

C# 基础知识 (五).变量类型和字符串处理

原创 Eastmount 最后发布于2014-08-23 00:18:37 阅读数 4724 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



Eastmount

¥9.90

去订阅

这篇文章是阅读《C#入门经典(Beginning C#)》书籍里面的内容,作者Karli Watson.主要包括自己缺乏的一些C#基础知识和在线笔记使用,文章主要包括C#简单变量类型和复杂变量类型、命名规则、隐式转换和显示转换、变量字符串处理等内容,都是非常基础的知识,希望对初学者有所帮助。

一. C#简单变量类型和命名规则

简单类型组成应用程序中基本构件的类型,其中主要有以下类型:

整数类型	sbyte(-128~127之间整数) byte(0~255之间整数) short(-32768~32767之间整数) ushort(0~65535之间整数) int(4字节,1Byte=8bit) uint(0~2^32-1之间整数) long(别名System.Int64) ulong(别名System.UInt64,u是unsigned缩写,非负数)
浮点数类型	float double(+/- m*2^e) decimal(+/- m*10^e)
3种简单类型	char(一个Unicode字符,存储0~65535之间整数) bool(布尔值true/false) string(一组字符)

C#变量命名规则:变量名的第一个字符必须是字母、下划线(_)或@,其后的字符可以是字母、下划线或数字。

命名约定:以前用Hungarian表示法-在变量名上加上小写形式的前缀,表示其类型,如int型iAge.但由于C# .NET Framework类型复杂,最好根据变量的作用命名它们,目前.NET Framework命名约定PascalCase和camelCase.在名称中使用大小写表示它们的用途,由多个单词组成。

PascalCase指定名称中每个单词除第一个字母大写外,其余小写.如Age/LastName/WinterOfDiscontent.

camelCase第一个单词以小写字母开头,如age、firstName、timeOfDeath.Microsoft建议对于简单变量使用camelCase规则,较高级命名使用PascalCase.建议大家在命名时尽量使用这种方法,方便他人阅读。

注意:下划线字符分割变量名中单词如month_time已经被淘汰。

逐字指定字符串:两个双引号之间的所有字符都包含在字符串中,包括行末字符和转义字符.唯一例外是双引号字符的转义必须指定,以免结束字符串,可以在该字符串前加@。

逐字指定的字符串在文件名中非常有用(反斜杠字符\),前面做过的工程都涉及磁盘目录、图片路径,可见其用途极为广泛.即: @"C:\Temp\MyDir\MyFile.doc" = "C:\\Temp\\MyDir\\MyFile.doc".

位运算包括&按位与 |按位或 ^按位异或 ~取反 <<左移 >>右移。

二. 隐式转换和显示转换

隐式转换:从类型A到类型B的转换可以在所有情况下进行,执行转换的规则简单,可以让编译器执行转换。

显示转换:从类型A到类型B的转换只能在某些情况下进行,转换的规则比较复杂,应进行某种类型的处理。

其中隐式转换规则是类型A取值范围完全包含在类型B的取值范围内,就可以转换为类型B.其中byte可以转换为short/int或float可以转换为double.short类型变量可以存储0~32767而byte可以存储的最大值255,所以当short转换byte时如果在256~32767就会出错。

而显示转换即明确要求编译器把数值从一种数据类型转换为另一种数据类型,其中编写代码的格式随着转换方法而异.相当于"我已经知道你对在这么做提出了警告,但我将对其后果负责".如:

