МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы № 5

по дисциплине “Кроссплатформенное Программирование”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Строганов В. А.

Севастополь

2023

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«Исследование способов интеграции интерфейса пользователя на языке QML и функциональности на языке C++»**

**Цель работы**

Исследование способов взаимодействия языка C++ и языка разметки QML. Приобретение навыков разработки приложений на основе QML интерфейса.

**Задачи**

1. Изучить способы организации взаимодействия QML и серверных классов на C++.
2. Разработать класс, реализующий функциональность по варианту задания, приведенному в Приложении.
3. Определить свойства и методы, необходимые для использования в QML разметке, с помощью соответствующих макросов.
4. Добавить класс в контекст Qt Quick приложения.
5. Дополнить разметку необходимыми элементами управления с вызовом соответствующих методов.
6. Исследовать эффективность работы полученного приложения, имитируя ошибки ввода/вывода.
7. Выполнить сравнительный анализ методов построения приложений в данной лабораторной работе и работе №3 по критерию трудоемкости проектирования и программирования.

**Вариант 1**

Добавить кнопки, позволяющие сохранять и загружать текст из выбранного поля ввода.

**Ход работы**

Ниже представлен интерфейс приложения, который видит пользователь при запуске программы (Рисунок 1). Также представлен результат выгрузки текста из загрузки из файла. (Рисунок 2).

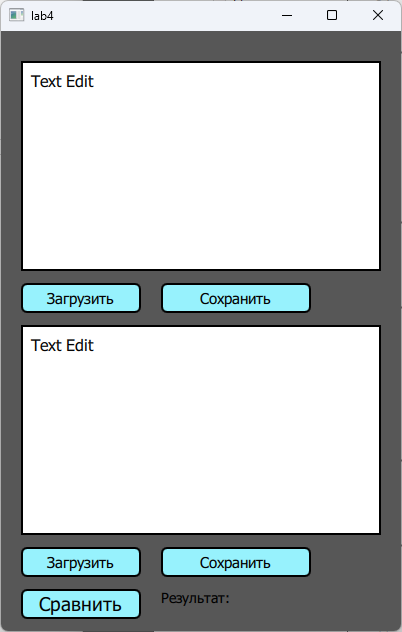


Рисунок 1 – Запуск программы

****

Рисунок 2 – Загрузка теста из файла

После нажатия кнопки “Сохранить” текст сохраняется в файлах file1.txt и file2.txt (Рисунок 3). Ниже представлен результат сохранения (Рисунок 4).



Рисунок 3 – Сохранение текста в файлы

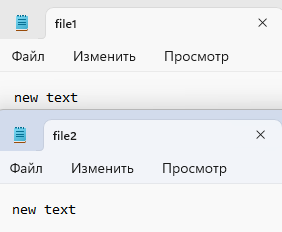


Рисунок 4 – Результат сохранения

**Текст программы:**

**Button.qml**

import QtQuick 2.0

Rectangle {

id: *button*

property int buttonHeight: 75

property int buttonWidth: 150

property string label:"Сравнить"

property real labelSize: 14

radius: 6

antialiasing: true

border {

width: 2

color: "#000"

}

width: *buttonWidth*;

height: *buttonHeight*

color: "#97f2fd"

Text {

id: *buttonLabel*

anchors.centerIn: *parent*

text: *label*

color: "#000"

font.pointSize: *labelSize*

Behavior on color {

ColorAnimation {

duration: 200

}

}

}

signal buttonClick()

MouseArea {

id: *buttonMouseArea*

anchors.fill: *parent*

onClicked: *buttonClick*()

hoverEnabled: true

onEntered: {

*parent*.color = "#0857c0"

*buttonLabel*.color = "#fff"

}

onExited: {

*parent*.color = "#97f2fd"

*buttonLabel*.color = "#000"

}

}

scale: *buttonMouseArea*.pressed ? 1.2 : 1

Behavior on scale {

NumberAnimation {

duration: 200

}

}

Behavior on color {

ColorAnimation {

duration: 200

}

}

}

**main.qml**

import QtQuick 2.12

import QtQuick.Window 2.12

Window {

id: *window*

width: 400

height: 600

color: "#575757"

visible: true

title: *qsTr*("lab4")

minimumWidth: 300

minimumHeight: 400

Rectangle {

id: *rectangle*

y: *parent*.height\*0.05

height: *parent*.height\*0.35

width: *parent*.width\*0.9

anchors.horizontalCenter: *parent*.horizontalCenter

color: "#ffffff"

border.width: 2

TextEdit {

id: *textEdit*

width: *parent*.width

height: *parent*.height

anchors.horizontalCenter: *parent*.horizontalCenter

color: "#000000"

text: *qsTr*("Text Edit")

font.pixelSize: 16

selectionColor: "#000000"

selectByMouse: true

textMargin: 10

}

}

Button {

id: *button1*

y: *parent*.height\*0.42

height: *parent*.height\*0.05

width: *parent*.width\*0.3

anchors.left: *parent*.left

anchors.leftMargin: *parent*.width/20

labelSize: 11

label: "Загрузить"

onButtonClick:

