Создание и использование DLL

Библиотека динамической компоновки (DLL) является исполняемым файлом, который выполняет функции общей библиотеки. Динамическая компоновка представляет способ вызова функции, который не является частью исполняемого кода. Исполняемый код функции расположен в библиотеке DLL, которая содержит несколько компилированных, связанных и отдельно сохраненных функций в используемых процессах. Библиотеки DLL часто упрощают процесс общего доступа к данным и источникам. Многочисленные приложения могут иметь одновременный доступ к нескольким содержаниям одной копии DLL в памяти.

Динамическая компоновка отличается от статической компоновки тем, что позволяет исполняемым модулям (таким как файл .dll или .exe) включать только необходимую информацию в среду выполнения и размещать исполняемый код в функции DLL. В статической компоновке компоновщик получает все указанные функции из библиотеки и размещает код в исполняемой среде.

Динамическая компоновка имеет некоторые преимущества над статической. Библиотеки DLL сохраняются в памяти, уменьшают количество обменов, занимают небольшой объем места на диске, упрощают процесс обновления, предоставляют вторичную поддержку, а также обеспечивают механизмом для расширения классов библиотеки MFC, поддерживают многоязыковые программы и упрощают создание международных версий.

Создание проекта библиотеки динамической компоновки DLL

Создайте новый проект, который имеет тип «Консольное приложение Win32». Укажите имя проекта, например, MyDLL и имя решения, например DynamicLibrary. На странице **Параметры приложения** в разделе **Тип приложения** выберите **Библиотека DLL** (Рисунок 1).

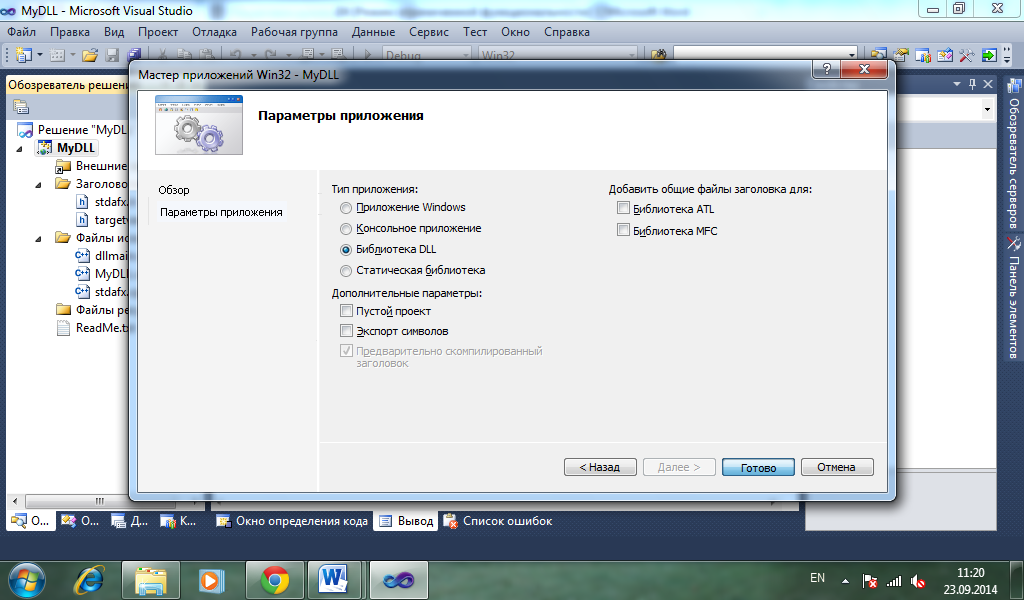


Рисунок 1 – Выбор типа приложения

Добавьте директиву включения головного файла #include “MyDLL.h” и описание функции Code в проект, в результате файл MyDLL.cpp будет выглядеть следующим образом:

#include "stdafx.h"

#include "MyDLL.h"

char \* Code\_Dec (char \*s, char Key)

{

for(int i=0; ; i++)

{

if (s[i] == '\0') break;

s[i] = s[i]^Key;

}

return s;

}

Далее создадим заголовочный файл библиотеки.

Чтобы создать файл заголовка, в меню **Проект** выберите **Добавить новый элемент**. В диалоговом окне **Добавить новый элемент** в левой области в разделе **Visual C++** выберите **Код**. В центральной области выберите **Заголовочный файл (.h)**. Укажите имя для файла заголовка MyDLL.h.

Запишите в файл следующий код:

#ifndef \_MYDLLH

#define \_MYDLLH

#ifdef MYDLL\_EXPORTS

#define MYDLL\_API \_\_declspec(dllexport)

#else

#define MYDLL\_API \_\_declspec(dllimport)

#endif

extern "C" MYDLL\_API char \* Code\_Dec (char \* s, char Key);

#endif

Когда символ MYDLL\_EXPORTS определен, символ MYDLL\_API установит модификатор \_\_declspec(dllexport) в объявлениях функции-члена в этом коде. Этот модификатор разрешает экспорт функции библиотекой DLL для использования ее другими приложениями. При неопределенном MYDLL\_EXPORTS MYDLL\_API определяет модификатор \_\_declspec(dllimport) в объявлениях функции-члена. Этот модификатор позволяет компилятору оптимизировать импорта функции из библиотеки DLL для использования в других приложениях. По умолчанию MYDLL\_EXPORTS определяется при сборке проекта MyDLL.

