Pascal风格[命名空间名称、类、函数、属性、文件夹命名]

每个单词首字母大写，其余字母均小写。如：HelloWorld

Camel风格[参数、变量]

第一个单词首字母小写，其余单词首字母大写。

## 数据类型

### 值类型[栈空间]

简单类型[1)、2)、3)、4)]、结构类型、枚举类型

1. **整数类型**（8种）

* 有无符号

sbyte/byte(有/无符号8位整数）

short/ushort:16位整数

int/uint:32位整数（若整数未声明，则默认为int类型）

long/ulong:64位整数

* 字符类型（Unicode字符集，一个字符占2字节）

char c1=’x’; //字符常量是用**单引号**括起来的一个字符

注：C#中用string表示字符串，不需要char数组表示字符串

1. **浮点数类型（实数类型）**

单精度(float):32位、7位有效 如：3.14F

双精度(double):64位浮点数、15~16位有效。（未声明时，默认double)

* **转移字符**

\’（单引号）；\0（空字符）；\a（感叹号）；\b（退格）；\e（取消）

\f（换页）；\n（换行）；\r（回车）；\t（水平制表符/横向跳格）\v（垂直制表符）;

1. **Decimal（十进制）类型**

128**高精度**十进制数表示法，7位有效

Decimal d=2.718M（或m）

1. **布尔类型（bool)**

**（适于逻辑运算，一般用于程序流程控制）**

占用1字节内存，结果为true或false

如：bool flag=true; //正确

bool flag=1; //错误，不能将一个整型数据赋给布尔类型的变量。

### 引用类型（类似于指针）[堆]

字符串、类、接口、委托、数组

声明字符串变量：string name=”TOM”

## 数据类型间转换

### 隐式转换

自动类型转换：低精度类型会自动转换为较高精度的类型。

如：double d=2; 2的精度低于变量d的精度，所以2会自动转换为2.0然后赋给d

如：int i=3.0; 变量i的精度低于3.0，但是i已经声明为int型的变量，（变量的值可以变，变量的类型不能变来变去）所以该命令会出错

（系统默认的，不需要加以声明就可进行的转换）

如：short st=250;

int i=st; //将短整型隐式转换成整型

1. 数值类型数据间的隐式转换
2. var类型数据隐式转换

var定义的数据类型由赋值的数据决定

如： var Name=”Johnson” 变量Name就是字符串类型进行了隐式转换

var intNum=250;

int i=intNum; //var型变量intNum隐式转换成整型

var Name=”Johnson”；

string strName=Name; //var型变量Name隐式转换成string型

float不能隐式转换成int类型

### 显示转换

（强制类型转换，需要用户明确地指定转换的类型）

double型强制转换为int型将失去小数部分

如： float pi=3.14f

Int i=(int)pi; //将单精度强制转换为整型来计算，i值变为3，造成数据丢失。

### Convert类转换

ToBoolean、ToChar、ToDateTime、ToDecimal、ToDouble、ToInt16、ToInt32、ToString

如：String MyString=”true”;

Bool MyBool=Convert.ToBoolean(MySting); //将String转换为Boolean型，MyBool=true

### 数值和字符串之间的转换

* ToString()方法：将数值类型数据转换为字符串

如：int num1 = 25;

string str1 = num1.ToString(); //num1的ToString()方法将num1转换为string赋给str1

* **Parse()方法**：将字符串转换为数值型

如：字符串转换为整型：int.Parse(string)

字符串转换为双精度浮点型使用double.Parse(string)

### 装箱和拆箱

拆箱：把“引用”类型转换成“值”类型

装箱：把“值”类型转换成“引用”类型

## 循环结构

### for循环

for(表达式1；表达式2；表达式3)

{

循环体

}

1. 计算表达式1，为循环变量赋初值。
2. 计算表达式2，检查循环条件，当表达式2的值为true,则执行一次循环体；若其为false,跳出循环
3. 执行完一次循环体后，计算表达式3，对控制变量进行增量或减量操作，再重复第（2）步操作。

### while循环

while(表达式)

{

循环体

}

### do-while循环（直到型）

do

{

循环体

}while(表达式)；

## 数组

### 一维数组

定义：数据类型[ ]数组名 如：int[ ] a

初始化：数组名 = new 数据类型[数组元素个数]{数组元素列表} 如a = new int[5]{1,2,3,4,5}

（初始化时若给出数组元素的初值列表，则可省略数组元素个数。如：a = new int[ ]{1,2,3}

或：数组名 = 数据类型[数组元素个数] 或a = new int[5]

定义的同时完成初始化：int[ ] a = new int[3]{1,2,3}

简化为：int[ ] a = {1,2,3} //只有定义的同时

数组属性：Length 表示数组的长度（即数组中的元素个数），类型整形。调用方法： 数组名.Length

### 二维数组

定义：数据类型[ , ]数组名； 如：int[ , ] b //定义二维整形数组

初始化：数组名 = new 数据类型[行数，列数]{数组元素列表} 如：b = new int[2,3]{{1,2,3}，{4,5,6}}

或：数组名 = new 数据类型[行数，列数] 如：b = new int[2,3]

定义的同时完成初始化：int b = new int[2,3]

.