

Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
По дисциплине: «ООТПиСП»
Тема: «Использование библиотеки элементов графического интерфейса Qt»

Выполнил:
Студент 3 курса
Группы ПО-7
Комиссаров А.Е.
Проверила:
Хацкевич М.В.

Цель: научиться использовать библиотеку элементов графического интерфейса Qt.

Задание (4 балла)

- пользуясь примером в каталоге lab08/02, создайте приложение с графическим интерфейсом, аналогичным представленному сверху
- используйте классы QLabel, QSpinBox, QSlider, QPlainTextEdit.

Знакомство с элементами интерфейса: добавим функциональность. (3 балла)

- Добавим функциональность созданному на предыдущем этапе приложению:
 - спин-боксы и слайдеры будут перемещаться синхронизировано в диапазоне значений от 0 до 255.
 - Цвет фона QPlainTextEdit будет меняться соответственно
- Выполнение:
 - Выставляем диапазон допустимых значений для QSpinBox и QSlider при помощи методов setMinimum() и setMaximum()
 - запрещаем ввод в текстовое поле: setEnabled(false)
 - Реализуем метод setColor() и слоты setRed(int), setGreen(int), setBlue(int)
 - к слотам подключаем сигналы QSlider::sliderMoved() и QSpinBox::valueChanged()
 - в реализации слотов синхронизируем значения слайдера и спин-бокса и вызываем setColor()
- Для изменения цвета фона текстового поля воспользуемся таблицами стилей для описания стиля элементов.
 - таблицы стилей используют синтаксис CSS
 - будем задавать цвет в виде строки типа #rrggbb
 - таким образом, надо задать QPlainTextEdit следующий стиль:
 - QPlainTextEdit { background: #rrggbb; }
 - задаём стиль при помощи метода setStyleSheet() (таблица стиля передаётся в виде строки).

Ход выполнения задания:

Файл main.cpp

```
#include <QApplication>
#include <QWidget>
#include <QLabel>
#include <QSpinBox>
#include <QSlider>
#include <QPlainTextEdit>
#include <QBoxLayout>
#include <QPushButton>
#include <iostream>
#include <sstream>

class QMyTextEdit : public QPlainTextEdit
{
public slots:
    QMyTextEdit(){m_red = 0; m_green = 0; m_blue = 0;}
    void setRed(int red){
        m_red = red;
        setColor();
    }
}
```

```

void setGreen(int green){
    m_green = green;
    setColor();
}
void setBlue(int blue){
    m_blue = blue;
    setColor();
}
public:
    void setColor(){
        QString hex = QString::fromStdString(rgb2hex(m_red, m_green, m_blue));
        QString start = "QPlainTextEdit { background: #";
        QString end = "; }";
        QString result = start + hex + end;
        this->setStyleSheet(result);
    }
private:
    std::string rgb2hex(int r, int g, int b){
        char hexColor[8];
        std::snprintf(hexColor, sizeof hexColor, "%02x%02x%02x", r, g, b);
        return hexColor;
    }
    int m_red;
    int m_green;
    int m_blue;
};

```

```

class QSlider2 : public QSlider{
public slots:
    void valueMax(){
        this->setValue(255);
    }
    void valueMid(){
        this->setValue(128);
    }
    void valueMin(){
        this->setValue(0);
    }
};

```

```

int main (int argc, char **argv)
{
    QApplication app(argc, argv);
    QWidget widget(0);
    QHBoxLayout hbox;
    widget.setLayout(&hbox);

    //Левый край, середина, правый край окна
    QVBoxLayout left_vbox, middle_vbox, right_vbox;

    hbox.addLayout(&left_vbox, 0);
    hbox.addLayout(&middle_vbox, 0);
    hbox.addLayout(&right_vbox, 1); //1 - разрешить растягивание

    //Левый край (текст)
    QLabel sb_label1, sb_label2, sb_label3;
    sb_label1.setText("Red");
    sb_label2.setText("Green");
    sb_label3.setText("Blue");
    left_vbox.addWidget(&sb_label1, Qt::AlignRight);
    left_vbox.addWidget(&sb_label2, Qt::AlignRight);
    left_vbox.addWidget(&sb_label3, Qt::AlignRight);
}

