Об’єктно орієнтоване програмування

**Лабораторна робота №8**

# Малювання з QPainter

# Теоретичні відомості

**Приклад.** Написати програму, яка хеш таблицю для зберігання пар (рядок, ціле число). Вміст хеш таблиці зв’язується з *QTableWidget* таким чином, щоб користувач міг бачити актуальний стан хеш-таблиці (рис. 7.2).

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; // return true if you do not want to have the child widget paint on its own afterwards, otherwise, return false.

}

return false;

}

|  |  |
| --- | --- |
|  |  |
|  |

Рисунок 8.2 – Результат роботи програми

**Лабораторна робота №8 (3-й семестр)**

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.

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, залишалося постійним.

Написати програму, в якій проводиться побудова заштрихованої зеленим діагональним штрихуванням і окресленої червоним штрих-пунктирним 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.

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.

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.

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.

Написати програму, в якій проводиться побудова заштрихованої блакитним горизонтальним штрихуванням і окресленої червоним пунктирним контуром області, яка максимально близько повторює символ, наведений на рис. 8.2ж. 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.

Написати програму, в якій проводиться побудова заштрихованої жовтим вертикальним штрихуванням і окресленої коричневим штриховим контуром області, яка максимально близько повторює символ, наведений на рис. 8.2з. 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.

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. 8.2і. 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| а | б | в | г | д |
| ω | γ |  |  |  |
| е | є | ж | з | і |

Рисунок 8.1 – Варіанти завдань