C#程序设计

第2章基本数据类型

杨琦
西安交通大学
计算机教学实验中心
http://ctec.xjtu.edu.cn

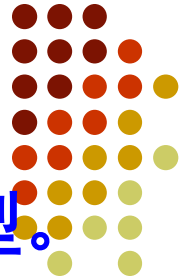
授课内容

- C#数据类型及转换
- C#运算符与表达式
- C#程序流程
- 编译与调试

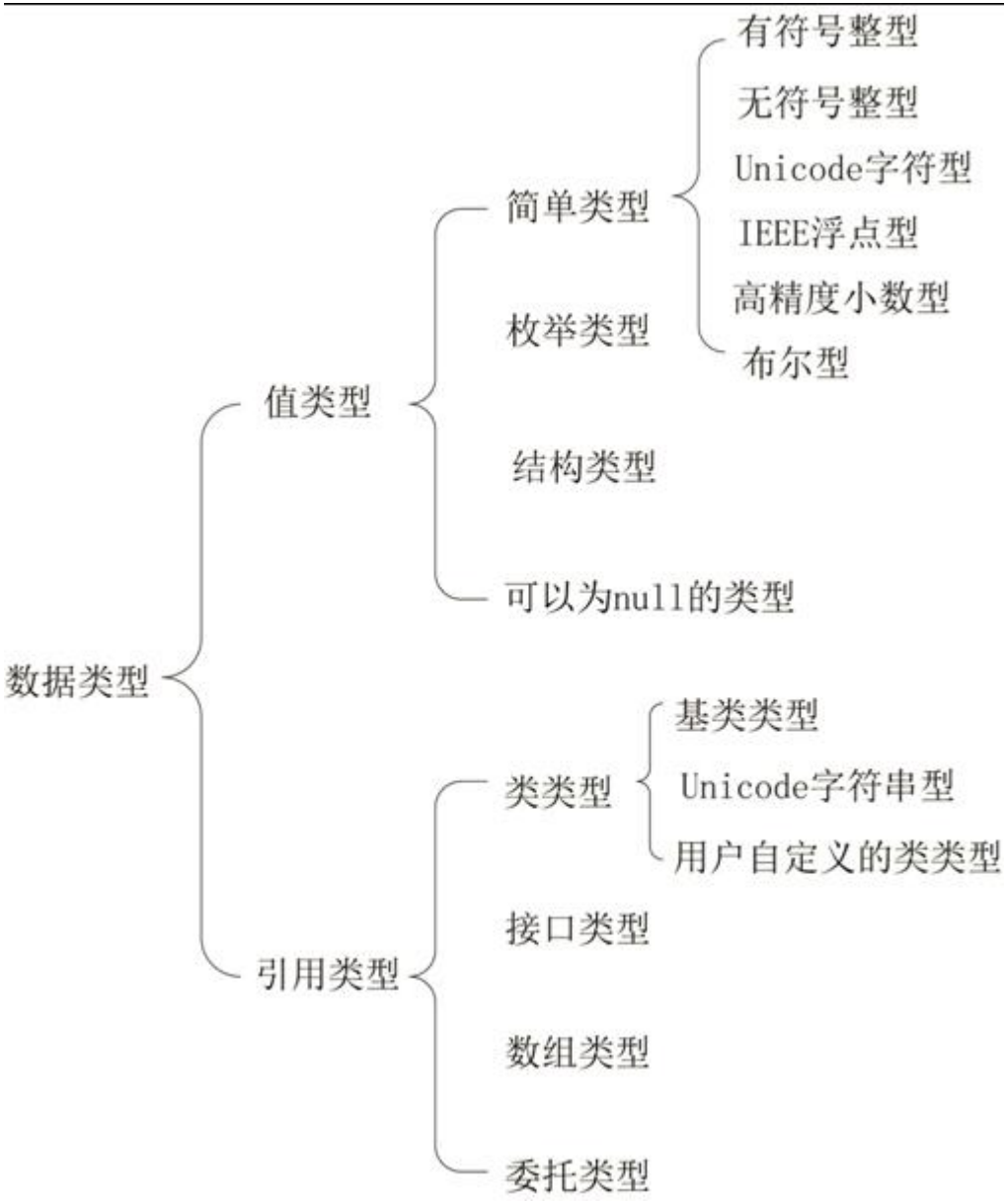
C#的基本字符集

- 数字：
  - 0 1 2 3 4 5 6 7 8 9
- 英文字母：
  - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  - a b c d e f g h i j k l m n o p q r
  - s t u v w x y z
- 特殊字符：
  - Space ！“ # $ % & ‘ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ { | } ~

2.1 数据类型



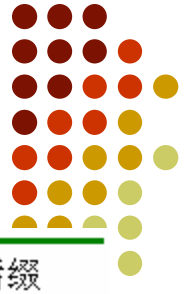
- C#语言的数据类型分为两大类：值类型和引用类型。


数据类型
值类型
简单类型
有符号整型
无符号整型
Unicode字符型
IEEE浮点型
高精度小数型
布尔型
枚举类型
结构类型
可以为null的类型
引用类型
类类型
基类类型
Unicode字符串型
用户自定义的类类型
接口类型
数组类型
委托类型

2.1 C#的简单数据类型

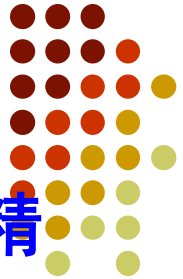
类型	举例
