**Лабораторная работа 1**

**Создание и использование статической библиотеки классов**

Оглавление

[1. Создание и использование демонстрационной статической библиотеки в соответствии с пошаговым руководством (проекты статической библиотеки и приложения находятся в одном решении). 1](#_Toc95081180)

[1.1. Создание проекта статической библиотеки 2](#_Toc95081181)

[1.2. Добавление класса в статическую библиотеку 3](#_Toc95081182)

[1.3. Создание приложения консоли C++, ссылающегося на статическую библиотеку 4](#_Toc95081183)

[1.4. Использование функциональных возможностей статической библиотеки в приложении 5](#_Toc95081184)

[1.5. Запуск приложения 6](#_Toc95081185)

[2. Использование демонстрационной библиотеки в отдельном приложении. 6](#_Toc95081186)

[2.1. Создание проекта MyExecRefsLib\_1 6](#_Toc95081187)

[2.2. Подключение к проекту MyExecRefsLib\_1 демонстрационной библиотеки 6](#_Toc95081188)

[2.3. Выполнение программы 6](#_Toc95081189)

[3. Внесение изменений в демонстрационную статическую библиотеку 7](#_Toc95081190)

[4. Использование изменённой библиотеки в отдельном приложении. 7](#_Toc95081191)

[Контрольные вопросы и задания 7](#_Toc95081192)

[Рекомендуемые источники информации 7](#_Toc95081193)

**Цель лабораторной работы** - приобретение студентами навыков выполнения в MS Visual Studio процессов создания и сопровождения статических библиотек и их использования в программах на C++.

Задачи, решаемые при выполнении:

* *Создание и использование* демонстрационной статической библиотеки в соответствии с пошаговым руководством (проекты статической библиотеки и приложения создаются в одном решении).
* *Использование* демонстрационной библиотеки в отдельном приложении.
* *Внесение изменений* в демонстрационную статическую библиотеку.
* *Использование* изменённой библиотеки в отдельном приложении.

# 1. Создание и использование демонстрационной статической библиотеки в соответствии с пошаговым руководством (проекты статической библиотеки и приложения находятся в одном решении).

Создаваемое решение содержит исходные коды включаемых в библиотеку классов и в дальнейшем может использоваться как «мастерская» для расширения библиотеки и внесения в неё изменений. Решение включает в себя два проекта: первый будем использовать для создания статической библиотеки (тип проекта – консольное приложение, статическая библиотека), а второй будем использовать для тестирования созданной библиотеки (тип проекта – консольное приложение, пустой проект).  
В пошаговом руководстве по созданию статической библиотеки описывается создание в MS Visual Studio 2012 статической библиотеки (файл .lib) для использования с приложениями C++.

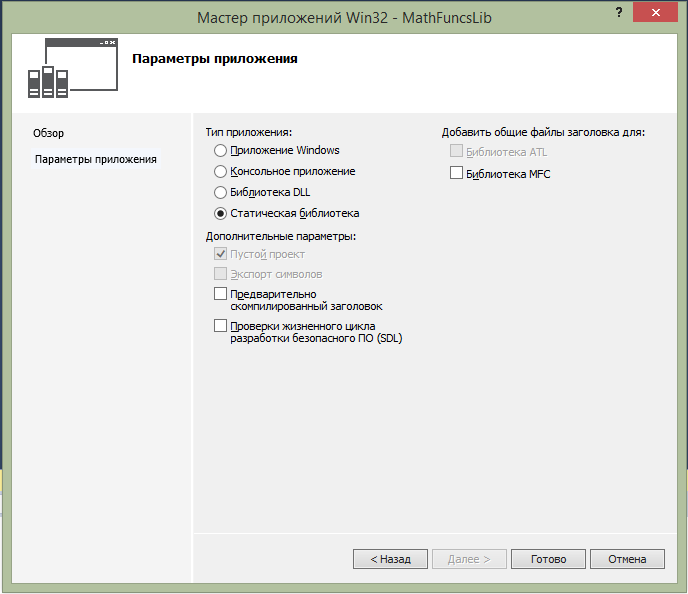
 Статические библиотеки являются хорошим способом повторного использования кода. Вместо повторного включения одних и тех же процедур в каждое приложение, требуемые функции объединяются в статическую библиотеку (файл .lib) и затем выполняется ссылка на этот файл из приложений. Код, связанный со статической библиотекой, становится частью вашего приложения и вам не нужно создавать другой файл для использования кода функций.