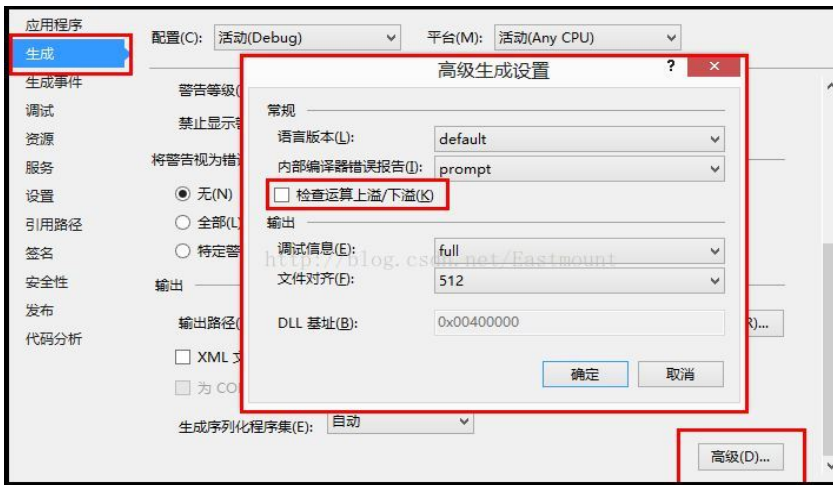
```
byte n; short m = 7; n = m;
```

它将会报错:无法将类型"short"隐式转换为"byte",存在一个显示转换(是否缺少强制转换?)

此时需要显示转换,最简单的方法是把short变量强制转换为byte,如n=(byte)m;但需要注意m大于255时会产生数据丢失。

C#提供表达式溢出检查上下文.使用checked和unchecked关键字,如n=checked((byte)m); 当m=281时会报错信息"算术运算导致溢出".

PS:配置程序表达式都包含checked,右击"解决方案资源管理器"项目->属性->生成->高级(Advanced)->选中"检查运算上溢/下溢",如下图所示:



另一种显示转换方法是使用Convert命令进行显示转换,要转换为int使用Convert.ToInt32(); 转换为字符串使用Convert.ToString(boolVal);结果输出"true/false".这种我在使用字符串和设置编码格式时经常使用.

三. 复杂的变量类型:枚举 结构 数组

复杂的变量类型主要包括:

枚举:变量类型,用户定义一组可能的离散值,这些值用人们理解的方式使用

结构:合成的变量类型,由用户定义的一组其他变量类型组成

数组:包含一种类型的多个变量,可以以索引方式访问各个数值

1. 枚举

有时希望变量提出的是一个固定集合中的值,如week类型存储星期7天,month存储12个月份.枚举允许定义一个类型,其中包含提供的限定值集合中的一个值,如枚举类型orientation可以存储north/south/east/west值.枚举默认类型是int,默认情况每个值都会根据定义的顺序(从0开始),自动赋值给对应的基本类型.没有赋值的任何值都会自动获得一个初始值,该值比上一个明确声明的值大1.举例:

```
namespace test
{
    // 定义枚举类型并指定每个枚举的实际值
    enum orientation : byte
    {
        north = 1,
        south = 2,
        east = 3,
        west = 4
    }
    class Program
    {
        static void Main(string[] args)
        {
            // 声明枚举类型变量
            orientation myDirection = orientation.south;
            Console.WriteLine("myDirection = {0} {1}", myDirection, (int)myDirection);
            Console.ReadKey();
        }
    }
}
```

PS:该程序输出结果为: "myDirection = south 2",实际项目中枚举应用如在制作画图软件时,自定义枚举类型UserSelect变量,然后判断其if (iUserSelect == (int)UserSelect.SELECT)鼠标选择的是线段、矩形、选中状态等.而C语言中经常考察枚举类型的赋值、占用空间地址与结构进行对比等题目.

2. 结构

struct,structure缩写.结构就是由几个数据组成的数据结构,这些数据可能具有不同的类型.根据这个结构可以定义自己的变量类型.如学生信息(学号+姓

名)、存储一段路程(方向+距离),假定东南西北四个方向.通常方法是定义:orientation myDirection; double myDistance;但是如果存储多个路径时定义变量比较麻烦,所以引入结构struct.主要包括结构体数据成员,其格式"<accessibility> <type> <name>;修饰 类型 名称".举例:

```
namespace test
{
    // 定义枚举类型并指定每个枚举的实际值
    enum orientation : byte
    {
        north,
        south,
        east,
        west
    }
    // 定义结构类型
    struct route
    {
        public orientation direction; // 方向
        public double distance;      // 距离
    }
    class Program
    {
        static void Main(string[] args)
        {
            // 定义结构类型变量及赋值
            route myRoute;
            myRoute.direction = orientation.west;
            myRoute.distance = 3.14;
            Console.WriteLine("direction = {0} distance = {1}", myRoute.direction, myRoute.distance);
            Console.ReadKey();
        }
    }
}
```

运行结果为"direction = west distance = 3.14".

