

Об'єктно орієнтоване програмування

Лабораторна робота №8

Малювання з QPainter



Теоретичні відомості

У більшості компонентів, присутній подія OnPaint. Ця подія викликається, коли є необхідність в перемальовуванні зображення на компоненті: при зміні розмірів компонента; коли вікно (форма) з компонентом перекривається іншими вікнами, а потім знову стає повністю видимим; і т. п. Відповідно, малювання в таких компонентах потрібно проводити при обробці цієї події.

При виведенні зображень досить часто виникає проблема їх коректного масштабування, тобто перекладу з системи координат моделі в систему координат «полотна». Оскільки прямі лінії при масштабуванні повинні залишатися прямими, то воно здійснюється шляхом лінійного перетворення координат:

$$\begin{aligned}x_c &= A \cdot x_m + B \\y_c &= C \cdot y_m + D\end{aligned}\tag{1}$$

де (x_m, y_m) – координати точки в системі координат моделі; (x_c, y_c) – координати точки в системі координат «полотна»; A, B, C, D – коефіцієнти масштабування.

Коефіцієнти масштабування можуть бути знайдені виходячи з відомих перетворень для граничних точок (рис. 8.1):

$$\begin{aligned}A &= (x_{cmax} - x_{cmin}) / (x_{mmax} - x_{mmin}) \\D &= x_{cmin} - A \cdot x_{mmin} \\C &= (y_{cmin} - x_{cmax}) / (y_{mmax} - y_{mmin}) \\D &= y_{cmin} - C \cdot y_{mmax}\end{aligned}\tag{2}$$

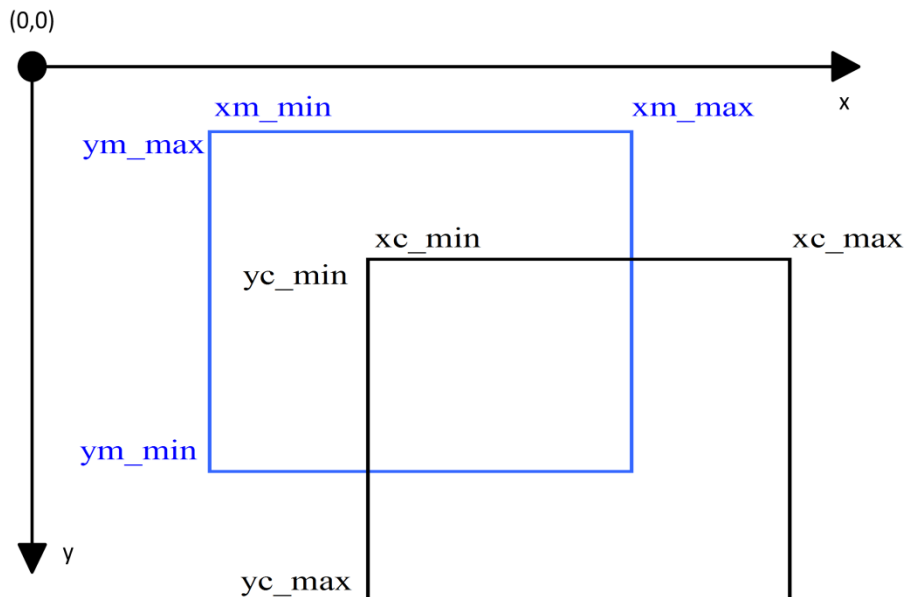


Рисунок 8.1 – Схема взаємозв'язку між системами координат моделі і «полотна»

Приклад. Написати програму, що малює трикутник синім суцільним контуром та заштрихований червоною подвійною діагональною штриховкою на компоненті. Трикутник має бути вписаний в прямокутну область (10,10, width-10, height-10) на «полотні» компонента та масштабуватись при зміні розміру вікна.

Лістинг 8.1 – Файл mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;

    const int xm_min=0, xm_max=1, ym_min=0, ym_max=1;

    double a, b, c, d;

    int _x(double x);
```

```

        int _y(double y);
        bool eventFilter(QObject* watched, QEvent* event);

};

#endif // MAINWINDOW_H

```

Лістинг 8.2 – Файл mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QPainter>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->widget->installEventFilter(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

int MainWindow::_x(double x)
{
    return int(a*x + b);
}

int MainWindow::_y(double y)
{
    return int(c*y + d);
}

bool MainWindow::eventFilter(QObject* watched, QEvent* event)
{
    if (watched == ui->widget && event->type() == QEvent::Paint)
    {
        QPainter p(ui->widget);

        int w = p.device()->width();
        int h = p.device()->height();

        int xc_min = 10;
        int xc_max = w - 10;
        int yc_min = 10;
        int yc_max = h - 10;
        a = (xc_max - xc_min) / (xm_max - xm_min);
        b = xc_min - a * xm_min;
        c = (yc_min - yc_max) / (ym_max - ym_min);
        d = yc_min - c * ym_max;
    }
}

