

前言

一、上机实验目的

上机实验的目的是提高学生的分析问题、解决问题的能力和动手能力，通过实践环节理解C#语言的基本结构和程序设计方法。通过亲手编程掌握C# 语言编程的方法。

二、实验基本内容

为了使学生在上机实验时目标明确，本实验指导书针对课程内容编写了10 个实验(不全，还有很多知识没包括)。学生可以在课内机时先完成指导书中给出的程序，理解所学的知识，在此基础上再编写其他应用程序。指导书中的8 个实验如下：

1. C# 程序运行环境的下载、安装与配置及C# 基本语法编程练习。
2. 类与对象编程练习
3. 继承与多态编程练习。
4. 操作符重载编程练习。
5. 结构体与接口编程练习。
6. 数组、索引器、集合编程练习。
7. 字符串、异常处理编程练习。
8. 委托与事件编程练习。

三、实验任务与时间安排

《C# 语言》是一门实践性很强的课程，除了在课内安排的实验外，鼓励同学在课外用相关技术进行编程练习。《C# 语言》上机实验时间为16学时，与教学内容同时进行上机实验。16个上机学时分为8个实验。

四、实验环境

Internet, Windows 7/8/10, Visual Studio 2005/2008/2010/2012/2013/2015

实验要求

一、可读性

编程高手都知道：一个好的程序要具备可读性，可方便自己也可方便别人。所以，要培养一个良好的编程习惯，可注意以下几方面：

1. 代码的缩进；
2. 有效使用空格；
3. 简明的注释；
4. 意义明确的命名；
5. 着重表示的常量。

二、亲手编写源程序

在编写C# 程序过程中，还可以利用一些可视化的开发工具，它们可以综合使用C# 的编译器和调试器等，例如Symantec 公司的Visual Café、Kawa、Sun 公司的C# Worshop 和C# Studio、微软的Visual Studio等，使用编程开发工具可以加快编程的速度。但在初始学习时还是最好亲手编写源程序，以便理解类和编程思想。

特别提示：本实验中的代码均为课本作者或教师提供的参考代码，上机过程中为了同样的实验目的大家可以应用不同的代码内容灵活实现。

来自CSDN的top programmer告诫：

- 不要看到别人的回复第一句话就说：给个代码吧！你应该想想为什么。当你自己想出来再参考别人的提示，你就知道自己和别人思路的差异；
- 请把书上的例子亲手敲到电脑上实践，即使配套光盘中有源文件，copy的代码和亲手敲出的代码永远从本质上不一样；
- 把在书中看到的有意义的例子扩充；并将其切实的运用到自己的工作中；
- 不要漏掉书中任何一个练习——请全部做完并记录下思路；
- 写程序不易，刚开始的孤独就像苦行的僧人，不过这种种值得恭喜的感觉，因为在此之后才可以成佛。

三、立即运行程序

编写的源程序要立即上机编译运行来检验程序中存在的问题。通过运行的结果验证程序的功能是否实现。即时有一些系统类的方法、变量也需要上机实验去了解它们的含义。

C#具有强大的应用能力和范围，在各类企业信息系统开发，电子商务网站开发甚至移动商务软件开发方面具有独特的优势。她可以高效地实现Windows窗体程序、Web项目和Web service项目的软件开发。如果你认真地投入到此实验全部过程的练习和思考，那么你将深刻理解C#这一面向对象程序设计语言并能学会开发出类似于俄罗斯方块、纸牌游戏、记事本、销售管理系统、网上商店、论坛等种种软件项目的过程和思路。

实验一(1) C# 程序运行环境的下载、安装与配置

【实验目的及要求】

1. 掌握 VISUAL STUDIO 2015 的下载与安装
2. 掌握环境变量设置方法
3. 学习使用记事本或 Visual Studio 2015 编辑 C#文件
4. 练习用 CSC 命令或 Visual Studio 2015 编译运行示例 C#源程序
5. 完成实验报告

【实验内容】

一. VISUAL STUDIO 2015 下载与安装

注意：以 visual studio 2005 为例，visual studio 2015 类似，可以百度。

1. VISUAL STUDIO 2005 的版本
2. 机器要求
3. 下载 VISUAL STUDIO 2005
4. 安装 VISUAL STUDIO 2005
5. VISUAL STUDIO 2005 中主要文件夹和文件

1. VISUAL STUDIO 2005 的版本和选择

Visual Studio 2005 是一个微软开发工具的套装，它包含了微软的开发工具，如 Visual Basic, C#, Visual C++, J# 等。

Visual Studio 有四个版本，按功能多少依次是：

- Visual Studio Team Edition
- Visual Studio Professional Edition
- Visual Studio Standard Edition
- Visual Studio Express Edition

(1) Visual Studio 2005 Express Edition

在 Visual Studio 2005 中，Express 系列版本是最低一级的版本，这些版本针对具体的语言，如 Visual C++ Express Edition, Visual C# Express Edition 等，Express 版本非常适合第一次接触软件和网站开发的人。

(2) Visual Studio 2005 Standard Edition

Standard 版是相应 Express 的更高一级的版本，拥有更多的特性（如类设计工具等），这些 Standard 版的组合就构成了 Visual Studio 2005 Standard Edition.

(3) Visual Studio 2005 Professional Edition

Visual Studio 2005 Professional Edition 除拥有 Standard Edition 的所有功能外,还具有一些 Enterprise Architect Developer 的大部分功能.

(4) Visual Studio 2005 Team Suite

包含 Professional Edition 的所有功能, 这是 Visual Studio 2005 实现团队开发的重要核心。通过高度整合的核心服务, Team Foundation Server 提供版本控制(version control), 工作项目追踪(work item tracking), 和自动构建(build automation)。此外, Team Foundation Server 整合了 Windows SharePoint Services 来提供门户网站和 SQL Server Reporting Services 提供专业的报表, 为团队开发和沟通提供了最佳的解决方案。

还包括如下三个版本:

- ◆ Team Edition for Software Architects: 该版本提供了许多 Microsoft 之前未商业化的功能(例如单元测试和代码分析)或是之前没有被整合到 Visual Studio IDE 的功能(例如 profiling 和 Managed 程序代码分析)。
- ◆ Team Edition for Software Developers: 该版本提供得分布式系统设计工具(Distributed System Designers)属于 Microsoft 进行中的一项大型计划, Dynamic Systems Initiative。这些设计工具和相应的工具代表的是第一波用来辅助设计和部署 service-oriented 应用程序的工具。
- ◆ Team Edition for Software Testers: 该版本是供专业软件测试工程师使用的, 包含了管理测试的功能、执行负载和压力测试、管理负载测试代理人、和整合 Virtual Server 2005 达到机器虚拟化。为了增进负载测试的能力, 测试经理可以通过购买 Visual Studio 2005 Team Test Load Agent 增加更多的负载测试代理人。

上述的三个版本中均包含有 Visual Studio 2005 Tools for the Microsoft Office System

选择: 对于一般用户来说, 安装 Visual Studio Express Edition 或者 Visual Studio Standard Edition 就足够了, 本课程实验在机房安装 Visual Studio Team Suit, 以下所实验的内容和演示也均为此环境下的结果。

2. 机器要求

(1) 下载要求:

参照各下载源要求, 下载文件一般 2G 多, 因此要准备好足够的下载空间。

(2) 安装要求:

- ◆ 安装支持的操作系统: Windows 2000 Service Pack 4; Windows Server 2003 Service Pack 1; Windows Vista; Windows XP Service Pack 2; Windows 7
- ◆ 处理器最低要求: 600 兆赫 (MHz) Pentium 处理器。推荐: 建议采用 1 千兆赫 (GHz) Pentium 处理器
- ◆ RAM 最低要求: 192 兆字节 (MB), 推荐: 256 MB
- ◆ 硬盘要求:

不安装 MSDN：安装驱动器上要有 2 GB 可用空间，系统驱动器上要有 1 GB 可用空间；
安装 MSDN：在完全安装 MSDN 的安装驱动器上要有 3.8 GB 的可用空间，在进行默认 MSDN 安装的安装驱动器上要有 2.8 GB 的可用空间。 系统驱动器上要有 1 GB 可用空间
CD 或 DVD 驱动器 没有要求；

- ◆ 显示器 最低要求：800 x 600 256 色，推荐：1024 x 768 增强色 – 16 位
- ◆ 鼠标： Microsoft 鼠标或兼容的指针设备

3. 下载源

- 电驴
- 迅雷
- 微软官方网站试用版本
- 其它搜索到的下载网站源

4. 安装

我是依照 Visual Studio Team Edition 的安装过程来讲解和截屏的，如果你手里有的是 Visual Studio 的其它版本，也别担心，因为无论是 Visual Studio 的哪个版本，安装过程几乎都是一样的，最多是在安装功能选项中，有些版本功能多，有些版本功能少而已。插入 Visual Studio 的安装光盘，你会看到如下画面。请点击最上面的链接 Install Visual Studio 2005。



