

## Пошаговое руководство. Создание и использование статической библиотеки

В этом пошаговом руководстве описывается создание статической библиотеки (LIB-файла) для использования с приложениями C++. Статические библиотеки являются хорошим способом повторного использования кода. Вместо того чтобы каждый раз реализовывать одни и те же подпрограммы для обеспечения той или иной функциональности в каждом создаваемом приложении, их можно создать единожды и затем вызывать из приложений. Код, подключенный из статической библиотеки, становится частью вашего приложения — для использования кода не нужно устанавливать еще какой-либо файл.

В этом пошаговом руководстве рассматриваются следующие задачи:

- [Создание проекта статической библиотеки](#)
- [Добавление класса в статическую библиотеку](#)
- [Создание консольного приложения C++, ссылающегося на статическую библиотеку](#)
- [Использование функциональности из статической библиотеки в приложении](#)
- [Запуск приложения](#)

### Предварительные требования

Для работы необходимо владеть основами языка C++.

### Создание проекта статической библиотеки

Инструкции по созданию проекта зависят от используемой версии Visual Studio. Чтобы ознакомиться с документацией по предпочтительной версии Visual Studio, используйте селектор **Версия**. Он находится в верхней части оглавления на этой странице.

### Создание проекта статической библиотеки в Visual Studio 2019

1. В строке меню выберите **Файл > Создать > Проект**, чтобы открыть диалоговое окно **Создание проекта**.
2. В верхней части диалогового окна для параметра **Язык** установите значение **C++**, для параметра **Платформа** — значение **Windows**, а для параметра **Тип проекта** — значение **Библиотека**.
3. В отфильтрованном списке типов проектов выберите пункт **Мастер классических приложений Windows**, а затем нажмите кнопку **Далее**.
4. На странице **Настроить новый проект** введите *MathLibrary* в поле **Имя проекта**. В поле **Имя решения** введите *StaticMath*. Нажмите

кнопку **Создать**, чтобы открыть диалоговое окно **Проект классического приложения Windows**.

5. В диалоговом окне **Проект классического приложения Windows** в разделе **Тип приложения** выберите **Статическая библиотека (.lib)**.
6. В разделе **Дополнительные параметры** снимите флажок **Предварительно откомпилированный заголовок**, если он установлен. Установите флажок **Пустой проект**.
7. Нажмите кнопку **ОК**, чтобы создать проект.

### Добавление класса в статическую библиотеку

### Добавление класса в статическую библиотеку

1. Чтобы создать файл заголовка для нового класса откройте контекстное меню проекта **MathLibrary** в обозревателе решений, а затем выберите **Добавить > Новый элемент**.
2. В диалоговом окне **Добавление нового элемента** выберите пункт **Visual C++ > Код**. В центральной области выберите **Заголовочный файл (.h)**. Укажите имя для файла заголовка, например *MathLibrary.h*, и нажмите кнопку **Добавить**. Отобразится почти пустой файл заголовка.
3. Добавьте объявление класса с именем **Arithmetic** для выполнения обычных арифметических операций, таких как сложение, вычитание, умножение и деление. Код должен выглядеть примерно так:

C++Копировать

```
// MathLibrary.h
```

```
#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);
```

```
};
```

```
}
```

4. Чтобы создать исходный файл для нового класса, откройте контекстное меню проекта **MathLibrary** в обозревателе решений, а затем выберите **Добавить > Новый элемент**.
5. В диалоговом окне **Добавление нового элемента** в центральной области выберите **Файл C++ (.cpp)**. Укажите имя исходного файла, например *MathLibrary.cpp*, и нажмите кнопку **Добавить**. Отобразится пустой исходный файл.
6. Используйте этот исходный файл для реализации функций класса Arithmetic. Код должен выглядеть примерно так:

C++Копировать

```
// MathLibrary.cpp
// compile with: cl /c /EHsc MathLibrary.cpp
// post-build command: lib MathLibrary.obj

#include "MathLibrary.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;
    }
}
```

7. Чтобы выполнить сборку статической библиотеки, выберите в строке меню команду **Сборка > Собрать решение**. В результате будет создана

статическая библиотека *MathLibrary.lib*, которая может использоваться другими программами.

### Примечание

При выполнении сборки из командной строки Visual Studio программа собирается в два этапа. Сначала запустите `cl /c /EHsc MathLibrary.cpp`, чтобы скомпилировать код и создать объектный файл с именем *MathLibrary.obj*. (Команда `cl` вызывает компилятор `Cl.exe`, а параметр `/c` дает указание компилировать без компоновки. Дополнительные сведения см. в разделе [Параметр /c \(компиляция без связывания\)](#)). Во-вторых, запустите файл `lib MathLibrary.obj`, чтобы связать код и создать статическую библиотеку *MathLibrary.lib*. (Команда `lib` вызывает диспетчер библиотек, `Lib.exe`. Дополнительные сведения см. в разделе [LIB Reference](#).)

