

Лабораторная работа №3
«ИССЛЕДОВАНИЕ СПОСОБОВ ПОСТРОЕНИЯ ИНТЕРФЕЙСА
ПОЛЬЗОВАТЕЛЯ НА БАЗЕ QTWIDGETS»

Цель работы

Изучить основные методики создания графического пользовательского интерфейса с использованием виджетов QtWidgets. Приобрести навыки разработки интерфейса пользователя для приложений на основе фреймворка Qt.

Постановка задачи

1. Создать проект Qt Gui Application.
2. Разработать графический интерфейс приложения согласно варианту задания: Форма добавления новости (заголовок (Line Edit), текст новости (Text Edit), дата (Date Edit));.
3. Используя механизм сигналов и слотов добавить функционал, согласно варианту задания: Добавление на форму таблицы и добавление в неё введенных данных..
4. Оформить отчет по работе.

Ход работы

1. Был разработан графический интерфейс приложения для создания новостей, состоящих из заголовка новости, текста новости и даты. Внешний вид приложения представлен на рисунке 1.

The screenshot shows a window titled "News poster" with a close button (X) in the top right corner. Inside the window, there are three input fields: "Title" (a single-line text box), "Text" (a multi-line text area), and "Date" (a date picker showing "31.12.1999"). To the right of the "Date" field is a "Post" button. Below these fields is a table with three columns: "Title", "Text", and "Date". The table is currently empty.

Рисунок 1 – Графический интерфейс приложения

2. Был создан функционал, позволяющий добавлять введённые данные о новости в таблицу. Результат корректного выполнения программы представлен на рисунке 2.

The screenshot shows the same "News poster" window, but now with data entered. The "Title" field contains "Post 2", the "Text" field contains "Lorem ipsum dolores sanctum. Avatus saxuh dolor sip apteme.", and the "Date" field shows "11.07.2030". The "Post" button is still visible. Below the input fields, the table now contains two rows of data:

	Title	Text	Date
1	Post 1	Lorem ipsum dolores sanctum	2050.06.21
2	Post 2	Lorem ipsum dolores sanctum. Avatus sax...	2030.07.11

Рисунок 2 – Результат выполнения программы

ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены основные методики создания графического пользовательского интерфейса с использованием виджетов QtWidgets.

Можно сделать вывод о том, что написание пользовательского интерфейса с использованием Designer позволяет довольно быстро и легко создать сам интерфейс, но для создания более сложных интерфейсов он не приспособлен. Поэтому, написание графического интерфейса пользователем используя код более предпочтительно при написании громоздких приложений. Такой метод позволяет создавать более гибкие и настраиваемые интерфейсы, также становится проще отладка. Также написание интерфейсов кодом может обеспечить более высокую производительность, ведь не требуется загрузка и интерпретация XML файлов, создаваемых Designer'ом. Также, одним из важных достоинств использования такого метода является то, что код интерфейса проще хранить в системе контроля версий, и, соответственно удобнее работать с такими интерфейсами в группе. Изучить основные методики создания графического пользовательского интерфейса с использованием виджетов QtWidgets.

ПРИЛОЖЕНИЕ А

«ИСХОДНЫЙ КОД ПРОГРАММЫ»

Листинг А1 – Файл mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

#include "QTableWidget"
#include "QPushButton"
#include "QTextEdit"
#include "QDateEdit"

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;
    QPushButton *postNews;
    QTableWidget *table;
    QTextEdit *newsText;
    QLineEdit *newsTitle;
    QDateEdit *newsDate;

private slots:
    void onPushClick();
};
#endif // MAINWINDOW_H
```

Листинг А2 – Файл mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include "QLineEdit"
#include "QTextEdit"
#include "QDateEdit"
#include "QLayout"
#include "QLabel"
#include "QPushButton"
```

```

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

    setWindowTitle("News poster");

    // Title LineEdit and Label vertical box
    newsTitle = new QLineEdit();
    QLabel *titleLabel = new QLabel("Title");
    QVBoxLayout *titleLabelLayout = new QVBoxLayout();
    titleLabelLayout->addWidget(titleLabel);
    titleLabelLayout->addWidget(newsTitle);

    // Text TextEdit and Label vertical box
    newsText = new QTextEdit();
    QLabel *textLabel = new QLabel("Text");
    QVBoxLayout *textLabelLayout = new QVBoxLayout();
    textLabelLayout->addWidget(textLabel);
    textLabelLayout->addWidget(newsText);

    // Text and Title vertical box
    QVBoxLayout *titleLabelAndTextLayout = new QVBoxLayout();
    titleLabelAndTextLayout->addLayout(titleLabelLayout);
    titleLabelAndTextLayout->addLayout(textLabelLayout);

    // Date DateEdit and Label vertical box
    newsDate = new QDateEdit();
    QLabel *dateLabel = new QLabel("Date");
    QVBoxLayout *dateLabelLayout = new QVBoxLayout();
    dateLabelLayout->addWidget(dateLabel);
    dateLabelLayout->addWidget(newsDate);

    // Date box and PushButton vertical box
    postNews = new QPushButton("Post");
    QVBoxLayout *dateAndConfirmLayout = new QVBoxLayout();
    dateAndConfirmLayout->addStretch(60);
    dateAndConfirmLayout->addLayout(dateLabelLayout);
    dateAndConfirmLayout->addWidget(postNews);
    dateAndConfirmLayout->addStretch(60);

    QHBoxLayout *formHLayout = new QHBoxLayout();
    formHLayout->addLayout(titleLabelAndTextLayout);
    formHLayout->addLayout(dateAndConfirmLayout);

    // Table
    table = new QTableWidgetItem(0,3);
    table->setShowGrid(false);
    table->setHorizontalHeaderLabels({"Title", "Text", "Date"});

    table->horizontalHeader()->setSectionResizeMode(QHeaderView::Stretch);

    connect(postNews, SIGNAL(clicked()), this, SLOT(onPushClick()));

```

```

    QVBoxLayout *centralLayout = new QVBoxLayout();
    centralLayout->addLayout(formHLayout);
    centralLayout->addWidget(table);

    QWidget *centralWidget = new QWidget();
    centralWidget->setLayout(centralLayout);
    this->setCentralWidget(centralWidget);
}

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

void MainWindow::onPushClick()
{
    int currentRowCount = table->rowCount();
    table->insertRow(currentRowCount);
    table->setItem(currentRowCount, 0, new
QTableWidgetItem(newsTitle->text()));
    table->setItem(currentRowCount, 1, new
QTableWidgetItem(newsText->toPlainText()));
    table->setItem(currentRowCount, 2, new
QTableWidgetItem(newsDate->date().toString("yyyy.MM.dd")));
}

```