

This is the program we need to write today

```

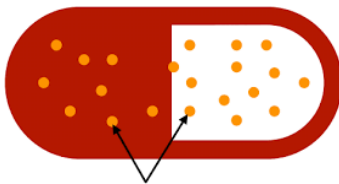
class ABBA
{
    static void Main(string[] args)
        // Here's a method called Main.
    {
        System.Console.WriteLine("ABBA!");
    }
}

```



So there's the keyword class. Unlike C++, in C# all code must be placed in a class.

Encapsulated in a class.

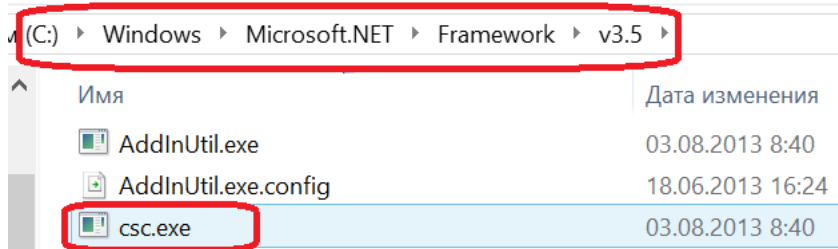


Medicines Inside Capsule



C# is case sensitive

C:\WINDOWS\Microsoft.NET\Framework\v3.5\ csc.exe



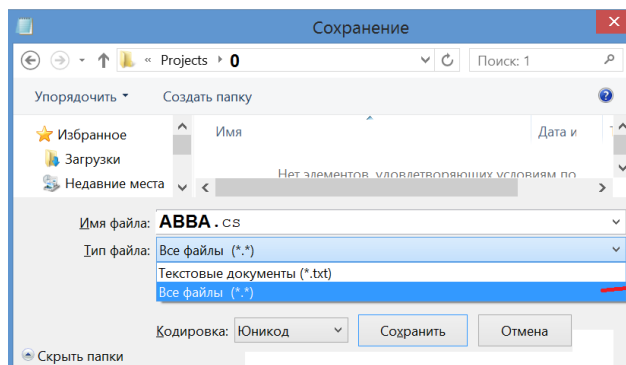
ABBA.cs => csc.exe = ABBA.exe  
C# Compiler

**Step 1.** And on my HDD, I also make a folder with the same name D:\ IT

**Step 2.** In the folder E:\ IT we make the folder of the **Projects** - E:\ IT**Projects**

And in the **Project** folder make folder **0** - E:\ IT**Projects**\0\ where our today's practical work will be stored

**Step 3.** As I mentioned above, C# is a built-in language of Windows. Notepad is enough to write a program



You need to switch from \*.txt (Text documents) to \*.\* (all files)

Otherwise, notepad with \*.txt extension

**Step 4.** Entering command mode

Start=>Run=>cmd

cd E: - After that go to the folder **IT/Projects/0/**  
cd IT - Then go to the folder **Projects**  
cd projects - Then go to the folder **0**  
cd 0 -

```

E:\>cd IT
E:\IT>cd Projects
E:\IT\Projects>cd 0
E:\IT\Projects\0>

```

### Step 5.

Now we need to compile the file using the compiler **csc.exe** which is in the folder  
C:\Windows\Microsoft.NET\Framework\v3.5

C:\WINDOWS\Microsoft.NET\Framework\v3.5\csc.exe ABBA.cs

the result is a file **ABBA.exe** that can already be run (which is located in the same folder). If you have taken the 1st step, then this means that the education-process has begun. This is victory. csc /target:library **ABBA.cs** - will make **ABBA.dll**.



**Step 6.** Modify the file as follows using System;

```

class ABBA
{
    static int Factorial(int n)
    {
        if (n == 1) return 1;
        return n * Factorial(n - 1);
    }
}

static void Main(string[] args)
// Here's a method called main.
{
    System.Console.WriteLine("ABBA -"+
        Factorial(4));
}

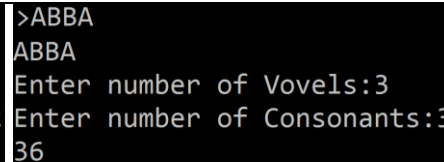
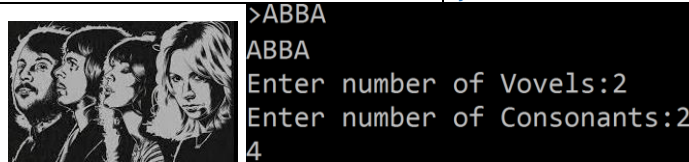
```