### Создание консольного приложения C++, ссылающегося на статическую библиотеку

#### Создание консольного приложения C++, ссылающегося на статическую библиотеку, в Visual Studio 2019

1. В обозревателе решений щелкните правой кнопкой мыши узел верхнего уровня **Решение StaticMath**, чтобы открыть контекстное меню. Выберите пункты **Добавить > Новый проект**, чтобы открыть диалоговое окно **Добавить новый проект**.
2. В верхней части диалогового окна задайте для фильтра **Тип проекта** значение **Консоль**.
3. В отфильтрованном списке типов проектов щелкните **Консольное приложение**, а затем нажмите кнопку **Далее**. На следующей странице в поле **Имя** введите имя проекта *MathClient*.
4. Нажмите кнопку **Создать**, чтобы создать клиентский проект.
5. После создания консольного приложения будет создана пустая программа. Имя исходного файла будет совпадать с ранее выбранным именем. В этом примере он имеет имя *MathClient.cpp*.

### Использование функциональности из статической библиотеки в приложении

#### Использование функциональности из статической библиотеки в приложении

1. Для использования математических процедур из статической библиотеки необходимо сослаться на эту библиотеку. В обозревателе решений откройте контекстное меню проекта **MathClient**, а затем выберите команду **Добавить > Ссылка**.

2. В диалоговом окне **Добавление ссылки** перечислены библиотеки, на которые можно создать ссылку. На вкладке **Проекты** перечислены проекты текущего решения и все библиотеки, на которые они ссылаются. На вкладке **Проекты** установите флажок **MathLibrary**, а затем нажмите кнопку **ОК**.
3. Для создания ссылки на файл заголовка **MathLibrary.h** необходимо изменить путь к каталогам включаемых файлов. В **обозревателе решений** щелкните правой кнопкой мыши проект **MathClient**, чтобы открыть контекстное меню. Выберите пункт **Свойства**, чтобы открыть диалоговое окно **Страницы свойств MathClient**.
4. В диалоговом окне **Страницы свойств MathClient** в раскрывающемся списке **Конфигурация** выберите пункт **Все конфигурации**. В раскрывающемся списке **Платформа** выберите пункт **Все платформы**.
5. Перейдите на страницу свойств **Свойства конфигурации > C/C++ > Общие**. В свойстве **Дополнительные каталоги включаемых файлов** укажите путь к каталогу **MathLibrary** или найдите этот каталог.

Чтобы найти путь к каталогу, выполните указанные ниже действия.

1. Откройте раскрывающийся список значений свойства **Дополнительные каталоги включаемых файлов**, а затем выберите **Изменить**.
2. В диалоговом окне **Дополнительные каталоги включаемых файлов** дважды щелкните в верхней части текстового поля. Нажмите кнопку с многоточием ( ... ) в конце строки.
3. В диалоговом окне **Выбор каталога** перейдите на уровень вверх и выберите каталог **MathLibrary**. Затем нажмите кнопку **Выбрать папку**, чтобы сохранить выбор.
4. В диалоговом окне **Дополнительные каталоги включаемых файлов** нажмите кнопку **ОК**.
5. В диалоговом окне **Страницы свойств** нажмите кнопку **ОК**, чтобы сохранить изменения в проекте.
6. Теперь в этом приложении можно использовать класс **Arithmetic**, включив в код заголовок `#include "MathLibrary.h"`. Замените содержимое **MathClient.cpp** на следующий код:

C++Копировать

```
// MathClient.cpp
```

```
// compile with: cl /EHsc MathClient.cpp /link MathLibrary.lib
```

```
#include <iostream>
```

```
#include "MathLibrary.h"
```

```
int main()
```

```
{
```

```

double a = 7.4;
int b = 99;

std::cout << "a + b = " <<
    MathLibrary::Arithmetic::Add(a, b) << std::endl;
std::cout << "a - b = " <<
    MathLibrary::Arithmetic::Subtract(a, b) << std::endl;
std::cout << "a * b = " <<
    MathLibrary::Arithmetic::Multiply(a, b) << std::endl;
std::cout << "a / b = " <<
    MathLibrary::Arithmetic::Divide(a, b) << std::endl;

return 0;
}

```

7. Чтобы выполнить сборку исполняемого файла, выберите в строке меню команду **Сборка > Собрать решение**.

## Запуск приложения

### Запуск приложения

1. Убедитесь в том, что проект **MathClient** выбран в качестве проекта по умолчанию. Чтобы выбрать его, в **обозревателе решений** откройте контекстное меню проекта **MathClient** и выберите команду **Назначить запускаемым проектом**.
2. Чтобы запустить проект, в строке меню выберите **Отладка > Запуск без отладки**. Выходные данные должны выглядеть примерно так:

```

OutputКопировать
a + b = 106.4
a - b = -91.6
a * b = 732.6
a / b = 0.0747475

```