然后你会进入下面的画面，这时候安装程序会运行一段时间，运行完以后，右下角的 Next 按钮就可以点击了。



点击 Next 按钮之后，你会被要求填写 product key，请将 I accept the terms of the License Agreement 勾上，然后填上你的 product key。至于 Name 栏，一般来说是填写该软件的购买者。所有填完之后再按 Next。

接下来出现的如下画面，让你可以选择你所要装的功能。如果你对 Visual Studio 还不是很熟，就直接勾 Default，Default 表示 Visual Studio 会自动帮你安装推荐的功能。如果你觉得还不放心，你可以选择 Full，Full 表示 Visual Studio 会安装所有的功能，一个不落。



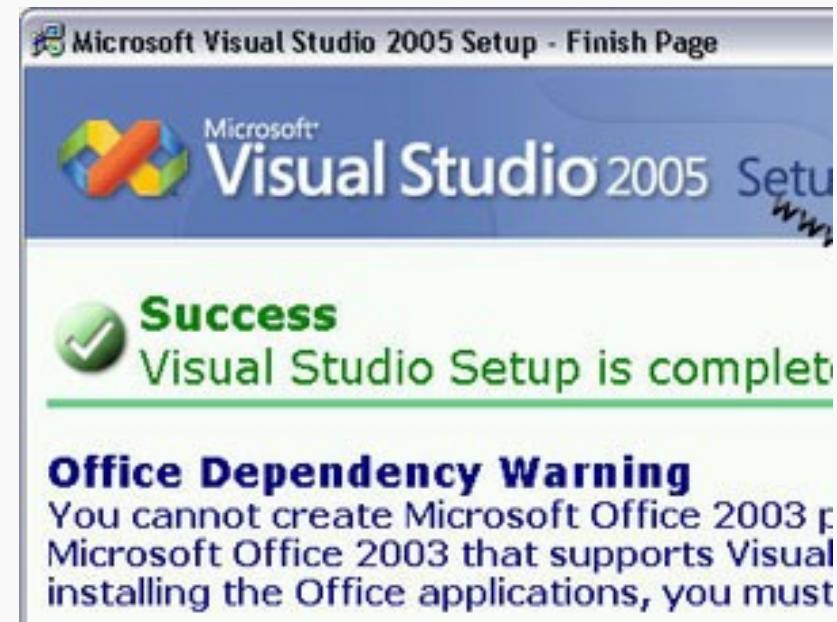
如果你是更加高级的用户，知道什么功能要，什么功能不要，你可以选择 Custom。选择 Custom 之后，你会看到如下画面，该画面列出了所有该版本 Visual Studio 的功能，打勾的表示要安装，打叉的表示不要安装。

我用的这个截图是 Visual Studio Team Edition，如果你用的是 Visual Studio 其它版本，你所看到的功能列表可能比这张图上的要少一些。



选择好你要装的功能之后，就开始安装了，你就等吧。这时候瘦人不妨去倒个茶吃个小点心，胖子则可以拿上哑铃举举，或者做个俯卧撑，仰卧起坐什么的。不胖不瘦的人也别傻盯着屏幕看安装进度，尽情望望野眼，为革命保护视力。

当屏幕出现如下画面，其中画面上方会显示 Success, Visual Studio Setup is completed 绿色字样，表示已经成功安装完毕。

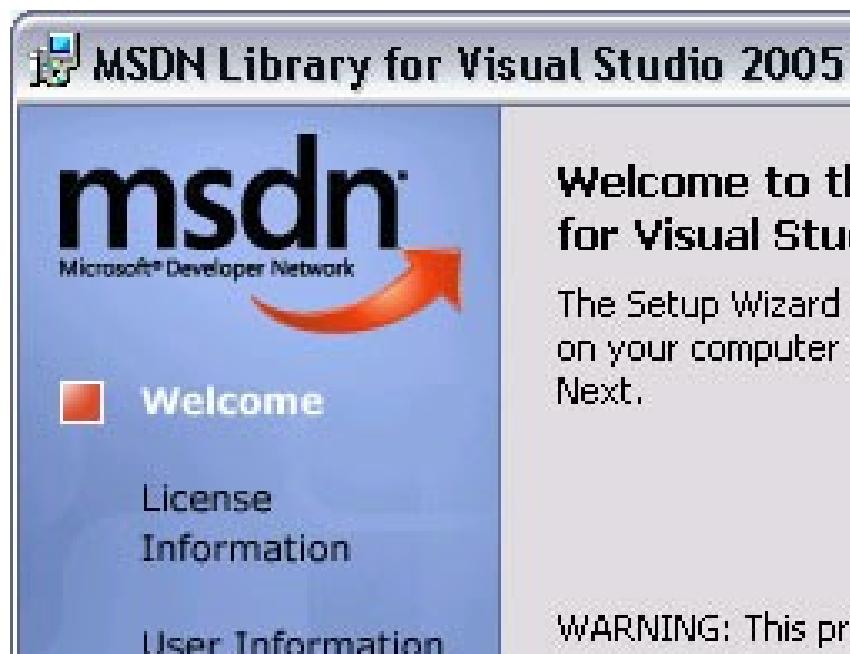


至于截图中的 Office Dependency Warning 是因为选择安装了 Visual Studio 有关 Office 工具的功能，但是机器上又没有安装 Office 造成的。系统就提示你，如果你要用 Visual Studio 中有关 Office 工具，要安装 Office 及其 Service Pack 。截图中的这条提示，在你的安装中可能根本就不会出现，忽略它就是了。

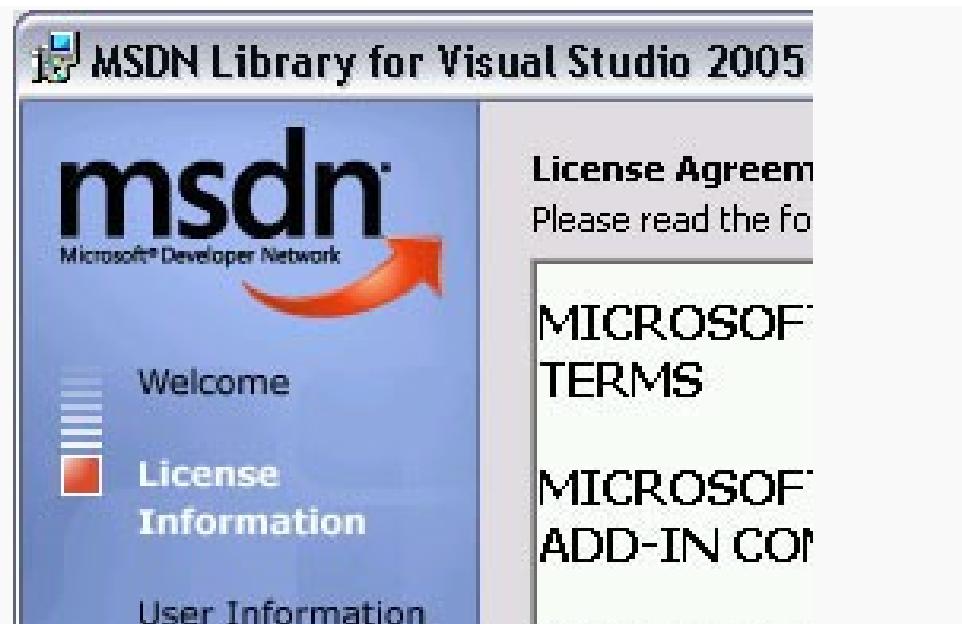
然后点击画面右下角的 Finish 按钮。点击了 Finish 按钮，安装程序会自动回到安装初始画面。这时候，画面提示你安装 MSDN，MSDN 就是微软的技术文档库。对于使用微软开发工具的人来说，没有 MSDN 就等于砍掉 6 个指头让他打键盘。