В пошаговом руководстве (оно приведено в [msdn.microsoft.com/ru](https://msdn.microsoft.com/ru)) рассматриваются следующие задачи.

* [Создание проекта статической библиотеки](https://msdn.microsoft.com/ru-ru/library/ms235627(v=vs.110).aspx#createlibproject)
* [Добавление класса в статическую библиотеку](https://msdn.microsoft.com/ru-ru/library/ms235627(v=vs.110).aspx#addclasstolib)
* [Создание приложения консоли C++, ссылающегося на статическую библиотеку](https://msdn.microsoft.com/ru-ru/library/ms235627(v=vs.110).aspx#createapptorefthelib)
* [Использование функциональных возможностей статической библиотеки в приложении](https://msdn.microsoft.com/ru-ru/library/ms235627(v=vs.110).aspx#uselibinapp)
* [Запуск приложения](https://msdn.microsoft.com/ru-ru/library/ms235627(v=vs.110).aspx#runapp)

### 1.1. Создание проекта статической библиотеки

1. В строке меню выберите **Файл**, **Создать**, **Проект**.
2. В левой области диалогового окна **Создать проект** разверните **Установленные**, **Шаблоны**,**Visual C++** и затем выберите Win32.
3. В центральной области, выберите **Консольное приложение Win32**.
4. Укажите имя для проекта, например, *MathFuncsLib* - в поле **Имя**.   
   Укажите имя для решения, например, *StaticLibrary* - в поле **Имя решения**.  
   В поле **Расположение** установите имя каталога для решения  
   Установите флажок **Создать каталог для решения**.   
   Нажмите кнопку **ОК**.



1. На странице **Обзор** диалогового окна **Мастер приложений Win32**, нажмите кнопку **Далее**.
2. На странице **Параметры приложения** в поле **Дополнительные параметры** установите флажок **Пустой проект** и снимите флажок **Предварительно скомпилированный заголовок**.
3. На странице **Параметры приложения** в поле **Тип приложения**, выберите **Статическая библиотека.**
4. Нажмите кнопку **Готово**, чтобы создать проект.

### 1.2. Добавление класса в статическую библиотеку

1. Для создания файла заголовка для добавляемых в библиотеку классов откройте в **Обозревателе решений** контекстное меню для проекта MathFuncsLib, а затем в поле **Добавить** выберите **Создать элемент**. В диалоговом окне **Добавление нового элемента** в левой панели выберите **Visual C++**, **Код**.   
   В центральной области выберите **Заголовочный файл (.h)**.  
   Укажите имя для заголовочного файла, например, MathFuncsLib.h, а затем нажмите кнопку **Добавить**. В окне проекта будет показан пустой заголовочный файл.
2. Добавьте в файл MathFuncsLib.h класс с именем MyMathFuncs, включающий общедоступные статические функции Add и Subtract.   
   Класс будет использоваться только для указания имени пространства имен.  
   Использование спецификатора static позволит вызывать из приложений функции, не создавая объект класса MyMathFuncs. Код должен выглядеть следующим образом:

// MathFuncsLib.h

namespace MathFuncs

{

class MyMathFuncs

{

public:

// Returns a + b

static double Add(double a, double b);

// Returns a - b

static double Subtract(double a, double b);

};

