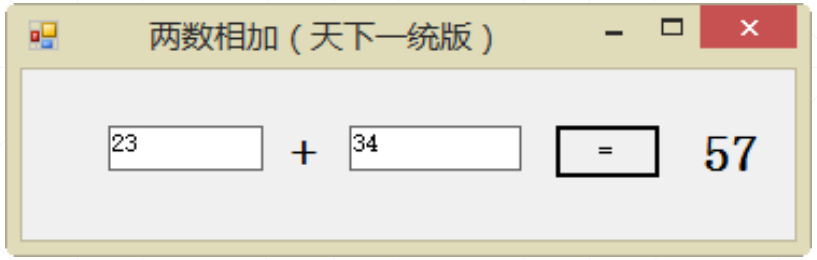
.NET组件化开发示例分析

从“天下一统”到“借鸡生蛋”

北京理工大学计算机学院
金旭亮

两数相加示例：天下一统版




```
1 reference
private void btnAdd_Click(object sender, EventArgs e)
{
    try
    {
        int num1 = Convert.ToInt32(txtNumber1.Text);
        int num2 = Convert.ToInt32(txtNumber2.Text);
        lblResult.Text = (num1+num2).ToString();
    }
    catch (Exception ex)
    {
        lblResult.Text = ex.Message;
    }
}
```

特征：
直接将功能代码写到事件响应方法中。

考虑：
用户希望在输入数字时，能直接看到结果而无需点击“=”按钮。

弊端：
引诱程序员Copy & Paste代码。

体制改革版

```
/// <summary>
/// 完成两数相加的功能
/// </summary>
3 references
private void Add()
{
    try
    {
        int num1 = Convert.ToInt32(txtNumber1.Text);
        int num2 = Convert.ToInt32(txtNumber2.Text);
        lblResult.Text = (num1 + num2).ToString();
    }
    catch (Exception ex)
    {
        lblResult.Text = ex.Message;
    }
}
```

将两数相加的功能抽取为独立的方法

在事件响应方法中只写方法名字

请问：这么干的好处是什么？

大幅度地减少重复代码！

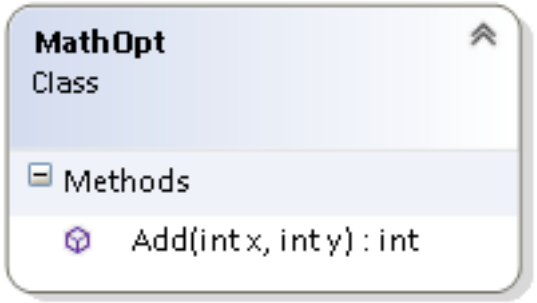
```
1 reference
private void btnAdd_Click(object sender, EventArgs e)
{
    Add();
}

1 reference
private void txtNumber1_TextChanged(object sender, EventArgs e)
{
    Add();
}

1 reference
private void txtNumber2_TextChanged(object sender, EventArgs e)
{
    Add();
}
```

体制改革版

```
/// <summary>
/// 完成两数相加的功能
/// </summary>
3 references
private void Add()
{
    try
    {
        int num1 = Convert.ToInt32(txtNumber1.Text);
        int num2 = Convert.ToInt32(txtNumber2.Text);
        lblResult.Text = (num1 + num2).ToString();
    }
    catch (Exception ex)
    {
        lblResult.Text = ex.Message;
    }
}
```

