

Лабораторна робота №13

Тема: Механізми сигналів і слотів. Класи QWidget та QDialog.

Мета: Навчитись застосовувати механізм сигналів і слотів фреймворку Qt, розробляти програмне забезпечення з використанням класів QWidget та QDialog.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Створити новий графічний Qt-проект.
2. В класі MainWindow оголосити два приватні слоти: OpenModalDialog() та OpenNotModalDialog(), які відображатимуть відповідно модальне і не модальне вікно:
 - за допомогою слота OpenModalDialog() відображати довільне повідомлення про критичну помилку (за допомогою класу QMessageBox());
 - за допомогою слота OpenNotModalDialog() відображати не модальне вікно для отримання назви файлу, який потрібно відкрити. Для цього використати метод QFileDialog::getOpenFileName().
3. Розмістити на формі вікна класу MainWindow дві кнопки pbOpenModalDialog і pbOpenNotModalDialog. За допомогою методу connect() підключити сигнал clicked() даних кнопок до визначених в другому пункті слотів.
4. Додати ще одну кнопку «Відключити», при кліку по якій відключити раніше підключені сигнали.
5. Переробити логіку роботи кнопки «Відключити», таким чином, щоб при повторному кліку по ній відбувалось підключення сигналів, якщо вони відключені, і відключення, якщо вони підключені. При цьому змінювати назву кнопки із «Відключити» на «Підключити» і наоборот.
6. Перевірити роботу програми.

Лістинг:

mainwindow.h:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT
```

```

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

private:
    Ui::MainWindow *ui;
    bool pbClicked;
private slots:
    void OpenModalDialog();
    void OpenNotModalDialog();
    void on_pushButton_clicked();
};
#endif // MAINWINDOW_H

```

mainwindow.cpp:

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDialog>
#include <QPushButton>
#include <QLabel>
#include <QMessageBox>
#include <QFileDialog>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    this->pbClicked = false;
    ui->setupUi(this);
    connect(ui->pbOpenModalDialog, &QPushButton::clicked, this,
    &MainWindow::OpenModalDialog);
    connect(ui->pbOpenNotModalDialog, &QPushButton::clicked, this,
    &MainWindow::OpenNotModalDialog);
}

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

void MainWindow::OpenModalDialog(){
    QMessageBox *modalWid = new QMessageBox(this);
    modalWid->setModal(true);
    modalWid->setText("Critical error");
    modalWid->show();
}

void MainWindow::OpenNotModalDialog(){
    QDialog *modalNotWid = new QDialog(this);
    QLabel *lb = new QLabel(modalNotWid);
    lb->setText(QFileDialog::getOpenFileName());
    lb->show();
    modalNotWid->show();
}

void MainWindow::on_pushButton_clicked()
{
    if(this->pbClicked){
        ui->pushButton->setText("Disconnect");
    }
}

```

```

        connect(ui->pbOpenModalDialog, &QPushButton::clicked, this,
&MainWindow::OpenModalDialog);
        connect(ui->pbOpenNotModalDialog, &QPushButton::clicked, this,
&MainWindow::OpenNotModalDialog);
        this->pbClicked = false;
        ui->pushButton->show();
    }
    else{
        ui->pushButton->setText("Disconnect");
        ui->pushButton->setText("Connect");
        disconnect(ui->pbOpenModalDialog, &QPushButton::clicked, this,
&MainWindow::OpenModalDialog);
        disconnect(ui->pbOpenNotModalDialog, &QPushButton::clicked, this,
&MainWindow::OpenNotModalDialog);
        this->pbClicked = true;
        ui->pushButton->show();
    }
}

```

main.cpp:

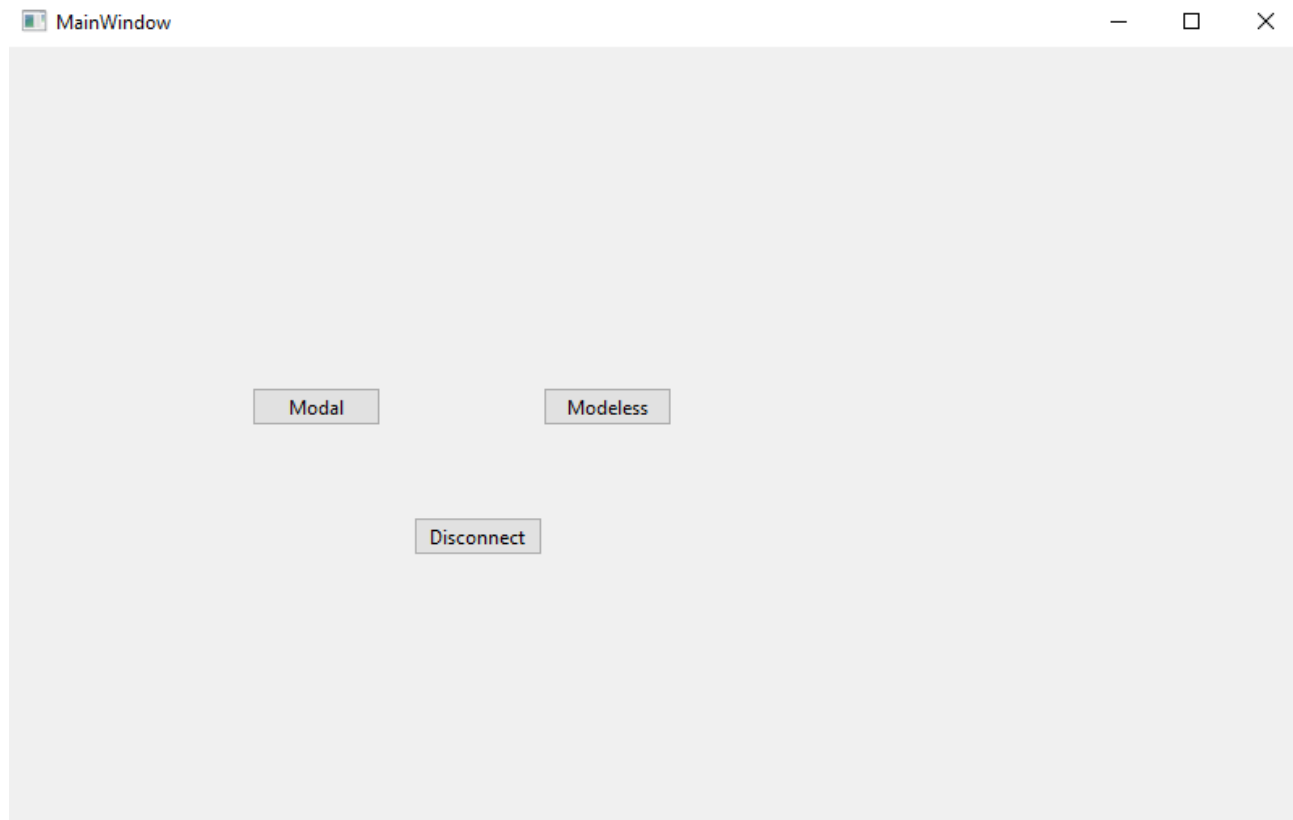
```

#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

```

Вікно форми:



Висновок: навчитися застосовувати механізм сигналів і слотів фреймворку Qt, розробляти програмне забезпечення з використанням класів QWidget та QDialog.