

使用Visual Studio编写C#程序

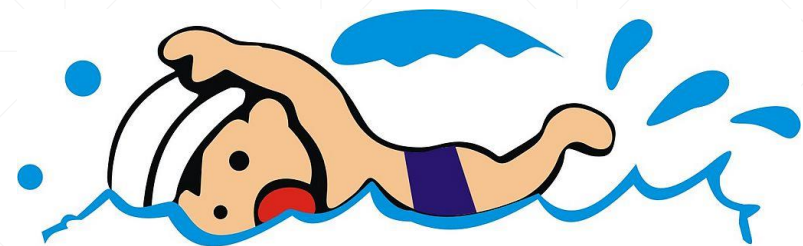
北京理工大学计算机学院
金旭亮

掌握Visual Studio是学习本课程的前提

要学会游泳，最重要的是你必须跳到水里去游！

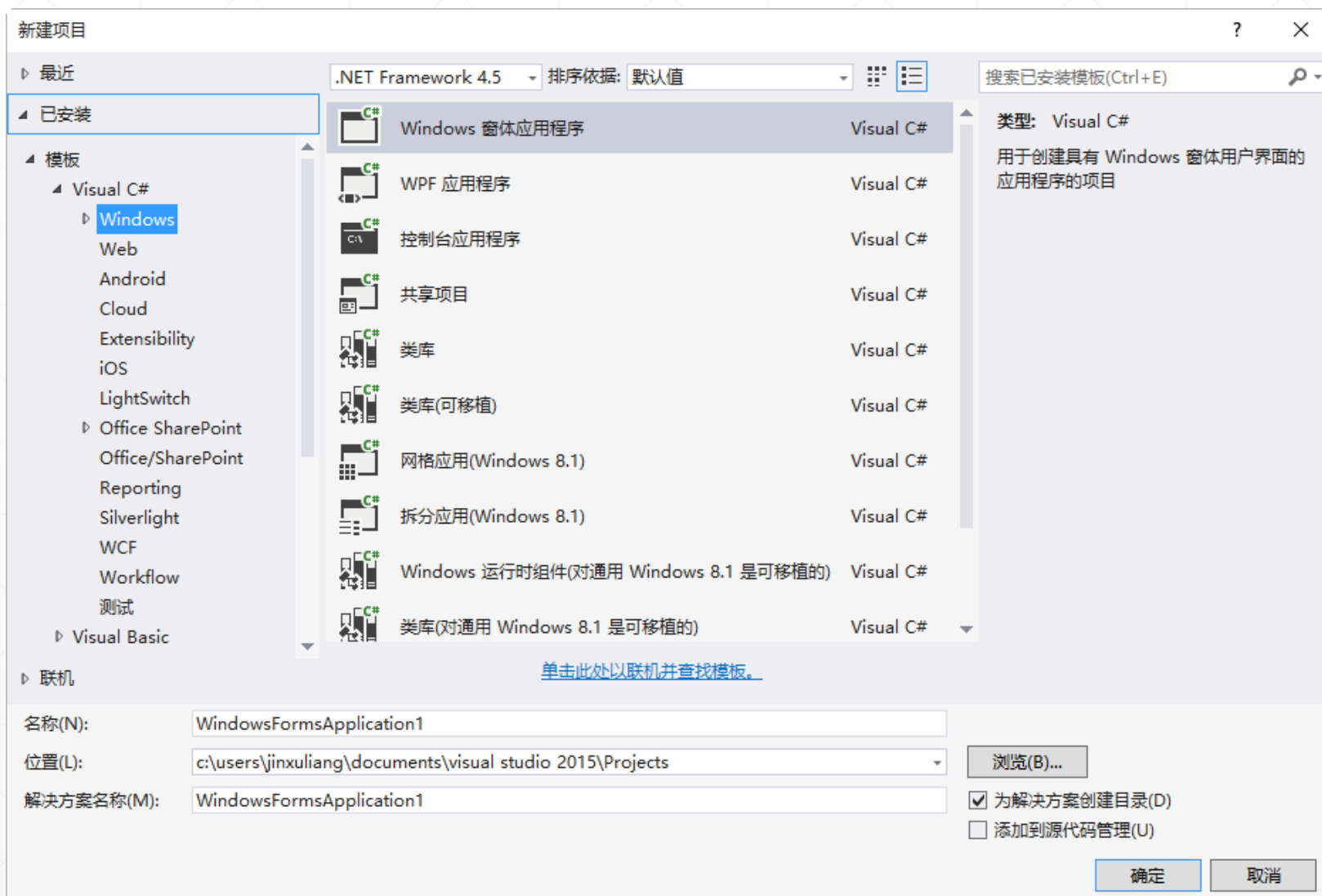
要想学会编程，你必须亲自动手编写代码！

要编写代码，你需要一个顺手的工具.....



Visual Studio，是一个功能强大易用的软件集成开发环境，它将成为我们编写代码的主要工具.....

使用Visual Studio可以创建多种类型的项目，每种项目各有各的应用场景.....