两个版本PK，你觉得哪个好？为什么？说出你的理由！

诸侯割据版

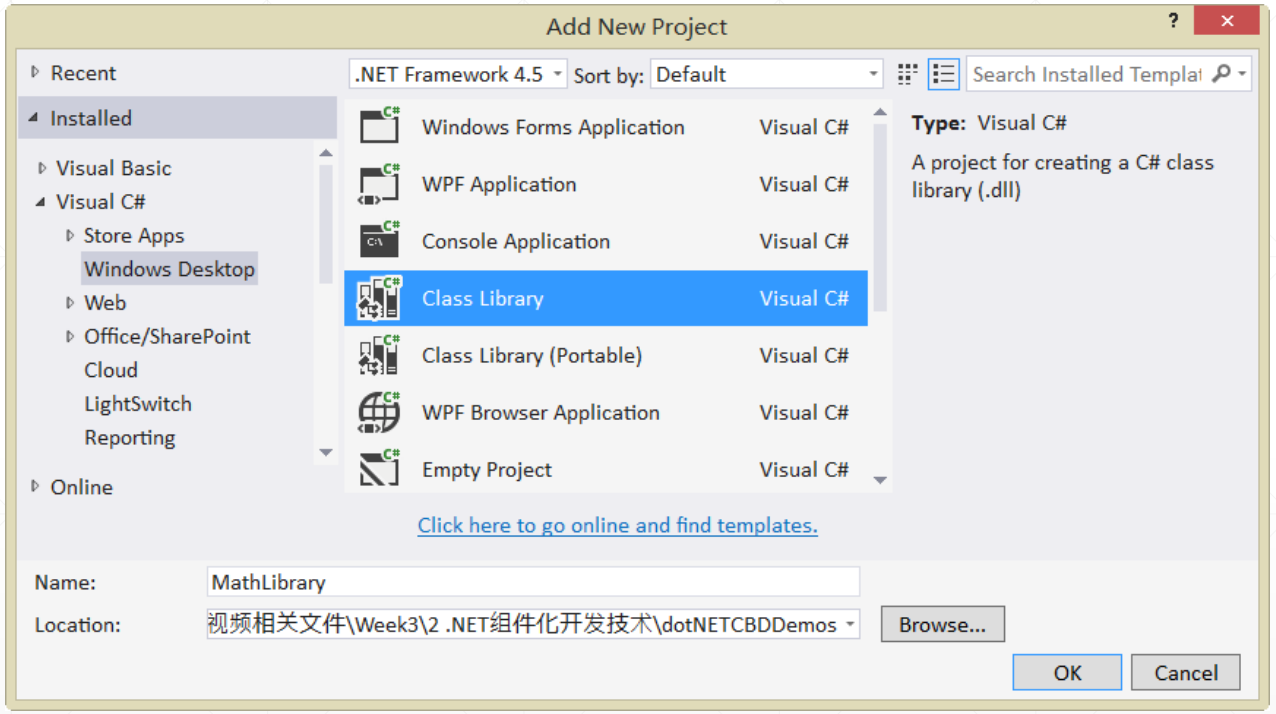



```
private MathOpt calculator = new MathOpt();
3 references
private void Add()
{
    try
    {
        int num1 = Convert.ToInt32(txtNumber1.Text);
        int num2 = Convert.ToInt32(txtNumber2.Text);
        calculator.Add(num1, num2);
    }
    catch (Exception ex)
    {
        lblResult.Text = ex.Message;
    }
}
```