>C:\WINDOWS\Microsoft.NET\Framework\v3.5\csc.exe ABBA.cs

>ABBA  
ABBA – 24

**Step 7.** Modify the file as follows

<pre> using System;  class ABBA {     static int Factorial(int n)     {         if (n == 1) return 1;         return n * Factorial(n - 1);     } } </pre>	<pre> public static void Main() {     System.Console.WriteLine("ABBA");     System.Console.Write("Enter number of Vowels:");     string s = Console.ReadLine();     int <b>vowels</b>=int.Parse(s);     System.Console.Write("Enter number of Consonants:");     s=System.Console.ReadLine();     int <b>consonants</b>=int.Parse(s);     Console.WriteLine(Factorial(<b>vowels</b>)*Factorial(<b>consonants</b>) ); } </pre>
---	---



1

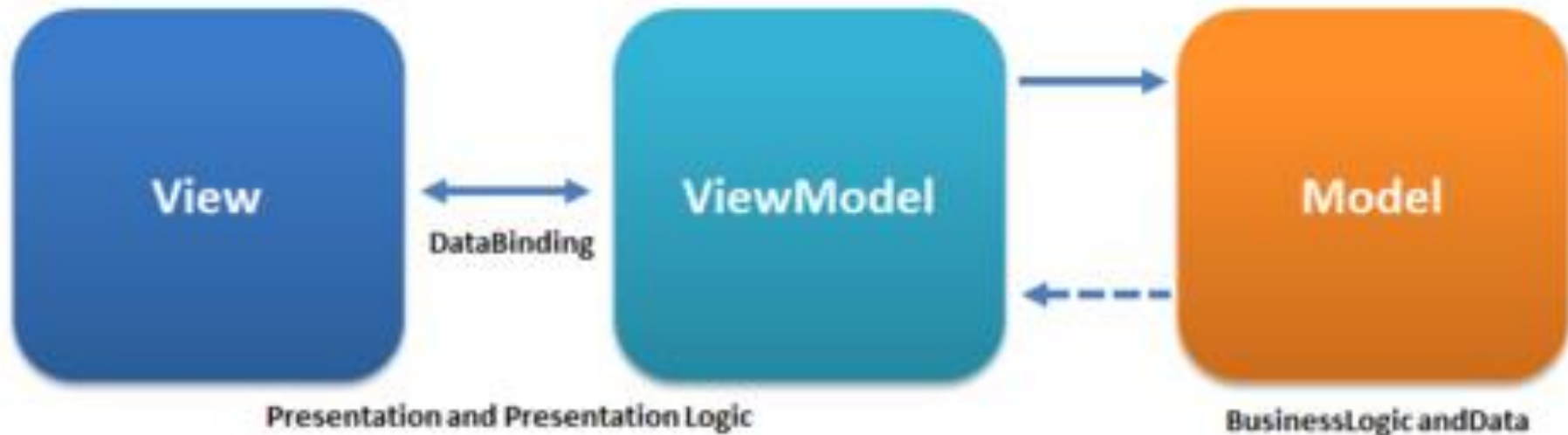
2

3

4



## Migrating from a console application to a WEB application.

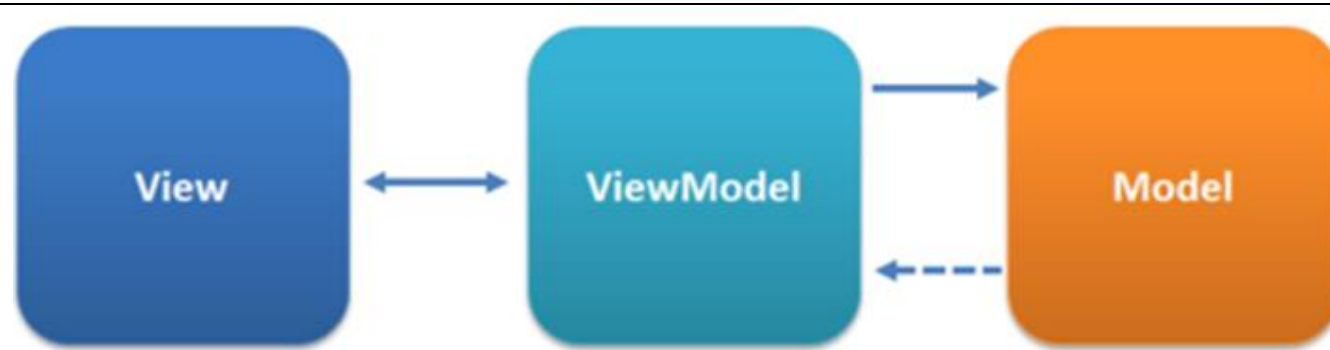


**Model–view–viewmodel (MVVM)** is an [architectural pattern](#) in computer software that facilitates the [separation](#) of the development of a [graphical user interface](#) (GUI; the *view*) from the development of the [business logic](#) or [back-end](#) logic (the *model*) such that the view is not dependent upon any specific model platform.

A little later we will be able to migrate from a console application to a WEB application.

To make this easier, it is better to write the program so that the part responsible for outputting information (so called View) is separated from the part responsible for logic .

This can be done by placing them in different methods (in Java Script this entity is called a "function"



```

using System;
class ABBA
{
    int vowels;
    int consonants;
    int result;
    int Factorial(int n)
    {
        if (n == 1) return 1;
        return n * Factorial(n - 1);
    }
}
  
```

```

void View(bool b);
{
    if (b) { System.Console.WriteLine("Enter
number of Vovels:");
        string s = Console.ReadLine();
        vowels=int.Parse(s);
        System.Console.WriteLine("Enter number of
Consonants:");
        s=System.Console.ReadLine();
        consonants=int.Parse(s);
    }
    else
    {
        System.Console.WriteLine(result);
    }
}
  
