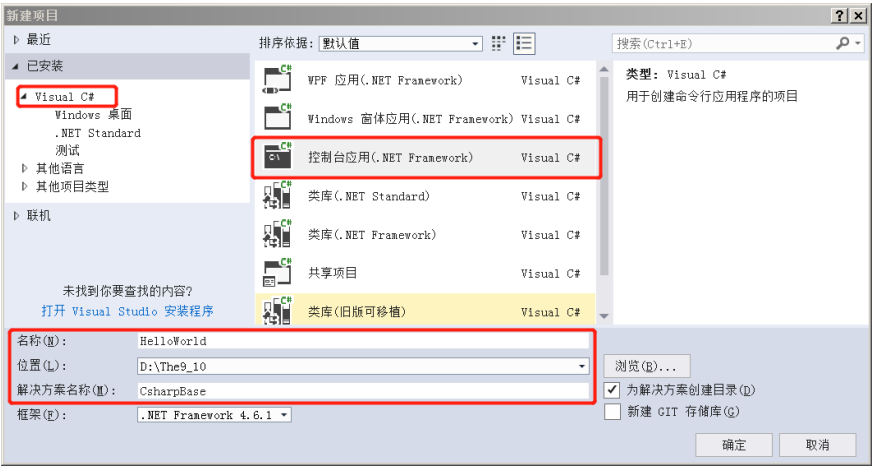


创建第一个控制台项目

打开VisualStudio 菜单>文件>新建>项目

注意:解决方案可以包含多个项目,下图我们创建了一个名叫HelloWorld的项目,同时也创建了一个名叫CsharpBase的解决方案,以后可以在该解决方案下添加更多的项目



了解C#程序的基本结构 (.cs文件的结构)

```
1 using System; //引用命名空间(可能会引用好几个命名空间)
2 namespace CsharpBase//定义命名空间
3 {
4     class Program//定义类
5     {
6         static void Main(string[] args)//定义方法
7         {
8             //这里是程序的入口点
9             //刚开始我们都在这里写代码
10        }
11    }
12 }
```

认识Main函数,它是程序的入口点

- 1. 一般情况一个项目只能有一个Main函数,多个会无法通过编译
- 2. 必须是静态的,(带有static关键字)
- 3. Main函数名字不能写错,必须是大写M开头的Main
- 4. 可以使用 svm Tab Tab 快速输入一个Main函数

控制台输出

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine("Hello world");
4 }
```

控制台输入

```
1 static void Main(string[] args)
2 {
3     //变量input存储着用户输入的内容
4     string input= Console.ReadLine();
5 }
```

C#编程注意事项:

1. Csharp是严格区分大小写的语言 按住Shift输入大写输入
2. 所有的标点符号均使用英文的,可以在输入法中设置
3. 写完代码记得保存
4. 使用键盘上Insert改变光标状态(插入or修改)
5. 所有的括号和引号成对输入 ,再到中间补内容
6. Ctrl+K+D (按D时可以松开K) 格式化代码(整理代码)
7. 写在方法体中的语句结尾需要加分号;

变量

1. 变量是内存中用于存储数据的容器
 2. 定义变量的语法: 数据类型 变量名
- 例如:

```
1 int num; //int是数据类型(这是装什么的容器) num是变量名(我们给这个容器取的名字)
2 或者:
3 int num = 1; //可以使用等号给变量赋予初始值
```

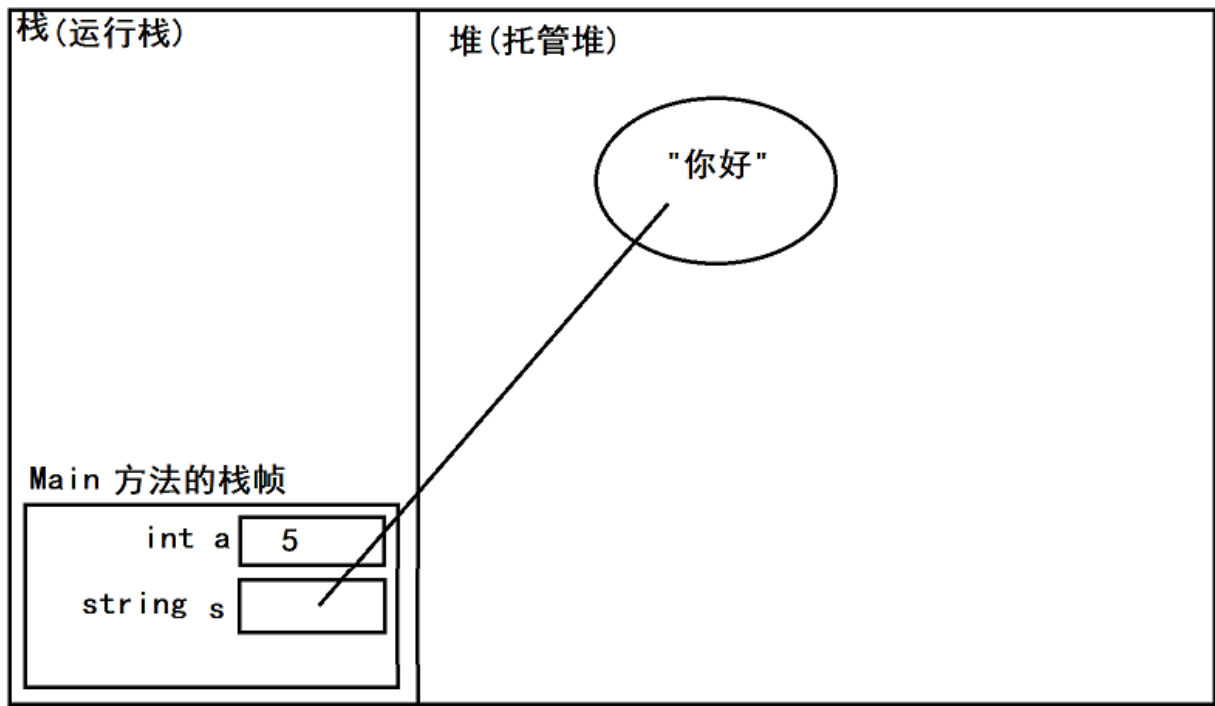
变量可以存实际的值,也可以存储内存地址

变量使用前必须赋值

变量命名规则:

- 不能以数字开头
- 不能以关键字作为变量名
- 第一个单词首字母小写,以后每个单词首字母大写
- 相同作用域内不能重名

变量在内存中的分配



数据类型

常用基本数据类型:

数据类型	名称	长度	是否有负数	注意
byte	字节	8	无符号	取值范围0-255
float	浮点数	32	有符号	书写时要在末尾加上f 例如1.5f
int	整数	32	有符号	整数/整数得整数 不会有小数点
string	字符串			书写时要加上引号例如: <code>string name="张三";</code>
bool	布尔值	8		表示真假只有true和false两个值

1	浮点数在内存中的表示方式:
2	种类-----符号位-----指数位-----尾数位-----
3	float---第31位(占1bit)---第30-23位(占8bit)----第22-0位(占23bit)
4	double--第63位(占1bit)---第62-52位(占11bit)---第51-0位(占52bit)

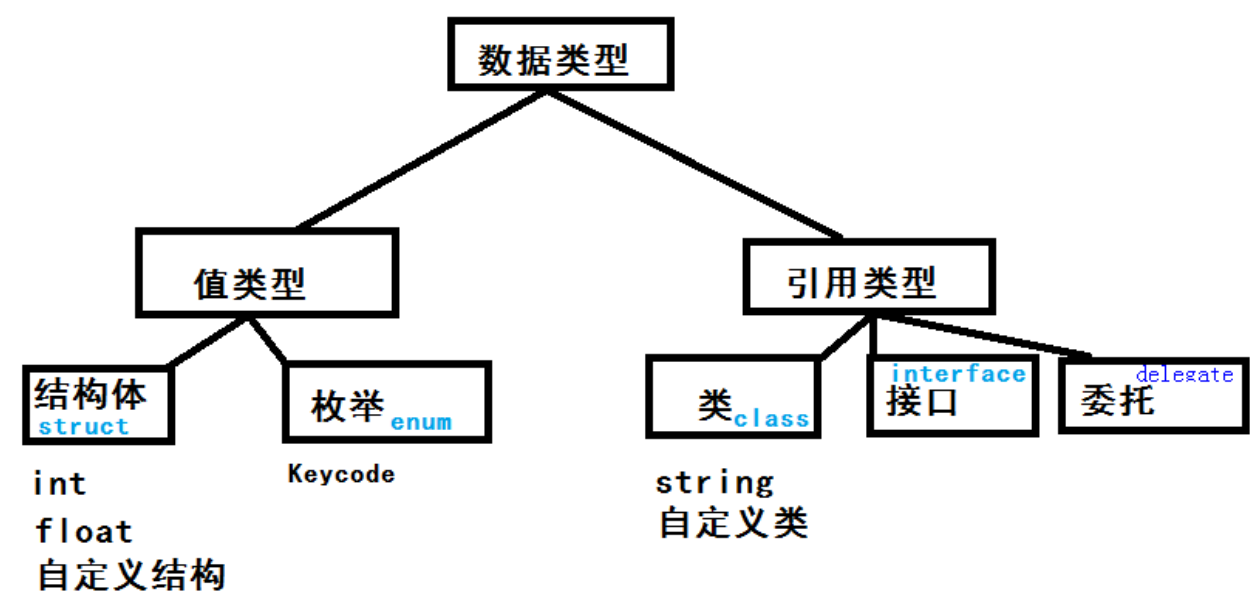
完整版基本数据类型:

bool	布尔值	True 或 False	FALSE
byte	8 位无符号整数	0 到 255	0
char	16 位 Unicode 字符	U +0000 到 U +ffff	'\0'
decimal	128 位精确的十进制值, 28-29 有效位数	$(-7.9 \times 10^{28} \text{ 到 } 7.9 \times 10^{28}) / 10^0 \text{ 到 } 28$	0.0M
double	64 位双精度浮点型	$(+/-)5.0 \times 10^{-324} \text{ 到 } (+/-)1.7 \times 10^{308}$	0.0D
float	32 位单精度浮点型	$-3.4 \times 10^{38} \text{ 到 } +3.4 \times 10^{38}$	0.0F
int	32 位有符号整数类型	-2,147,483,648 到 2,147,483,647	0
long	64 位有符号整数类型	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807	0L
sbyte	8 位有符号整数类型	-128 到 127	0
short	16 位有符号整数类型	-32,768 到 32,767	0
uint	32 位无符号整数类型	0 到 4,294,967,295	0
ulong	64 位无符号整数类型	0 到 18,446,744,073,709,551,615	0
ushort	16 位无符号整数类型	0 到 65,535	0

划分数据类型长度的意义: 节省内存

数据类型派系

决定变量什么时候存值什么时候存内存地址



基本数据类型的 类型转换

隐式转换

低取值范围转高取值范围