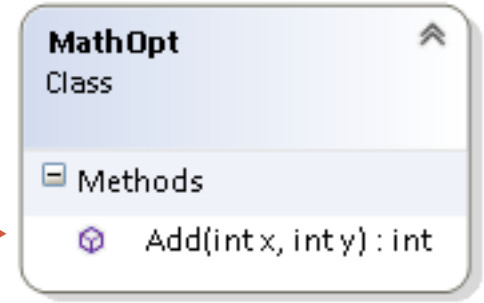
.NET平台上的组件化开发是基于程序集的

“类库(Class Library)”项目可用于创建程序集

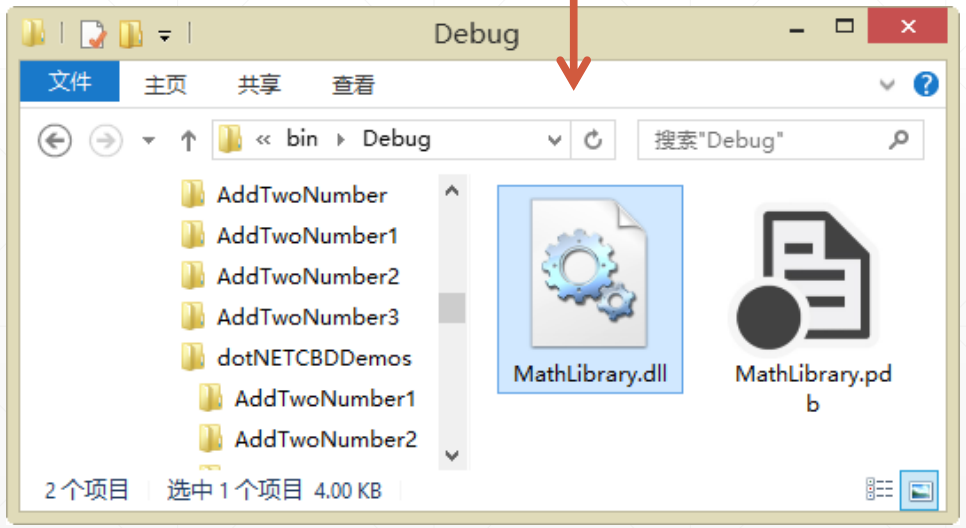