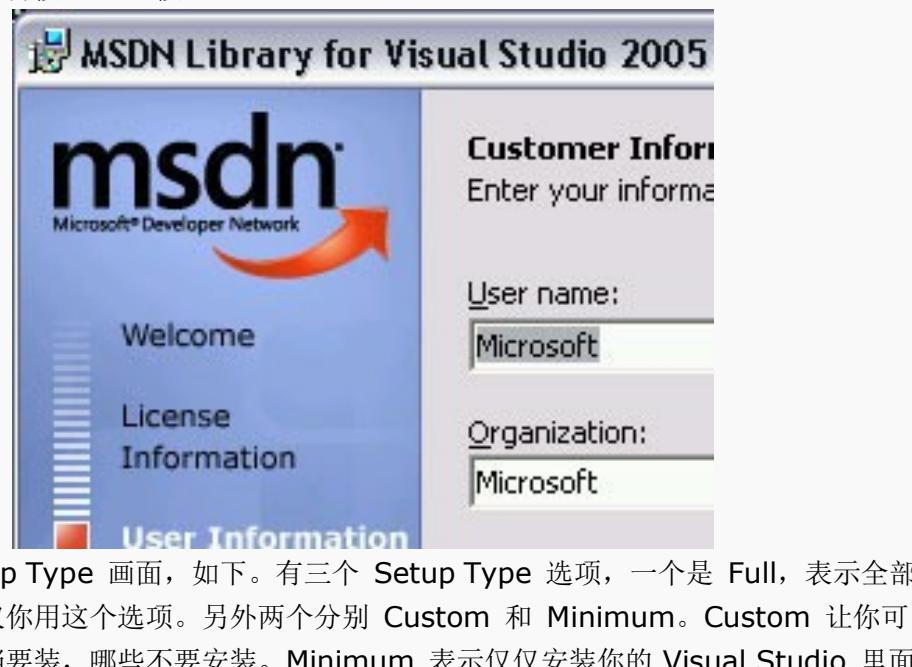
点击 Install Product Documentation, 会出现如下画面, 按 Next 按钮。



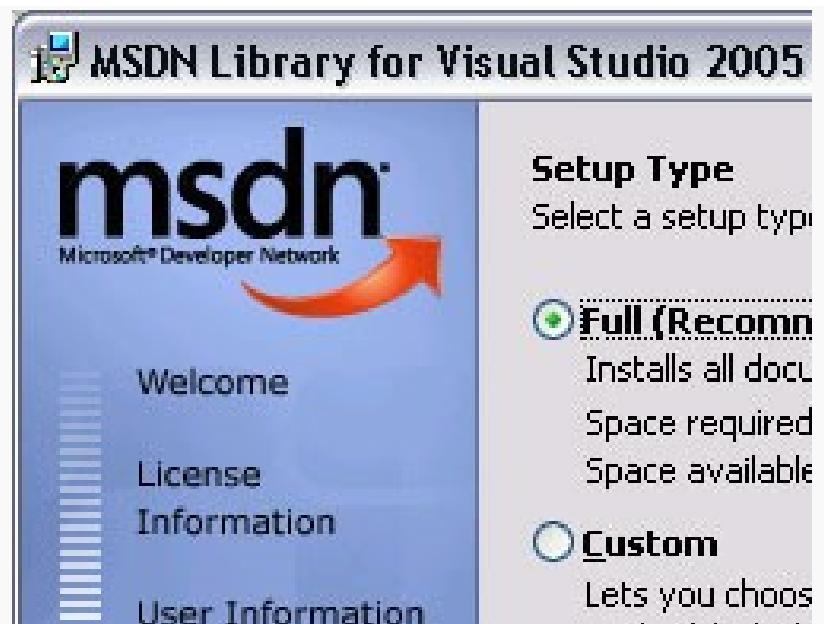
然后会出现如下画面, 请勾上 I accept the terms in the license agreement.



然后填上你的 User Name 信息和 Organization 信息，通常是填写该软件的购买者信息。填完之后，再按 Next 按钮。



然后出现 Setup Type 画面，如下。有三个 Setup Type 选项，一个是 Full，表示全部安装，强烈建议你用这个选项。另外两个分别 Custom 和 Minimum。Custom 让你可以选择哪些文档要装，哪些不要安装。Minimum 表示仅仅安装你的 Visual Studio 里面所选功能的相关文档。



然后出现目的目录选择画面，如下。如果你不喜欢安装在缺省 Program Files 目录下，你可以按 browser 按钮，选择一个目录。



选好目的目录后，会出现如下画面。请点击画面的 **Install** 按钮。



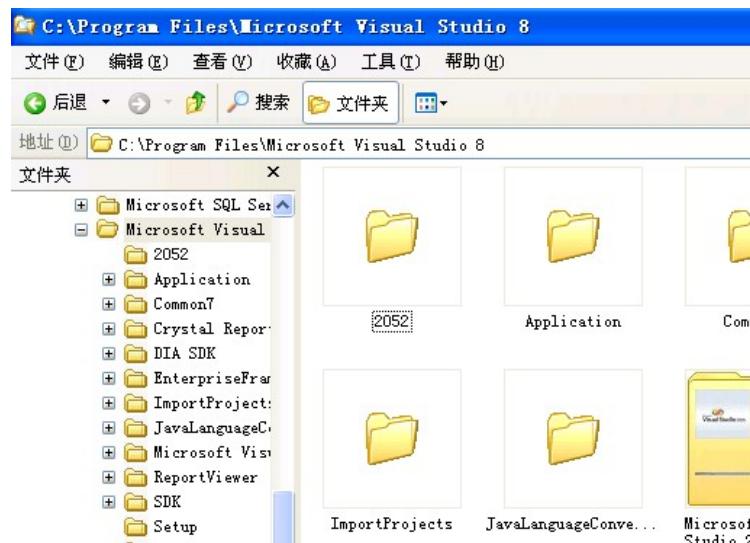
安装完毕后，会出现如下画面，按 Finish 即可。



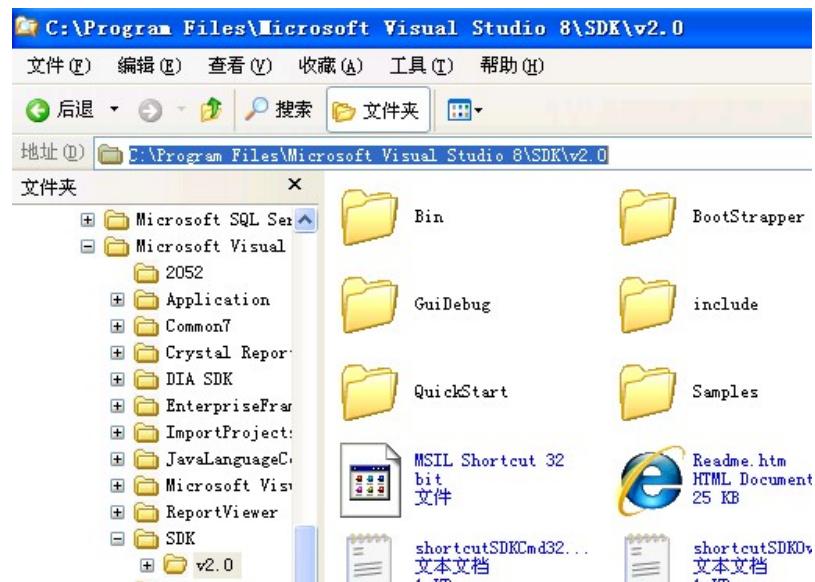
至此，你已经成功安装好了 Visual Studio 和 MSDN。你真聪明！

5. 安装后的文件夹和文件

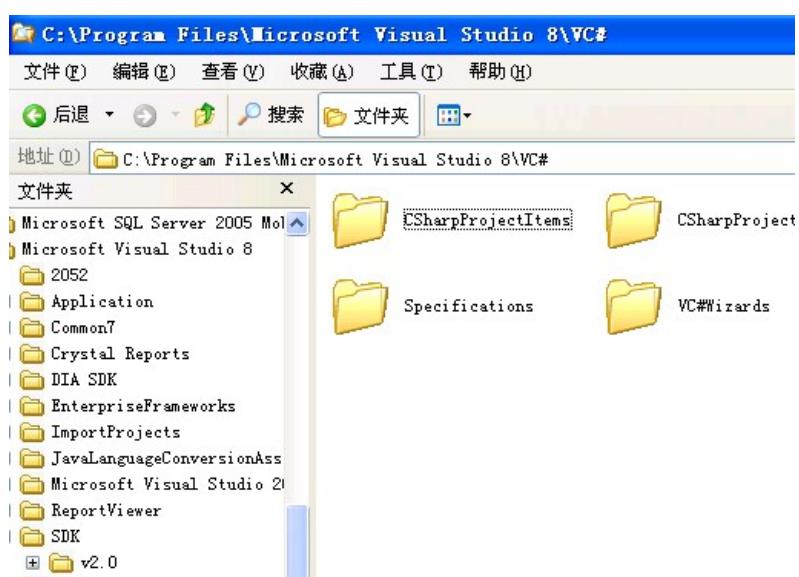
安装成功后，d:\c#\Visual Studio 2005\1.5 中文件和子目录结构如图 1.1 所示。其中 bin 文件夹中包含编译器（c#c.exe）、解释器（c#.exe）、Applet 查看器（appletviewer.exe）等可执行文件，lib 文件夹中包含了所有的类库以便开发 C# 程序使用，sample 文件夹包含开源代码程序实例，src 压缩文件中包含类库开源代码。