```

```

//середина (спинбоксы)
QSpinBox sb1, sb2, sb3;
sb1.setRange(0, 255);    sb2.setRange(0, 255);    sb3.setRange(0, 255);
middle_vbox.addWidget(&sb1);    middle_vbox.addWidget(&sb2);    middle_vbox.addWidget(&sb3);
//правый край (слайдеры, а также их connect или привязка)
QSlider2 rb1, rb2, rb3;
rb1.setOrientation(Qt::Horizontal);
rb2.setOrientation(Qt::Horizontal);
rb3.setOrientation(Qt::Horizontal);
right_vbox.addWidget(&rb1);    right_vbox.addWidget(&rb2);    right_vbox.addWidget(&rb3);
rb1.setRange(0,255);    rb2.setRange(0,255);    rb3.setRange(0,255);
QSlider::connect(&rb1, &QSlider::valueChanged, &sb1, &QSpinBox::setValue);
QSlider::connect(&rb2, &QSlider::valueChanged, &sb2, &QSpinBox::setValue);
QSlider::connect(&rb3, &QSlider::valueChanged, &sb3, &QSpinBox::setValue);
QSpinBox::connect(&sb1, &QSpinBox::valueChanged, &rb1, &QSlider::setValue);
QSpinBox::connect(&sb2, &QSpinBox::valueChanged, &rb2, &QSlider::setValue);
QSpinBox::connect(&sb3, &QSpinBox::valueChanged, &rb3, &QSlider::setValue);
//большой квадрат справа
QMyTextEdit text_edit;
text_edit.setEnabled(false);
hbox.addWidget(&text_edit, 1);
text_edit.setColor();
QSpinBox::connect(&sb1, &QSpinBox::valueChanged, &text_edit, &QMyTextEdit::setRed);
QSpinBox::connect(&sb2, &QSpinBox::valueChanged, &text_edit, &QMyTextEdit::setGreen);
QSpinBox::connect(&sb3, &QSpinBox::valueChanged, &text_edit, &QMyTextEdit::setBlue);
QPushButton bb1, bb2, bb3;
QPushButton::connect(&bb3, &QPushButton::clicked, &rb1, &QSlider2::valueMax);
QPushButton::connect(&bb3, &QPushButton::clicked, &rb2, &QSlider2::valueMax);
QPushButton::connect(&bb3, &QPushButton::clicked, &rb3, &QSlider2::valueMax);
QPushButton::connect(&bb2, &QPushButton::clicked, &rb1, &QSlider2::valueMid);
QPushButton::connect(&bb2, &QPushButton::clicked, &rb2, &QSlider2::valueMid);
QPushButton::connect(&bb2, &QPushButton::clicked, &rb3, &QSlider2::valueMid);
QPushButton::connect(&bb1, &QPushButton::clicked, &rb1, &QSlider2::valueMin);
QPushButton::connect(&bb1, &QPushButton::clicked, &rb2, &QSlider2::valueMin);
QPushButton::connect(&bb1, &QPushButton::clicked, &rb3, &QSlider2::valueMin);

hbox.addWidget(&bb1); hbox.addWidget(&bb2); hbox.addWidget(&bb3);

//подгоняем свойства главного окна
widget.setWindowTitle("Lab1 Zad1");
widget.resize(500, 200);

widget.show();
return app.exec();
}

```

Результат работы программы:

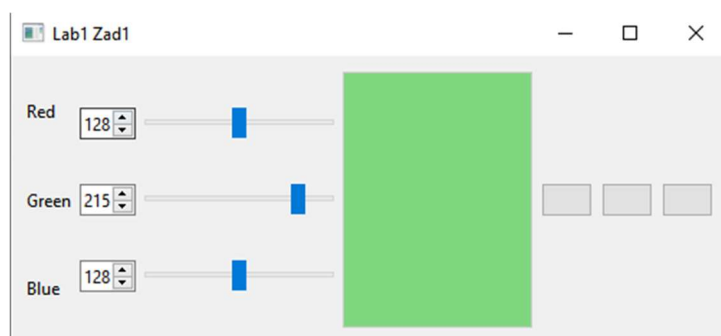


Рис. 1 – Результат работы программы (1)

Диалоги (3 балла)

Библиотека графических элементов Qt предлагает набор из нескольких полнофункциональных диалоговых окон, позволяющих выполнять некоторые стандартные операции. Среди них такие, как выбор файла в файловой системе, выбор шрифта, выбор цвета, диалог печати и некоторые другие. Мы рассмотрим в настоящей работе пример использования диалога выбора файла - QFileDialog.

Создаём простейший обозреватель текстовых файлов.