借鸡生蛋版



在类库项目中编写类

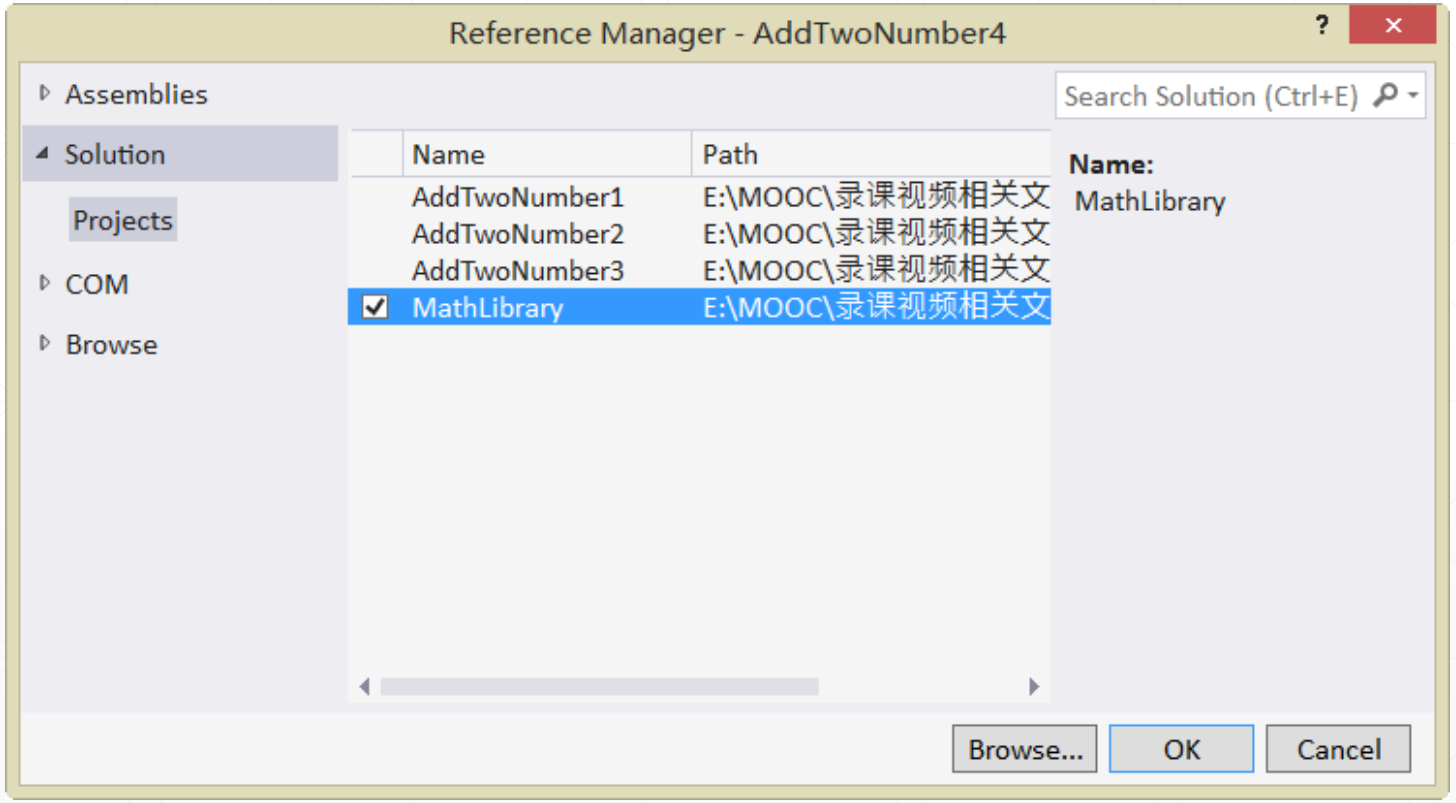


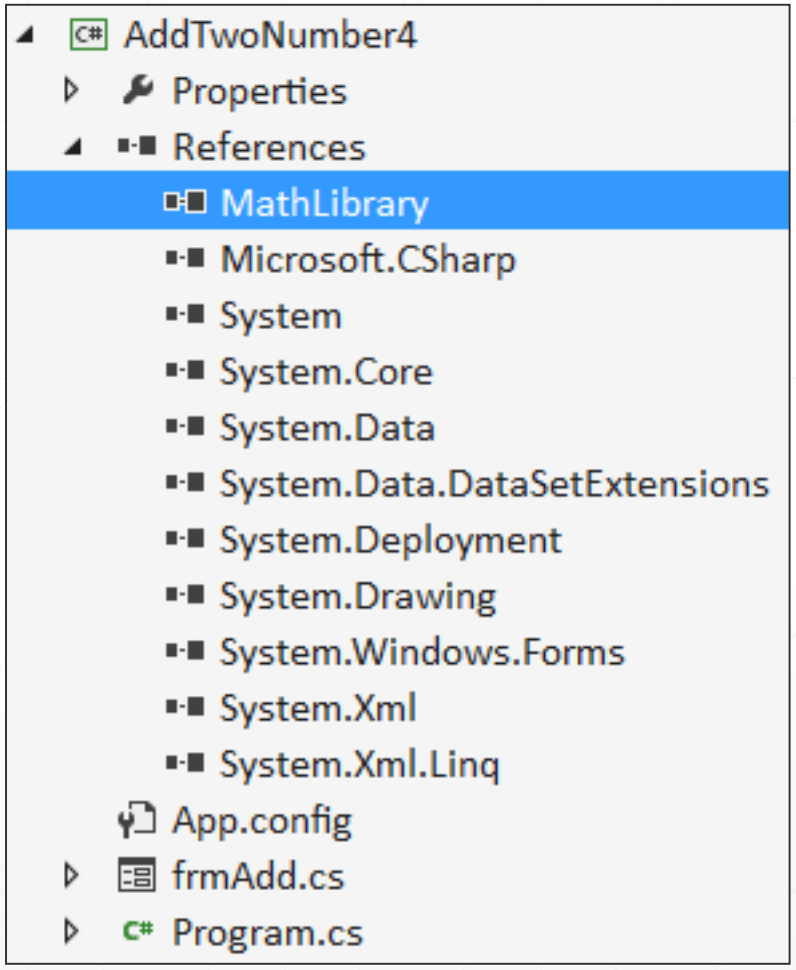
类库项目编译之后生成