本课程学习过程中用到的项目类型



Windows窗体应用程序

- 带窗体的程序，编译之后，生成扩展名为“.exe”可执行程序文件，能直接在资源管理器中双击启动运行。很多我们日常使用的PC软件可以归属于此类。



控制台应用程序

- 生成的程序运行于黑底白字的命令提示符窗口，多用于学习，在真实软件项目中用得不算多。

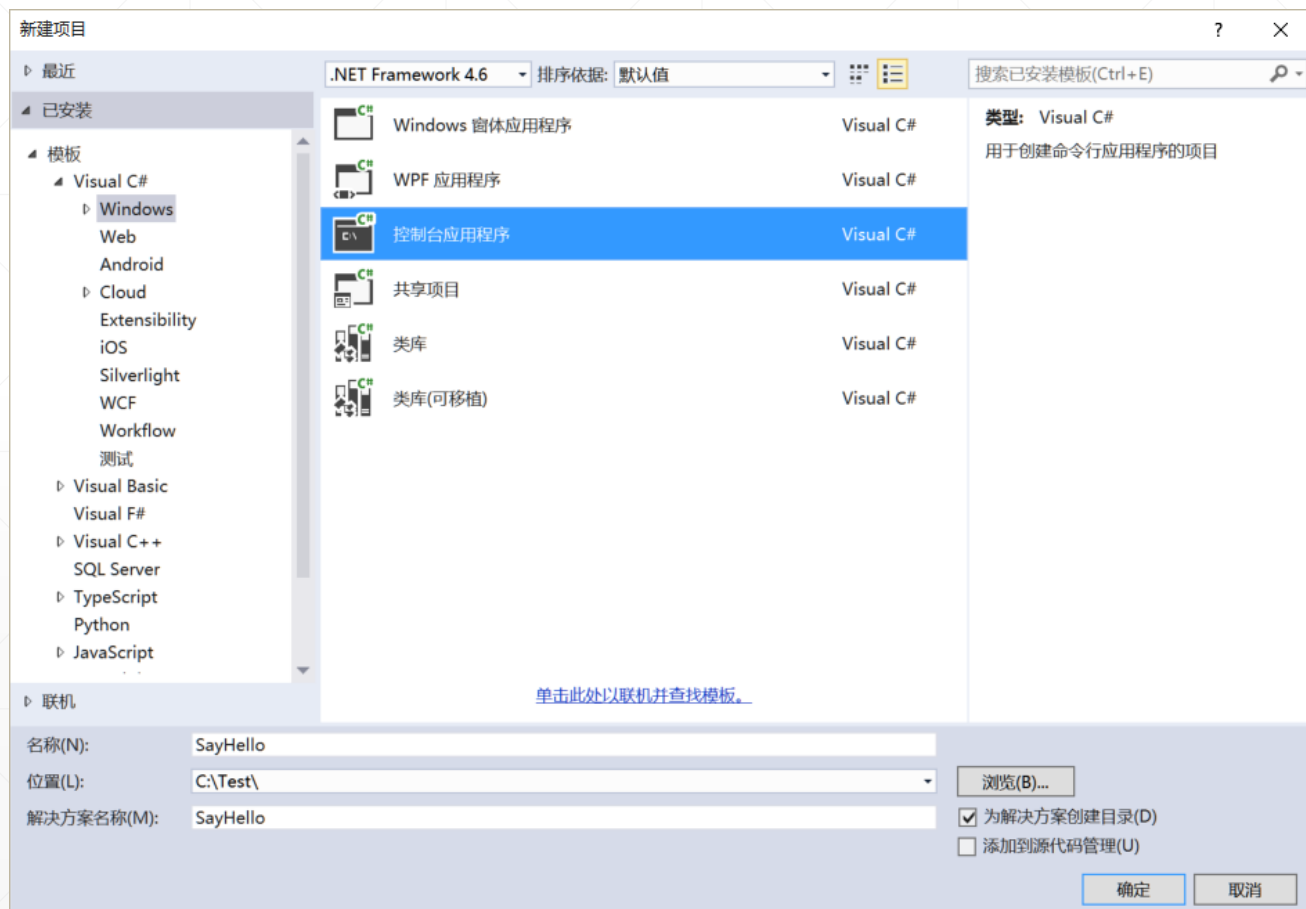


类库

- 编译之后，生成一种扩展名为“.dll”的文件，主要用于提供搭建“软件大厦”的“砖块”，在实际开发中用得很多。

我们先从最简单的“Console Application（控制台应用程序）”开始……

菜单命令：文件/新建/项目…



动手编写第一个示例