```

```

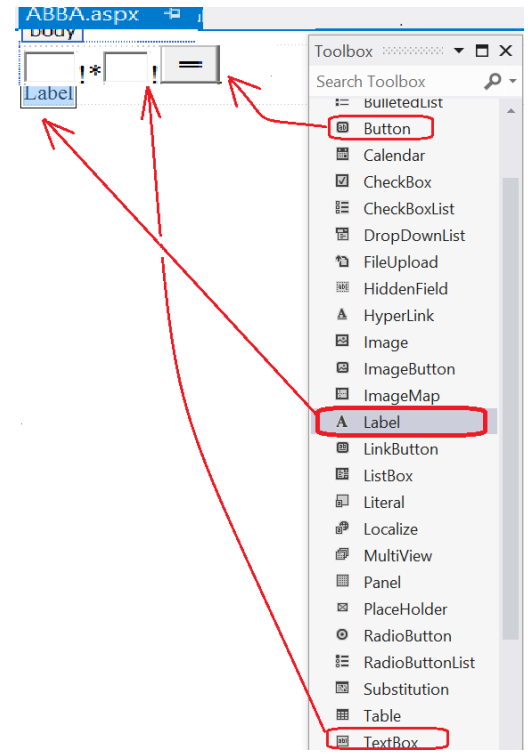
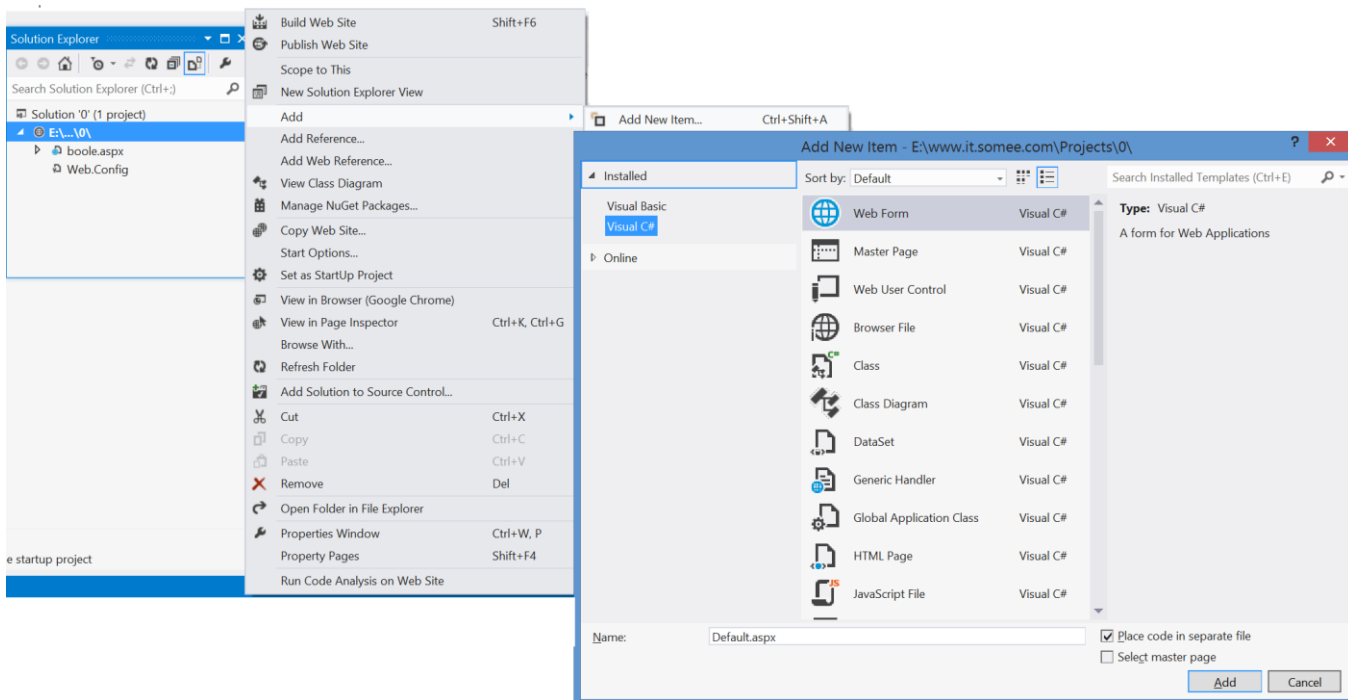
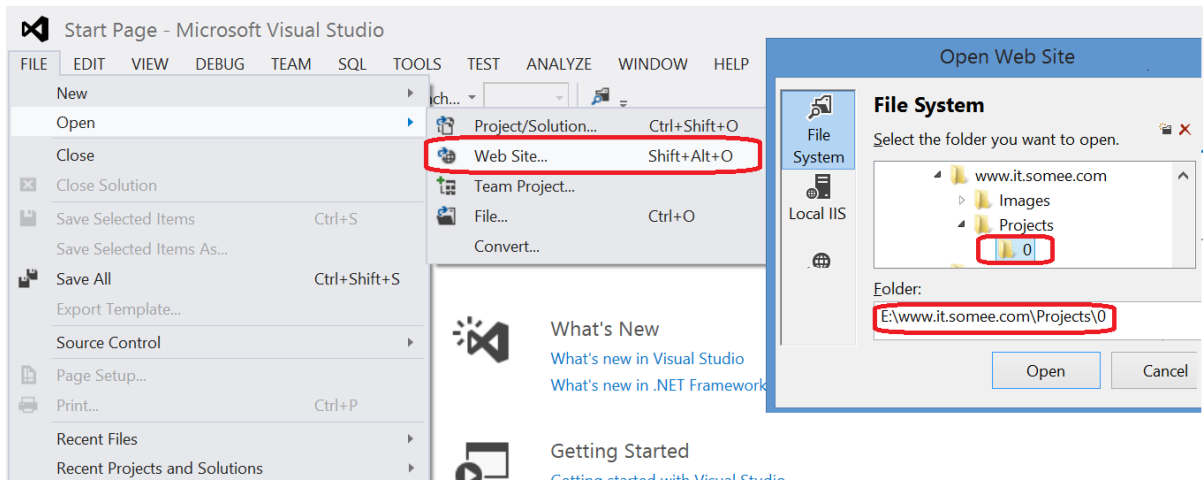
static void
ViewModel() {
    View(true);
    Model();
    View(false);
}
  
```

```

void Model();
{
    result =
    Factorial(vowels)*
    Factorial(consonants);
}
  
