

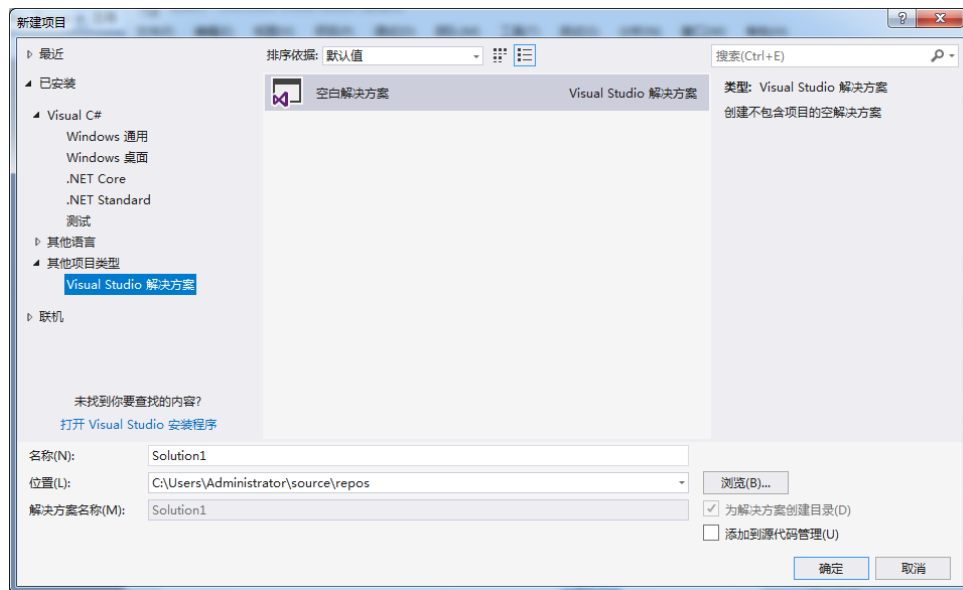
Lab1 C#概述

学习目标:

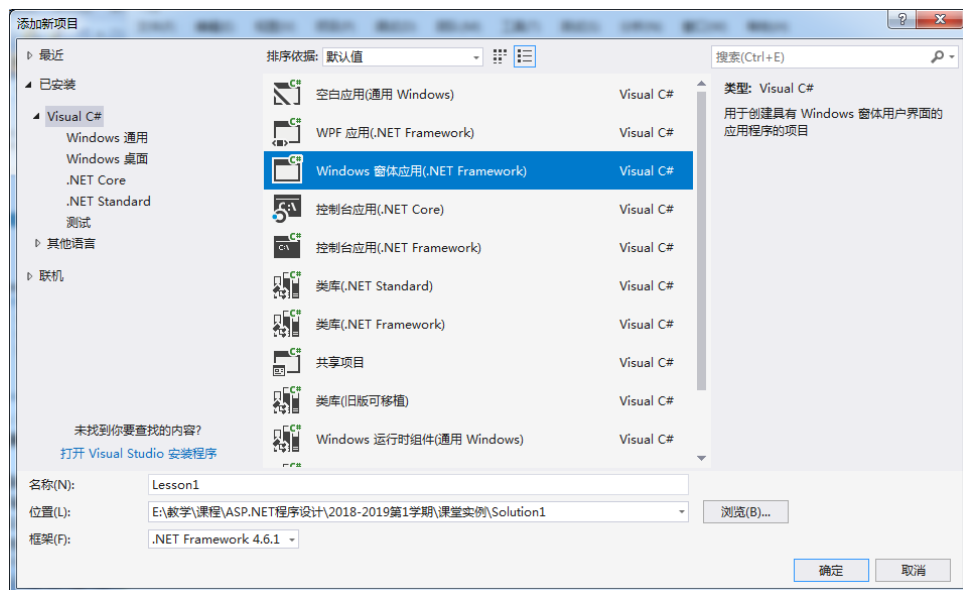
- 1.了解.NET 体系与 C#语言
- 2.掌握开发环境搭建
- 3.掌握控制台程序和 Windows 程序的编写
- 4.理解.NET 基本概述

一、基本操作（要熟练操作）

1、新建解决方案：文件——新建——项目——其他项目类型——Visual Studio 解决方案——空白解决方案——在下面位置和名称中修改自己想要存储的路径的解决方案名称，**注意名称不要使用中文**



2.新建项目：右击右侧“解决方案资源管理器”中的解决方案——添加——新建项目——Windows 窗体应用——在下面修改项目名称（**注：建议命名要有意义，例如 Lesson1**）



3.编写 Windows 应用程序

拖拽——右击——双击——F5

- (1) 从左侧“工具箱”中拖拽一个 Button 按钮到 Form1 中，即创建了一个按钮对象
- (2) 右击按钮对象——修改属性，Text: 问好（注：也可以修改其它相关属性）
- (3) 双击按钮对象，即跳转到代码页——编写代码 `MessageBox.Show(“你好！”);`

注意：其他代码是平台自动生成的，自动生成的代码不要修改。

```
namespace Lesson1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show(“你好!”);
        }
    }
}
```

注意：符号要使用英文，语句结束要使用分号

- (4) F5（或者按启动按钮），每点击一次按钮都会调用 button1_click 事件，弹出消息框。



4.编写控制台应用程序

显示行号（便于调试和排错）：点击“工具”→“选项”，选择“文本编辑器”→“C#”→勾选“行号”选项，可以在代码前面显示行号。（注：显示行号方便后期排错）

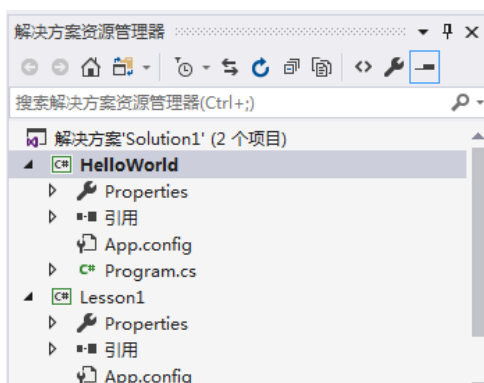
- (1) 右击解决方案——新建项目，选择控制台应用(.NET Framework)，命名为“HelloWorld”
- (2) 在主方法 Main()中编写程序，参考下图：