SDK 中的内容：



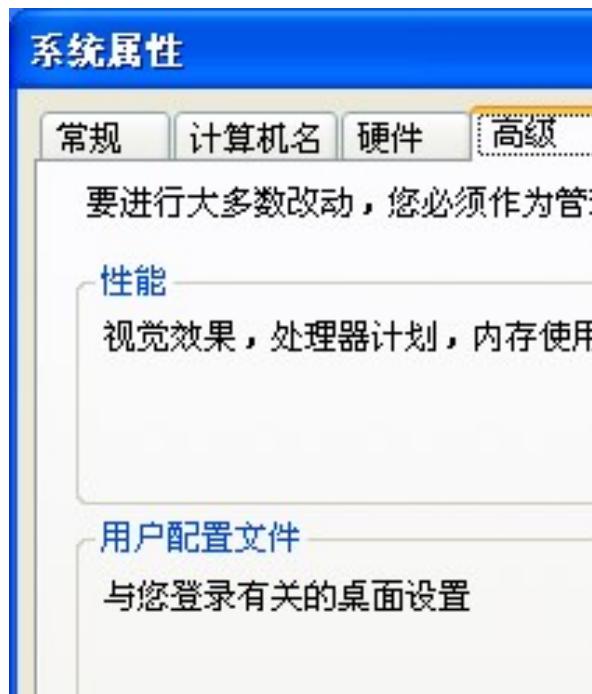
VC#中的内容



二. 设置环境变量

VISUAL STUDIO 2015 中的工具都是命令行工具，需要从命令行即 MS-DOS 提示符下运行它们。很多人可能会不习惯，但这是微软特意采取的策略，为的是把精力更多投入到 C# 语言本身而不是花在开发工具上。以 Windows XP 为例说明设置过程（Windows 其他版本类似）。

右击桌面“我的电脑”图标，选择“属性”菜单



在“高级”选项卡中单击“环境变量”按钮，将出现“环境变量”设置界面



在“系统变量”框中点击“新建”按钮，在出现的“编辑系统变量”对话框中，在“变量名”栏的文本框内输入“csc”，在变量值栏的文本框内输入“C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727”（即 Visual Studio 2005 安装的主目录）



在“系统变量”框中选择“path”，然后单击“编辑”按钮，在出现的“编辑系统变量”对话框中，在“变量值”栏的命令后添加“;%csc%”。



设置完成后，单击“开始”->“所有程序”->“附件”->“命令提示符”打开 DOS 窗口，在命令行提示符输入“csc”回车后，如果出现下面提示信息，则安装正确。

```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.

C:\>Documents and Settings\Administrator>c:
Microsoft (R) Visual C# 2005 编译器 版本 :
用于 Microsoft (R) Windows (R) 2005 Frame
版权所有 (C) Microsoft Corporation 2001-2005

fatal error CS2008: 未指定输入

C:\>Documents and Settings\Administrator>
```

三. 使用记事本和 Visual Studio 2005

要编写 C# 程序，需要有文本编辑器。可使用操作系统提供的记事本作为编辑器，或者选择 Visual Studio 2005 编辑软件，它可以使用不同颜色显示 c# 的关键字和类名，简单好用，同时可以对 c# 源程序进行编译和运行。

1. 使用记事本编辑 C# 代码文件

打开记事本，在其中输入 C#代码，并将文件保存为扩展名为.cs 的文件，本例文件名为“3-1.cs”就生成了 c#文件。



```
3-1.cs - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#region Using directives

using System;
using System.Collections.Generic;
using System.Text;

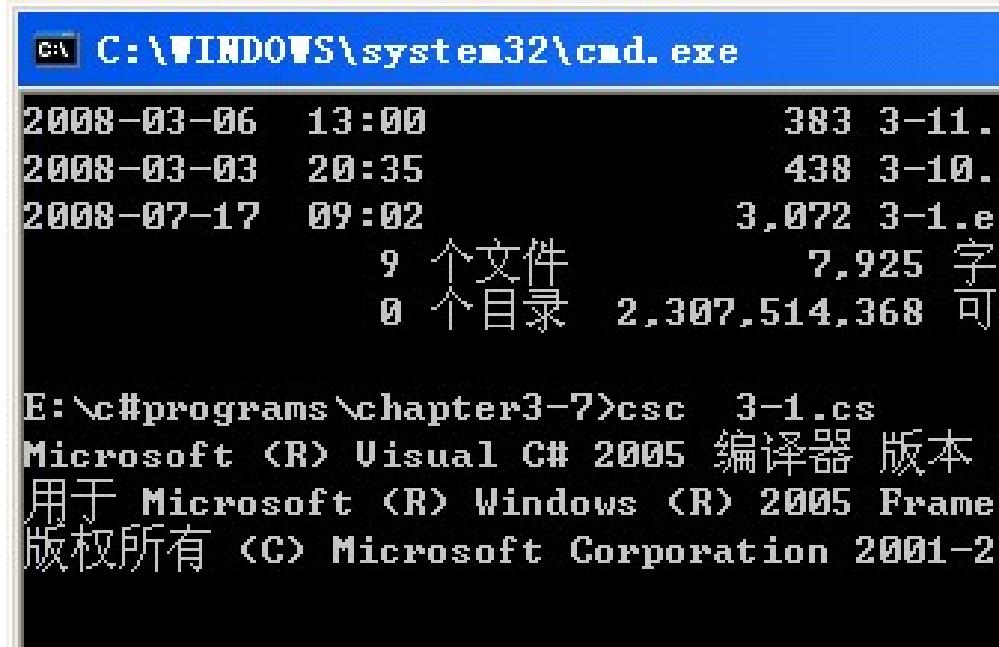
#endregion

namespace InitializingVariables
{
```

C#源程序编写后，要使用 C#编译器（csc.exe）进行编译，将 C#源程序编译成可执行的程序代码。C#源程序都是扩展名为.cs 的文件。编译时，首先读入 C#源程序，然后进行语法检查，如果出现问题就终止编译。语法检查通过后，生成可执行程序代码。要想编译和运行本程序需要在环境变量配置好的情况下在 MS-DOS 中输入以下命令：

CSC 3-1.cs，然后敲入回车，cs 文件进行编译并生产 3-1.exe 文件。在命令提示符窗口键入生产的可执行文件按回车键即开始执行程序并输出结果。

输入 3-1.exe 或 3-1 进行执行程序，结果如下：



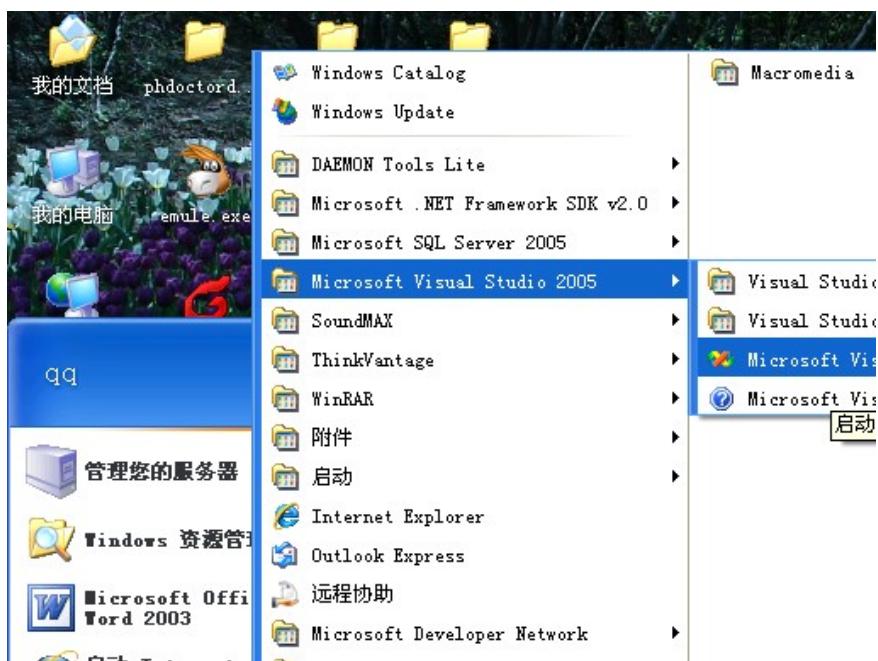
```
C:\WINDOWS\system32\cmd.exe
2008-03-06 13:00          383 3-11.
2008-03-03 20:35          438 3-10.
2008-07-17 09:02          3,072 3-1.e
                           9 个文件          7,925 字
                           0 个目录  2,307,514,368 可
E:\c#\programs\chapter3-7>csc 3-1.cs
Microsoft (R) Visual C# 2005 编译器 版本
用于 Microsoft (R) Windows (R) 2005 Frame
版权所有 (C) Microsoft Corporation 2001-2.
```