```

```

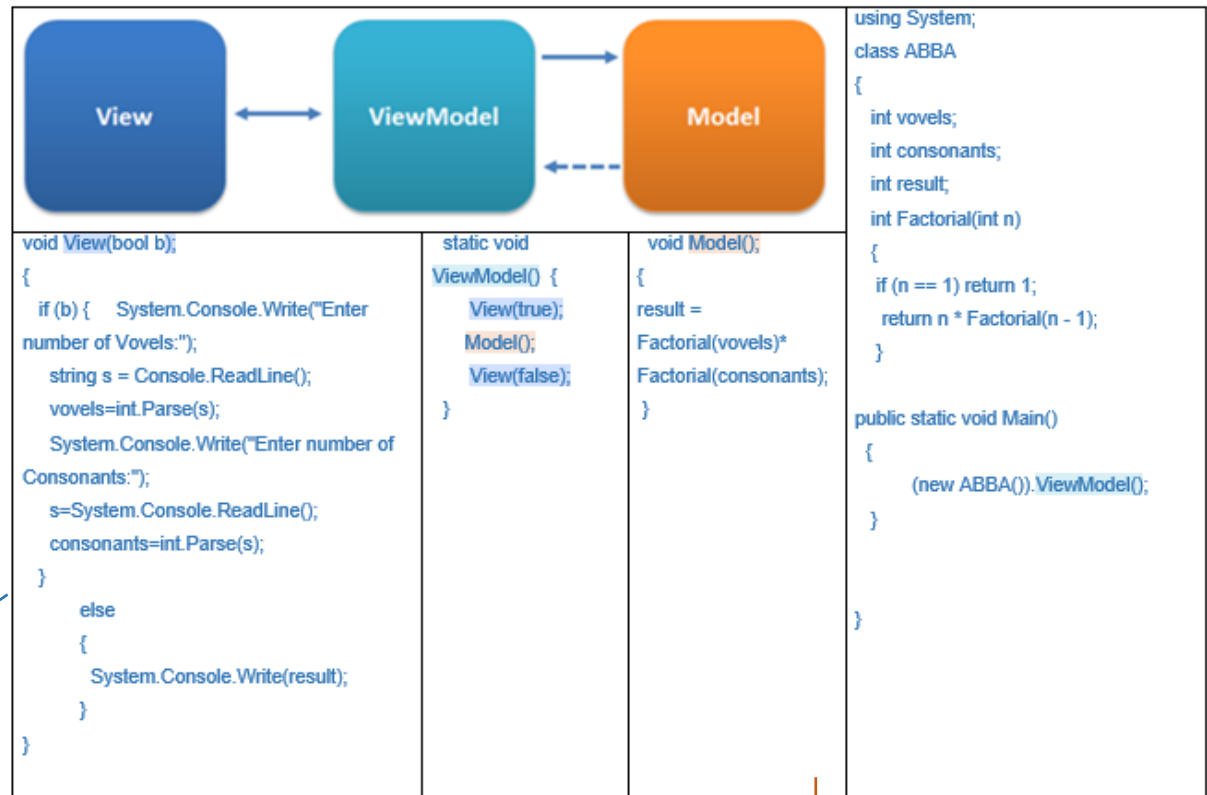
public static void Main()
{
    (new ABBA()).ViewModel();
}
  
```



```

public partial class ABBA:
    System.Web.UI.Page
{
    int vowels;
    int consonants;
    int result;
    protected void
    int Factorial(int n)
    {
        if (n == 1) return 1;
        return n * Factorial(n - 1);
    }
    protected void
    cmdGo_Click(object sender, EventArgs e)
    {
        ViewModel() ;
    }
    public void ViewModel()
    {
        View(true);
        Model();
        View(false);
    }
}

```



```

using System;
class ABBA
{
    int vowels;
    int consonants;
    int result;
    int Factorial(int n)
    {
        if (n == 1) return 1;
        return n * Factorial(n - 1);
    }
    public static void Main()
    {
        (new ABBA()).ViewModel();
    }
}

```

```

void View(bool b);
{
    if (b){
        string s = TextBox1.Text;
        vowels = int.Parse(s);
        s = TextBox2.Text;
        consonants = int.Parse(s);
    }
}

```

```

else
{
    Label1.Text=
        result.ToString();
}
}

```

```

void Model();
{
    result = Factorial(vovels) *
    Factorial(consonants);
}

```