

Библиотеки DLL

Библиотеки DLL используются для переноса какой-либо программы из приложения в приложение.

Невозможно создать сколько-нибудь серьезное приложение под Windows без использования библиотек динамической компоновки **dll** (Dynamic Link Library). **dll** – это наборы функций и данных, собранные в отдельную библиотеку. Во время выполнения приложения и обращения к библиотечным функциям **dll**-библиотеки, последняя загружается в память. Потом приложение использует функции и данные библиотеки, и когда они больше не нужны - выгружает библиотеку. В этом заключается ее динамичность, это позволяет экономить память компьютера.

Создание и использование библиотек **DLL** в платформе .NET не представляет никакой сложности. Для создания библиотеки в Visual Studio есть специальный тип проекта, «библиотека классов».

Не обязательно писать библиотеки на языке C#, подойдет любой другой язык, интегрированный с платформой .NET.

```
public class Person          // public - обязательно
{
    private string _name;
    private string _surname;
    private int _age;

    public Person()
    {
    }

    public string Name
    {
        get { return _name; }
        set { _name = value; }
    }

    public string Surname
    {
        get { return _surname; }
        set { _surname = value; }
    }

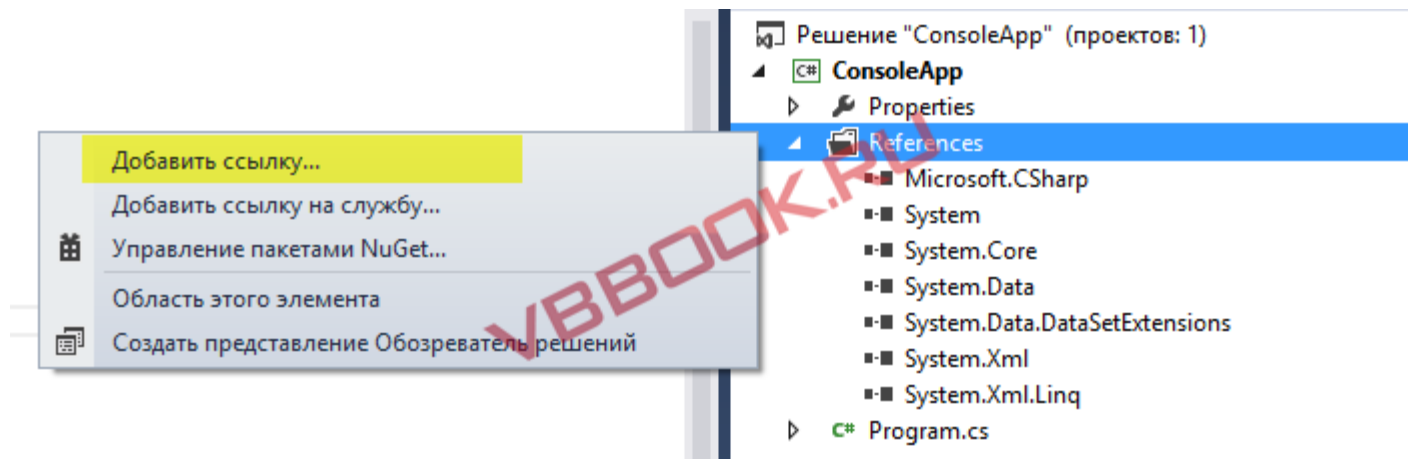
    public int Age
    {
        get { return _age; }
        set { _age = value; }
    }
}
```

