Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – “Создание и использование DLL-библиотек”

Отчет по лабораторной работе “Лабораторная работа №4”

по дисциплине Операционные системы

Выполнил

Студент, номер группы ИСМб-19-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Д.Солопов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В.Пестерев

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2021 г.

Содержание

[Постановка задач 3](#_Toc69565281)

[Решение задач 4](#_Toc69565282)

[1 Реализация приложения “Калькулятор” 4](#_Toc69565283)

[2 Статически подключаемая библиотека 16](#_Toc69565284)

[3 Динамически загружаемая библиотека 18](#_Toc69565285)

Постановка задач

Лаб. 4. Создание и использование DLL-библиотек

1. Разработать калькулятор, реализующий простые функции: сложение, вычитание, умножение, деление чисел. Использовать одно поле для ввода и для вывода. Исключить введение в поле каких-либо символов, кроме цифр. Размещение кнопок свободное на усмотрение исполнителя (допускается использование кнопок на форме, в панели инструментов и т.д.). Добавить кнопку «О программе». Нажатие на которую вызывает форму с информацией о исполнителе: ФИО, группа.

2. Функции для вычисления и вызываемая форма должны экспортироваться из **статически** подключаемой dll-библиотеки, назовите ее «stat.dll».

3. Предусмотреть включение «Расширенного режима» (включение на усмотрение исполняемого: кнопкой на форме, кнопкой на панели инструментов и т.д). При его включении добавляются 4 функции: sin, cos, tg, ctg вводимого числа. При выключении «Расширенного режима» они убираются. Функции для этих вычислений должны экспортироваться из **динамически** подключаемой dll-библиотеки, назовите ее «dyn.dll».

4. Оформить отчет по стандартам оформления ИрНИТУ, в отчете приложить теорию по DLL-библиотекам, их использованию и типам подключения. Назначение вашего приложения и описание функций, которые оно выполняет. Исходный код и примеры выполнения функций.

Для разработки приложения и создания библиотек будет использована среда Qt, Visual Studio Community и язык программирования C++.

Решение задач

1 Реализация приложения “Калькулятор”

