МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

 «БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

ЛАБОРАТОРНАЯ РАБОТА №3

По дисциплине: «ОСИСП»

Выполнил:

Студент ФЭИС

3-го курса, группы ПО-5

Белко В.А.

Проверила:

Дряпко А.В.

Брест 2021

**Цель работы:** научиться разрабатывать и использовать динамические библиотеки (DLL) с использованием Qt.

Вариант 2

**Задание:**

Доработать программу, разработанную в лабораторной работе №1-2, внеся следующие изменения:

1) Выбрать 3 вспомогательные функции и вынести их описание и реализацию в динамическую

библиотеку helper.dll. В основном приложении осуществить загрузку реализованных функций

во время работы программы (at run-time, с использованием объекта QLibrary) и их вызов.

2) Выбрать вспомогательный класс и вынести его описание и реализацию в динамическую биб-

лиотеку helper\_class.dll. В основном приложении осуществить загрузку реализованного класса

во время компиляциии..

3) Реализовать окно «О программе» в виде объекта динамической библиотеки about.dll с указанием автора программы, группы, курса и краткого описания разработанного приложения.

Осуществить импорт указанной библиотеки и отображение соответствующего окна при выборе

пункта меню «О программе».

4) Реализовать расширения для приложения, позволяющие изменять оформление пунктов меню (шрифт, размер, начертание и т.д.). Соответствующие изменения должны происходить при

выборе специального пункта меню. Создать как минимум три расширения такого типа.

**mainwindow.h:**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <iostream>

#include <math.h>

#include <string>

#include <QDialog>

#include <QMessageBox>

#include <QLibrary>

#include <QDebug>

#include <QDir>

#include <QPluginLoader>

#include "interface.h"

using namespace std;

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

void **load\_mode\_plugins**();

~***MainWindow***();

private slots:

void **on\_pushButton\_clicked**();

void **on\_pushButton\_2\_clicked**();

void **on\_pushButton\_3\_clicked**();

void **on\_pushButton\_4\_clicked**();

void **on\_pushButton\_5\_clicked**();

void **Two\_Ten**();

void **Eight\_Ten**();

void **Two\_Six**();

void **Eight\_Six**();

void **Ten\_Six**();

void **on\_pushButton\_6\_clicked**();

void **on\_pushButton\_7\_clicked**();

void **applyPlugin**(int index);

private:

Ui::MainWindow \*ui;

int Flag = 0;

int Value = 0;

QVector< Interface\* > mPlugins;

};

#endif // MAINWINDOW\_H

**main.cpp:**

#include "mainwindow.h"

#include <QApplication>

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

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent): QMainWindow(parent), ui(new Ui::MainWindow)

{

typedef QString (\*Helper\_set\_name\_window)();

typedef QString (\*Helper\_set\_name\_about)();

typedef QString (\*Helper\_set\_name\_btn)();

ui->setupUi(this);

load\_mode\_plugins();

// установка имени окна приложения

QLibrary \*helper\_set = new QLibrary("Helper");

Helper\_set\_name\_window set\_window\_name\_ = (Helper\_set\_name\_window)helper\_set->resolve("set\_window\_name");

QString windowTitle = set\_window\_name\_();

delete helper\_set;

this->setWindowTitle(windowTitle);

// установка имени кнопки

QLibrary \*helper\_set\_n\_btn = new QLibrary("Helper");

Helper\_set\_name\_btn set\_btn\_name\_ = (Helper\_set\_name\_btn)helper\_set\_n\_btn->resolve("set\_window\_btn\_name");

ui->pushButton->setText(set\_btn\_name\_());

delete helper\_set\_n\_btn;

// установка имени About

QLibrary \*helper\_set\_n\_about = new QLibrary("Helper");

Helper\_set\_name\_about set\_window\_name\_first = (Helper\_set\_name\_about)helper\_set\_n\_about->resolve("set\_name\_About");

ui->pushButton\_7->setText(set\_window\_name\_first());

delete helper\_set\_n\_about;

}

void MainWindow::load\_mode\_plugins() {

QDir dir(qApp->applicationDirPath());

dir.cd("plugins");

QStringList filter;

filter << "\*.dll";

QPluginLoader loader;

foreach(QFileInfo info, dir.entryInfoList(filter)) {

loader.setFileName(info.absoluteFilePath());

Interface\* mode = qobject\_cast<Interface\*>(loader.instance());

if(mode) {

this->mPlugins.push\_back(mode);

QString name = loader.metaData().value("MetaData").toObject().value("Mode\_name").toString();

ui->comboBox->addItem(name);

connect(ui->comboBox, SIGNAL(activated(int)), this, SLOT(applyPlugin(int)));

}

}

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::Two\_Ten() {

if(Flag==1) {

QString inputNum = ui->lineEdit->text();

QByteArray ba = inputNum.toLocal8Bit();

char \*s= ba.data();

char \*end;

long a = strtol (s, &end, 2);

QString num = QString::number(a);

ui->lineEdit->setText(num);

}

}

void MainWindow::Two\_Six() {

if(Flag==1) {

QString inputNum = ui->lineEdit->text();