整数类型	sbyte、byte、short、ushort、int、uint、long、ulong 和 char
浮点型	float 和 double
十进制类型	decimal
布尔类型	true 或 false 值，指定的值
空类型	可为空值的数据类型

2.1 C#的简单数据类型



简单数据类型	表示数据	字节长度	取值范围	默认初值	后缀
bool	布尔型	1	True 或 False	False	
sbyte	字节型	1	-128 ~ 127	0	
byte	无符号字节型	1	0 ~ 255	0	
short	短整型	2	-32,768 ~ +32767	0	
ushort	无符号短整型	2	0 ~ 65535	0	
int	整型	4	$-2^{31} \sim 2^{31}-1$	0	
uint	无符号整型	4	$0 \sim 2^{32}-1$	0	U
long	长整型	8	$-2^{63} \sim 2^{63}-1$	0	L
ulong	无符号长整型	8	$0 \sim 2^{64}-1$	0	UL
char	字符型	2	0 ~ 65535	null	
float	单精度浮点数	4	1.40E-45 ~ 3.40E+38	0.0	F
double	双精度浮点数	8	4.940E-324 ~ 1.798E+308	0.0	D
decimal	十进制数类型	16	$1.0\times10^{-28} \sim 7.9\times10^{28}$	0.0	M

decimal类型



- 适合财务和货币计算的128位数据类型。有更高的精度和更小的范围。

- decimal myMoney = 100.34m;