这样就完成了一个利用记事本来编辑程序文件，用编译器命令来编译 cs 文件并执行的过程。

2. 使用 Visual Studio 2005

打开装 Visual Studio 2005 开发环境软件

从开始→所有程序→Microsoft Visual Studio 2005→ Microsoft Visual Studio 2005



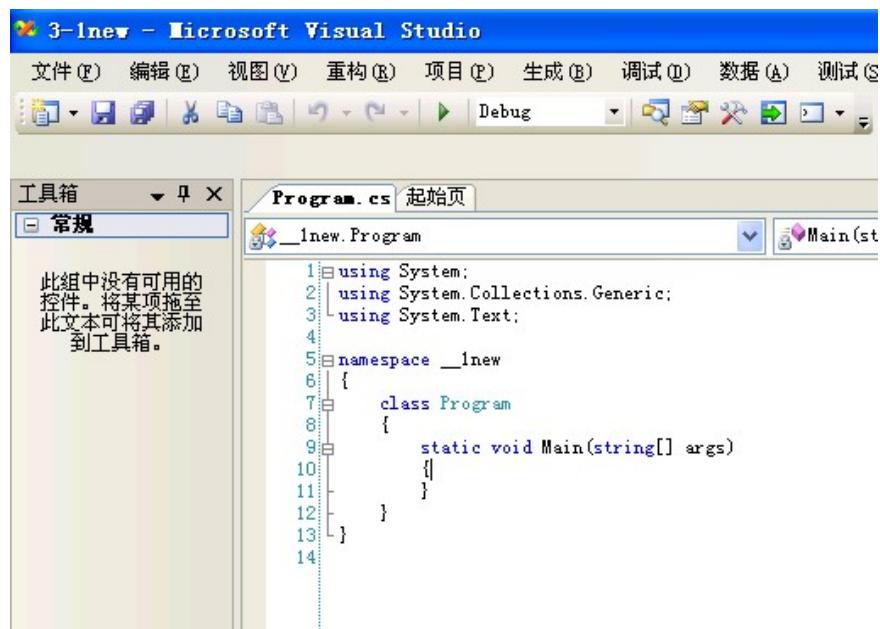
打开开发环境图标后进入以下起始页面：



因为 3-1 所示例子为控制台程序，需要新建控制台项目。操作为从文件菜单进入新建→项目→Visual C#项目如下图



选择“控制台应用程序”并命名为“3-1new”，单击确定后进入编辑页面。

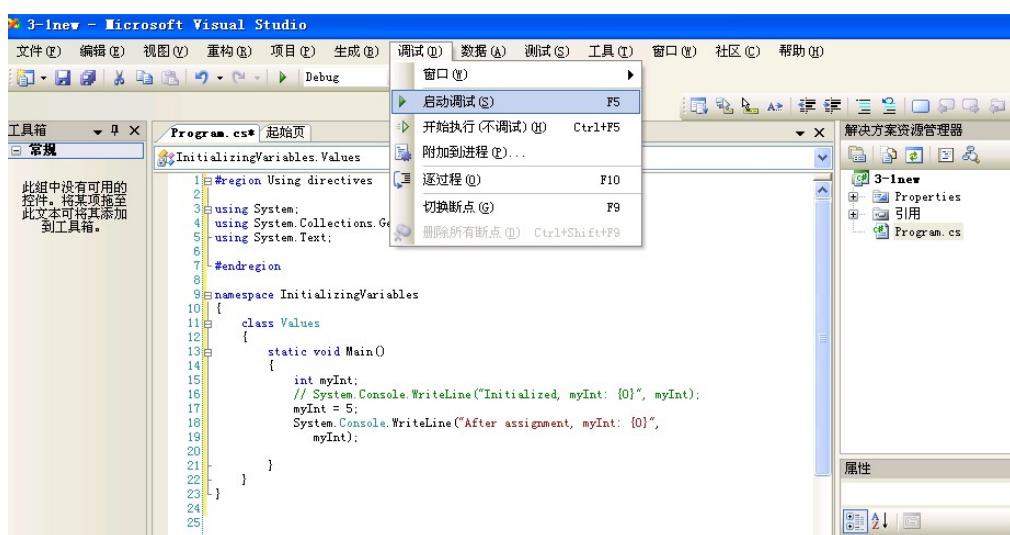


在文件中输入和 3-1 一样的代码：

The screenshot shows the Microsoft Visual Studio 2005 interface. The title bar says "3-new - Microsoft Visual Studio". The menu bar includes "文件 (F)", "编辑 (E)", "视图 (V)", "重构 (R)", "项目 (P)", "生成 (G)", "调试 (D)", "数据 (A)", and "测试 (T)". The toolbar has various icons for file operations. The code editor window is titled "Program.cs* 起始页". It contains the following C# code:

```
1 // #region Using directives
2
3 using System;
4 using System.Collections.Generic;
5 using System.Text;
6
7 //endregion
8
9 namespace InitializingVariables
10 {
11     class Values
12     {
13         static void Main()
14         {
15             int myInt;
16             // System.Console.WriteLine("Initialized");
17             myInt = 5;
```

然后进入程序的编译和运行环节，Visual Studio 2005 提供了编译选项为：



选择 F5 或者 Ctrl+F5 或者单击菜单，程序编译并运行结果如下：

The screenshot shows a Windows Command Prompt window with the title "C:\WINDOWS\system32\cmd.exe". The output of the program is displayed:

```
c:\ After assignment, myInt: 5
请按任意键继续... . . .
```

【思考题】

- (1) 环境变量 path 的作用。
- (2) 如何在记事本和 Visual Studio 2015 中如何编写、编译和运行 C#文件。

环境变量 Path 是 shell 命令扫描的位置，由于程序的定位。

记事本：

先使用 dotnet new console 命令创建一个.net core console 项目，之后编写代码，使用 dotnet run 命令运行。

Visual Studio 2015 创建项目，编写代码，然后按绿色的按钮编译、生成和运行代码。

实验一（2） C#基本语法编程练习

【实验目的】

1. 了解 C#的数据类型
2. 掌握各种变量的声明方式
3. 理解运算符的优先级
4. 掌握 c#基本数据类型、运算符与表达式的使用方法
5. 理解 C#程序语法结构，掌握顺序结构、选择结构和循环结构语法的程序设计方法

【实验要求】

1. 写出程序，并调试程序，要给出测试数据和实验结果。
2. 整理上机步骤，总结经验和体会。
3. 完成实验日志和上交程序。

【实验内容】

1. 编写一个程序，输出两行结果：Hello, My name is “自己的姓名”，This is my first csharp program；我的 C#学习之旅开始了。
2. 编写一个声明 c#不同数据类型变量的程序
3. 编写一个使用运算符、表达式、变量的程序
4. 编写一个使用 c#数据的程序
5. 编写表达式语句、复合语句的程序
6. 编写使用不同选择结构的程序
7. 编写使用不同循环结构的程序