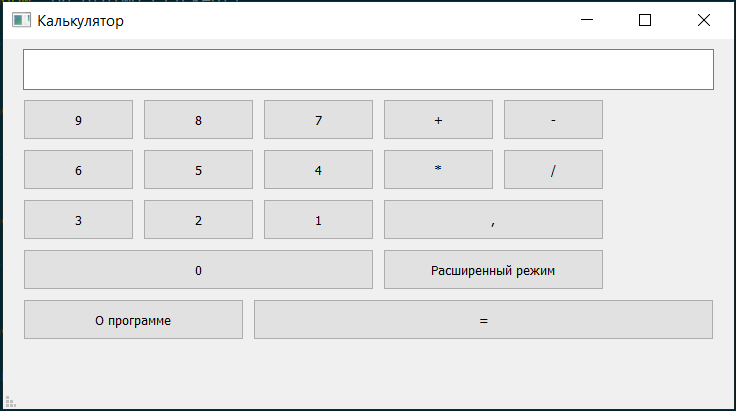


Рисунок 1 – Интерфейс приложения “Калькулятор” без расширенного режима

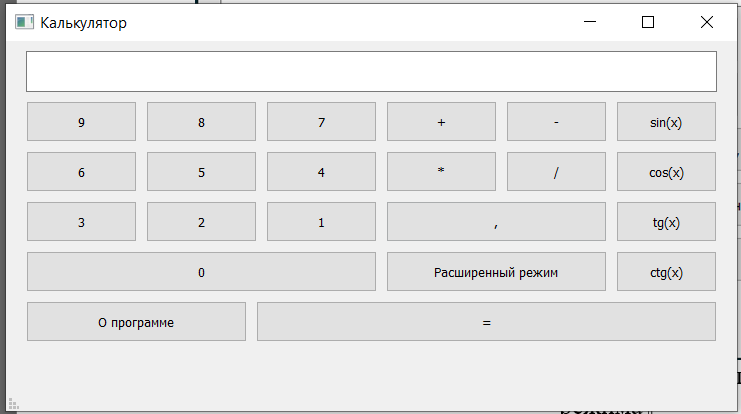


Рисунок 2 – Интерфейс приложения “Калькулятор” с расширенным режимом

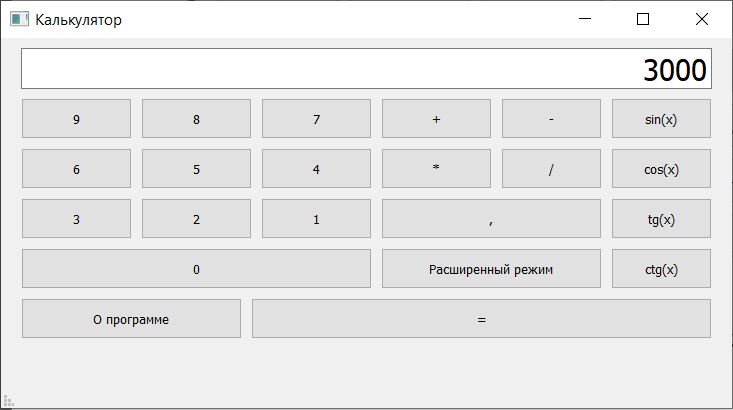


Рисунок 3 – Пример вычислений (результат деления 15000 на 5)

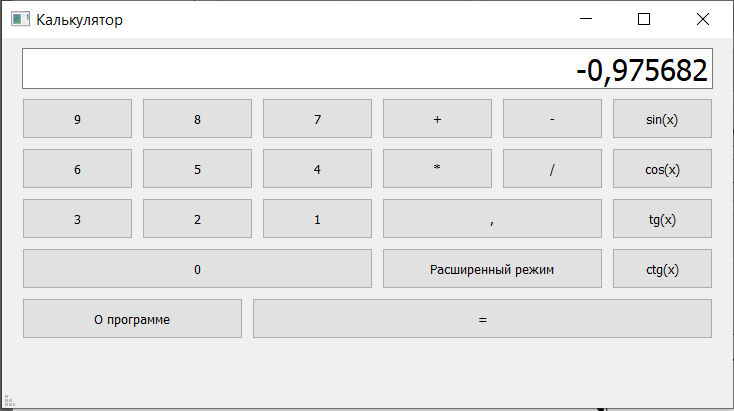


Рисунок 4 – Пример вычисления cos(x), из результатов, полученных на рисунке 3

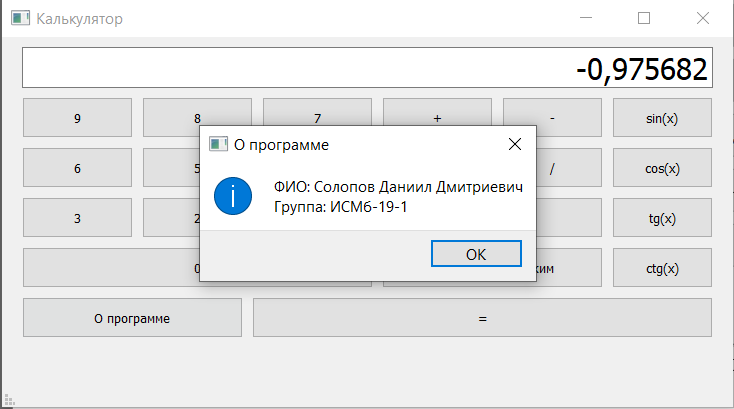


Рисунок 5 – Сообщение после нажатия на кнопку “О программе”