QByteArray ba = inputNum.toLocal8Bit();

char \*s= ba.data();

char \*end;

long a = strtol (s, &end, 2);

string digits[16] = {"0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"};

string hex;

do {

hex.insert (0, digits[a % 16]);

a /= 16;

}

while (a!=0);

QString num = QString::fromUtf8((hex).c\_str());

ui->lineEdit->setText(num);

}

}

void MainWindow::Eight\_Six(){

if(Flag==2) {

QString inputNum = ui->lineEdit->text();

string numb = inputNum.toUtf8().constData();

int res = 0;

for (int i = 0; i < (int)numb.size(); ++i) {

res \*= 8;

res += (numb[i] - '0');

}

string digits[16] = {"0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"};

string hex;

do {

hex.insert (0, digits[res % 16]);

res /= 16;

}

while (res!=0);

QString num = QString::fromUtf8((hex).c\_str());

ui->lineEdit->setText(num);

}

}

void MainWindow::Ten\_Six(){

if (Flag==3) {

QString inputNum = ui->lineEdit->text();

QByteArray ba = inputNum.toLocal8Bit();

char \*s= ba.data();

char \*end;

long a = strtol (s, &end, 10);

string digits[16] = {"0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"};

string hex;

do {

hex.insert (0, digits[a % 16]);

a /= 16;

}

while (a!=0);

QString num = QString::fromUtf8((hex).c\_str());

ui->lineEdit->setText(num);

}

}

void MainWindow::Eight\_Ten() {

if(Flag==2) {

QString inputNum = ui->lineEdit->text();

string numb = inputNum.toUtf8().constData();

int res = 0;

for (int i = 0; i < (int)numb.size(); ++i) {

res \*= 8;

res += (numb[i] - '0');

}

QString num = QString::fromUtf8(to\_string(res).c\_str());

ui->lineEdit->setText(num);

}

}

void MainWindow::on\_pushButton\_clicked()

{

Flag = 1;

}

void MainWindow::on\_pushButton\_2\_clicked() {

Flag = 2;

}

void MainWindow::on\_pushButton\_3\_clicked() {

Flag = 3;

}

void MainWindow::on\_pushButton\_4\_clicked() {

Value = 1;

if(Flag==1) {

Two\_Six();

} else if (Flag==2) {

Eight\_Six();

} else if (Flag==3) {

Ten\_Six();

}

}

void MainWindow::on\_pushButton\_5\_clicked() {

Value = 2;

if(Flag==1) {

Two\_Ten();

} else if (Flag==2) {

Eight\_Ten();

}

}

void MainWindow::on\_pushButton\_6\_clicked() {

ui->lineEdit->clear();

}

void MainWindow::on\_pushButton\_7\_clicked() {

typedef void (\*about)();

QLibrary \*aboutLib = new QLibrary("about");

about showWindowAboutProgramm = (about)aboutLib->resolve("about");

showWindowAboutProgramm();

delete aboutLib;

}

void MainWindow::applyPlugin(int index) {

QFont pallete = this->mPlugins[index]->changeView();

QApplication::setFont(pallete);

}

**Mode.h**

#ifndef MODE\_H

#define MODE\_H

#include <QObject>

#include <interface.h>

class Mode : public QObject, public Interface

{

Q\_OBJECT

Q\_PLUGIN\_METADATA(IID "calc.Interface" FILE "Interface.json")

Q\_INTERFACES(Interface)

public:

Mode(QObject \*parent =0);

~Mode();

QString name();

virtual QFont changeView();

};

#endif // MODE\_H

***Mode.cpp***

#include "mode.h"

#include <QDebug>

Mode::Mode(QObject \*parent) : QObject(parent)

{

qDebug() << name() << "created";

}

Mode::~Mode()

{

qDebug() << name() << "destroy";

}

QString Mode::name()

{

return "mode 2";

}

QFont Mode::changeView()

{

QFont palette("Times New Roman", 10, QFont::Bold, true);

return palette;

}

***Interface.h***

#ifndef INTERFACE\_H

#define INTERFACE\_H

#include <QObject>

#include <QString>

#include <QFont>

class Interface

{

public:

virtual QFont changeView() = 0;//плагин выполняет действия над объектами в структуре Styles

};

Q\_DECLARE\_INTERFACE(Interface, "calc.Interface");

#endif // INTERFACE\_H

***Helper.h***

#ifndef HELPER\_H

#define HELPER\_H

#include "helper\_global.h"

#include <QString>

extern "C" HELPER\_EXPORT QString set\_window\_name();

extern "C" HELPER\_EXPORT QString set\_name\_About();

extern "C" HELPER\_EXPORT QString set\_window\_btn\_name();

#endif // HELPER\_H

***Helper.cpp***

#include "helper.h"

QString set\_window\_btn\_name() {

return QString("Двоичная");

}

QString set\_window\_name() {

return QString("CALCULATOR");

}

QString set\_name\_About() {

return QString("О программе");

}

***About.cpp***

#include "about.h"

#include <QMessageBox>

void about()

{

QMessageBox msgBox;

msgBox.setWindowTitle("About programm");

msgBox.setText("Эту программу сделал Белко Владислав");

msgBox.exec();

}

Вывод: научился разрабатывать и использовать динамические библиотеки (DLL) с использованием Qt.