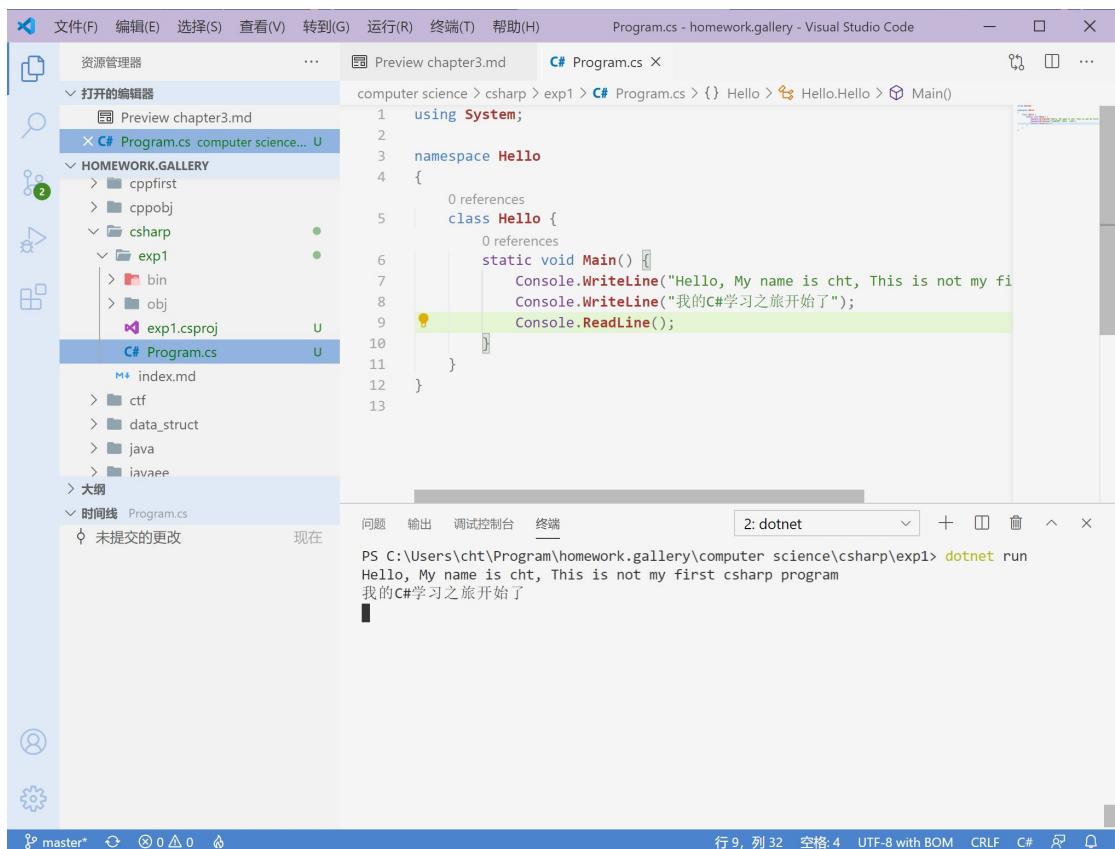
一. 编写程序，输出实验内容 1 要求的结果

参考代码如下：

```
#region Using directives
using System;
using System.Collections.Generic;
using System.Text;
#endregion
namespace Hello
```

```
{
    class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello, My name is LiNing, This is my first csharp
program");
            Console.WriteLine("我的c#学习之旅开始了");
            Console.ReadLine();
        }
    }
}
```

在最后将 `Console.ReadLine();` 行加上注释，看输出界面有何变化，掌握此语句控制命令操作符界面的使用技巧。



如果将最后的语句注释，将直接结束程序的运行，即直接返回到终端。

二. 声明不同数据类型的变量

1. 编写声明不同数据类型变量的程序文件datatype.cs，源代码如下：

```
#region Using directives
using System;
using System.Collections.Generic;
```

```
using System.Text;
#endregion
namespace DataType
{
    class DataType
    {
        static void Main()
        {
            int myInt = 3;
            short myShort = 32765;
            uint myUint = 1;
            float myFloat = 100.15f;
            double myDouble = -99;
            long myLong = 10000;
            decimal myDecimal = -1.88m;

            Console.WriteLine("myInt: {0}, myShort: {1}, myUint: {2}, myFloat: {3}", myInt, myShort,
                myUint, myFloat);
            Console.WriteLine("myDouble: {0}, myLong: {1}, myDecimal", myDouble, myLong, myDecimal
            );
            Console.ReadLine();
        }
    }
}
```

2. 编译并运行该程序并修改各数据类型不同的数值，检验教材上所示数据范围。

3. 将两行中的WriteLine改为Write查看输出结果。

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** DataType.cs - exp1 - Visual Studio Code.
- Explorer:** Shows the project structure under EXP1: bin, obj, C# DataType.cs (selected), exp1.csproj, C# Program.cs.
- Editor:** Displays the C# code for DataType.cs. The code defines a class `DataType` with a static void `Main` method. It declares variables of various types and prints them to the console.
- Terminal:** Shows the command `dotnet run` being run in the terminal, resulting in errors and warnings related to variable names.
- Status Bar:** master*, 1: dotnet, Ln 23, Col 2, Spaces: 4, UTF-8, CRLF, C#, 🔍, ⚙️.

This screenshot is nearly identical to the one above, but the cursor is positioned at a different line in the code editor (line 18).

4. 分析以下程序功能:

```
short shortResult, shortVal = 4;
```

```

int    integerVal = 67;
long   longResult;
float  floatVal = 10.5F;
double doubleResult, doubleVal = 99.999;
string stringResult, stringVal = "17";
bool   boolVal = true;

Console.WriteLine("Variable Conversion Examples\n");
doubleResult = floatVal * shortVal;
Console.WriteLine("Implicit, -> double: {0} * {1} -> {2}", floatVal,
    shortVal, doubleResult);
shortResult = (short)floatVal;
Console.WriteLine("Explicit, -> short: {0} -> {1}", floatVal,
    shortResult);
stringResult = Convert.ToString(boolVal) +
    Convert.ToString(doubleVal);
Console.WriteLine("Explicit, -> string: \"{0}\" + \"{1}\" -> {2}",
    boolVal, doubleVal, stringResult);
longResult = integerVal + Convert.ToInt64(stringVal);
Console.WriteLine("Mixed,      -> long: {0} + {1} -> {2}",
    integerVal, stringVal, longResult);
Console.ReadLine();

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main window displays the C# file `dataType.cs` containing the provided code for variable conversion examples.
- Explorer:** Shows the project structure with files `bin`, `obj`, `exp1.csproj`, and `Program.cs`.
- Terminal:** The terminal window at the bottom shows the command-line output of running the application with the command `dotnet run`. It displays the results of implicit and explicit conversions for various data types.
- Status Bar:** Shows the current branch is `master*`, the file is `exp1`, and the status bar includes `Ln 43, Col 1`, `Spaces: 4`, `UTF-8`, `CRLF`, `C#`, and icons for search, refresh, and other tools.

第一行说明Short兼容向Float，Float向Double的隐式转换，

第二行说明当高一级的数据转向第一级的数据时，需要进行强制转换，
第三行说明了所有类型都有ToString()以及字符串的拼接运算符+，
第四行说明了String向long的转换（等价于long.Parse()），以及int到long的隐式转换。
5. 用enum定义字节类型方位常量，打印出某一方位并将此方位值转化为字节类型，字符串类型值。分析输出结果的原因。回答以下问题：
Enum的缺省类型是什么？直接输出myDirection和(byte)myDirection有何区别。

```
class Variables
{
    enum orientation : byte
    {
        north = 1,
        south = 2,
        east = 3,
        west = 4
    }
    static void Main(string[] args)
    {
        byte directionByte;
        string directionString;
        orientation myDirection = orientation.north;
        Console.WriteLine("myDirection = {0}", myDirection);
        directionByte = (byte)myDirection;
        directionString = Convert.ToString(myDirection);
        Console.WriteLine("byte equivalent = {0}", directionByte);
        Console.WriteLine("string equivalent = {0}", directionString);
        Console.ReadLine();
    }
}
```

```

class DataType
{
    // static void Main()
    // {
    //     // int myInt = 3;
    //     // short myShort = 32765;
    //     // uint myUint = 1;
    //     // float myFloat = 100.15f;
    //     // double myDouble = -99;
    //     // long myLong = 10000;
    //     // decimal myDecimal = -1.88m;

    //     // Console.WriteLine("myInt:{0},myShort:{1},myUint:{2},myFlo
    //     // Console.WriteLine("myDouble:{0},myLong:{1},myDecimal",myD
    //     // Console.ReadLine();

    //     // short shortResult, shortVal = 4;
}

Explicit, -> short: 10.5 -> 10
Explicit, -> string: "True" + "99.999" -> True99.999
Mixed, -> long: 67 + 17 -> 84

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
datatype.cs(7,21): error CS0017: 程序定义了多个入口点。使用 /main (指定包含入口点的类型)
进行编译。 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1\exp1.csproj]

```

Enum的缺省类型是int，直接输出myDirection将调用Enums.ToString()，即输出字面量，而byte (myDireciton)将强制转换枚举到存储类型。

二. 使用关系运算符和逻辑运算符

1. 建立使用关系运算符和逻辑运算符的程序文件。Main方法中实例代码如下

```

static void Main(string[] args)
{
    Console.WriteLine("Enter an integer:");
    int myInt = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Integer less than 10? {0}", myInt < 10);
    Console.WriteLine("Integer between 0 and 5? {0}",
        (0 <= myInt) && (myInt <= 5));
    Console.WriteLine("Bitwise AND of Integer and 10 = {0}", myInt & 10);
    Console.ReadLine();
}

```

2. 编译运行该程序。并尝试myInt输入不同范围整数，非10和10时的输出差异。

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Program.cs - exp1 - Visual Studio Code.
- Explorer:** Shows the project structure with files: DataType.cs, Variables.cs, Program.cs, and exp1.csproj.
- Code Editor:** Displays C# code for a program that reads an integer, checks if it's less than 10 or between 0 and 5, and performs a bitwise AND operation.
- Terminal:** Shows the command PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run being run, followed by the program's output: "Enter an integer: 10", "Integer less than 10? False", "Integer between 0 and 5? False", "Bitwise AND of Integer and 10 = 10".
- Status Bar:** master*, Ln 15, Col 13, Spaces: 4, UTF-8 with BOM, CRLF, C#, etc.

三. 使用表达式语句与复合语句

- 建立包含表达式语句程序，要求定义两个double型数据，从控制台输入你的名字和两个double数据，计算这两个数据的和、差、乘、除。源代码如下。