**Код заголовочного файла MainWindow.h:**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <Windows.h>

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private slots:

void on\_btnNight\_clicked();

void on\_btnEight\_clicked();

void on\_btnSeven\_clicked();

void on\_btnSix\_clicked();

void on\_btnFive\_clicked();

void on\_btnFour\_clicked();

void on\_btnThree\_clicked();

void on\_btnTwo\_clicked();

void on\_btnOne\_clicked();

void on\_btnZero\_clicked();

void on\_btnUpgrade\_clicked();

void on\_btnAboutProgram\_clicked();

void on\_btnEqual\_clicked();

void on\_btnSin\_clicked();

void on\_btnCos\_clicked();

void on\_btnTg\_clicked();

void on\_btnCtg\_clicked();

void on\_btnPoint\_clicked();

void on\_btnSum\_clicked();

void on\_btnDifference\_clicked();

void on\_btnMultiplication\_clicked();

void on\_btnDivision\_clicked();

private:

Ui::MainWindow \*ui;

bool \_upgrade; //расширенный режим

HMODULE \_hDLL; //динамически загружаемая библиотека

char \_currentOperation; //текущая операция

QString oneValue; //первое число

private:

double (\*calculateSinT)(double);

double (\*calculateCosT)(double);

double (\*calculateTgT)(double);

double (\*calculateCtgT)(double);

void calculateFunction();

};

#endif // MAINWINDOW\_H

**Код файла MainWindow.cpp:**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QDoubleValidator>

#include <QMessageBox>

#include <StaticLib1/StaticCalc.h>

typedef double (\*FUNC)(double);

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

ui->lineEdit->setStyleSheet("font-size: 30px;");

ui->lineEdit->setAlignment(Qt::AlignRight);

ui->btnSin->hide();

ui->btnCtg->hide();

ui->btnTg->hide();

ui->btnCos->hide();

this->\_upgrade = true;

ui->lineEdit->setValidator( new QDoubleValidator(0, 100, 20, this));

\_hDLL = LoadLibraryA("Dll1.dll");

if(\_hDLL != NULL){

this->calculateSinT = (FUNC)GetProcAddress(\_hDLL, "CalculateSin");

this->calculateCosT = (FUNC)GetProcAddress(\_hDLL, "CalculateCos");

this->calculateTgT = (FUNC)GetProcAddress(\_hDLL, "CalculateTg");

this->calculateCtgT = (FUNC)GetProcAddress(\_hDLL, "CalculateCtg");

}

\_currentOperation = ' ';

}

MainWindow::~MainWindow()

{

if(\_hDLL != NULL){

FreeLibrary(\_hDLL);

}

delete ui;

}

void MainWindow::on\_btnNight\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "9");

}

void MainWindow::on\_btnEight\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "8");

}

void MainWindow::on\_btnSeven\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "7");

}

void MainWindow::on\_btnSix\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "6");

}

void MainWindow::on\_btnFive\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "5");

}

void MainWindow::on\_btnFour\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "4");

}

void MainWindow::on\_btnThree\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "3");

}

void MainWindow::on\_btnTwo\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "2");

}

void MainWindow::on\_btnOne\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "1");

}

void MainWindow::on\_btnZero\_clicked()

{

ui->lineEdit->setText(ui->lineEdit->text() + "0");

}

void MainWindow::on\_btnUpgrade\_clicked()

{

this->\_upgrade = !this->\_upgrade;

if(this->\_upgrade){

ui->btnSin->hide();

ui->btnCtg->hide();

ui->btnTg->hide();

ui->btnCos->hide();

}else{

ui->btnSin->show();

ui->btnCtg->show();

ui->btnTg->show();

ui->btnCos->show();

}

}

void MainWindow::on\_btnAboutProgram\_clicked()

{

(new QMessageBox("О программе",

"ФИО: Солопов Даниил Дмитриевич\nГруппа: ИСМб-19-1",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

}

void MainWindow::on\_btnEqual\_clicked()

{

calculateFunction();

}

void MainWindow::on\_btnSin\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Нет входных данных для вычисления!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

try{

QString data = ui->lineEdit->text().replace(",", ".");

double value = this->calculateSinT(data.toDouble());

data = QString::number(value).replace(".", ",");

ui->lineEdit->setText(data);

}catch(...){

(new QMessageBox("Ошибка!", "Не удалось выполнить операцию!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