В свойствах компоновщика укажите название **Библиотеки импорта** MyDLL.lib.

Скомпилируйте библиотеку динамической компоновки.

Если вы захотите теперь протестировать вашу DLL и выполните команду Run, то получите сообщение (Рисунок 2).

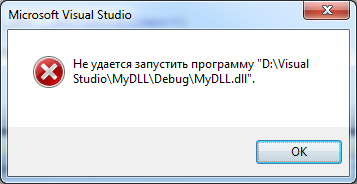


Рисунок 2 – Сообщение при запуске DLL

Это означает, что сначала нужно определить тестирующее приложение.

Создание приложения, ссылающегося на DLL

Чтобы создать приложение, которое будет ссылаться и использовать ранее созданную библиотеку DLL, создайте новый проект (Приложение Windows Forms). Введите имя проекта TestDLL, в поле **Решение** выберите **Добавить в решение**. В результате новый проект будет добавлен в то же решение, содержащее DLL.

Для использования в приложении функций, созданных в библиотеке DLL, необходимо сослаться на эту библиотеку. Для этого в **Обозревателе решений** выберите проект TestDLL, затем в меню **Проект** выберите **Ссылки**. В диалоговом окне **Страницы свойств**, на вкладке общие свойства, выберите **.NET Framework и ссылки** и нажмите кнопку **Добавить новую ссылку**.

В диалоговом окне **Добавление ссылки** перечислены библиотеки, на которые можно создать ссылку. На вкладке **Проект** перечислены все проекты текущего решения и включенные в них библиотеки, если они есть. Выберите проект MyDLL и нажмите кнопку OK.

Для создания ссылки на файлы заголовков DLL необходимо изменить путь к каталогам включения. Для этого в диалоговом окне **Окна свойств** последовательно развернуть **Свойства конфигурации**, **C/C++**, а затем выберите **Общие**. В поле **Дополнительные каталоги включаемых файлов** укажите путь к месту размещения файла заголовков MyDLL.h.

На форме проекта TestDLL разместите окно редактирования и кнопку. В обработчике нажатия кнопки поместите следующий код:

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

textBox1->Text = CharToSystemString(Code\_Dec(SystemStringToChar(textBox1->Text),'A'));

}

Добавьте две функции преобразования типов:

public: char\* SystemStringToChar(System::String^ string)

{

return (char\*)(void\*)System::Runtime::InteropServices::Marshal::StringToHGlobalAnsi(string);

}

System::String^ CharToSystemString(char\* ch)

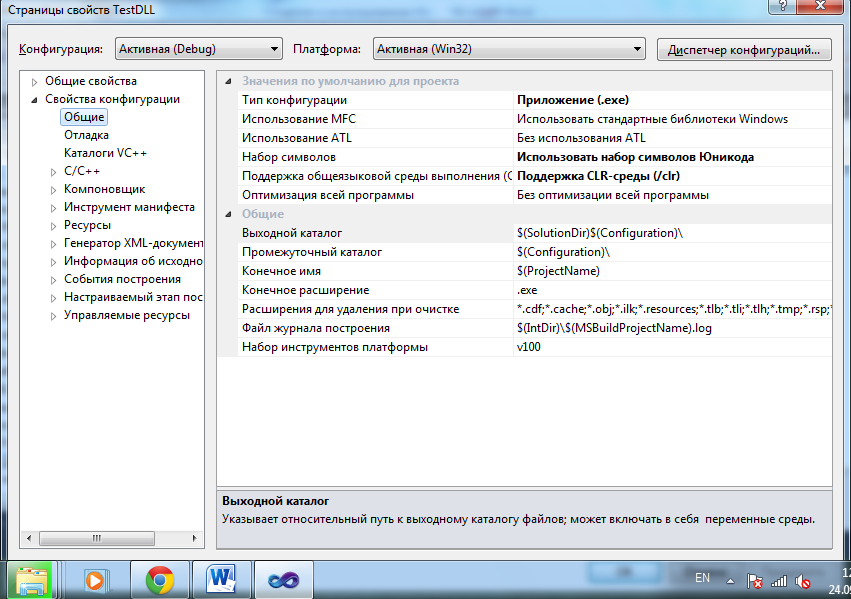
{

return gcnew String(ch);

}

Он берет текст, занесенный пользователем в окно редактирования textBox1, кодирует его с помощью функции Code\_Dec и возвращает закодированную строку в textBox1. Включите в модуль директиву, подключающую заголовочный файл библиотеки.

Измените в свойствах оконного приложения опцию компиляции на /clr.



Выполните построение решения. Перед запуском приложения убедитесь, что проект TestDLL выбран в качестве проекта по умолчанию. В **Обозревателе решений** выберите TestDLL и затем в меню **Проект** выберите **Назначить запускаемым проектом**.