注意：//后面是注释

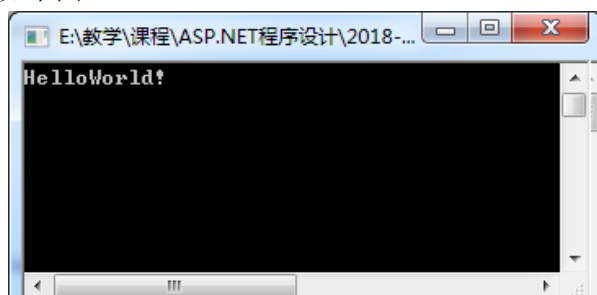
```
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("HelloWorld!")//用于向控制台输出字符串“HelloWorld!”
            Console.ReadLine(); //用于暂停程序，也可以按Ctrl+F5
        }
    }
}
```

(3) 在“解决方案资源管理器”中右击 HelloWorld 项目——设为启动项目

注意：设为启动项目的项目名称会“加粗”显示，如下图中的 HelloWorld 为启动项目。想要运行哪个项目，就将哪一个项目设为启动项目。



(4) F5 运行，效果如下图



二、.NET 基本概述（重点理解和掌握）

1. 解决方案与项目

(1) 解决方案是构成某个软件包（应用程序）的所有项目集。一个解决方案可以由几个项目共同组成（上课前打开解决方案，每次课会新建 1 个或多个项目，所有课上项目在一个解决方案中即可）。(2) 项目是一组要编译到单个程序集中的源文件和资源。

2. 命名空间

为了便于组织和管理，C#语言引入了命名空间的概念。命名空间相当于一个容器，包含一组定义的类或结构。命名空间也可以嵌套在另一个命名空间中，具有相同名称的类位于不同的命名空间。要调用类，可以使用以下两种方法，推荐使用第二种方法。

(1) 通过命名空间直接调用

命名空间. 类名（名实例名）. 方法名（参数）

具体示例：`System.Console.WriteLine("HelloWorld!");`

该语句调用了 System 命名空间下 Console 类的静态方法 `WriteLine()`，用来在控制台输出字符串“HelloWorld!”。

(2) 先引入命名空间后再调用（using 使用的意思），好处是以后代码可以简写

using 命名空间;

具体示例：`using System;`

`Console.WriteLine("HelloWorld!");`

该语句在程序的开头使用关键字 using 来引入命名空间，然后在需要时直接调用该命名空间下的 Console 类。

3. Main()方法

一个程序只能有一个 Main()方法入口。程序的功能是通过执行方法的代码来实现的，每个方法都是从第一行开始至最后一行，中间可以调用其他方法，以完成各种各样的操作。

每个应用程序都可能包含了多个方法，但入口方法只有一个。每个 C#控制台和 Windows 应用程序中，必须有一个类包含名为 **Main** 的静态方法。例如：

```
static void Main(string[] args)
{
}
```

Main 方法首字母必须大写。`string[]` 是声明一个字符串数组类型，`args` 就是这个数组的变量名称，也是该方法的参数。可以直接写 `svm`+两次 **Tab** 键，VS 会自动帮你生成代码

Main 的最简单形式如下：

```
static void Main()
{ }
```

4.代码注释

为源代码加上注释，不但可以提高程序的可读性，还便于后期维护。C#语言中常规的注释方法有以下 3 种。

（1）单行注释

单行注释通常用于对程序中的某一行代码进行解释，用符号“//”表示，后面为被注释的内容，具体示例如下。

```
Console.WriteLine("HelloWorld!"); //用于向控制台输出字符串“HelloWorld!”
```

（2）多行注释

多行注释就是注释中的内容可以为多行，它以符号“/*”开头，以符号“*/”结尾，具体示例如下。

```
/* 用于向控制台输出字符串“HelloWorld!”
   用于暂停程序 */
```

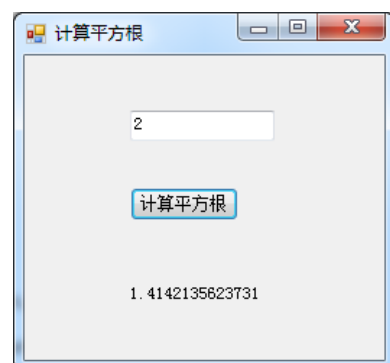
（3）文档注释

文档注释用于对类或方法进行说明和描述。在类或方法前面连续输入 3 个“/”，就会自动生成相应的文档注释，用户需要手动填写类或方法的描述信息，来完成文档注释的内容，具体示例如下。

```
/// <summary>
/// 控制台输出字符串“HelloWorld!”
/// </summary>
```

三、课后练习：计算平方根

- 1.在前面已建的 Windows 项目中新建一个 Windows 窗体；
- 2.在窗体中添加 3 个控件：TextBox（文本框）、Button（按钮）和 Label（标签）
- 3.为 Button 对象添加点击事件代码，实现从 TextBox 中获得用户输入的数字，计算数字的平方根，然后在 Label 中输出结果。效果如下图：



四、阅读教材，了解.NET 体系与 C#语言 思考：如何运行一个项目中的不同窗体??