**Практическая работа № 7**

***Подключение библиотеки DLL***

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

***Цель*** практической работы – научиться работать с *библиотеками DLL*, написать программу с использованием *библиотеки DLL*.

***Оснащение*:**конспект, компьютер.

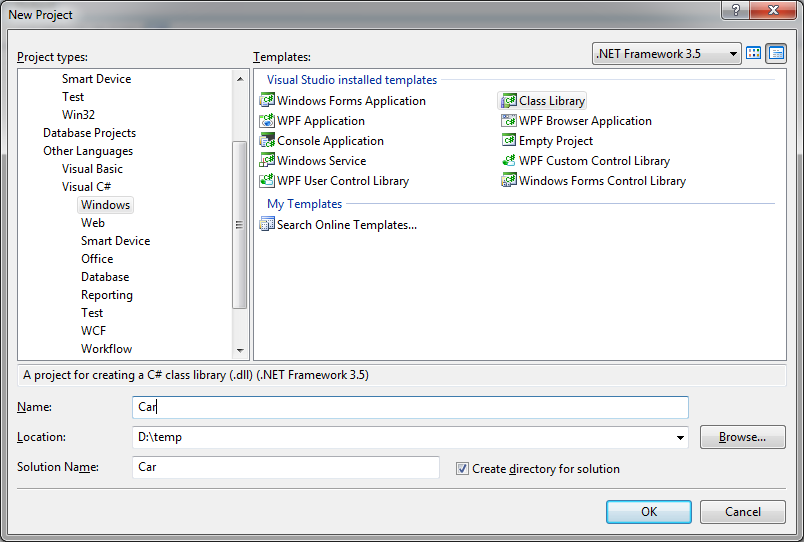
**Краткие теоретические сведения:**

DLL - это просто наборы функций, собранные в библиотеки. Однако, в отличие от своих статических родственников (файлов . lib), библиотеки DLL не присоединены непосредственно к выполняемым файлам с помощью редактора связей. В выполняемый файл занесена только информация об их местонахождении. В момент выполнения программы загружается вся библиотека целиком. Благодаря этому разные процессы могут пользоваться совместно одними и теми же библиотеками, находящимися в памяти. Такой подход позволяет сократить объем памяти, необходимый для нескольких приложений, использующих много общих библиотек, а также контролировать размеры ЕХЕ-файлов.

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.

**Создание файла dll**

Для начала выберем тип нового создаваемого проекта. Выбираем Class Library, то есть создаем файл динамической библиотеки (dll)



Код dll.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Car

{

public class class1

{

}

}

Пространство имён: Car и класс: Class1. заменим Class1 на BMW, и добавим метод, который будет выводить имя нашего класса:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Car

{

public class BMW

{

public void NameofClasses()