【例2-1】根据三边长求三角形面积

算法 利用海伦公式：

$$A=\sqrt{s(s-a)(s-b)(s-c)}$$

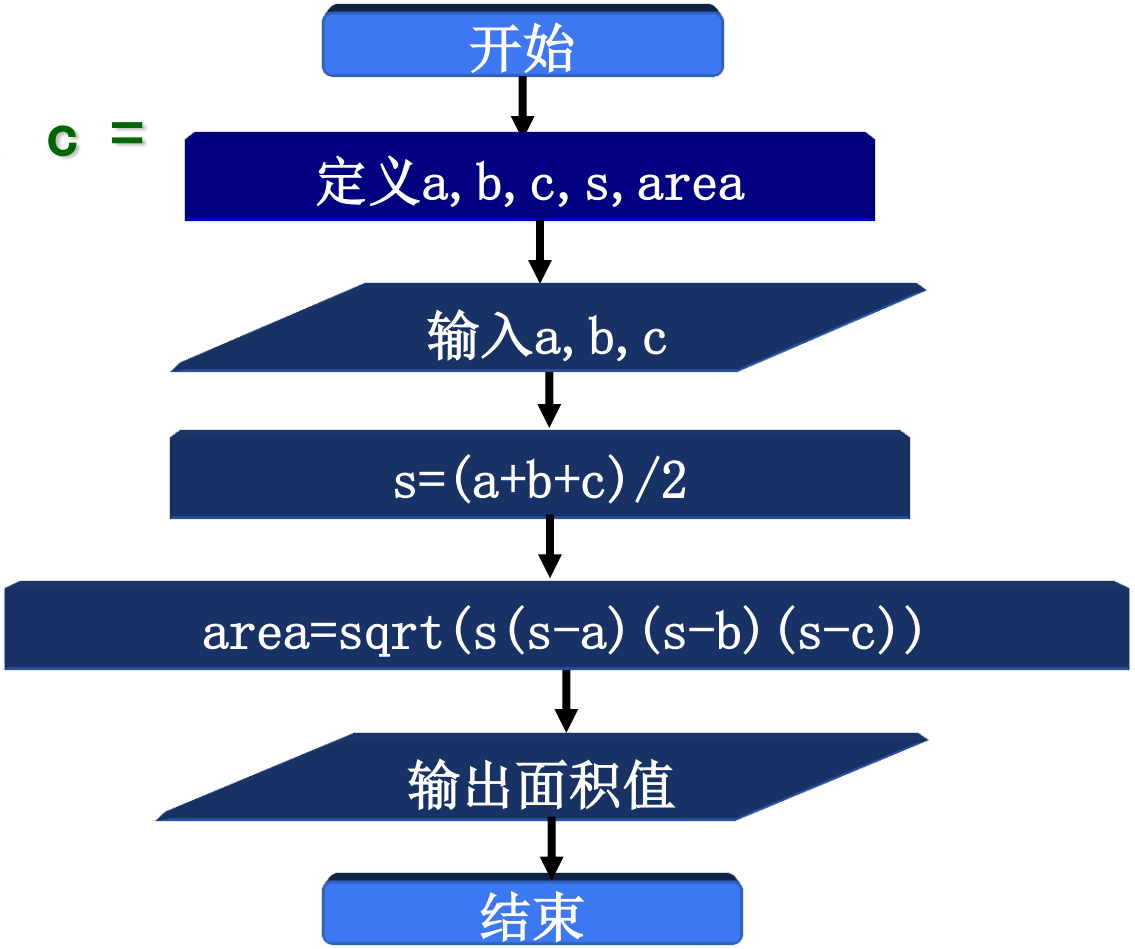
其中a, b, c分别为三角形三条边的长度，。

$$s=\frac{1}{2}(a+b+c)$$

【例2-1】根据三边长求三角形面积

输入输出

```
Please input a, b, c =
3 4 5
area = 6
```



【例2-1】根据三边长求三角形面积

```
1.  using System;
2.  class My{
3.      static int Main()    {
4.          double a, b, c, s, area;
5.          Console.WriteLine("Please input a, b, c =");
6.          string str1 = Console.ReadLine();
7.          string []split=str1.Split (' ');
8.          a = Convert.ToDouble(split[0]);
9.          b = Convert.ToDouble(split[1]);
10.         c = Convert.ToDouble(split[2]);
11.         s = (a + b + c) / 2;
```

【例2-1】根据三边长求三角形面积

```
1.          area = Math.Sqrt(s * (s - a) * (s - b) * (s - c));
2.          Console.WriteLine("area = {0}", area);
3.           return 0;
4.      }
5.  }
```

2.2 常量

- **整型常量**
  - 8进制常量，例04400，0777，0100
  - 10进制常量，例2304
  - 16进制常量，0x900，0xABC，0xffff

- **实型常量**
  - 0.0，–2.68, 3.141593, 637.312, 32767.0, –32768.0,…
  - 0.0E0, –6.226E–4, 1.267E20, …