- Создайте новый виджет и поместите на него элемент QTextEdit.
- Добавьте кнопку QPushButton и подключите её сигнал clicked() к слоту openFile()
- Реализуйте в слоте выбор имени файла пользователем: QFileDialog::getOpenFileName()
- Откройте QFile в соответствии с выбранным названием
- Прочитайте его содержимое и поместите в виде текста в элемент QTextEdit

Ход выполнения задания:

Файл main.cpp

```
#include <QApplication>
#include <QWidget>
#include <QLabel>
#include <QSpinBox>
#include <QSlider>
#include <QTextEdit>
#include <QBoxLayout>
#include <QPushButton>
#include <QFileDialog>
#include <iostream>
#include <sstream>

class QMyTextEdit : public QTextEdit{
public slots:
    void openFile(){
        QString result;
        QString file_name = QFileDialog::getOpenFileName();
        QFile file(file_name);
        if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
            return;

        QTextStream in(&file);
        while (!in.atEnd()) {
            QString line = in.readLine();
            QString newline = "\n";
            result = result + line + newline;
        }
        this->setText(result);
    }
};

////////////////////////////////////
int main (int argc, char **argv)
{
    QApplication app(argc, argv);
    QWidget widget(0);
```

```

QHBoxLayout hbox;
widget.setLayout(&hbox);

QMyTextEdit text_edit;
QPushButton button1;
button1.setText("Open File");

QPushButton::connect(&button1, &QPushButton::clicked, &text_edit, &QMyTextEdit::openFile);

hbox.addWidget(&text_edit, 7);
hbox.addWidget(&button1, 1);

widget.setWindowTitle("Lab1 Zad1");
widget.resize(500, 500);

widget.show();
return app.exec();
}

```

Результат работы программы:

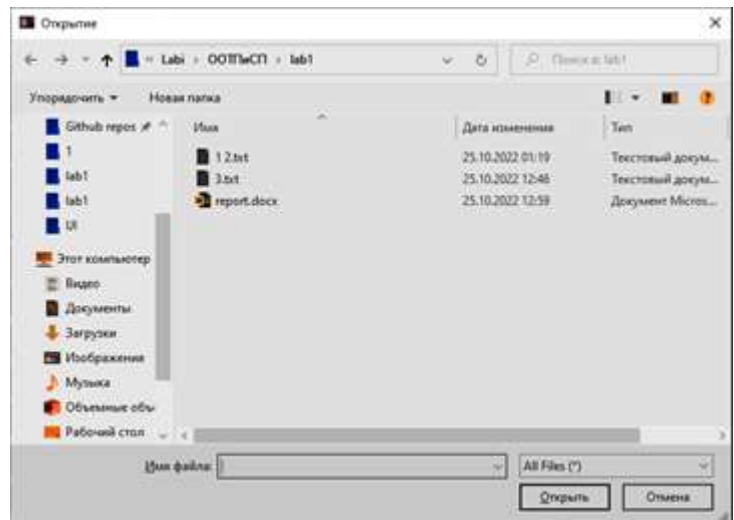
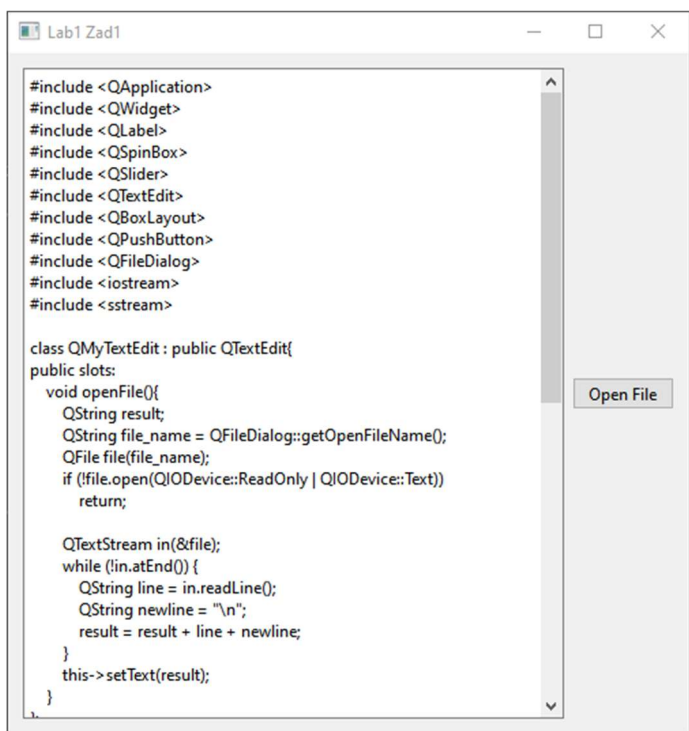


Рис. 1, 2 – Результат работы программы (2)

Вывод: я научился использовать библиотеку элементов графического интерфейса Qt.