}

void MainWindow::on\_btnCos\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Нет входных данных для вычисления!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

try{

QString data = ui->lineEdit->text().replace(",", ".");

double value = this->calculateCosT(data.toDouble());

data = QString::number(value).replace(".", ",");

ui->lineEdit->setText(data);

}catch(...){

(new QMessageBox("Ошибка!", "Не удалось выполнить операцию!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

}

void MainWindow::on\_btnTg\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Нет входных данных для вычисления!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

try{

QString data = ui->lineEdit->text().replace(",", ".");

double value = this->calculateTgT(data.toDouble());

data = QString::number(value).replace(".", ",");

ui->lineEdit->setText(data);

}catch(...){

(new QMessageBox("Ошибка!", "Не удалось выполнить операцию!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

}

void MainWindow::on\_btnCtg\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Нет входных данных для вычисления!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

try{

QString data = ui->lineEdit->text().replace(",", ".");

double value = this->calculateCtgT(data.toDouble());

data = QString::number(value).replace(".", ",");

ui->lineEdit->setText(data);

}catch(...){

(new QMessageBox("Ошибка!", "Не удалось выполнить операцию!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

}

void MainWindow::on\_btnPoint\_clicked()

{

if(ui->lineEdit->text().indexOf(",") >= 0){

(new QMessageBox("Ошибка!", "Запятая в записи числа уже присутствует!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

ui->lineEdit->setText(ui->lineEdit->text() + ",");

}

void MainWindow::calculateFunction(){

if(oneValue.size() == 0){

ui->lineEdit->setText("0");

return;

}

QString data2 = ui->lineEdit->text().replace(",", ".");

double value2 = data2.toDouble();

QString data1 = oneValue.replace(",", ".");

double value1 = data1.toDouble();

if(\_currentOperation == '+'){

value1 = MathLibrary::Arithmetic::Add(value1, value2);

}else if(\_currentOperation == '-'){

value1 = MathLibrary::Arithmetic::Subtract(value1, value2);

}else if(\_currentOperation == '\*'){

value1 = MathLibrary::Arithmetic::Multiply(value1, value2);

}else if(\_currentOperation == '/'){

value1 = MathLibrary::Arithmetic::Divide(value1, value2);

}

data1 = QString::number(value1).replace(".", ",");

ui->lineEdit->setText(data1);

\_currentOperation = ' ';

}

void MainWindow::on\_btnSum\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Не введено число, для арифметической операции!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

if(\_currentOperation != ' '){

calculateFunction();

}

oneValue = ui->lineEdit->text();

\_currentOperation = '+';

ui->lineEdit->setText("");

}

void MainWindow::on\_btnDifference\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Не введено число, для арифметической операции!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

if(\_currentOperation != ' '){

calculateFunction();

}

oneValue = ui->lineEdit->text();

\_currentOperation = '-';

ui->lineEdit->setText("");

}

void MainWindow::on\_btnMultiplication\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Не введено число, для арифметической операции!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

if(\_currentOperation != ' '){

calculateFunction();

}

oneValue = ui->lineEdit->text();

\_currentOperation = '\*';

ui->lineEdit->setText("");

}

void MainWindow::on\_btnDivision\_clicked()

{

if(ui->lineEdit->text().size() == 0){

(new QMessageBox("Ошибка!", "Не введено число, для арифметической операции!",

QMessageBox::Information,

QMessageBox::NoButton,

QMessageBox::NoButton,

QMessageBox::Ok| QMessageBox::Escape))->exec();

return;

}

if(\_currentOperation != ' '){

calculateFunction();

}

oneValue = ui->lineEdit->text();

\_currentOperation = '/';

ui->lineEdit->setText("");

}

2 Статически подключаемая библиотека

**Статические библиотеки** — это набор уже скомпилированных подпрограмм или объектов, которые подключаются к исходной программе в виде объектных файлов. Этот набор выглядит как архив из нескольких объектных файлов, который умеет распаковывать gcc.