```
using System;
namespace Variables
{
    class Variables
    {
        static void Main(string[] args)
        {
            double firstNumber, secondNumber;
            string userName;
            Console.WriteLine("Enter your name:");
            userName = Console.ReadLine();
            Console.WriteLine("Welcome {0}!", userName);
            Console.WriteLine("Now give me a number:");
            firstNumber = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("Now give me another number:");
            secondNumber = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("The sum of {0} and {1} is {2}.", firstNumber,
                secondNumber, firstNumber + secondNumber);
        }
    }
}
```

```

        Console.WriteLine("The result of subtracting {0} from {1} is {2}.",
                           secondNumber, firstNumber, firstNumber - secondNumber);
        Console.WriteLine("The product of {0} and {1} is {2}.", firstNumber,
                           secondNumber, firstNumber * secondNumber);
        Console.WriteLine("The result of dividing {0} by {1} is {2}.",
                           firstNumber, secondNumber, firstNumber / secondNumber);
        Console.WriteLine("The remainder after dividing {0} by {1} is {2}.",
                           firstNumber, secondNumber, firstNumber % secondNumber);
        Console.ReadLine();
    }
}
}

```

2. 注意总结：Console.ReadLine 方法，Convert.ToDouble 方法的使用。

3. 查阅 MSDN 找出一系列转化成其它类型的 Convert 方法的使用。

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files: `bin`, `obj`, `exp1.csproj`, `Program.cs`, and `Variables.cs`.
- Editor:** The `Variables.cs` file is open, containing the following C# code:


```

C# Variables.cs > {} Variables > Variables.Variables
30     Console.WriteLine("Enter your name. ");
31     userName = Console.ReadLine();
32     Console.WriteLine("Welcome {0}!", userName);
33     Console.WriteLine("Now give me a number:");
34     firstNumber = Convert.ToDouble(Console.ReadLine());
35     Console.WriteLine("Now give me another number:");
36     secondNumber = Convert.ToDouble(Console.ReadLine());
37     Console.WriteLine("The sum of {0} and {1} is {2}.", firstNumber,
38                     secondNumber, firstNumber + secondNumber);
39     Console.WriteLine("The result of subtracting {0} from {1} is {2}.",
40                     secondNumber, firstNumber, firstNumber - secondNumber);
41     Console.WriteLine("The product of {0} and {1} is {2}.", firstNumber,
42                     secondNumber, firstNumber * secondNumber);
43     Console.WriteLine("The result of dividing {0} by {1} is {2}.",
44                     firstNumber, secondNumber, firstNumber / secondNumber);
45     Console.WriteLine("The remainder after dividing {0} by {1} is {2}.",
46                     firstNumber, secondNumber, firstNumber % secondNumber);
47     Console.ReadLine();
48
49
      
```
- Terminal:** Shows the command `dotnet run` being executed, followed by the program's output:


```

PS C:\Users\cht\Program\homework\computer science\csharp\exp1> dotnet run
Enter your name:
cht
Welcome cht!
Now give me a number:
9
Now give me another number:
10
The sum of 9 and 10 is 19.
The result of subtracting 10 from 9 is -1.
The product of 9 and 10 is 90.
The result of dividing 9 by 10 is 0.9.
The remainder after dividing 9 by 10 is 9.
      
```
- Status Bar:** Shows the current branch is `master*`, the file is `exp1.cs`, and the terminal tab is active.

四. 使用选择语句

1. 从键盘输入两个数进行比较，并定义一个字符串变量，当数1小于数2时，字符串变量为“less than”，当数1等于数2时字符串变量为“equal to”，当数1大于数2时字符串变量为“greater than”。

```

static void Main(string[] args)
{
    string comparison;
}

```

```

Console.WriteLine("Enter a number:");
double var1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Enter another number:");
double var2 = Convert.ToDouble(Console.ReadLine());
if (var1 < var2)
    comparison = "less than";
else
{
    if (var1 == var2)
        comparison = "equal to";
    else
        comparison = "greater than";
}
Console.WriteLine("The first number is {0} the second number.", comparison);
Console.ReadLine();
}

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Program.cs - exp1 - Visual Studio Code. The code is as shown in the previous block.
- Explorer:** Shows files in the project: OPEN EDITORS (C# DataType.cs, C# Variables.cs), EXP1 (bin, obj, C# DataType.cs, C# Program.cs, C# Variables.cs), and exp1.csproj.
- Terminal:**
 - Output: The remainder after dividing 9 by 10 is 9.
 - PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
 - Program.cs(24,21): error CS0017: 程序定义了多个入口点。使用 /main (指定包含入口点的类型) 进行编译。 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1\exp1.csproj]
 - 生成失败。请修复生成错误并重新运行。
 - PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
 - Enter a number: 9
 - Enter another number: 10
 - The first number is less than the second number.
- Status Bar:** master*, 26, Col 25, Spaces: 4, UTF-8 with BOM, CRLF, C#, 8, 8.

3. 定义三个常量字符串“karli”，“angelina”，“ploppy”，并从键盘输入一个名字，当名字与“karli”相同时输出我们的名字相同，当和“angelina”名字相同时输出你的名字太性感了，当和“ploppy”相同时输出这名字真傻。

```

static void Main(string[] args)
{

```

```

const string myName = "karli";
const string sexyName = "angelina";
const string sillyName = "ploppy";
string name;
Console.WriteLine("What is your name?");
name = Console.ReadLine();
switch (name.ToLower())
{
    case myName:
        Console.WriteLine("You have the same name as me!");
        break;
    case sexyName:
        Console.WriteLine("My, what a sexy name you have!");
        break;
    case sillyName:
        Console.WriteLine("That's a very silly name.");
        break;
}
Console.WriteLine("Hello {0}!", name);
Console.ReadLine();
}

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Program.cs - exp1 - Visual Studio Code.
- Explorer Panel:** Shows the project structure with files: dataType.cs, Variables.cs, and Program.cs (the active file).
- Code Editor:** Displays the C# code provided above, with syntax highlighting and a yellow cursor indicating the current position.
- Terminal:** Shows command-line output from running the program with `dotnet run`. It prompts for user input ("请输入温度(摄氏):") and displays responses for different inputs (30, 21, 5050) corresponding to the names myName, sexyName, and sillyName respectively.
- Status Bar:** master*, 0△0, exp1, Line 24, Col 39, Spaces: 4, UTF-8 with BOM, CRLF, C#.

五. 使用循环语句

1. for 循环语句练习

(1) 程序功能要求：按5 度的增量打印出一个从摄氏温度到华氏温度的转换表。

```
static void Main(string[] args)
{
    double Fa,Cel;
    Cel =0;
    for(Cel=0;Cel<100;Cel+=5)
    {
        Fa = Cel * 9/5;
        Console.WriteLine(Fa);
        Console.ReadLine();
    }
}
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Program.cs - exp1 - Visual Studio Code. The code is displayed in the center editor area.
- Explorer:** Shows the project structure under EXP1, including bin, obj, and files like DataType.cs, exp1.csproj, Program.cs, and Variables.cs. Program.cs is currently selected.
- Terminal:** At the bottom, it shows a PowerShell terminal with the following history:

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1>
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1>
```

and status information: Ln 18, Col 10, Spaces: 4, UTF-8 with BOM, CRLF, C#.

(2) 自行改造以上程序。

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Program.cs - exp1 - Visual Studio Code.
- Explorer:** Shows the project structure under EXP1, including bin, obj, C# DataType.cs, exp1.csproj, C# Program.cs (selected), and C# Variables.cs.
- Code Editor:** Displays the following C# code:

```
0 references
class Program
{
    static void Main(string[] args)
    {
        double Fa,Cel;
        Cel = 0;
        Console.WriteLine("Cel\tFa");
        for(Cel = 0; Cel < 100; Cel += 5)
        {
            Fa = Cel * 9/5;
            Console.WriteLine($"{Cel}\t{Fa}");
        }
        Console.ReadLine();
    }
}
```
- Terminal:** Shows the command 1: dotnet.
- Status Bar:** master*, 30 54, 35 63, 40 72, 45 81, 50 90, 55 99, 60 108, 65 117, 70 126, 75 135, 80 144, 85 153, 90 162, 95 171. Ln 18, Col 10, Spaces: 4, UTF-8 with BOM, CRLF, C#.