2.2 常量

- 字符型常量
  - 符号常量，例#define PI 3.1415926
  - 字符常量，例'a', 'A', '1', ' ', '+',
  - 转义常量 '\n'(换行), '\r'(回车), '\t'(横向跳格), '\''(单引号), …
- 字符串常量
  - "Visual C++", "12.34", "This is a string.\n", …
  - string b = @"hello, world";
- 符号常量
  - const double PI=3.1415926;

表2-1 常用转义字符

转义序列	含义
\\	\ 字符
\'	' 字符
\"	" 字符
\?	? 字符
\a	Alert 或 bell
\b	退格键（Backspace）
\f	换页符（Form feed）
\n	换行符（Newline）
\r	回车
\t	水平制表符 tab
\v	垂直制表符 tab
\ooo	一到三位的八进制数
\xhh . . .	一个或多个数字的十六进制数

【例2-2】大小写转换

- 输入一个字符，判断它是否为大写字母，如是，将其转换为对应的小写字母输出；否则，不用转换直接输出。

- 输入输出
  - 请输入一个字母:
  - D
  - 将大写转换为小写后，该字母为：d

【例2-2】大小写转换

```
1.  class My{
2.      static int Main()   {       char ch;
3.          Console.WriteLine("请输入一个字母: ");
4.          ch = Convert.ToChar(Console.Read());
5.          ch = char.ToLower(ch);
6.          Console.WriteLine("将大写转换为小写后，该字母为："
7.  + ch);
8.          return 0;
9.      }
10. }
```

Console.Read()方法

- Read方法从标准输入流读取下一个字符。
- 其返回值是输入流中下一个字符的Unicode编码值，返回值类型是System.Int32；
- 如果当前没有更多的字符可供读取，则返回-1。
- 当用户按Enter 键时该方法才会终止。

- 其方法定义为：
- public static int Read();

C# 运算符

- 算术运算符
- 关系运算符
- 逻辑运算符
- 位运算符
- 赋值运算符
- 杂项运算符

位运算符

运算符	描述
&	如果同时存在于两个操作数中，二进制 AND 运算符复制一位到结果中。
|	如果存在于任一操作数中，二进制 OR 运算符复制一位到结果中。
^	如果存在于其中一个操作数中但不同时存在于两个操作数中，二进制异或运算符复制一位到结果中。
~	二进制补码运算符是一元运算符，具有"翻转"位效果，即0变成1，1变成0。
<<	二进制左移运算符。左操作数的值向左移动右操作数指定的位数。
>>	二进制右移运算符。左操作数的值向右移动右操作数指定的位数。

杂项运算符

- sizeof：返回数据类型的大小
- typeof ： 返回 class 的类型
- ？:如果条件为真？则为 X : 否则为 Y
- is判断对象是否为某一类型
- as强制转换
- 逗号运算符
  - var = (count=19, incr=10, count+1);

【例2-3】 解一元二次方程程序

- 输入输出
- Please intput a, b, c =
- 1 6 9
- x1 = -3,x2 = -3

【例2-3】 解一元二次方程程序

```
1.  class My {
2.      static int Main()    {
3.          double a, b, c, delta, p, q;
4.          Console.WriteLine( "Please intput a, b, c = " );
5.          string str1 = Console.ReadLine();
6.          string []split=str1.Split (' ');
7.          a = Convert.ToDouble(split[0]);
8.          b = Convert.ToDouble(split[1]);
9.          c = Convert.ToDouble(split[2]);
10.         delta = b * b - 4 * a * c;
11.
```

【例2-3】 解一元二次方程程序

```
1.          p = -b / (2 * a);
2.        q = Math.Sqrt(Math.Abs(delta)) / (2 * a);
3.        if (delta >= 0)
4.            Console.WriteLine( "x1 = {0},x2 = {1}" , p + q, p - q)
5.        else
6.        {
7.            Console.WriteLine( "x1 = {0} + {1}i", p, q);
8.            Console.WriteLine( "x2 = {0} - {1}i", p, q );
9.        }
10.       return 0;
11.   }
12. }
```