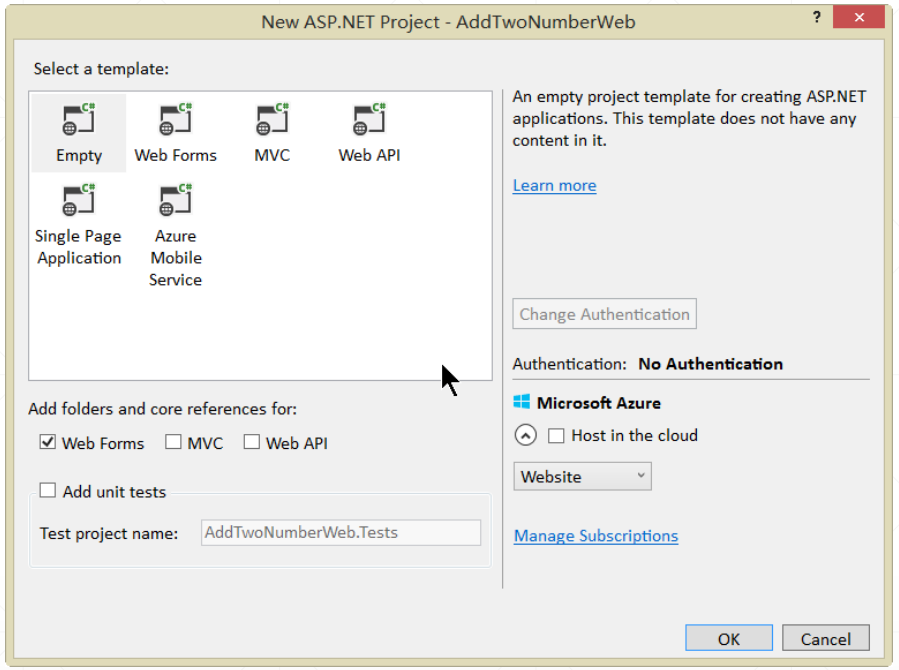
程序集实际上是一个.dll（或.exe）文件

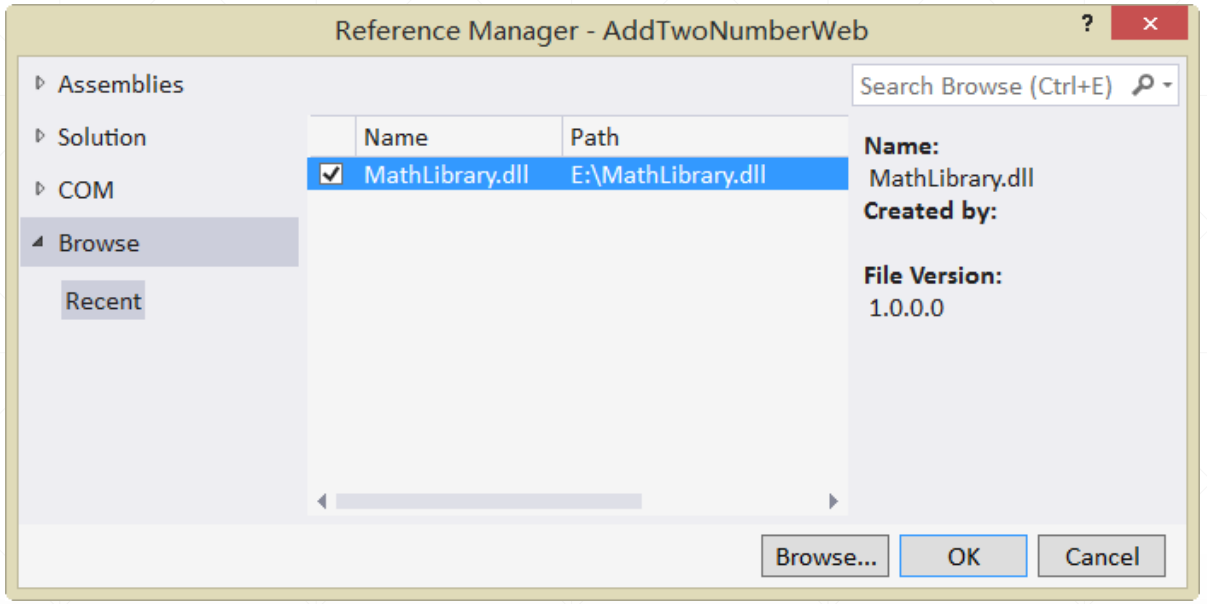
.NET项目通过“**添加引用（Add Reference）**”使用程序集组件