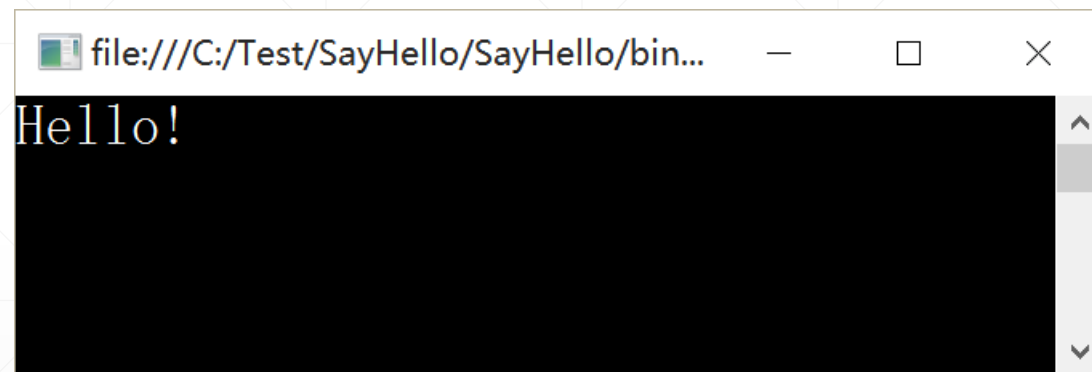
输入以下代码：

```
using System;

namespace SayHello
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello!");
            Console.ReadKey(true);
        }
    }
}
```

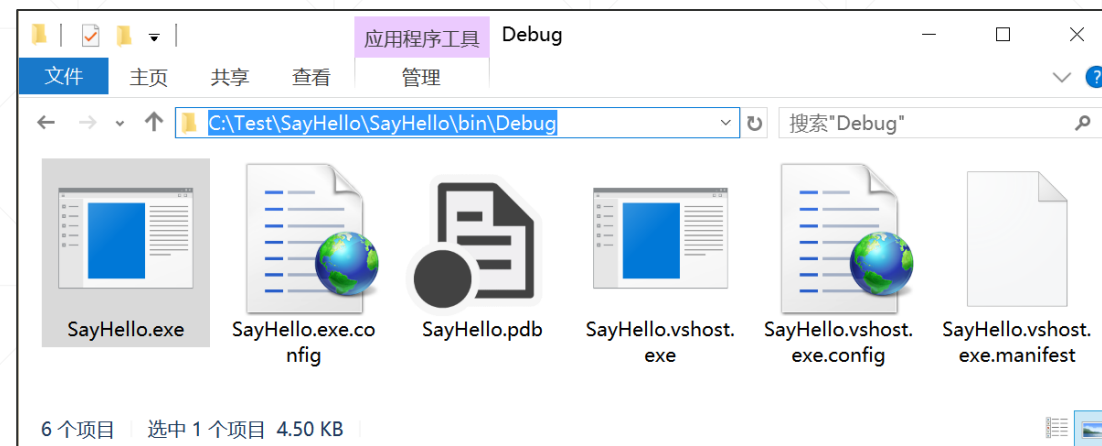
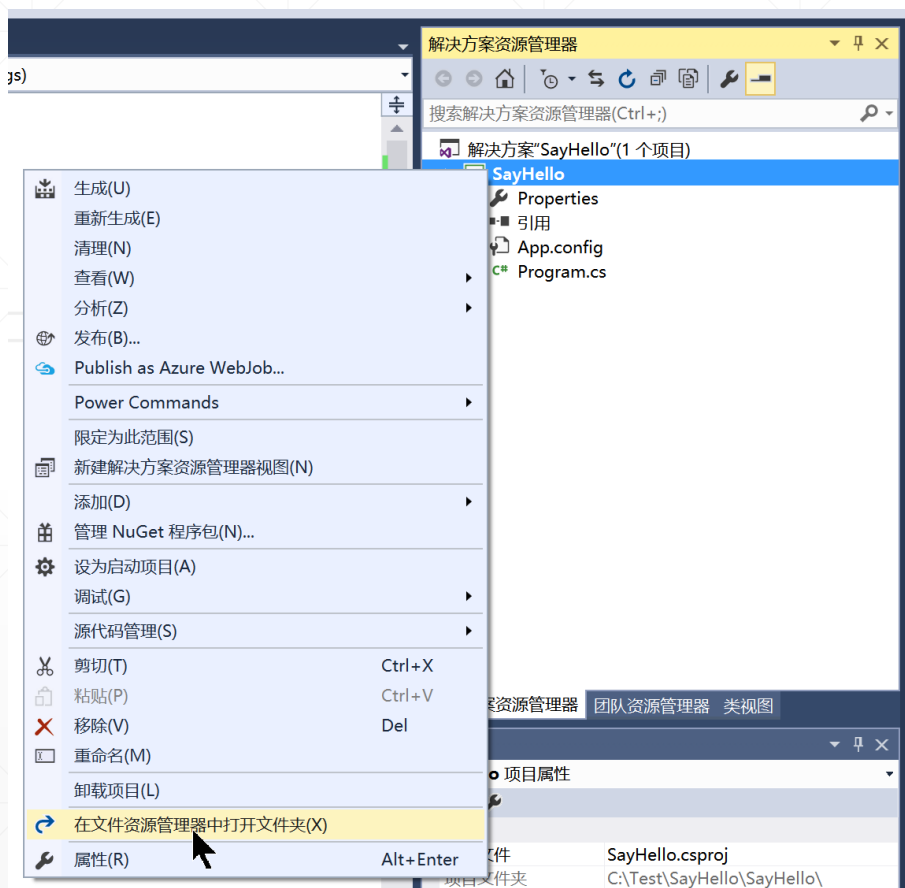
- * Console是一个类，代表控制台窗口
- * WriteLine是Console类的方法，用于输出字符串
- * ReadKey是Console类的另一个方法，用于接收按键