【例2-4】求绝对值

- 输入输出
- 请输入一个实数：
- -5
- |-5|=5

【例2-4】求绝对值

```
1.  using System;
2.  class My{
3.      static int Main()    {
4.          double x, y;
5.          Console.WriteLine( "请输入一个实数：" );
6.          x=Convert.ToDouble( Console.ReadLine() );
7.          y = x > 0 ? x : -x;
8.          Console.WriteLine( "|" + x + "|=" + y);
9.          return 0;
10.     }
11. }
```

【例2-5】反序输出这四位数

- 输入与输出：
- 请输入一个界于1000与9999之间的数：1234
- 反序输出前的数为：1234
- 反序输出后的数为：4321

【例2-5】反序输出这四位数

- 将n%10的值即个位数字存入c1中
- 将n/10%10的值即十位数字存入c2中
- 将n/100%10的值即百位数字存入c3中
- 将n/1000的值即千位数字存入c4中

【例2-5】反序输出这四位数

```
1.  class My{
2.      static int Main()    {
3.          int n, m, c1, c2, c3, c4;
4.          Console.WriteLine("请输入一个界于与之间的数:");
5.          n = Convert.ToInt32(Console.ReadLine());
6.          Console.WriteLine("反序输出前的数为:" + n);
7.          c1 = n % 10;                    //分离个位数字
8.          c2 = n / 10 % 10;               //分离十位数字
9.          c3 = n / 100 % 10;              //分离百位数字
10.         c4 = n / 1000;                  //分离千位数字
11.         m = ((c1 * 10 + c2) * 10 + c3) * 10 + c4;
```

【例2-5】反序输出这四位数

```
1.          Console.WriteLine("反序输出后的数为:" + m);
2.          return 0;
3.      }
4.  }
```

【例2-6】取整型变量的最低4位

输入与输出：

请输入一个整数:
255
255的最低位对应的十进制数是:15

【例2-6】取整型变量的最低4位

```
1.  using System;
2.  class My{
3.      static int Main()
4.      {
5.          int i;
6.          Console.WriteLine( "请输入一个整数: ");
7.          i=Convert.ToInt32(Console.ReadLine());
8.          Console.WriteLine(  i + "的最低位对应的十进制数是:" +
9.  (i & 0X0F) );
10.         return 0;
11.     }
12. }
```

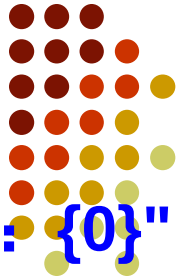
【例2-7】找零钱问题

- 输入与输出：
- 请输入要找给顾客的零钱（以分为单位）72
- 找给顾客的五角硬币个数为：1
- 找给顾客的壹角硬币个数为：2
- 找给顾客的伍分硬币个数为：0
- 找给顾客的贰分硬币个数为：1
- 找给顾客的壹分硬币个数为：0

【例2-7】找零钱问题

```
1.  class My{
2.      static int Main()   {
3.          int change;                    //存放零钱的变量
4.          Console.WriteLine( "请输入要找给顾客的零钱（以分为
5.          change=Convert.ToInt32(Console.ReadLine());
6.          Console.WriteLine( "找给顾客的五角硬币个数为：{0}",
7.          change = change % 50;
8.          Console.WriteLine( "找给顾客的壹角硬币个数为：{0}" ,
9.          change = change % 10;
10.         Console.WriteLine( "找给顾客的伍分硬币个数为：{0}" ,
11.         change = change % 5;
```

【例2-7】找零钱问题



```
1.          Console.WriteLine("找给顾客的贰分硬币个数为：{0}" ,c
    / 2);
2.          change = change % 2;
3.          Console.WriteLine("找给顾客的壹分硬币个数为：{0}"
    ,change);
4.          return 0;
5.      }
6.  }
```

结　束　语

- 学好程序设计语言的唯一途径是

- 你的编程能力与你在计算机上投入的时间成

正比