{

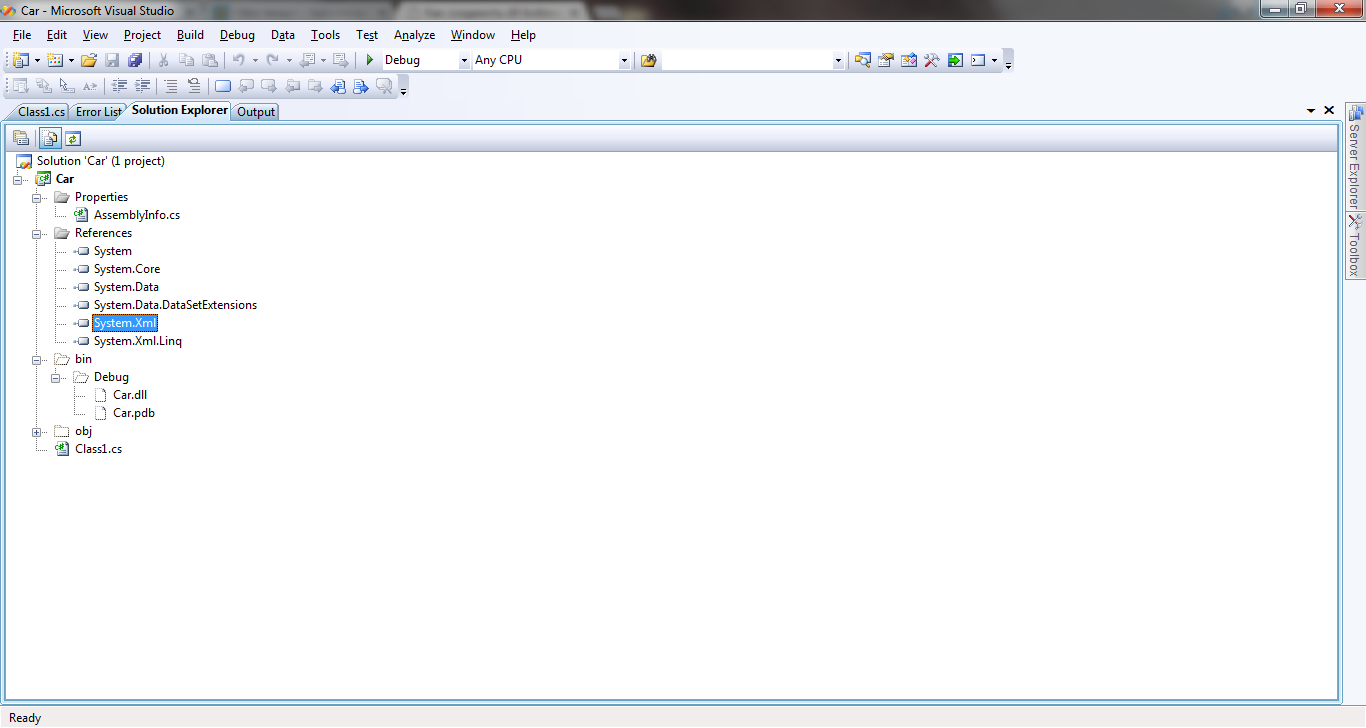
Console.WriteLine("BMW");

}

}

}

Чтобы получить сборку, выполним компиляцию, нажав F7. Следует заметить, что создается файл динамической библиотеки (dll), а не исполняемый файл (exe), который не может быть запущен. Поэтому компиляция через F5 неприемлема. После компиляции в директории bin\Debug появится файл Car.dll. Чтобы проверить был ли создан файл библиотеки, воспользуйтесь кнопкой Show All Files на вкладке Solution Explorer.



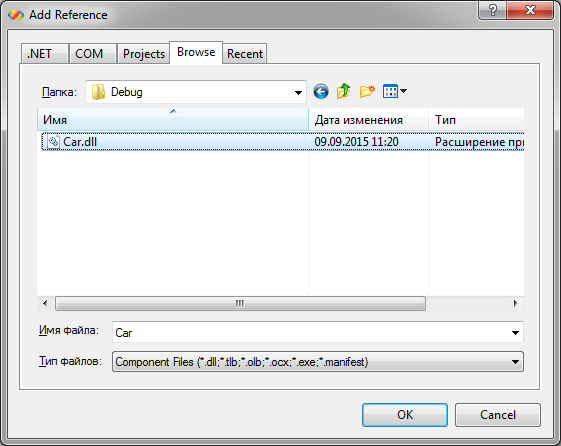
Сборка в виде файла динамической библиотеки успешно создана.

Для того, чтобы быстро переместиться в текущую директорию проекта, в том же Solution Explorer можно воспользоваться пунктом Open Folder in Windows Explorer

Закроем текущий проект и создадим новый. Но, на этот раз выберем тип проекта — консольное приложение. Подключим в текущий проект, нашу библиотеку (Car.dll)

**Подключение dll**

Для этого на папке References, в окне Solution Explorer нужно нажать правую кнопку мыши и выбрать пункт Add Reference, откроется окно:



1. Выберите вкладку Browse

2. Укажите папку, в которой лежит файл созданной библиотеки (в нашем примере — Car.dll)

3. Выделите файл нужной библиотеки и нажмите кнопку «ОК»

Теперь в текущий проект была успешно добавлена ссылка на сборку Car.dll. В папку bin\Debug была помещёна копия созданной библиотеки.