单击工具栏上的绿色小三角编译并运行程序



黑底白字的命令提示符窗口，称为“**控制台 (Console)**”

生成EXE文件在哪儿？



在项目文件夹的Debug目录下，可以找到"项目名称.exe"文件，双击即可运行它。

可以随意地将其复制，只要目标计算机安装有相应版本的.NET，程序就能顺利地运行.....

Console类使用指南

Console是.NET所提供的的一个类，它代表控制台窗口

Console类中定义了一些“属性”，可以用于设定控制台窗口的一些特性：

属性名	说明
Title	控制台窗口的标题
ForegroundColor	控制台文字的前景色
BackgroundColor	控制台文字的背景色

学习指南——小步快跑

请在视频的指导下，在自己的计算机上跟随老师，把整个示例的开发过程走一遍，以形成对编程的感性认识。

Console类的常用方法：

Console类还提供了一些方法，这些方法能实现特定的功能

- 它的ReadLine()方法可用于读取用户输入：

```
//等待用户从键盘输入信息，敲回车后，  
//用户输入的信息保存在userInput变量中  
String userInput = Console.ReadLine();
```

- 它的WriteLine()/Write()方法用于输出，最后自动换行：

```
//WriteLine()方法输出变量的内容，自动换行  
//Write()方法则不会输出  
Console.WriteLine(userinput);
```

格式化输出

可以使用占位符（如“{0}”）来格式化输出，还可以使用“\n”（换行），“\t”（跳格）之类的特殊标记控制输出的格式：

```
Console.Write("请输入一个字符串:");  
String userInput = Console.ReadLine();  
Console.WriteLine(  
    "\t用户输入了: {0}, 共有 {1} 个字符\n", userInput, userInput.Length);
```

动手动脑

请上互联网搜索资料，看看除了“\n”和“\t”之外，还可以找到哪些特殊标记。

- 暂停输出，等待用户击键：

Console.ReadKey(true);



当给Readkey()方法传入一个true参数时，程序将停下来等待用户击键，但不会在屏幕上显示用户敲入的键。

- 让电脑发声，通知用户某事发生了……

Console.Beep();

学习提示

到目前为止，PPT和视频中都介绍了一些典型的功能和典型的代码，请注意积累这些代码段（比如可以创建一个Word文档、建立专门的文件夹），以便在需要时可以快速地找到它们。

优秀的程序员，通常都有自己收集与整理各种技术资料的习惯，许多人还建立了自己的代码仓库，由于平时就把相关资料整理得井井有条，所以开发效率就比较高。

优秀的程序员，比之差的程序员，其工作绩效可能要高十倍甚至更多！

亲自动手试一试.....