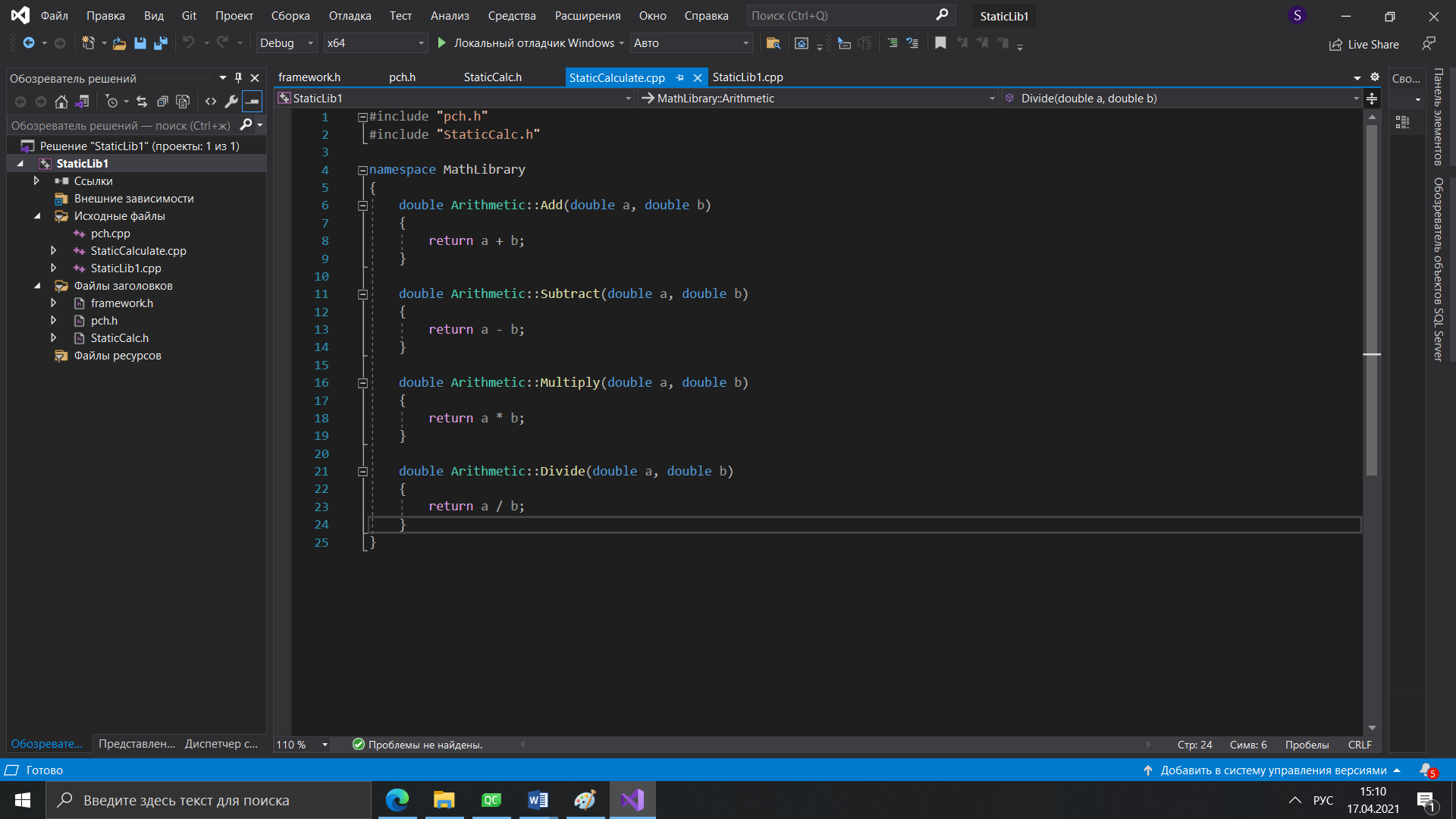


Рисунок 6 – Реализация статической библиотеки

**Код файла StaticCalc.h статической библиотеки:**

#pragma once

namespace MathLibrary

{

class Arithmetic

{

public:

// Returns a + b

static double Add(double a, double b);

// Returns a - b

static double Subtract(double a, double b);

// Returns a \* b

static double Multiply(double a, double b);

// Returns a / b

static double Divide(double a, double b);

};

}

**Код файла StaticCalculate.cpp статической библиотеки:**

#include "pch.h"

#include "StaticCalc.h"

namespace MathLibrary

{

double Arithmetic::Add(double a, double b)

{

return a + b;

}

double Arithmetic::Subtract(double a, double b)

{

return a - b;

}

double Arithmetic::Multiply(double a, double b)

{

return a \* b;

}

double Arithmetic::Divide(double a, double b)

{

return a / b;

}

}

Данная статическая библиотека подключалась через среду разработки Qt, при подключении использовался файл собранный средой разработки Visual Studio Community с расширением \*.lib.

**Код подключения статической библиотеки в \*.pro файле проекта Qt:**

win32:CONFIG(release, debug|release): LIBS += -L$$PWD/StaticLib1/x64/release/ -lStaticLib1

else:win32:CONFIG(debug, debug|release): LIBS += -L$$PWD/StaticLib1/x64/debug/ -lStaticLib1

INCLUDEPATH += $$PWD/StaticLib1

DEPENDPATH += $$PWD/StaticLib1

win32-g++:CONFIG(release, debug|release): PRE\_TARGETDEPS += $$PWD/StaticLib1/x64/release/libStaticLib1.a

else:win32-g++:CONFIG(debug, debug|release): PRE\_TARGETDEPS += $$PWD/StaticLib1/x64/debug/libStaticLib1.a

else:win32:!win32-g++:CONFIG(release, debug|release): PRE\_TARGETDEPS += $$PWD/StaticLib1/x64/release/StaticLib1.lib

else:win32:!win32-g++:CONFIG(debug, debug|release): PRE\_TARGETDEPS += $$PWD/StaticLib1/x64/debug/StaticLib1.lib

3 Динамически загружаемая библиотека

DLL - это сокращение от Dynamic Link Library (динамически загружаемая библиотека).

С формальной точки зрения DLL - особым образом оформленный относительно независимый блок исполняемого кода.

Особый способ оформления предполагает наличие в DLL так называемых секций импорта и экспорта. Секция экспорта указывает те идентификаторы объектов (функций, классов, переменных), доступ к которым предоставляет данная DLL. В этом случае мы говорим об экспортировании идентификаторов из DLL. В общем случае, именно секция экспорта предоставляет особый интерес для разработчиков.