{

*textEdit*.text = *file*.load("file1.txt")

}

}

Button {

id: *button2*

y: *parent*.height\*0.42

height: *parent*.height\*0.05

anchors.left: *button1*.right

anchors.leftMargin: *parent*.width/20

labelSize: 11

label: "Сохранить"

onButtonClick:

{

*file*.save("file1.txt", *textEdit*.text)

}

}

Rectangle {

id: *rectangle1*

y: *parent*.height\*0.49

height: *parent*.height\*0.35

width: *parent*.width\*0.9

anchors.horizontalCenter: *parent*.horizontalCenter

color: "#ffffff"

border.width: 2

TextEdit {

id: *textEdit1*

width: *parent*.width

height: *parent*.height

anchors.horizontalCenter: *parent*.horizontalCenter

color: "#000000"

text: *qsTr*("Text Edit")

font.pixelSize: 16

selectionColor: "#000000"

selectByMouse: true

textMargin: 10

}

}

Button {

id: *button3*

y: *parent*.height\*0.86

height: *parent*.height\*0.05

width: *parent*.width\*0.3

anchors.left: *parent*.left

anchors.leftMargin: *parent*.width/20

label: "Загрузить"

labelSize: 11

onButtonClick:

{

*textEdit1*.text = *file*.load("file2.txt")

}

}

Button {

id: *button4*

y: *parent*.height\*0.86

height: *parent*.height\*0.05

anchors.left: *button3*.right

anchors.leftMargin: *parent*.width/20

label: "Сохранить"

labelSize: 11

onButtonClick:

{

*file*.save("file2.txt", *textEdit1*.text)

}

}

Button {

id: *button*

y: *parent*.height\*0.93

height: *parent*.height\*0.05

width: *parent*.width\*0.3

anchors.left: *parent*.left

anchors.leftMargin: *parent*.width/20

label: "Сравнить"

onButtonClick:

{

*text1*.text = *textEdit*.text == *textEdit1*.text ? "Текст совпадает" : "Текст не совпадает"

}

}

Text {

id: *text1*

y: *parent*.height\*0.93

height: *parent*.height\*0.05

text: *qsTr*("Результат:")

anchors.left: *button*.right

font.pixelSize: *parent*.width/28

anchors.leftMargin: *parent*.width/20

}

}

**main.cpp**

#include <QGuiApplication>

#include <QQmlApplicationEngine>

#include <QQmlContext>

#include <QQmlEngine>

#include <QQuickView>

#include "file.h"

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

{

#if QT\_VERSION < QT\_VERSION\_CHECK(6, 0, 0)

QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);

#endif

QGuiApplication app(*argc*, argv);

QQmlApplicationEngine engine;

file file;

const QUrl url(QStringLiteral("qrc:/main.qml"));

QObject::connect(&engine, &QQmlApplicationEngine::objectCreated,

&app, [url](QObject \*obj, const QUrl &objUrl) {

if (!obj && url == objUrl)

QCoreApplication::exit(-1);

}, Qt::QueuedConnection);

engine.rootContext()->setContextProperty("file", &file);

engine.load(url);

return app.exec();

}

**file.cpp**

#include "file.h"

file::**file**(QObject \*parent) : QObject(parent)

{

}

QString file::**load**(QString fileName)

{

QFile file(fileName);

QString result = "";

if (file.*open*(QFile::ReadWrite))

{

QTextStream inStream(&file);

result = inStream.readAll();

}

file.*close*();

return result;

}

void file::**save**(QString fileName, QString text)

{

QFile file(fileName);

if (file.*open*(QFile::ReadWrite | QIODevice::Truncate))

{

QTextStream outStream(&file);

outStream << text;

}

file.*close*();

}

**file.h**

#ifndef FILE\_H

#define FILE\_H

#include <QObject>

#include <QQuickItem>

#include <QtGui/QGuiApplication>

#include <QFile>

#include <QTextStream>

class **file** : public QObject

{

Q\_OBJECT

public:

**file**(QObject \*parent = 0);

Q\_INVOKABLE QString **load**(QString fileName);

Q\_INVOKABLE void **save**(QString fileName, QString text);

};

#endif // FILE\_H

**Выводы**

В ходе выполнения лабораторной работы был изучен способ интеграции C++ кода с QML. Данный метод построения приложений следует применять в следующих случае необходимости получить доступ к функциональности извне окружения QML, для повышения эффективности выполнения, для более сложного кода.