Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

ОТЧЕТ

по лабораторной работе №8  
Дисциплина «ООТПиСП»   
Тема: «Создание интерфейса простых приложений»

Выполнила:

студ. гр.ПО-3

Дмитрук В.Г.

Проверил:

Миндер

Брест 2020

**Цель работы:** изучить графические примитивы библиотеки QT.

**Вариант 9**

****

**Код программы:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QtWidgets/QMainWindow>

#include "ui\_mainwindow.h"

#include "qpaintwiget.h"

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

explicit **MainWindow**(QWidget \*parent = 0);

~***MainWindow***();

private:

Ui::MainWindow ui;

// ”казатель на объект нашего класса

QPaintWidget \*wgt;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

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

QMainWindow(parent)

{

ui.setupUi(this);

wgt = new QPaintWidget(this);

ui.verticalLayout->addWidget(wgt);

}

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

{

}

**qpaintwiget.h**

#ifndef QPAINTWIDGET\_H

#define QPAINTWIDGET\_H

#include <QtWidgets/QWidget>

#include <QTimer>

class **QPaintEvent**;

class **QPaintWidget** : public QWidget

{

Q\_OBJECT

int offset;

QTimer \*timer;

QRectF rect1=QRectF(0, 0, 50, 20);

public:

**QPaintWidget**(QWidget \* parent = 0);

QRectF **showRect**(QRectF &rect);

QPainter **showEllipse**(QPainter &el);

protected:

void ***paintEvent***(QPaintEvent \*);

private slots:

void **UpdatePixmap**();

};

#endif // QPAINTWIDGET\_H

**qpaintwiget.cpp**

#include "qpaintwiget.h"

#include <QPainter>

#include <QTextStream>

#include <QtMath>

#include <QPainterPath>

QTextStream cout(stdout);

QPaintWidget::**QPaintWidget**(QWidget \* parent) : QWidget(parent)

{

timer = new QTimer();

connect(timer, SIGNAL(timeout()), this, SLOT(UpdatePixmap()));

timer->start(100); // И запустим таймер

}

void QPaintWidget::***paintEvent***(QPaintEvent \*)

{

QPainter p(this); // Создаём новый объект рисовальщика

p.setPen(QPen(Qt::red,6,Qt::SolidLine)); // Настройки кисти

p.setFont(QFont("Yu Gothic", 20, QFont::DemiBold)); // Настройки шрифта

QTransform transform; // создание объекта трансформации transform

transform.rotate(90,Qt::ZAxis);

transform.rotate(0,Qt::YAxis);

transform.rotate(0,Qt::XAxis);

transform.translate(60,-20);

p.setTransform(transform); // устанавливаем наши трансформации transform к рисовальщику.

p.drawText(rect(), "OOTPiSP lab №8"); // рисуется с учётом transform

p.resetTransform();

p.setPen(QPen(Qt::blue,3,Qt::SolidLine)); // Настройки кисти

p.setRenderHint(QPainter::Antialiasing, true);

QPainterPath Path;

for(float Ux = 0; Ux < width(); Ux++)

Path.lineTo(QPointF(Ux, pow(Ux,3)/100));

p.drawPath(Path);

//в том случае если картинка не меняется

p.drawRect(showRect(*rect1*));

p.drawEllipse(rect1.x()+10,rect1.y()+20,25,25);

}

QRectF QPaintWidget::**showRect**(QRectF &rect)

{

float Ux= rect.x() + 1;

float Uy= pow(Ux,3)/100;

rect.moveTopLeft(QPointF(Ux,Uy));

return rect;

}

void QPaintWidget::**UpdatePixmap**()

{

update();

}

**main.cpp**

#include "mainwindow.h"

#include <QApplication>

int **main**(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

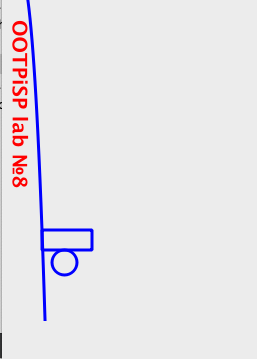
MainWindow w;

w.show();

return a.exec();

}

**Результат работы**:



**Вывод:** изучили графические примитивы библиотеки QT.