}

Чтобы создать файл источника (.cpp) для нового класса откройте в **Обозревателе решений** контекстное меню для проекта MathFuncsLib, а затем в поле **Добавить** выберите **Создать элемент**. В диалоговом окне **Добавление нового элемента** в левой панели выберите **Visual C++**, **Код**. В центральной области, выберите   
**Файл C++ (.cpp)**. Укажите имя для источника, например, MathFuncsLib.cpp, а затем нажмите кнопку **Добавить**. В окне проекта будет показан пустой файл источника. Для реализации функциональности для MyMathFuncs код должен выглядеть следующим образом:

// MathFuncsLib.cpp

#include "MathFuncsLib.h"

using namespace std;

namespace MathFuncs

{

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

{

return a + b;

}

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

{

return a - b;

}

}

Компилировать статическую библиотеку, выбрав в строке меню **Построение**, **Построить решение**. В результате будет создана статическая библиотека, которая может использоваться другими программами

**Примечание**

### **1.3. Создание приложения консоли C++, ссылающегося на статическую биб**лиотеку

1. В строке меню выберите **Файл**, **Создать**, **Проект**.
2. В левой панели выберите **Visual C++**, выберите **Win32**.
3. В центральной области, выберите **Консольное приложение Win32**.
4. Укажите в поле **Имя** имя проекта, например, MyExecRefsLib.  В раскрывающемся списке **Решение**, выберите **Добавить в решение**.

Нажмите кнопку **ОК**.

После этого новый проект будет добавлен в решение, содержащее статическую библиотеку.

1. На странице **Обзор** диалогового окна **Мастер приложений Win32**, нажмите кнопку **Далее**.
2. На странице **Параметры приложения** в поле **Тип приложения**, выберите **Консольное приложение**.
3. На странице **Параметры приложения** в поле **Дополнительные параметры**, снимите флажок **предкомпилированный заголовок** и установите флажок **Пустой проект**.
4. Нажмите кнопку **Готово**, чтобы создать проект.

### 1.4. Использование функциональных возможностей статической библиотеки в приложении

1. Откройте в **Обозревателе решений** контекстное меню для проекта MyExecRefsLib и добавьте в него пустое приложения консоли с именем MyExecRefsLib.cpp.
2. Для использования в консольном приложении MyExecRefsLib.cpp математических функций из статической библиотеки, необходимо связать это приложение с библиотекой MathFuncsLib, добавив в него ссылки на файлы .lib и .h библиотеки.
3. Для ссылки на файл MathFuncsLib.lib откройте в **Обозревателе решений** контекстное меню для проекта MyExecRefsLib, а затем пункт **Ссылки**. В диалоговом окне **Страницы свойств** MyExecRefsLib разверните узел **Общие свойства**, выберите **.NET Framework и ссылки** и затем нажмите кнопку **Добавить новую ссылку**.
4. В диалоговом окне **Добавить ссылку** перечислены библиотеки, на которые можно ссылаться. На вкладке **Проекты** перечисляются проекты текущего решения и все библиотеки, что они содержат. На вкладке **Проекты** установите флажок MathFuncsLib, а затем нажмите кнопку **ОК**.
5. Для ссылки на файл заголовка MathFuncsLib.h необходимо изменить путь к включаемым файлам на путь к каталогу с файлом MathFuncsLib.h.  Для этого в диалоговом окне **Страницы свойств** для MyExecRefsLib разверните узел **Свойства конфигурации**, разверните узел **C/C++**, а затем выберите **Общие**. Рядом с окном **Дополнительные каталоги включаемых файлов** откройте раскрывающийся список значений свойств, а затем выберите **Изменить**. В появившемся диалоговом окне **Дополнительные каталоги включаемых файлов** в пустом текстовом поле выполните двойной щелчок в верхней строке и затем нажмите кнопку с многоточием (**…**) в конце незаполненной полосы. В диалоговом окне **Выбор каталога** выберите каталог MyExecRefsLib, в котором находится файл MathFuncsLib.h (!!! имя каталога должно завершаться символом ‘\’ ). Для сохранения выбора и закрытия диалогового окна нажмите кнопку **Выбор папки**.  В диалоговом окне **Дополнительные каталоги включаемых файлов** нажмите кнопку **ОК.**
6. Для сохранения в проекте изменений, связанных с созданием ссылок на статическую библиотеку, в диалоговом окне **Страницы свойств**, нажмите кнопку **ОК.**
7. Теперь можно использовать функции, определенные в классе MyMathFuncs библиотеки MathFuncsLib, в созданном приложении. Для этого замените содержимое файла MyExecRefsLib.cpp следующим кодом:

