Об’єктно орієнтоване програмування

**Лабораторна робота №3**

# Графіки Qt

Модуль Qt Charts надає набір простих у використанні компонентів діаграми. Він використовує Qt Graphics View Framework, тому діаграми можна легко інтегрувати в сучасні користувальницькі інтерфейси. Графіки Qt можна використовувати як типи QWidgets, QGraphicsWidget або QML. Користувачі можуть легко створювати графіки, вибравши одну з тем діаграм.

Для використовувати в програмі класів Qt Charts, необхідно підключити бібліотеку:

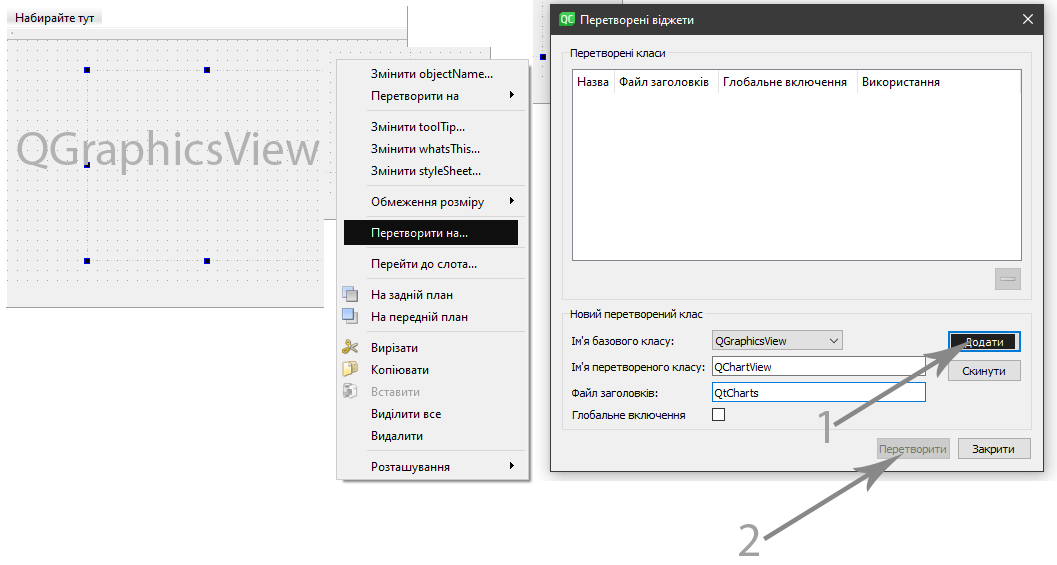
#include <QtCharts>

using namespace QtCharts;

Щоб встановити зв'язок із модулем Qt Charts, додайте цей рядок у файл проекту qmake (файл проєтку \*.pro):

QT += charts

Для того, щоб додати компонент QChart на форму, потрібно виконати дії, як зображено на рисунку 3.1.

Рисунок 3.1 – Додавання компоненту QChart на форму

Файл: *polynomial.h*

#ifndef POLYNOMIAL\_H

#define POLYNOMIAL\_H

#include <QString>

#include <QLineSeries>

class Polynomial

**{**

//Q\_OBJECT

int n**;**

double **\***coef**;**

double lb**,** rb**,** plot\_step**;**

QT\_CHARTS\_NAMESPACE**::**QLineSeries **\***series**;**

public**:**

Polynomial**(**int n**,** double coef**[]);**

**~**Polynomial**();**

double eval**(**double x**);**

QT\_CHARTS\_NAMESPACE**::**QLineSeries **\*** plot**();**

QString getName**();**

**};**

#endif // POLYNOMIAL\_H

Файл: *polynomial.cpp*

#include <math.h>

#include <QString>

#include <QLineSeries>

#include "polynomial.h"

Polynomial**::**Polynomial**(**int n**,** double coef**[])**

**{**

lb **=** **-**5**;**

rb **=** 5**;**

plot\_step **=** **(**rb **-** lb**)/**100**;**

**this->**n **=** n**;**

**this->**coef **=** **new** double**[**n**];**

**for** **(**int i **=** 0**;** i **<** n**;** i**++)**

**this->**coef**[**i**]** **=** coef**[**i**];**

series **=** **new** QT\_CHARTS\_NAMESPACE**::**QLineSeries**();**

**}**

Polynomial**::~**Polynomial**()**

**{**

**if** **(** coef **!=** **nullptr)**

**delete** **[]**coef**;**

**delete** series**;**

**}**

double Polynomial**::**eval**(**double x**)**

**{**

double value **=** 0**;**

**for** **(**int i **=** 0**;**i **<** n**;**i**++)**

**{**

value **+=** coef**[**i**]** **\*** pow**(**x**,** i**);**

**}**

**return** value**;**

**}**

QT\_CHARTS\_NAMESPACE**::**QLineSeries **\*** Polynomial**::**plot**()**

**{**

series**->**clear**();**

double x **=** **this->**lb**;**

**while** **(**x **<** **this->**rb**)**

**{**

series**->**append**(**x**,** **this->**eval**(**x**));**

x **+=** plot\_step**;**

**}**

series**->**setName**(this->**getName**());**

**return** series**;**

**}**

QString Polynomial**::**getName**()**

**{**

QString name**;**

**for** **(**int i **=** 0**;**i **<** n**;**i**++)**

{

if (coef[i] > 0 and i!=0)

name += "+";

name += QString::number(coef[i], 'f', 1);

if (i == 1)

name += "x";

if (i > 1)

name += "x^"+QString::number(i);