请准备两到三首古诗依次显示在控制台窗口中。具体要求如下：

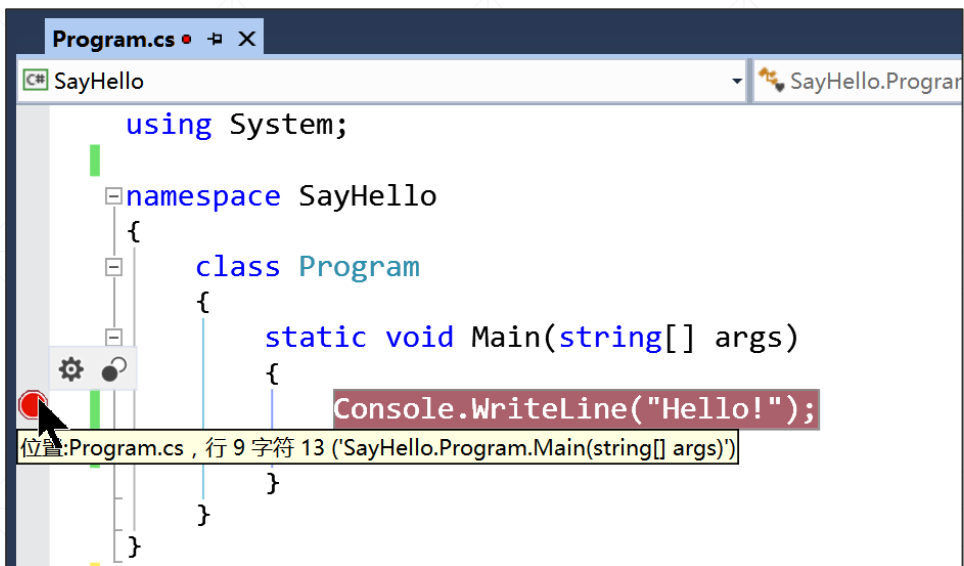
程序运行时，先显示必要的提示信息，然后等待用户击键。

用户每击一次键，程序在控制台窗口中显示一首古诗内容，同时在控制台窗口标题上显示古诗的名字。

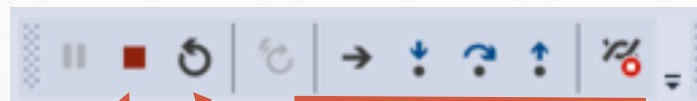
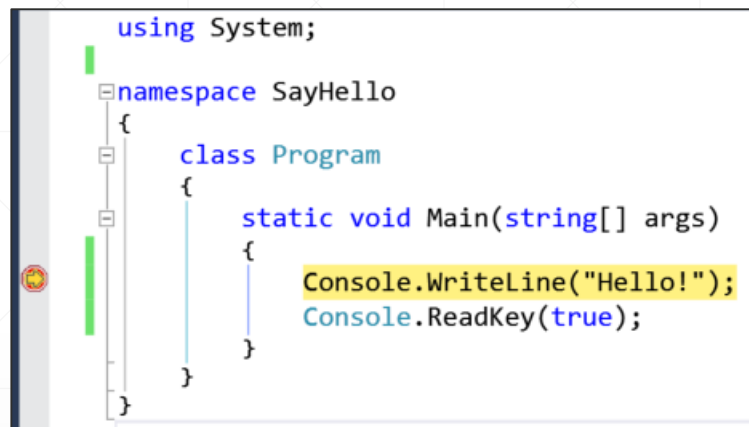
你可以选择合适的前景色、背景色，在合适的地方换行，对齐文字，让你的程序的输出比较美观。

掌握最基本的调试功能

程序运行时在“断点”处暂停



在代码左边点击，可以设置并配置“断点”



停止
程序

重新启动
程序

用于程序调试：
下一语句，逐语
句，逐过程.....

学习指南

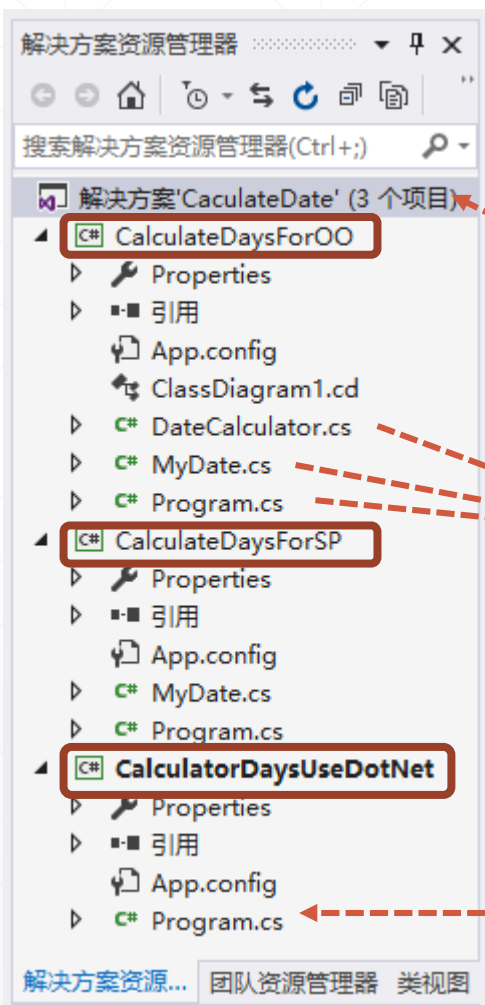
有过开发经验的人都有这个体会：其实只要你想好想清楚了，写代码并不难，真正有挑战性的往往是程序中出现了预料之外的错误，你需要找出原因进行更正。这个查找错误并更正的过程，称为“调试（Debug）”，很能体现一个程序员功底与水平。

Visual Studio支持相当强大的调试功能，在视频中仅仅只是介绍了皮毛，尽管如此，仅仅掌握这些，你也已经可以调试一些简单的应用了，在初学编程阶段，这几招基本上够用。

大致了解C#基本编程规则

- **“标识符**（ identifier ：即程序中拥有特殊含义的单词 ）**”区分大小写**
- **每条语句以分号结尾，多条语句可以使用“{”和“}”组合为一个语句块。**
- **编写的代码放在“类（ class ）”中，类是C#编程的基本单元**
- **存放C#类的源代码文件其扩展名为“.cs”，一个“.cs”文件中可以存放多个C#类。**