```
using System;

namespace exp1
{
    class Program
    {
        static void Main(string[] args)
        {
            double Fa,Cel;
            Cel = 0;
            Console.WriteLine("Cel\tFa");
            for(Cel = 0; Cel < 100; Cel += 5)
            {
                Fa = Cel * 9/5;
                Console.WriteLine($"{Cel}\t{Fa}");
            }
            Console.ReadLine();
        }
    }
}
```

2. while 循环语句练习

(1) 程序功能要求：运行程序后从键盘输入数字1/2/3 后，可显示抽奖得到的奖品：恭喜

你得了一辆汽车；不错啊，一台笔记本电脑；没白来，一个MP3；如果输入其它数字或字符显示“没有奖品给你！”示例代码如下：

```
int choice;
choice = Convert.ToInt32(Console.ReadLine());
while (choice == 1)
{
    Console.WriteLine("恭喜你得了一辆汽车");
    break;
}
while (choice == 2)
{
    Console.WriteLine("不错啊，一台笔记本电脑");
    break;
}
while (choice == 3)
{
    Console.WriteLine("没白来，一个MP3");
    break;
}
while (choice != 1 && choice != 2 && choice != 3)
{
    Console.WriteLine("没有奖品给你");
    break;
}
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Program.cs - exp1 - Visual Studio Code. The code is as follows:

```
C# Program.cs > {} exp1 > exp1.Program > Main(string[] args)
    Console.WriteLine("不错啊，一台笔记本电脑");
    break;
}
while (choice == 3)
{
    Console.WriteLine("没白来，一个MP3");
    break;
}
while (choice != 1 && choice != 2 && choice != 3)
{
    Console.WriteLine("没有奖品给你");
    break;
}
```

- Bottom Status Bar:** master*, 1: powershell, Ln 32, Col 10, Spaces: 4, UTF-8 with BOM, CRLF, C#.

(2) 改造以上程序实现此功能；尝试将choice==1或2或3中的“==”改为一个“=”，看效果如何？并分析错误。

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Program.cs - exp1 - Visual Studio Code. The code is as follows, with a red squiggle under the '==' operator in the second while loop condition:

```
C# Program.cs > {} exp1 > exp1.Program > Main(string[] args)
    static void Main(string[] args)
    {
        int choice;
        choice = Convert.ToInt32(Console.ReadLine());
        while (choice == 1)
        {
            Console.WriteLine("恭喜你得了一辆汽车");
            break;
        }
        while (choice == 2)
        {
            Console.WriteLine("不错啊，一台笔记本电脑");
            break;
        }
        while (choice == 3)
        {
            Console.WriteLine("没白来，一个MP3");
            break;
        }
    }
}
```

- Bottom Status Bar:** master*, 1: powershell, Ln 16, Col 28, Spaces: 4, UTF-8 with BOM, CRLF, C#.

The terminal output shows a compilation error:

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
1
恭喜你得了一辆汽车
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
Program.cs(16,20): error CS0029: 无法将类型“int”隐式转换为“bool” [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1\exp1.csproj]
```

赋值表达式的值是变量的值，因此出现int->bool的错误。

3. do...while 循环语句练习

程序功能要求：输入你现有的存款和当前的年利率及你期望将来得到的存款，计算出存款多少年后才可以变成你期望的存款额。注意，若为一年输出year为year，若为多年输出year为years。参考代码如下：

```
static void Main(string[] args)
{
    double balance, interestRate, targetBalance;
    Console.WriteLine("What is your current balance?");
    balance = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("What is your current annual interest rate (in %)?");
    interestRate = 1 + Convert.ToDouble(Console.ReadLine()) / 100.0;
    Console.WriteLine("What balance would you like to have?");
    targetBalance = Convert.ToDouble(Console.ReadLine());

    int totalYears = 0;
    do
    {
        balance *= interestRate;
        ++totalYears;
    }
    while (balance < targetBalance);
    Console.WriteLine("In {0} year{1} you'll have a balance of {2}.",
        totalYears, totalYears == 1 ? "" : "s", balance);
}
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Program.cs - exp1 - Visual Studio Code.
- Explorer:** Shows the project structure under EXP1: bin, obj, C# DataType.cs, exp1.csproj, C# Program.cs (selected), and C# Variables.cs.
- Code Editor:** Displays the C# code for Program.cs. The code prompts the user for their current balance, annual interest rate, and target balance, then calculates the total years required to reach the target balance using a do...while loop.
- Terminal:** Shows the command PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run being run, and the output of the program running in powershell.
- Status Bar:** master*, 27500, 2.75, 3, 27500, 32000, In 6 years you'll have a balance of 32361.129927754886, PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1>, Ln 23, Col 24, Spaces: 4, UTF-8 with BOM, CRLF, C#, etc.

【思考题】

1. while和do...while的区别。

while是先判断条件再执行，do...while是先执行再判断条件，后者至少执行一次。

2. 使用if...else 语句编写以下程序

(1) 程序功能要求：使用if...else 语句构造多分支，判断某一年是否为闰年。闰年的条件是符合下面二者之一：能被4 整除，但不能被100 整除；能被4 整除，又能被100 整除。

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Program.cs - exp1 - Visual Studio Code.
- Explorer:** Shows the project structure under EXP1, including bin, obj, C# DataType.cs, exp1.csproj, C# Program.cs (selected), and C# Variables.cs.
- Code Editor:** Displays the C# code for Program.cs, which checks if a year is a leap year based on the rules: divisible by 400, or by 100 but not 400, or by 4 but not 100.
- Terminal:** Shows the command PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run being run, followed by user input and the program's output: "Enter a number: 9", "Enter another number: 10", and "The first number is less than the second number."
- Status Bar:** master*, 1: powershell, Ln 31, Col 10, Spaces: 4, UTF-8 with BOM, CRLF, C#, etc.

```
using System;
```

```
namespace exp1
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("please input a year:");
            int year = int.Parse(Console.ReadLine());
            bool result;
            if (year % 400 == 0)
            {
                result = true;
            }
            else if (year % 100 == 0)
            {
                result = false;
            }
            else if (year % 4 == 0)
            {
                result = true;
            }
        }
    }
}
```

```

        else
        {
            result = false;
        }
        var resultStr = result ? "是" : "不是";
        Console.WriteLine($"{year}年{resultStr}闰年");

    }
}
}

```

3. 使用switch语句编写以下程序

(1) 程序功能要求：在不同温度时显示不同的解释说明：有点冷，多穿衣服；正合适，出去玩吧；太热了，开空调。

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files: EXP1, bin, obj, C# DataType.cs, exp1.csproj, C# Program.cs (selected), and C# Variables.cs.
- Code Editor:** Displays the following C# code in Program.cs:


```

float temprature = float.Parse(Console.ReadLine());
int tempratureL = (int)temprature / 10;
string result = null;
switch(tempratureL){
    case 3:
        result = "太热了";
        break;
    case 2:
        result = "正合适";
        break;
    default:
        result = "太冷了";
        break;
}
Console.WriteLine(result);
      
```
- Terminal:** Shows command-line interaction:


```

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
please input a year:2020
2020年是闰年
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
请输入温度(摄氏): 30
太热了
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
请输入温度(摄氏): 21
正合适
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1>
      
```
- Status Bar:** Shows master*, 1: powershell, Ln 24, Col 13, Spaces: 4, UTF-8 with BOM, CRLF.

```

using System;

namespace exp1
{
    class Program
    {
        static void Main()
        {
            Console.Write("请输入温度(摄氏): ");
            float temprature = float.Parse(Console.ReadLine());
      
```

```

int tempratureL = (int)temprature / 10;
string result = null;
switch(tempratureL){
    case 3:
        result = "太热了";
        break;
    case 2:
        result = "正合适";
        break;
    default:
        result = "太冷了";
        break;
}
Console.WriteLine(result);
}
}
}

```

4. 用do...while语句实现程序功能：求 $1+2+\dots+100$ 之和，并将求和表达式与所求的和显示出来。

```

File Edit Selection View Go Run Terminal Help Program.cs - exp1 - Visual Studio Code
EXPLORER ... C# DataType.cs C# Variables.cs C# Program.cs X
> OPEN EDITORS EXP1
> bin
> obj
C# DataType.cs U C# Program.cs > {} exp1 > exp1.Program > Main()
3 namespace exp1
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main()
10        {
11            int i = 0;
12            int sum = 0;
13            do {
14                sum += ++i;
15            } while(i < 100);
16            Console.WriteLine(sum);
17        }
18    }
19
PROBLEMS OUTPUT 调试控制台 TERMINAL 1: powershell +
10
The first number is less than the second number.

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
please input a year:2020
2020年是闰年
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
请输入温度(摄氏): 30
太热了
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
请输入温度(摄氏): 21
正合适
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1> dotnet run
5050
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp1>

```

using System;

```
namespace exp1
{
    class Program
    {
        static void Main()
        {
            int i = 0;
            int sum = 0;
            do {
                sum += ++i;
            } while(i < 100);

            Console.WriteLine(sum);
        }
    }
}
```