}

return name;

}

Файл: *mainwindow.cpp*

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "polynomial.h"

MainWindow**::**MainWindow**(**QWidget **\***parent**)** **:**

QMainWindow**(**parent**),**

ui**(new** Ui**::**MainWindow**)**

**{**

ui**->**setupUi**(this);**

**}**

MainWindow**::~**MainWindow**()**

**{**

**delete** ui**;**

**}**

void MainWindow**::**on\_pushButton\_clicked**()**

**{**

static double coef**[]** **=** **{-**5**,** 1**};**

static Polynomial p**(**2**,** coef**);**

static double coef1**[]** **=** **{-**1**,** **-**3**,** 1**};**

static Polynomial p1**(**3**,** coef1**);**

QChart **\***chart **=** **new** QChart**();**

chart**->**addSeries**(**p**.**plot**());**

chart**->**addSeries**(**p1**.**plot**());**

chart**->**createDefaultAxes**();**

ui**->**chart**->**setChart**(**chart**);**

ui**->**chart**->**setRenderHint**(**QPainter**::**Antialiasing**);**

**}**

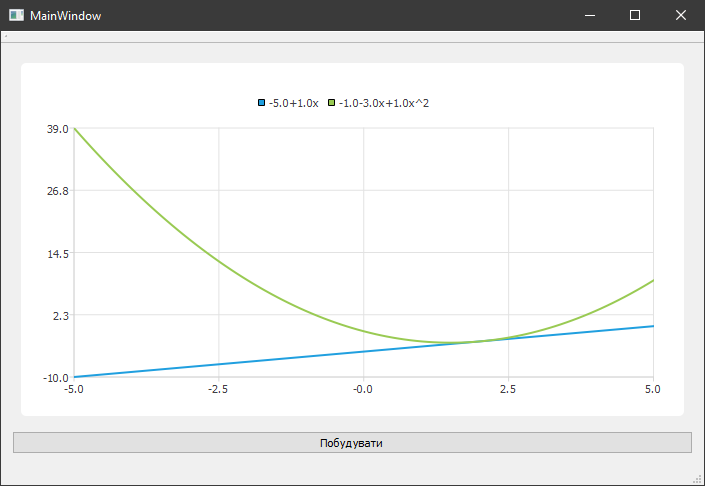


Рисунок 3.1 – приклад роботи програми

**Лабораторна робота №3 (3-й семестр)**

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються три екземпляри класу «поліном», що описують функції *5, x + 1, x2 - 5*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [0; 2]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються чотири екземпляра класу «поліном», що описують функції: *4x + 1, x - 5, x3 + 1, - x2*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-2; 3]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються два примірника класу «поліном», що описують функції: *x2 + 2x + 1, x4 + 1*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-2π; 2π]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються три екземпляри класу «поліном», що описують функції: *x3 + 3x2-3x + 5, 3x-7, 4 - x2*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-5; 2]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються чотири екземпляра класу «поліном», що описують функції: *2, x + 1, 2x2 - 1, 1 - 3x + x6*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-5; 5]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються три екземпляри класу «поліном», що описують функції: *-1 + x, - x + 1, x4 + x2 + 1*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [0; 2π]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення поліномів (+, -, \*). У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються два примірника класу «поліном», що описують функції: *-2x + 7, x + 1 + x2, -5*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-π; 7]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються три екземпляри класу «поліном», що описують функції: *5+ x2 + 1, x4 + x2 - 5, 2 + x3*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-2.5; 3.63]. Діапазон побудови може змінювати користувач.

Написати програму, в якій реалізований клас, що описує поліном ступеня n. У класі визначено оператори для додавання, віднімання і множення (+, -, \*) поліномів. У класі також перевизначений оператор << для виведення графіка полінома на компоненті TChart.

* 1. При запуску програми створюються чотири екземпляра класу «поліном», що описують функції: *3 - 3x + 3x2 - 3x3, 3 - 3x + 3x2, 3 - 3x, 3*.
  2. Користувач може змінювати існуючі поліноми, створювати нові і виконувати зазначені операції над ними.
  3. Побудова графіків за умовчанням проводиться в діапазоні [-4.35; 4.45]. Діапазон побудови може змінювати користувач.

**Варіант \* (Додаткове завдання за додаткові бали)**

Розробити методи класу поліном та продемонструвати їх роботу, які дозволяють

1. Ділити поліноми.
2. Знаходити похідну та первісну.
3. Визначати корені поліному.