// MyExecRefsLib.cpp

#include <iostream>

#include "MathFuncsLib.h"

using namespace std;

int main()

{

double a = 7.4;

int b = 99;

cout << "a + b = " <<

MathFuncs::MyMathFuncs::Add(a, b) << endl;

cout << "a - b = " <<

MathFuncs::MyMathFuncs::Subtract(a, b) << endl;

system(“pause”);

return 0;

}

1. Создайте исполняемый файл, выбрав **Построение**, **Построить решение** в строке меню.

### 1.5. Запуск приложения

1. Чтобы запустить проект MyExecRefsLib, его нужно назначить проектом по умолчанию.  
   Для этого в **Обозревателе решений** откройте контекстное меню для проекта MyExecRefsLib, а затем выберите **Назначить запускаемым проектом**.
2. Результат выполнения должен выглядеть так:

a + b = 106,4   
a - b = -91,6

## **2. Использование демонстрационной библиотеки в отдельном приложении**.

### 2.1. Создание проекта MyExecRefsLib\_1

Создайте проект MyExecRefsLib\_1 и добавьте в него файл MyExecRefsLib.cpp из проекта MyExecRefsLib.

### 2.2. Подключение к проекту MyExecRefsLib\_1 демонстрационной библиотеки

Для подключения к проекту библиотеки MathFuncsLib выполните следующие действия:  
- скопируйте в ту папку проекта MyExecRefsLib\_1, в которой находится файл MyExecRefsLib.cpp, файлы MathFuncsLib.h и MathFuncsLib.lib из проекта MathFuncsLib решения *StaticLibrary* и добавьте их к проекту MyExecRefsLib\_1как существующие;

Копирование библиотечных файлов в папку проекта (а не использование ссылок на папки библиотеки) сделает программу перемещаемой и поможет избежать ошибок, связанных с поиском файлов, при компиляции программы. Однако при изменении библиотеки потребуется замена старых файлов на новые.

### 2.3. Выполнение программы

Выполните программу MyExecRefsLib.cpp в созданном проекте.

## 3. Внесение изменений в демонстрационную статическую библиотеку

Добавьте в библиотеку функции Multiply и Divide для выполнения операций умножения и деления чисел типа double. Для сопровождения библиотеки (внесения изменений и тестирования) используйте решение StaticLibrary.

## **4. Использование изменённой библиотеки в отдельном приложении**.

Выполните в проекте MyExecRefsLib\_1 с измененной библиотекой приложение MyExecRefsLib1.cpp, в котором используются функции для выполнения сложения, вычитания, умножения и деления. Прежде чем делать подумайте, как это нужно сделать.

# Контрольные вопросы и задания

1. Как создать решение для создания и сопровождения статической библиотеки C++?   
2. Как подключить статическую библиотеку к консольному приложению?   
3. Как вызвать для выполнения в приложении метод класса из статической библиотеки?

## Рекомендуемые источники информации

1. Р. Лафоре. Объектно-ориентированное программирование в С++. Издательство ПИТЕР, 2004 г. – 532с.
2. Г. Шилдт “Полный справочник по С++”: Пер. с англ.- М., Издательский дом “Вильямс”, 2006 г. – 800с
3. MSDN Library for Visual Studio 2012 (<http://msdn.microsoft.com/ru-ru/library/> , раздел: Справочник по С++.)
4. Эккель Б. Философия С++. Введение в стандартный С++. 2-е изд.- СПб.: Питер, 2004.- 572с.: ил.