Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Факультет электронно-информационных систем

Кафедра ИИТ

Лабораторная работа №3

за 5 семестр

По дисциплине: «ОСиСП»

Выполнила:

студентка 3 курса

группы ПО-4(1)

Грицак A.В.

Проверила:

Дряпко А.В.

2021

Лабораторная работа №3

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

Вариант 7

Задание:

Доработать программу, разработанную в лабораторной работе No1-2, 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.

**Текст программы:**

1. **helper.h**

#ifndef HELPER\_H

#define HELPER\_H

#include "helper\_global.h"

#include "QString"

extern "C" HELPER\_EXPORT QString **GetStart**();

extern "C" HELPER\_EXPORT QString **Start**();

extern "C" HELPER\_EXPORT QString **Exit**();

#endif // HELPER\_H

1. **helper.cpp**

#include "helper.h"

QString **GetStart**()

{

return QString("&New Game");

}

QString **Start**()

{

return QString("&Start");

}

QString **Exit**()

{

return QString("&Exit");

}

1. **helper\_global.h**

#ifndef HELPER\_GLOBAL\_H

#define HELPER\_GLOBAL\_H

#include <QtCore/qglobal.h>

#if defined(HELPER\_LIBRARY)

# define HELPER\_EXPORT Q\_DECL\_EXPORT

#else

# define HELPER\_EXPORT Q\_DECL\_IMPORT

#endif

#endif // HELPER\_GLOBAL\_H

1. **helper.pro**

QT += widgets

TARGET = Helper

TEMPLATE = lib

DEFINES += HELPER\_LIBRARY

CONFIG += c++11

DEFINES += QUILIB\_LIBRARY

# You can make your code fail to compile if it uses deprecated APIs.

# In order to do so, uncomment the following line.

#DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \

helper.cpp

HEADERS += \

helper\_global.h \

helper.h

# Default rules for deployment.

unix {

target.path = /usr/lib

}

!isEmpty(target.path): INSTALLS += target

**Подключение и использование в приложении:**

#include <../../helper/helper.h>

newg = new QRadioButton(GetStart());

start = new QPushButton (Start());

start->setEnabled(false);

exit = new QPushButton (Exit());

2) Выбрать вспомогательный класс и вынести его описание и реализацию в динамическую библиотеку helper\_class.dll. В основном приложении осуществить загрузку реализованного класса во время компиляции (at compile-time).

**Текст программы:**

1. **mybutton.h**

#ifndef MYBUTTON\_H

#define MYBUTTON\_H

#include "helper\_class\_global.h"

#include <QPushButton>

#include <QMouseEvent>

class HELPER\_CLASS\_EXPORT **MyButton** : public QPushButton

{

Q\_OBJECT

public:

explicit **MyButton**(QWidget \*parent = 0);

~***MyButton***();

signals:

void **rightClicked**();

void **leftClicked**();

void **doubleClicked**();

private slots:

void ***mousePressEvent***(QMouseEvent \*event);

void ***mouseDoubleClickEvent***(QMouseEvent \*event);

};

#endif // MYBUTTON\_H

1. **mybutton.cpp**

#include "mybutton.h"

#include <iostream>

MyButton::**MyButton**(QWidget \*parent) : QPushButton(*parent*)

{

}

MyButton::~***MyButton***()

{

}

void MyButton::***mousePressEvent***(QMouseEvent \*event)

{

if (event->button() == Qt::RightButton)

emit rightClicked();

if (event->button() == Qt::LeftButton)

emit leftClicked();

if (event->button() == Qt::MiddleButton)

emit doubleClicked();

}

void MyButton::***mouseDoubleClickEvent***(QMouseEvent \*event)

{

if (event->button() == Qt::LeftButton)

emit doubleClicked();

}

1. **helper\_global.h**

#ifndef HELPER\_CLASS\_GLOBAL\_H

#define HELPER\_CLASS\_GLOBAL\_H

#include <QtCore/qglobal.h>

#if defined(HELPER\_CLASS\_LIBRARY)

# define HELPER\_CLASS\_EXPORT Q\_DECL\_EXPORT

#else

# define HELPER\_CLASS\_EXPORT Q\_DECL\_IMPORT

#endif

#endif // HELPER\_CLASS\_GLOBAL\_H

1. **helper\_class.pro**

QT += widgets

TEMPLATE = lib

DEFINES += HELPER\_CLASS\_LIBRARY

CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.

# In order to do so, uncomment the following line.

#DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \

mybutton.cpp

HEADERS += \

helper\_class\_global.h \

mybutton.h

# Default rules for deployment.

unix {

target.path = /usr/lib

}

!isEmpty(target.path): INSTALLS += target

**Подключение и использование в приложении:**

#include "../../helper\_class/mybutton.h"

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».