```
1 int i = 5;
2 float f = i;
```

强制转换

高取值范围转低取值范围

```
static void Main(string[] args)
{
    float f = 5.0f;
    int i = f; //高精度转低精度需要强制转换
}
```

(局部变量) float f

无法将类型“float”隐式转换为“int”。存在一个显式转换(是否缺少强制转换?)

```
1 当遇到以上情况需要强制转换
2     static void Main(string[] args)
3     {
4         float f = 5.0f;
5         int i = (int) f;
6     }
```

Convert转换

```
1 float f = 5.0f;
2 int i = Convert.ToInt32(f);
```

```
1 ()强制转换 和Convert转换的区别
2
3 ()强制转换转换失败得到一个没有意义的数字
4 Convert转换失败会抛出异常
5
```

Parse转换:字符串转数值类型比如 字符串转成int:

```
1     int a = int.Parse("456");
```

使用Parse转换需要转换的字符串必须长得像目标类型 比如: 字符串"ab"不能转换成int和float

任意类型转字符串

直接调用ToString()

占位符的使用

- 1 传统占位符 例如:{0}
- 2 \$占位符(Unity目前不支持) C# 6.0新特性

案例练习:

- 1 让用户输入 学生姓名(字符串) 学号(整数) 成绩(浮点数) 并以占位符方式组成一句话输出
- 2 例如:张三的学号是45成绩是96.5分

补充:

1命令行自动关机

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>shutdown -s -t 3600 3600秒后关机

C:\Users\Administrator>shutdown -a 取消
```

2环境变量

- 1 右键我的电脑-属性-高级系统设置-环境变量
- 2 当我们在Windows命令窗口输入 命令的时候 会从系统环境变量的Path 中的所有路径中查找跟命令名称相同的exe或快捷方式文件执

VS快捷键

- 1 F1打开帮助文档
- 2 Ctrl+F5 开始运行不调试
- 3 F5 开始运行调试
- 4 F11逐行运行调试
- 5 F10逐方法(过程)调试
- 6 Shift+F5停止调试
- 7
- 8 F12 查看定义的位置 (跟踪代码)
- 9 Ctrl+K-D 整理代码
- 10 Ctrl+K-C 注释代码
- 11 Ctrl+K-U 取消注释