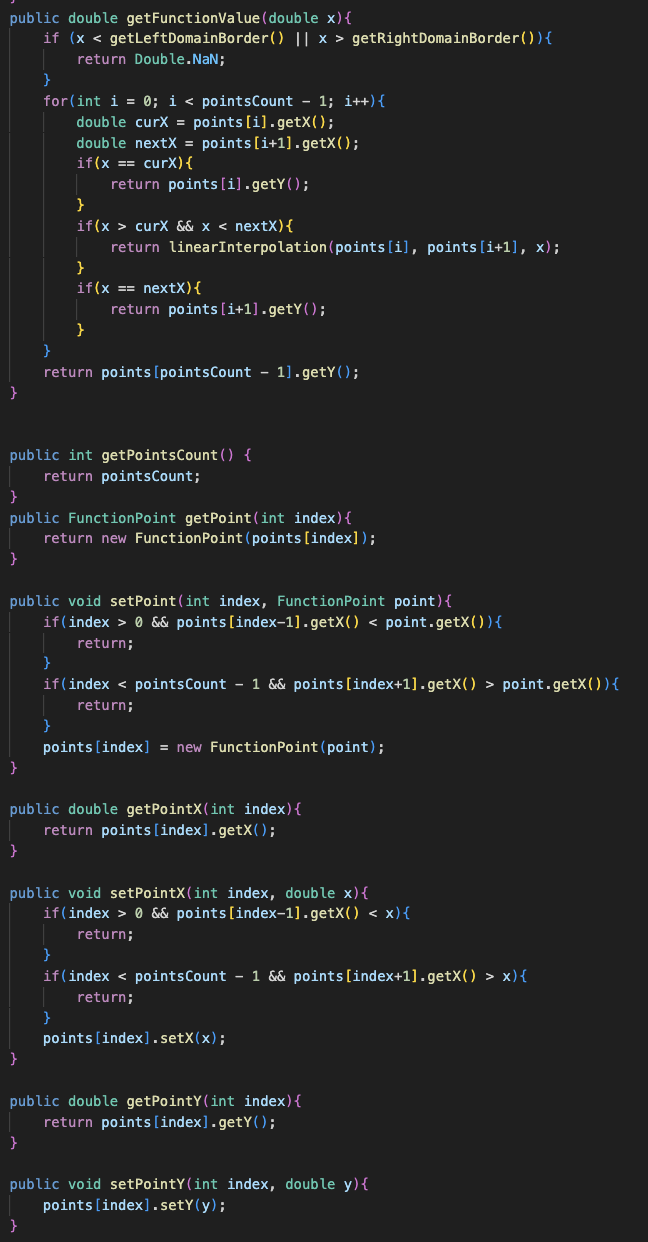




Рисунок 4 – Еще методы в классе TabulatedFunction.

Задание 7.

Был создан main вне пакета function. Там я написал экземпляр класса TabulatedFunction который задает табулированные значения функции f(x) = x^2. А также написал вывод в консоль значения функций на ряде точек. Вывел результат программы после: удаления, добавления и изменения точек.

Вот что вывелось у меня в консоль:

1. СОЗДАНИЕ ФУНКЦИИ f(x) = x^2 на [0, 4]

Функция f(x) = x^2

Количество точек: 5

Область определения: [0.0, 4.0]

2. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИИ

f(x) = не определено (вне области)

f(x) = 0.0

f(x) = 0.5

f(x) = 1.0

f(x) = 2.5

f(x) = 4.0

f(x) = 6.5

f(x) = 9.0

f(x) = 12.5

f(x) = 16.0

f(x) = не определено (вне области)

3. ИЗМЕНЕНИЕ СУЩЕСТВУЮЩИХ ТОЧЕК

После setPointY(2, 10.0): f(2.0) = 10.0

После setPoint(3, (3.5; 20.0)): f(3.5) = 12.5

4. ДОБАВЛЕНИЕ НОВЫХ ТОЧЕК

После addPoint((-1.0; 1.0)): точек = 6, f(-1.0) = 1.0

После addPoint((1.5; 2.25)): f(1.5) = 2.25

После addPoint((5.0; 25.0)): точек = 8, f(5.0) = 25.0

После addPoint((2.0; 100.0)) - замена: functions.TabulatedFunction@1722011b

f(2.0) = 100.0

5. УДАЛЕНИЕ ТОЧЕК

После deletePoint(0): точек = 7, левая граница = 0.0

После deletePoint(2): точек = 6

После deletePoint(5): точек = 5, правая граница = 4.0