```

```

QPen pen(Qt::GlobalColor::blue);
pen.setWidth(2);
p.setPen(pen);

QBrush brush(Qt::GlobalColor::red,
Qt::DiagCrossPattern);

QPainterPath path;
path.moveTo(_x(0), _y(0));
path.lineTo(_x(0.5), _y(1));
path.lineTo(_x(1), _y(0));
path.lineTo(_x(0), _y(0));

p.drawPath(path);
p.fillPath(path, brush);

return true; // повернути true, якщо ви не хочете, щоб
після цього віджет самостійно малював, інакше поверніть false.
}
return false;
}

```

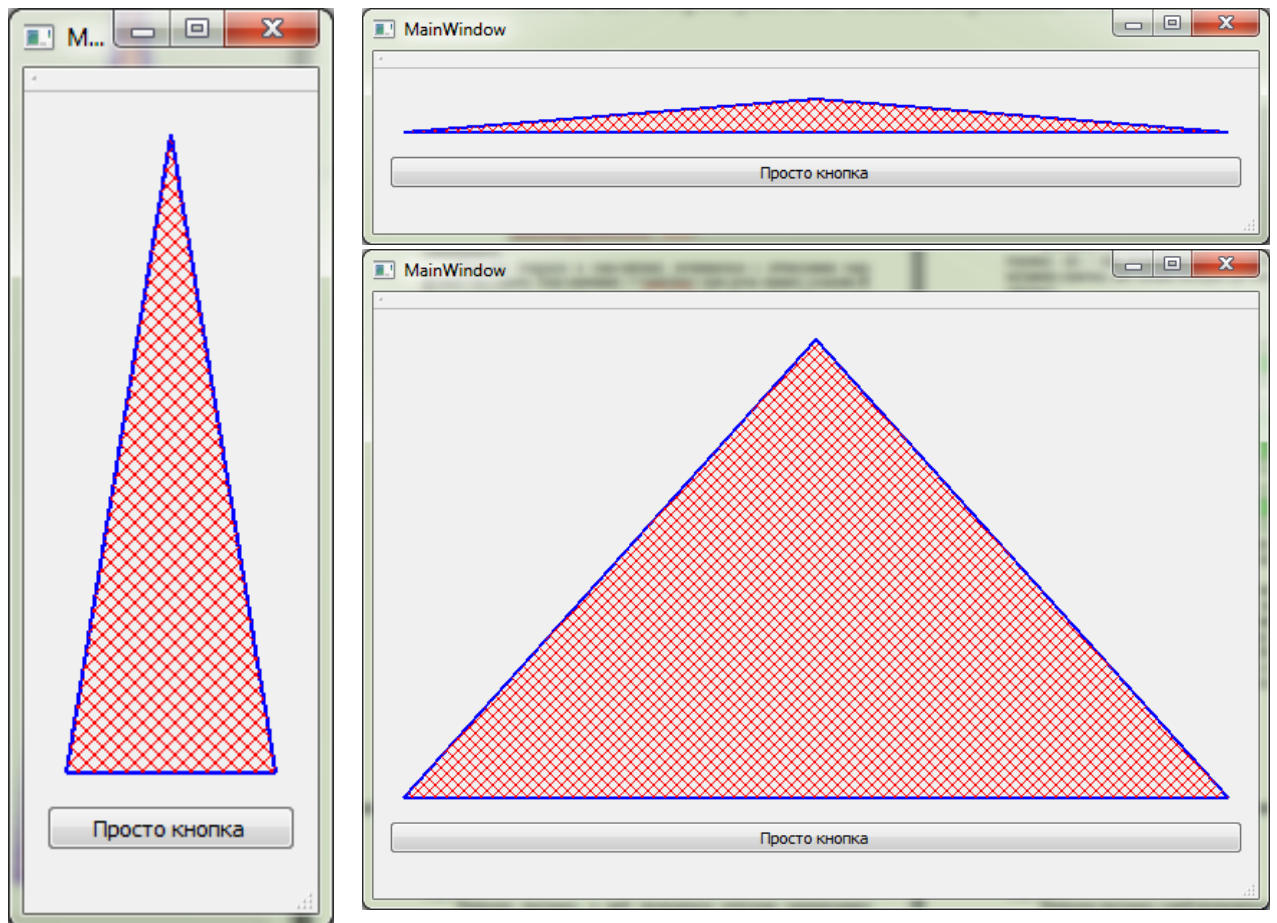


Рисунок 8.2 – Результат роботи програми

Лабораторна робота №8 (3-й семестр)

Варіант 1.

Написати програму, в якій проводиться побудова заштрихованої синім горизонтальним штрихуванням і окресленої зеленим пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2а. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 2.