3. 数组

数组是一个变量的索引列表,存储在数组类型的变量中,存储类型相同的多个值.与C略有不同,C#声明数组的方法是:<baseType>[] <name> 如:int[]

myArray;

初始化的两种方法:

int[] myArray = {1,2,3,4,5};或者int[] myArray = new int[5];

其中new显示初始化数组大小为5,数组元素默认值0.如 Console.WriteLine("aArray = {0} , bArray = {1}",aArray[3],bArray[2]);输出数字"aArray = 4 , bArray = 0".数组索引从0开始,定义Name[5]即索引为0-4.多维数组的声明如double[,] length = new double[3,4]表示3*4的数组.这些基础知识就不在强调了,应该大家都会.

四. 字符串处理

<string>.ToCharArray()获取一个可写的char数组,该字符串数组中存放当前字符串的所有字符,将字符串复制到指定字符串数组中.如:string str = "This is a string"; char[] chs = str.ToCharArray();

<string>.Length获取元素的个数,给出字符串的字符数.

<string>.ToLower|<string>.ToUpper()把字符串转换为小写形式和大写形式,在比较或赋值时使用,如:if(useResponse.ToLower() == "yes").

<string>.Trim()删除输入字符串的空白字符,注意是去掉首尾空格.如:char[] chs = { ' ', 'e', 's' }; string str = " This is a Yeeees "; str = str.Trim(chs);输出"This is a Y".从字符串的前面或后面删除所有空格、字母e和s 而中间的空格\e\s不变.

<string>.TrimStart()|<string>.TrimEnd()把字符串前面或后面的空格去掉.

<string>.PadLeft()|<string>.PadRight()字符串的左边或右边添加空格,使字符串达到指定长度.该方法用于列中对齐字符串,数据库查询显示信息时常用.如: str = "abcdefg"; str = str.PadLeft(10); //输出" abcdefg" str = str.PadRight(10,'X'); //输出"abcdefgXXX".

<string>.Split()把string字符串转换为string数组,在指定位置分隔并删除分隔符,下面是使用空格分隔string str = "This is a string!" char[] chs

```
= {' '}; string[] words; words = str.Split(chs); //输出This is a string!四个单词
```


PS:还有很多字符串处理函数作者不在续写,用户可以键入代码第一个字符时,IDE帮助用户提供输入关键字、变量名、类型名等,即IntelliSense功能(智能感应).而且查询用法和关键字时按住Ctrl键可以继续阅读被挡住部分代码(变透明).这个Ctrl我也是丛书中知道.

总结:该文章主要是关于C#基础知识的变量类型和字符串处理,其中主要参照书籍《C#入门经典》,推荐大家学习C#的入门书籍.写这种文章感觉有时候自己想得挺多的,可能也有点水,但问心无愧.其实我也更愿意写些实际项目的东西或一些项目经验、项目真正的技术文章,但由于自己的阅历不够、项目能力还一般,还需要继续学习实践,而这种基础文章更多是在线笔记!有时写起来有点淡淡的伤感.希望以后自己能沉下心来真正去学些东西、做些东西,但现在总是有点迷茫,感觉很多东西需要学习又有些乏力,不管如何,先脚踏实地的学些东西,沉浸到代码中去,慢慢寻找自己的人生路吧!自己给自己鼓励,如果有错误或不足之处还请海涵!有建议或教导也可以提出来分享~

(By:Eastmount 2014-8-23 夜12点 原创CSDN <http://blog.csdn.net/eastmount/>)

👍 点赞 5 ☆ 收藏 ➦ 分享 ...



Eastmount  博客专家

发布了450 篇原创文章 · 获赞 6243 · 访问量 497万+

他的留言板

关注