**Текст программы:**

1. **about.h**

#ifndef ABOUT\_H

#define ABOUT\_H

#include "about\_global.h"

extern "C" ABOUT\_EXPORT void **About**();

#endif // ABOUT\_H

1. **about.cpp**

#include "about.h"

#include <QMessageBox>

void **About**()

{

QMessageBox msgBox;

msgBox.setWindowTitle("About programm");

msgBox.setText("The programm was developed by Gritsak Alexandra\nCourse : 3\nGroup: PO-4\n2021\nGood Game:)");

msgBox.*exec*();

}

1. **about\_global.h**

#ifndef ABOUT\_GLOBAL\_H

#define ABOUT\_GLOBAL\_H

#include <QtCore/qglobal.h>

#if defined(ABOUT\_LIBRARY)

# define ABOUT\_EXPORT Q\_DECL\_EXPORT

#else

# define ABOUT\_EXPORT Q\_DECL\_IMPORT

#endif

#endif // ABOUT\_GLOBAL\_H

1. **about.pro**

QT += widgets

TEMPLATE = lib

DEFINES += ABOUT\_LIBRARY

CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.

# In order to do so, uncomment the following line.

#DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \

about.cpp

HEADERS += \

about\_global.h \

about.h

# Default rules for deployment.

unix {

target.path = /usr/lib

}

!isEmpty(target.path): INSTALLS += target

**Подключение и использование в приложении:**

#include "../../about/about.h"

menu\_about = new QMenu("&About");

menu\_about\_action = new QAction("&About", *menu\_about*);

menu\_about->addAction(*menu\_about\_action*);

menu\_about\_action->setShortcut(QKeySequence(Qt::Key\_F4));

menuBar()->addMenu(*menu\_about*);

connect(menu\_about\_action, &QAction::triggered, this, &Game::menu\_about\_game);

void Game::**menu\_about\_game**()

{

About();

firstCheck = false;

}

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.

menu\_style = new QMenu("&Style");

menu\_style\_action = new QAction("&Style",*menu\_style*);

menu\_style->addAction(*menu\_style\_action*);

menu\_style\_action->setShortcut(QKeySequence(Qt::Key\_F6));

menuBar()->addMenu(*menu\_style*);

connect(menu\_style\_action, &QAction::triggered, this, &Game::menu\_style\_game);

void Game::**menu\_style\_game**()

{

bool Changed;

QFont newFont = QFontDialog::getFont(*&Changed*);

if (Changed) {

QApplication::setFont(newFont);

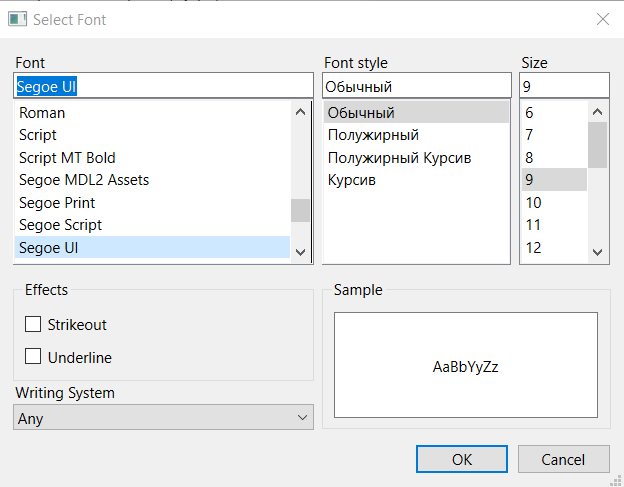
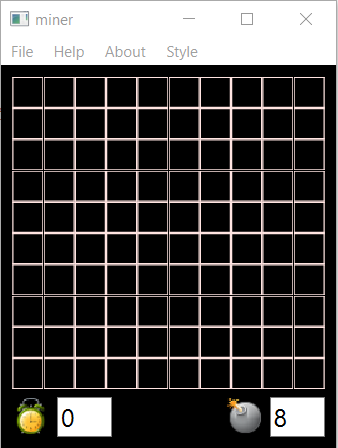
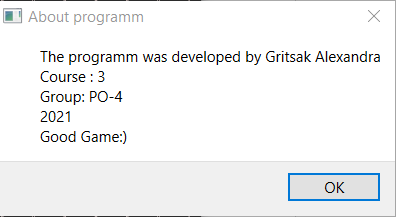
QSettings settings(this);

settings.setValue("VIEWF", newFont);

}

}

**Результаты тестирования программы:**



**Вывод:** Изучила создание динамических библиотек в Qt.