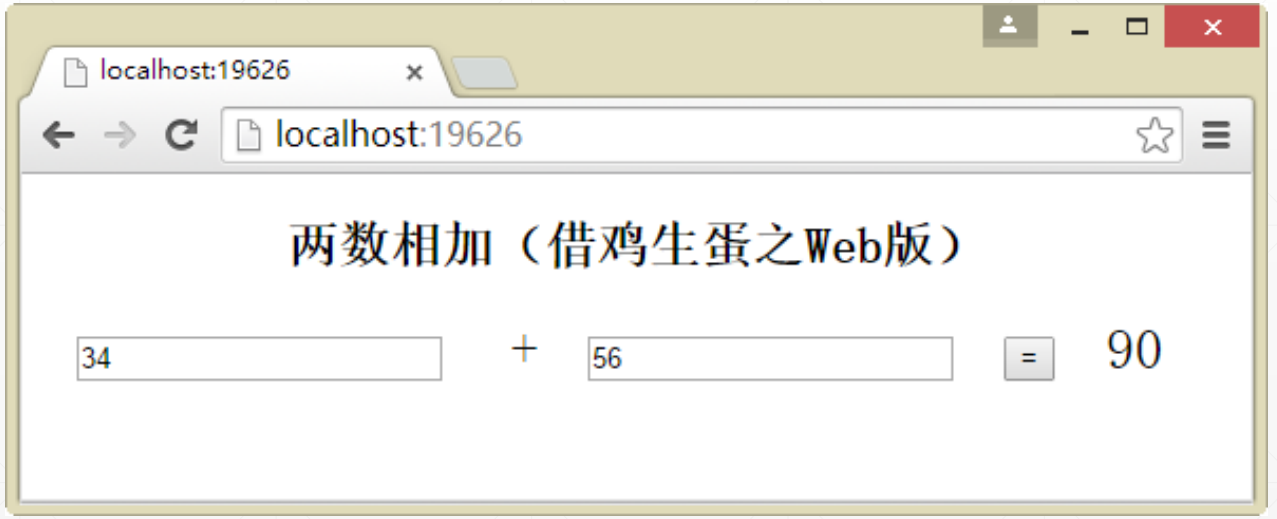


两数相加之Web版





在Web项目中使用程序集与普通桌面应用方式一致



回顾：

天下一统

- 所有代码均放到控件的事件响应方法
- 存在大量的重复代码

体制改革

- 将重复代码抽取为独立的方法，在控件的事件响应代码中直接调用方法
- 有效地消除了同一个类中的重复代码

诸侯割据

- 将需要“跨类”重用的代码移到独立的类中
- 使用者只需new一个对象，就能方便地调用这些代码。
- 实现了“跨越类边界”的代码重用

借鸡生蛋

- 用于代码重用的类封装到类库项目中，编译为二进制的程序集
- 使用者只需引用此程序集，就能重用其中的代码。
- 实现了“跨项目”的代码重用

我们得到了哪些启发？

1. 不在控件的事件响应代码中写功能代码
2. 尽可能少地编写重复代码
3. 界面代码与功能代码相分离
4. 复用二进制的程序集，而不是复用源代码