Компиляция

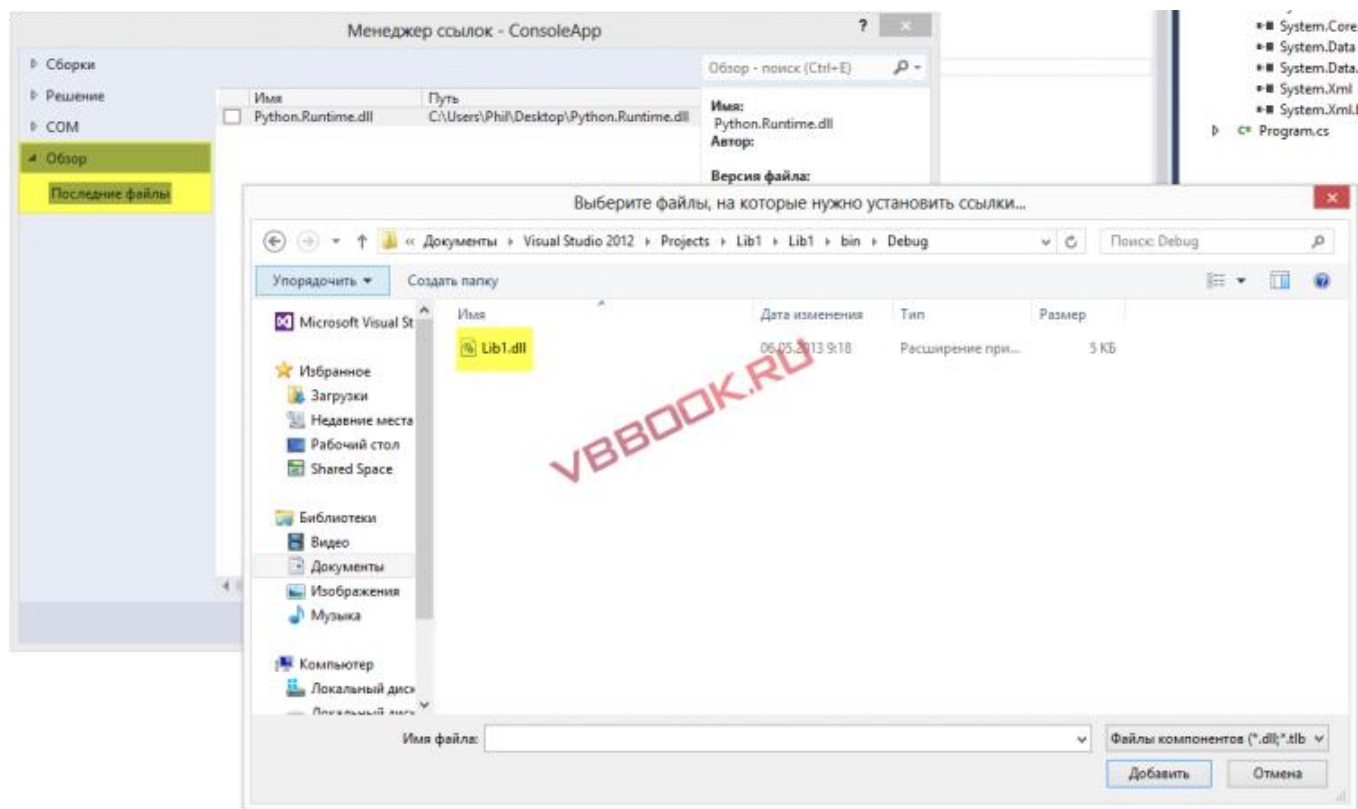
Выберите конфигурацию «Release» и нажмите клавишу F6. В директории binRelease будет лежать библиотека.

Подключение


Создайте проект типа Console Application. Выполните команды “References -> Add reference”:



В окне «Add reference» выберите «Обзор» и укажите путь к вашей библиотеке:



Выделите библиотеку и нажмите «Добавить». После этого в списке «References» должна появиться ваша библиотека.



Мы не указывали пространство имен в библиотеке, поэтому обращаться к классам можно следующим образом:

```
var $VarName$ = new $ClassName$;
```

Если же вы указали пространство имен, обращайтесь конструкцией вида:

```
var $VarName$ = new $namespace$. $ClassName$;
```

Пример работы с классом Person:

```
class Program
{
    static void Main(string[] args)
    {
        var p = new Person(); // у нас конструктор без параметров

        p.Name = "Andrey";
        p.Surname = "Andreyev";
        p.Age = 27;

        Console.WriteLine(p.Name);
        Console.WriteLine(p.Surname);
        Console.WriteLine(p.Age);

        Console.ReadKey();
    }
}
```