Visual Studio中的文件组织方式



“项目 (Project)” 是Visual Studio编译的最小单元。

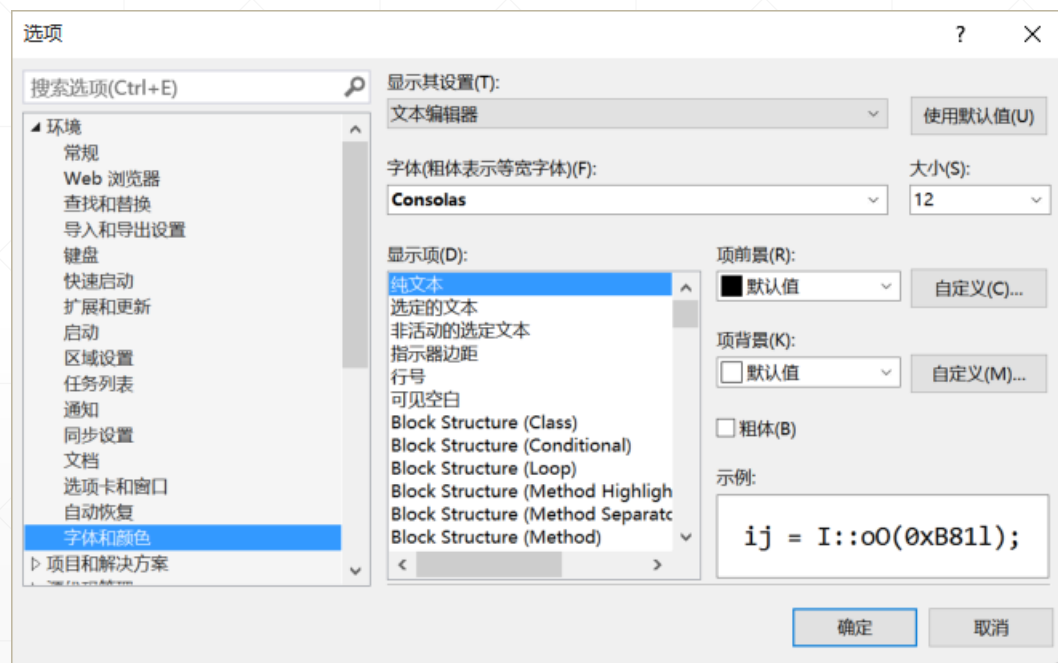
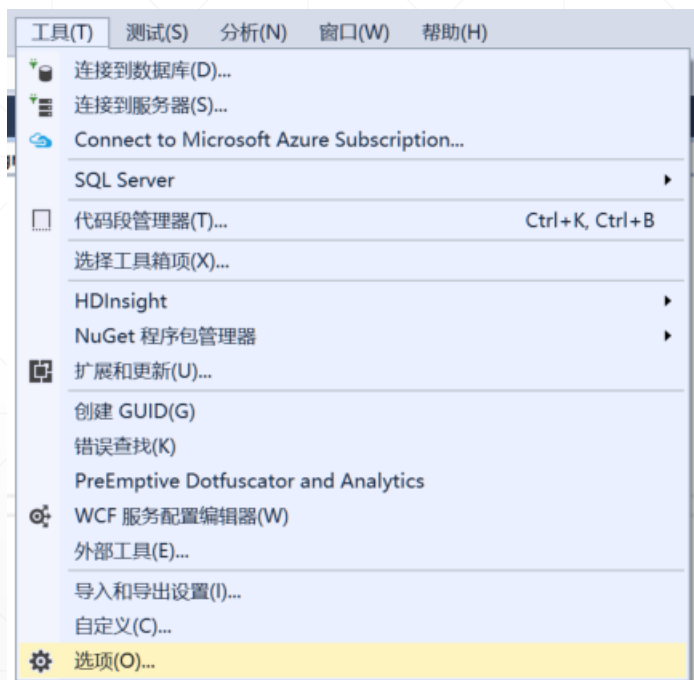
一个“解决方案 (solution)” 中可以有多个项目，显示为粗体的是启动项目。

