

Лабораторная работа №2

Создание приложения для отображения графика интерполяционного полинома.

1. Добавить в проект лабораторной работы №1 отдельный модуль Converting. В модуле создать класс Converter для выполнения преобразования координат из декартовой системы в экранную и наоборот. Класс должен содержать следующие элементы.
 - а. Вещественнозначные приватные поля для хранения границ отображаемых отрезков по осям абсцисс и ординат.
 - б. Целочисленные приватные поля для хранения ширины и высоты изображения в пикселях, в рамках которого будут выполняться преобразования.
 - в. Конструкторы:
 - с 4 параметрами – границами отрезков в декартовой системе координат;
 - с 6 параметрами – границами отрезков в декартовой системе координат + ширина + высота изображения в пикселях.
 - г. Приватные вспомогательные методы для получения значения плотности экранных точек на 1 деление в декартовой системе координат.
 - д. Публичные методы преобразования координат:
 - xScr2Crt() – для преобразования абсциссы из экранной системы в декартовую;
 - xCrt2Scr() – для преобразования абсциссы из декартовой системы в экранную;
 - yScr2Crt() – для преобразования ординаты из экранной системы в декартовую;
 - yCrt2Scr() – для преобразования ординаты из декартовой системы в экранную.
2. Разработать тесты на JUnit5 для тестирования всех методов класса Converter.
3. Создать артефакт .jar из модуля Converter для получения возможности использования класса преобразования в других проектах, путём подключения к ним скомпилированного .jar-файла.
4. В основном модуле проекта разработать класс, наследник JFrame из библиотеки Swing для формирования основного окна приложения. Окно приложение должно быть примерно таким как указано на рис. 1 ниже.

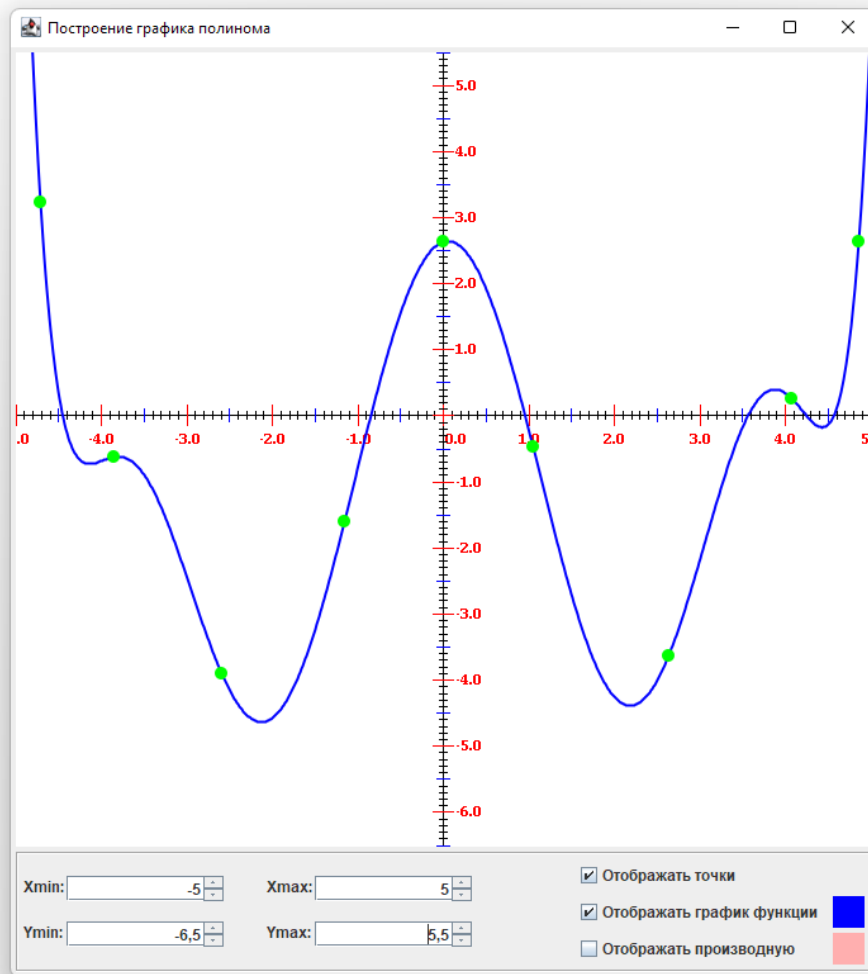


Рис. 1. Пример окна приложения для отображения графика интерполяционного полинома.

5. Разработать класс `CartesianPainter`, предназначенный для отображения декартовой системы координат. Класс должен реализовывать интерфейс `Painter`:

```
public interface Painter {  
    Dimension getSize();  
    void setSize(Dimension d);  
    void setSize(int width, int height);  
    void paint(Graphics g);  
}
```

6. Разработать класс `FunctionPainter`, также реализующий интерфейс `Painter` и предназначенный для вывода графика функции полинома.

Функции, которое должно поддерживать приложение.

1. Отображение декартовой системы координат с осями и делениями по целым числам (обязательно). Также опционально¹ приложение должно выводить в системе координат:
 - а. числовые метки рядом с целыми делениями;
 - б. более мелкие деления с шагом 0,5 ед. и еще более мелкие – с шагом 0,1 ед;
 - в. деления с переменным шагом, зависящим от ширины и высоты панели и длин отображаемых отрезков, таким образом, чтобы все деления были хорошо видны пользователю.
2. Добавлять точки, через которые должен проходить график интерполяционного полинома при помощи щелчков мышью по панели (JPanel), где отображается график. Также дополнительно приложение должно позволять:
 - а. выполнять проверку добавляемой точки и не допускать создания второй точки, относительно уже существующей в некоторой окрестности от нее по оси абсцисс;
 - б. удалять точки при щелчке по ним правой кнопкой мыши.
3. Отрисовывать график интерполяционного полинома, проходящего через указанные пользователем точки.
4. С помощью панели управления, расположенной в нижней части окна давать возможность пользователю задавать границы отображаемых отрезков по осям абсцисс и ординат. При этом программа должна контролировать минимального и максимального значений, не давать им возможности поменяться местами или стать равными.
5. Опционально. Добавить возможность выбора отображаемых элементов на графике: точек, которые указал пользователь, а также графика производной построенного полинома.
6. Опционально. Добавить возможность выбора пользователем цвета графика интерполяционного полинома, а также точек, через которые он проходит и графика производной (при их наличии).

NB: Ввиду особенностей построения интерполяционных полиномов и применяемых типов данных (*double*), при создании полиномов высоких степеней могут возникать погрешности, накапливающиеся с ростом степени. Они могут приводить к ошибкам отображения графика. Для того, чтобы приложение

¹ Опциональные и дополнительные функции можно не реализовывать, но за их выполнение могут начисляться дополнительные баллы на усмотрение преподавателя.

можно было считать корректно работающим, оно должно поддерживать добавления порядка 40 точек в диапазоне $[-5; 5] \times [-5; 5]$.