МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

 «БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

ЛАБОРАТОРНАЯ РАБОТА №3

# По дисциплине:

# « Операционные Системы и Системное Программирование»

Выполнил:

Студент ФЭИС

3-го курса, группы ПО-5

Кныш К.О.

Проверила:

Дряпко А.В.

Брест 2021

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

Вариант 18

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.

2) 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».

3) Реализовать расширения для приложения, позволяющие изменять оформление пунктов меню (шрифт, размер, начертание и т.д.). Соответствующие изменения должны происходить приьвыборе специального пункта меню. Создать как минимум три расширения такого типа.

Ход работы:

Код основной программы:

**Sudoku.h**

**…**

#include <QPluginLoader>

#include <fontPluginInterface.h>

typedef void (\*AboutFunction)(QWidget\* parent);

public:

...

void **loadPlugins**();

public slots:

...

void **changeAppFont**(QFont font);

private:

...

AboutFunction showAboutWindow = nullptr;

private slots:

...

void **about**();

};

#endif // SUDOKU\_H

**Main.cpp**

...

#include <QLibrary>

typedef QString (\*SetNameWindow)();

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

{

...

QLibrary \*helper\_set = new QLibrary("D:\\Programming\\Qt\\projects\\build-helper-Desktop\_Qt\_6\_2\_1\_MinGW\_64\_bit-Debug\\debug\\helper.dll");

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

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

delete helper\_set;

w.setWindowTitle(windowTitle);

...

return a.exec();

}

**Sudoku.cpp**

...

ui(new Ui::Sudoku) {

...

QObject::connect(ui->actionAbout, SIGNAL(triggered()), this, SLOT(about()));

…

}

…

void Sudoku::**about**()

{

if (showAboutWindow == nullptr)

{

QLibrary aboutLib("D:/Programming/Qt/projects/build-about-Desktop\_Qt\_6\_2\_1\_MinGW\_64\_bit-Debug/debug/about.dll");

showAboutWindow = (AboutFunction)aboutLib.resolve("showAboutWindow");

}

showAboutWindow(this);

}

...

void Sudoku::**changeAppFont**(QFont font)

{

QApplication::setFont(font);

}

void Sudoku::**loadPlugins**()

{

QDir dir(qApp->applicationDirPath() + "/plugins");

QStringList filter("\*.dll");

QPluginLoader loader;

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

{

loader.setFileName(info.absoluteFilePath());

FontPluginInterface\* fontPlugin = qobject\_cast<FontPluginInterface\*>(*loader.instance()*);

if(fontPlugin)

{

QString name = loader.metaData().value("MetaData").toObject().value("fontName").toString();

QAction\* actionFontPlugin = ui->plugins->addAction(name);

connect(actionFontPlugin, &QAction::triggered, this, [this, fontPlugin]{changeAppFont(fontPlugin->*getFont*());});

}

}

}

Код расширения для шрифтов:

**Fontplugin.h**

#ifndef FONTPLUGIN\_H

#define FONTPLUGIN\_H

#include <fontPluginInterface.h>

class **FontPlugin** : public QObject, public FontPluginInterface

{

Q\_OBJECT

Q\_INTERFACES(**FontPluginInterface**)

Q\_PLUGIN\_METADATA(**IID** "Sudoku.FontPluginInterface" **FILE** "fontPlugin.json")

public:

virtual QFont ***getFont***();

};

#endif // FONTPLUGIN\_H

**Fontplugin.cpp**

#include "fontPlugin.h"

QFont FontPlugin::***getFont***()

{

QFont font("Helvetica", 7, QFont::Medium);

font.setCapitalization(QFont::AllUppercase);

font.setStretch(QFont::Expanded);

return font;

}

**fontPluginInterface.h**

#ifndef FONTPLUGININTERFACE\_H

#define FONTPLUGININTERFACE\_H

#include "pluginInterface\_global.h"

#include <QFont>

class **FontPluginInterface**

{

public:

virtual QFont ***getFont***() = 0;

};

Q\_DECLARE\_INTERFACE(FontPluginInterface, "Sokoban.FontPluginInterface");

#endif // FONTPLUGININTERFACE\_H

**pluginInterface\_global.h**

#ifndef PLUGININTERFACE\_GLOBAL\_H

#define PLUGININTERFACE\_GLOBAL\_H

#include <QtCore/qglobal.h>

#if defined(PLUGININTERFACE\_LIBRARY)

# define PLUGININTERFACE\_EXPORT Q\_DECL\_EXPORT

#else

# define PLUGININTERFACE\_EXPORT Q\_DECL\_IMPORT

#endif

#endif // PLUGININTERFACE\_GLOBAL\_H

Код вспомогательной функции:

**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

**Windowtitle.h**

#ifndef WINDOWTITLE\_H

#define WINDOWTITLE\_H

#include "helper\_global.h"

#include <QString>

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

#endif // WINDOWTITLE\_H

**Windowtitle.cpp**

#include "windowtitle.h"

QString **set\_window\_name**() {

return QString("Sudokuuu");

}

Код динамической библиотеки “о программе”:

**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

**About.h**

#ifndef ABOUT\_H

#define ABOUT\_H

#include "about\_global.h"

#include <QWidget>

extern "C" void **showAboutWindow**(QWidget\* parent);

#endif // ABOUT\_H

**About.cpp**

#include "about.h"

#include <QMessageBox>

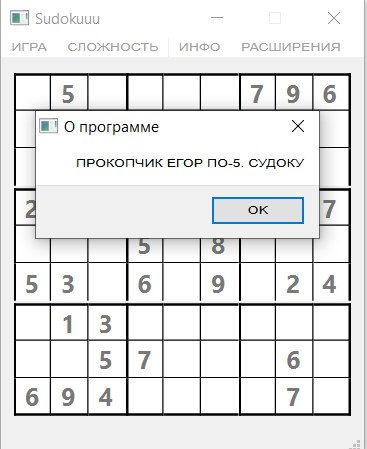
void **showAboutWindow**(QWidget\* parent)

{

QMessageBox::about(*parent*, "О программе", "Прокопчик Егор ПО-5. Судоку");

}

**Результат:**

****

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