一个项目中可以有多个类文件。一个类只有被包括在一个项目中，才可以被Visual Studio编译。

程序的入口点是Program类中的**Main()**方法

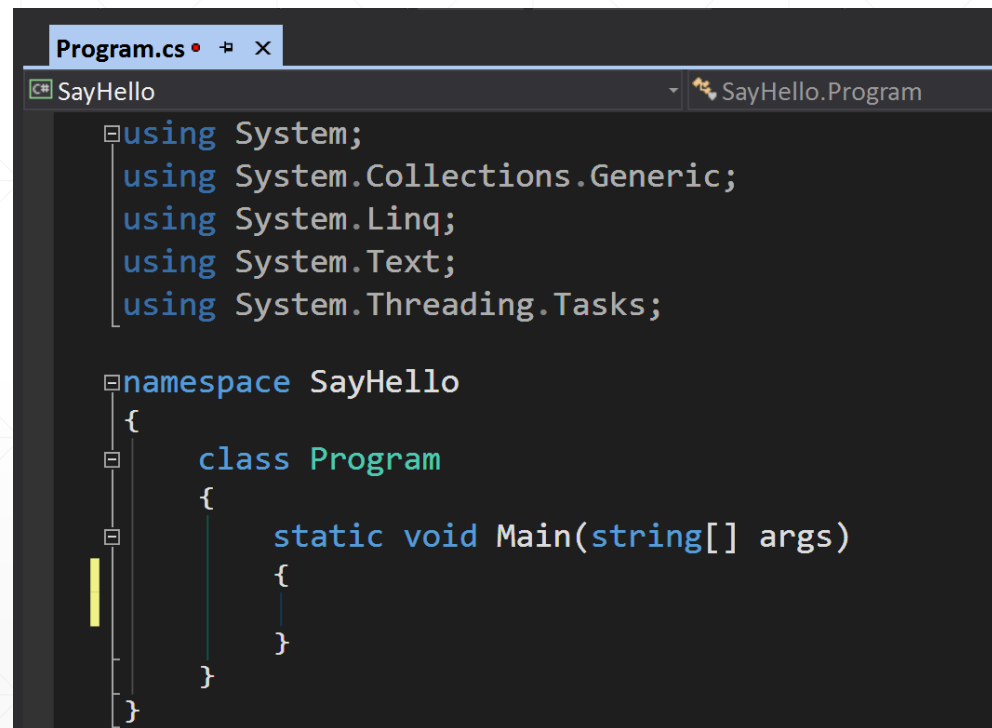
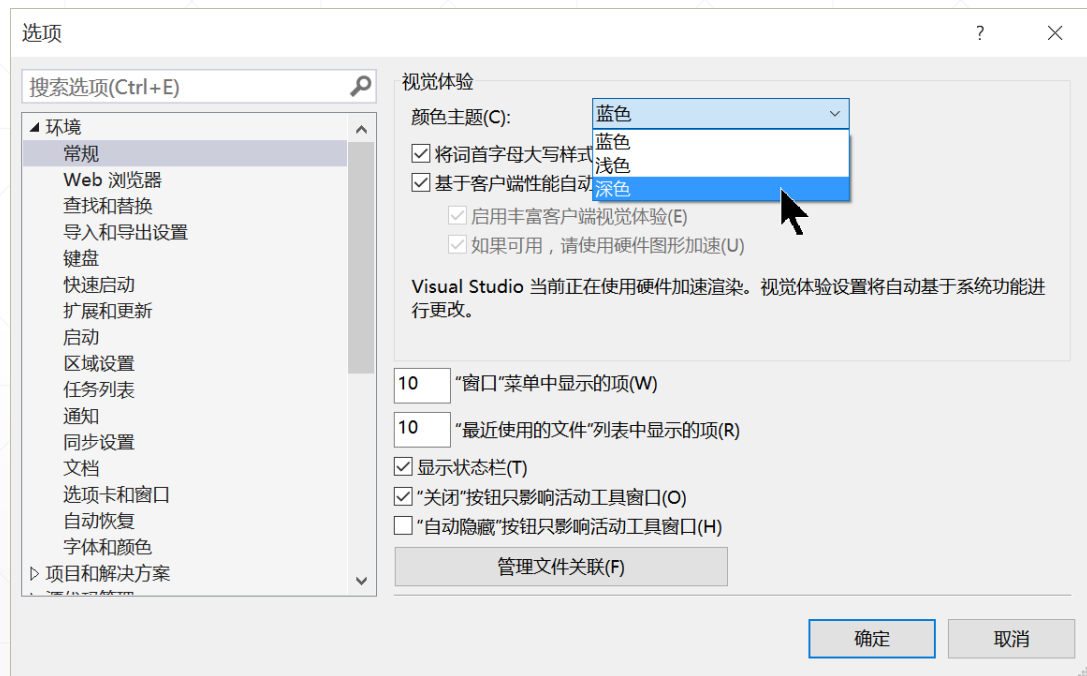
补充材料：让你的开发环境更舒适

调整字体与大小，让代码美观.....