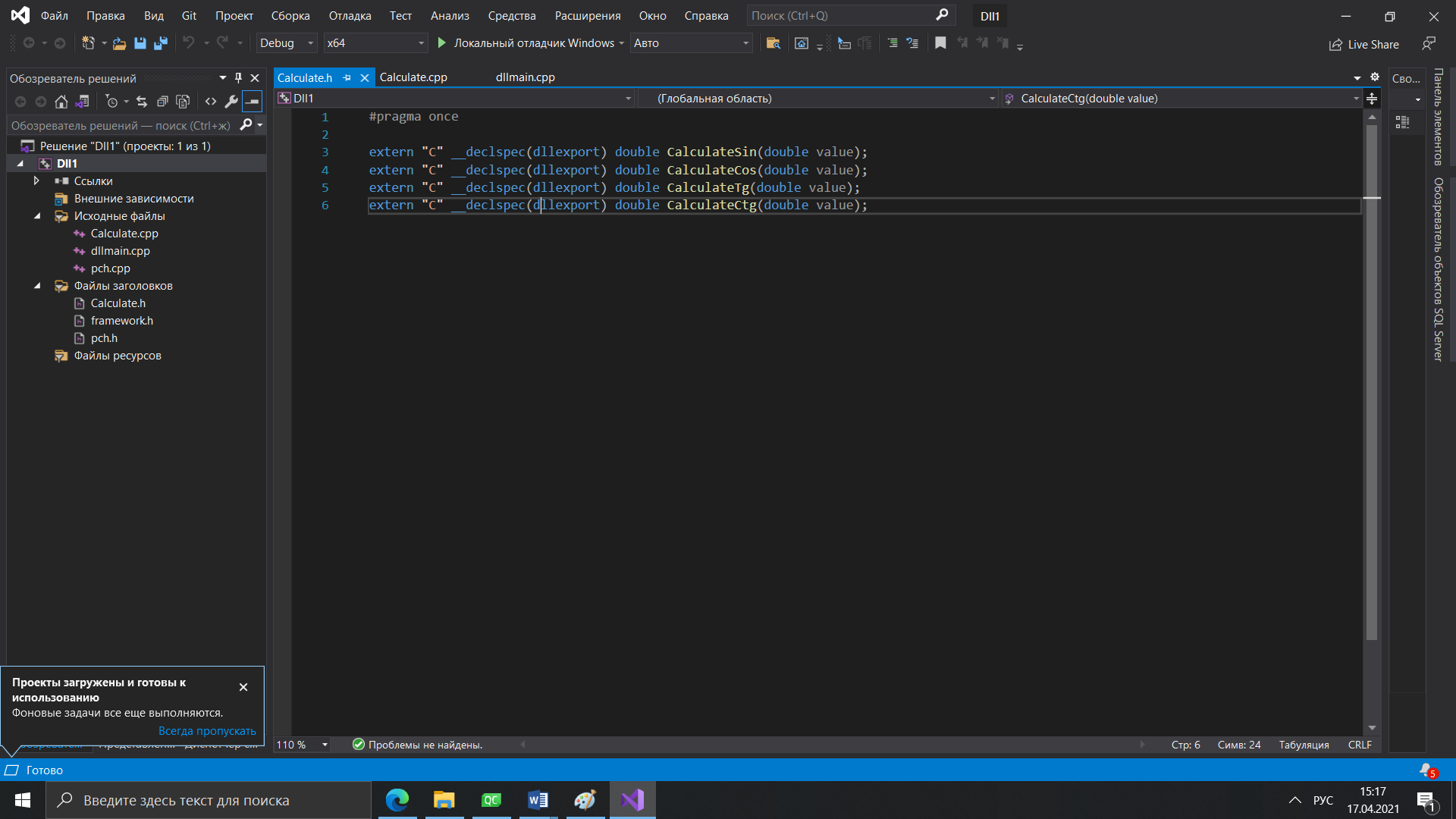


Рисунок 7 – Реализация динамически загружаемой библиотеки

**Код файла Calculate.h:**

#pragma once

extern "C" \_\_declspec(dllexport) double CalculateSin(double value);

extern "C" \_\_declspec(dllexport) double CalculateCos(double value);

extern "C" \_\_declspec(dllexport) double CalculateTg(double value);

extern "C" \_\_declspec(dllexport) double CalculateCtg(double value);

**Код файла Calculate.cpp:**

#include "pch.h"

#include <math.h>

#include "Calculate.h"

double CalculateSin(double value) {

return sin(value);

}

double CalculateCos(double value) {

return cos(value);

}

double CalculateTg(double value) {

return tan(value);

}

double CalculateCtg(double value) {

return (1 / tan(value));

}

Подключается данная библиотека явным образом в проекте Qt, без использования инструментария среды разработки, но с использованием библиотеки Windows.h:

\_hDLL = LoadLibraryA("Dll1.dll");

if(\_hDLL != NULL){

this->calculateSinT = (FUNC)GetProcAddress(\_hDLL, "CalculateSin");

this->calculateCosT = (FUNC)GetProcAddress(\_hDLL, "CalculateCos");

this->calculateTgT = (FUNC)GetProcAddress(\_hDLL, "CalculateTg");

this->calculateCtgT = (FUNC)GetProcAddress(\_hDLL, "CalculateCtg");

}