Чтобы посмотреть, какое пространство имен, типы, или члены содержит dll библиотека, можно воспользоваться таким инструментом Visual Studio, как Object Browser. Для этого перейдите на вкладку Solution Explorer, откройте папку References, и щёлкните правой кнопкой мыши по добавленной ранее библиотеке, в нашем случае — это файл (Car.dll) и выберите пункт View in Object Browser. В окне Object Browser можно посмотреть содержимое нашей сборки.

Добавим, с помощью ключевого слова using пространство имен Car из созданной нами библиотеки Car.dll, после чего создадим объект класса BMW и выполним метод NameofClasses().

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Car;

namespace MyApplication

{

class Program

{

static void Main(string[] args)

{

BMW my = new BMW();

my.NameofClasses();

Console.ReadKey();

}

}

}

**Порядок выполнения работы:**

1. Ознакомиться с определением библиотеки DLL .
2. Создать библиотеку DLL в соответствии с вариантом задания .
3. Написать демонстрационную программу, в которой подключается созданная библиотека DLL.
4. Составить отчет.
5. Сделать выводы по работе.
6. Ответить на контрольные вопросы.

**Варианты заданий**

В рамках каждого индивидуального задания требуется создать библиотеку DLL и приложение, демонстрирующее использование библиотеки. Индивидуальное задание описывает требования к библиотеке. Тестирующее приложение может иметь произвольный вид. Главное требование к нему – полноценная демонстрация использования созданной библиотеки.

1). Создать библиотеку DLL, содержащую функции для перевода чисел из десятичной системы в двоичную и обратно.

2). Создать библиотеку DLL, содержащую функции для перевода чисел из десятичной системы в восьмеричную и обратно.

3). Создать библиотеку DLL, содержащую функции для перевода чисел из десятичной системы в шестнадцатеричную и обратно.

4). Создать библиотеку DLL, содержащую функции для поиска минимального и максимального элементов двумерного массива.

5). Создать библиотеку DLL, содержащую функции для сортировки элементов одномерного массива по возрастанию и убыванию.

6). Создать библиотеку DLL, содержащую функции для решения квадратных уравнений.

7). Создать библиотеку DLL, содержащую функцию для определения количества прожитых человеком лет по дате его рождения.

8). Создать библиотеку DLL, содержащую функцию для сбора личных данных и записи их в текстовые файлы. Имена файлов должны состоять из фамилии, имени и отчества.

9). Создать библиотеку DLL, содержащую функцию для поиска заданного элемента в двумерном массиве. Результатом должен стать номер строки и столбца с найденным элементом.

10). Создать библиотеку DLL, содержащую функции для реализации операций объединения, вычитания, пересечения множеств.

11). Создать библиотеку DLL, содержащую функцию для вычисления значения произвольной функции и сохраняющую результат в текстовый файл.

12). Создать библиотеку DLL, содержащую функцию для сохранения значений двумерного массива в текстовом файле.

13). Создать библиотеку DLL, содержащую функцию для нахождения наибольшего общего делителя элементов целочисленного массива.

14). Создать библиотеку DLL, содержащую функцию, находящую разложение натурального числа на простые множители. Количество множителей возвращается в целой переменной, а сами множители в виде целочисленного массива.

15). Создать библиотеку DLL, содержащую функцию для округления элементов двумерного массива до требуемого знака после запятой.

16). Создать библиотеку DLL, содержащую функцию для поиска и замены в двумерном целочисленном массиве заданного элемента на ноль.

17). Создать библиотеку DLL, содержащую функцию для замены в двумерном массиве двух строк местами. Входные параметры – номера строк.

18). Создать библиотеку DLL, содержащую функцию для вычисления среднего арифметического элементов заданной строки в двумерном массиве.

19). Создать библиотеку DLL, содержащую функцию для определения числа слов в строке. Слова разделяются пробелом.

20). Создать библиотеку DLL, содержащую функцию для определения парности круглых скобок в строке.

**Содержание отчета**

1. Название и цель работы.
2. Оснащение для выполнения лабораторной работы.
3. Вариант задания
4. Демонстрационная программа.
5. Выводы по работе.
6. Ответы на контрольные вопросы.

**Контрольные вопросы**.

1. Что такое DLL?

2.Что представляет динамическая компоновка ?

3.Чем динамическая компоновка отличается от статической компоновки?