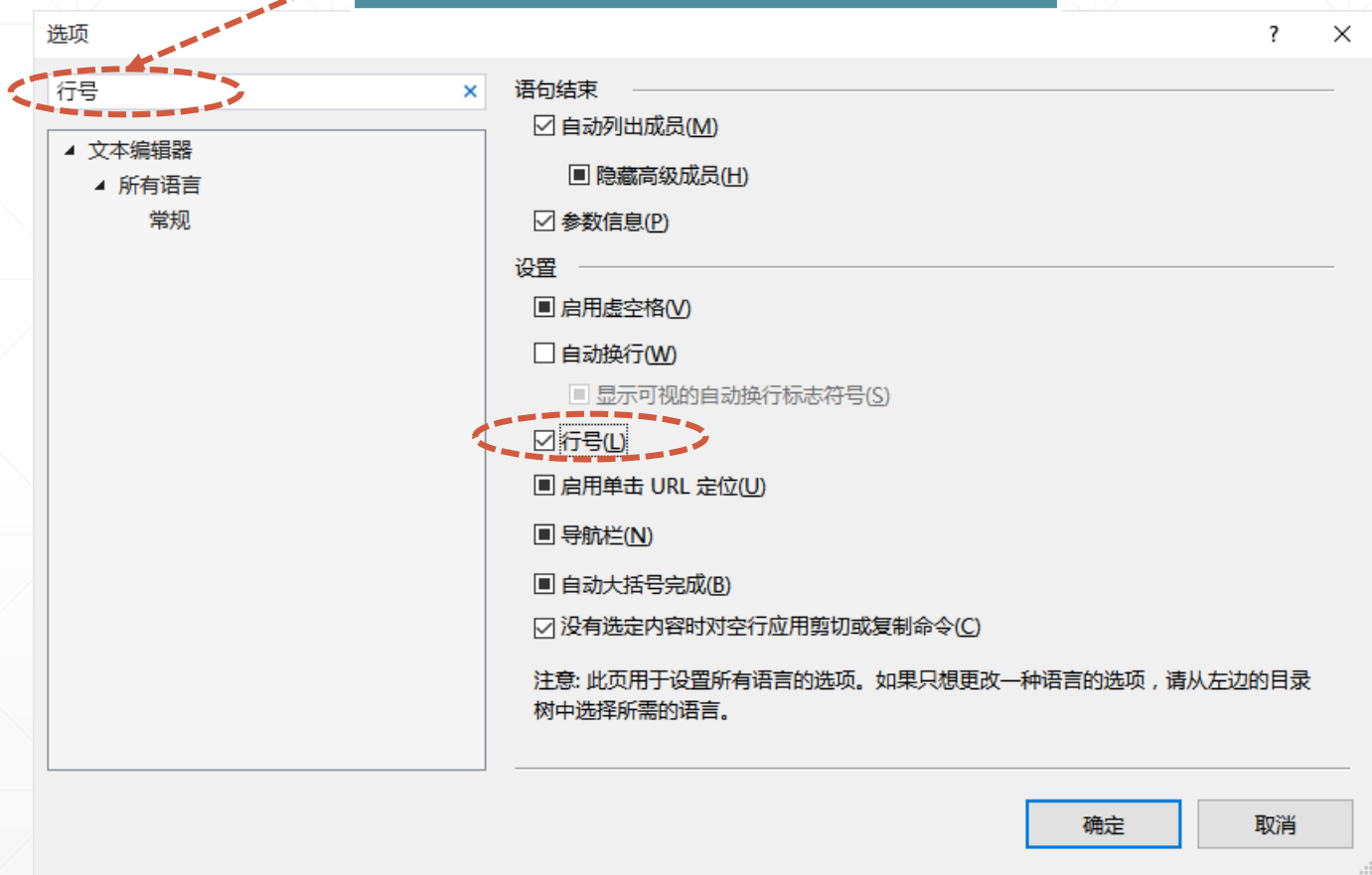
补充材料：让你的开发环境更舒适

启用“黑底白字”，保护眼睛.....

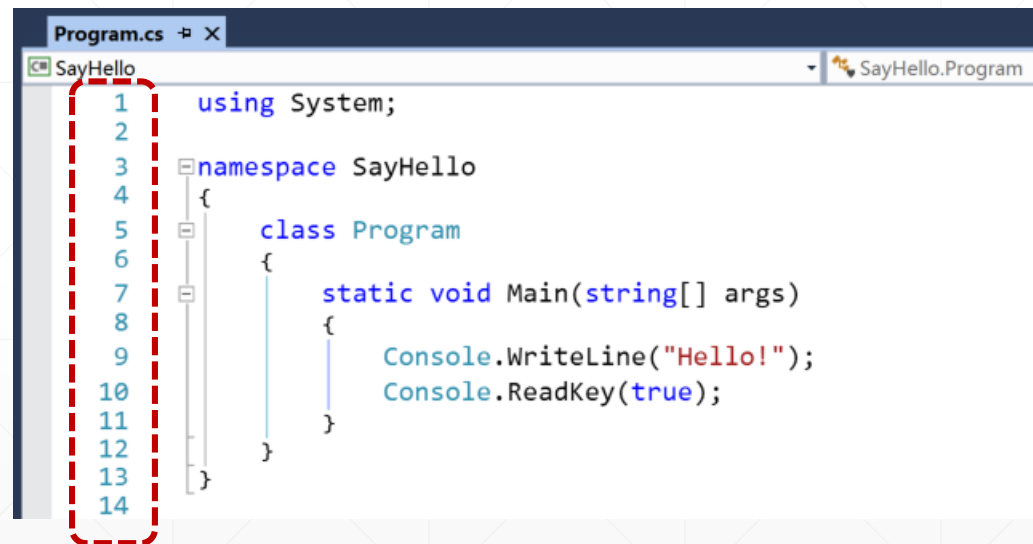


添加行号

在这里输入“行号”进行搜索



给代码添加“行号”



Visual Studio的使用还有许多技巧，我们将在后面的课程中逐步介绍，你也可以通过互联网收集一些有关Visual Studio使用技巧的资料主动学习。

对开发工具越熟悉，就越能发挥它的威力，就越有可能高效快速地完成特定的软件开发任务。

课后作业

- (1) 跟着视频的演示，将本讲老师所敲的代码和示例，自己重新敲一遍。**编程从模仿开始!**
- (2) 应用本讲所学的知识，你能否在控制台窗口中输出以下这个由星号组成的简单图形?

```
*****  
  *****  
    *****  
  *****  
*****
```

学了不用，等于白学!