Написати програму, в якій проводиться побудова заштрихованої червоним вертикальним штрихуванням і окресленої сірим штриховим контуром області, яка максимально близько повторює символ, наведений на рис. 8.2б. При зміні розмірів форми повинно виконуватися автоматичне зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 3.

Написати програму, в якій проводиться побудова заштрихованої зеленим діагональним штрихуванням і окресленої червоним штрихпунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2в. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 4.

Написати програму, в якій проводиться побудова заштрихованої сірим горизонтальним штрихуванням і окресленої чорним пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2г. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 5.

Написати програму, в якій проводиться побудова заштрихованої зеленим вертикальним штрихуванням і окресленої жовтим штриховим контуром області, яка максимально близько повторює символ, наведений на рис. 8.2д. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 6.

Написати програму, в якій проводиться побудова заштрихованої чорним діагональним штрихуванням і окресленої помаранчевим штрих-пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2е. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 7.

Написати програму, в якій проводиться побудова заштрихованої блакитним горизонтальним штрихуванням і окресленої червоним пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2є. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 8.

Написати програму, в якій проводиться побудова заштрихованої жовтим вертикальним штрихуванням і окресленої коричневим штриховим контуром області, яка максимально близько повторює символ, наведений на рис. 8.2ж. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

Варіант 9.

Написати програму, в якій проводиться побудова заштрихованої рожевим діагональним штрихуванням і окресленої блакитним штрих-пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2з. При зміні розмірів форми відбувається автоматична зміна розмірів виведеного малюнка, таким чином, щоб відстань від кордонів прямокутної області, в яку вписаний малюнок, до відповідних меж форми, залишалася постійною.

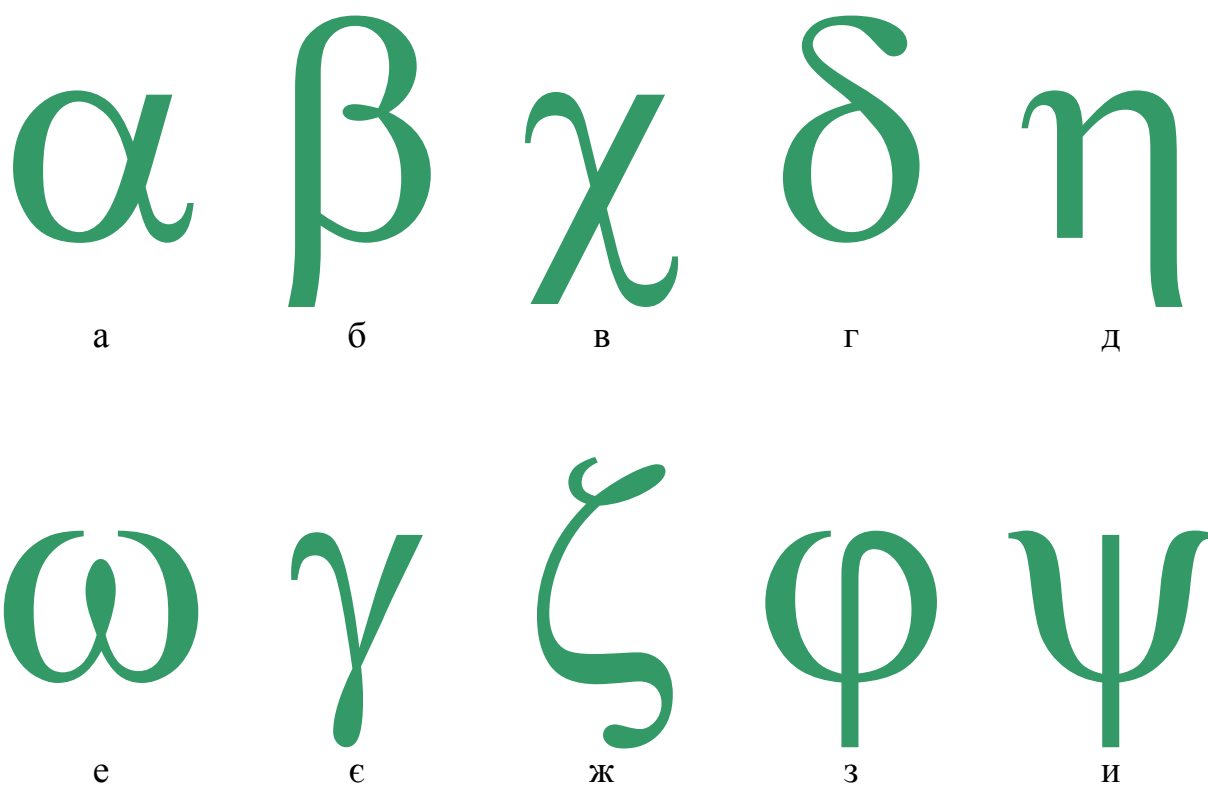


Рисунок 8.1 – Варіанти завдань