C#基础补充。

2020-5-14 3:38 | 82 | 0 | C#

C#一些基础知识

创建.NET程序的步骤。

使用某种.net (如C#) 语言编写程序。
将代码编译成CIL (通用中间语言–CIL) ，存储在程序集中。
在执行代码时，首先必须 使用JIT编译器 (即时编译器，Just-In-Time) 将代码编译为本机代码。
在托管的CLR环境 (公共语言运行库 (common language runtime,CLR)) 下运行本机代码，以及其他应用程序或者进程.

C#的应用

桌面应用程序
Windows store应用程序
云/Web应用程序Web API — 这是建立REST风格的HTTP服务的理想框架，支持许多客户端，包括移动设备和浏览器。
WCF服务。

用VS编写C#入门

VS的各个组成部分

1. Program中的各个组成部分
引用命名空间：
using System;
using System.Collections.Generic;
using System.Linq;项目名：namespace类：class函数和方法：
main函数是程序的主入口
其他地方可以写方法

2.两种文件格式

源文件夹里的文件：

.sln：解决方案文件。里面包含着整个解决方案的信息，可以双击运行。

csproj：项目文件，包含着项目的信息

3.两行代码作用

函数代码

Console.WriteLine(“HelloWorld”); // 在控制台输出双引号内的内容

Console.ReadKey(); // 让当前程序程序暂停，等待用户输入任意键，显示在控制台当中

4.生成解决方案

菜单栏–>生成–>生成解决方案。

也就是编译，可以检查错误

5.设置解决方案初始启动项目

右键解决方案–>属性–>启动项目 ： 选择 当前启动项目。

6.文本注释

///三个斜线可以启动文本注释

常用快捷键

ctrl+K+D 快速对齐代码

ctrl+z: 撤销

ctrl+s: 保存

ctrl+J 快速弹出智能提示

Shift+End , Shift+Home 控制光标在一行代码的首尾位置。home=begin end=end

ctrl+K+C: 注释所选代码

ctrl+K+U: 取消对所选代码的注释

f1: 转到帮助文档

基础语法

输入输出

Console.ReadLine();

要用一个数据类型接收。string or something输出语句的书写Console.WriteLine(“{0}同学你已经{1}岁了而且你是{2},name,age,sex”);

//{} 表示占位符，索引从0开始

Console.WriteLine(“你叫{0}，你今年{1}岁，你是{2}”,sname,sage,ssex);

Console.ReadKey();

变量命名规则

变量名出现波浪线。

1.如果你的代码中出现了红色的波浪线，意味着你的代码中出现了

语法错误。

2.如果你的代码中出现了绿色的波浪线，说明你的代码语法并没有错误，

只不过提示你有可能会出现错误，但是不一定会出现错误。警告线变量的使用规则
如果你要是用变量的话，应该要先声明再赋值再使用。

给变量起名字的时候要满足两个命名规范：
1、Camel 骆驼命名规范。要求变量名首单词的首字母要小写，其余每个单词的首字母要大写。
多用于给变量命名。
2、Pascal 命名规范：要求每个单词的首字母都要大写，其余字母小写。多用于给类或者方法命名。HighSchoolStudent

占位符

使用方法：先挖个坑，再填个坑。
使用占位符需要注意的地方：
1、你挖了几个坑，就应该填几个坑，如果你多填了，没效果。
如果你少填了，抛异常。
2、输出顺序：按照挖坑的顺序输出。
例如：Console.WriteLine(“{0},你的总成绩是{1}” +
“，平均成绩是{2}”, name, chinese + math + english, (chinese + math + english) / 3);
如果占位符占位的是小数，则{0:0.00}就可以设置保留几位小数

转义字符

转义符指的就是一个‘\’+一个特殊的字符，组成了一个具有特殊意义的字符。
\n:表示换行
\”:表示一个英文半角的双引号
\t:表示一个tab键的空格

\b:表示一个退格键,就是删除一个，放到字符串的两边没有效果。
\r\n:windows操作系统不认识\n,只认识\r\n
\:表示一个\

@符号

1、取消\在字符串中的转义作用，使其单纯的表示为一个'\'
2、将字符串按照编辑的原格式输出

类型转换

隐式类型转换：
我们要求等号两遍参与运算的操作数的类型必须一致，如果不一致，满足下列条件会发生。
自动类型转换，或者称之为隐式类型转换。
两种类型兼容
例如：int 和 double 兼容(都是数字类型)
目标类型大于源类型
例如：double > int 小的转大的显示类型转换：
1、两种类型相兼容 int–double
2、大的转成小的 double—-int
语法：
(待转换的类型)要转换的值;
总结：
自动类型转换：int—->double
显示类型转换：double—>int

Convert类型转换

类型如果相兼容的两个变量，可以使用自动类型转换或者强制类型转换。
但是，如果两个类型的变量不兼容，比如 string与int或者string 与double。
这个时候我们可以使用一个叫做Convert的转换工厂进行转换。
注意：
使用Convert进行类型转换，也需要满足一个条件：面儿上必须要过的去。也就是说要符合基本逻辑。

算数运算符

一元运算符++ ：分为前++和后++，不管是前++还是后++，最终的结果都是给这个变量加一。
区别表现表达式当中，如果是前++，则先给这个变量自身加一，然后带着这个加一后的值去参与运算。
如果是后++，则先拿原值参与运算，运算完成后，再讲这个变量自身加一。
一元运算符–：同上。

一元运算符优先级高于二元运算符。运算的时候优先计算一元运算符的变量。

流程控制

变量的作用域

变量的作用域就是你能够使用到这个变量的范围。
变量的作用域一般从声明它的那个括号开始到那个括号所对应的结束的括号结束。
在这个范围内，我们可以访问并使用变量。超出这个范围就访问不到了。

if else; switch case; while; do while; 和java一样。

关键字Continue(s)

立即结束本次循环然后判断循环条件，如果成立则进入下一次循环，否则退出循环。

调试快捷键

1、F11逐语句调试(单步调试)
2、F10逐过程调试
3、断点调试 : 程序运行到断点出，就不再向下执行了

三元表达式

表达式1?表达式2:表达式3;
表达式1一般为一个关系表达式。
如果表达式1的值为true，那么表达式2的值就是整个三元表达式的值。
如果表达式1的值为false，那么表达式3的值就是整个三元表达式的值。
注意：表达式2的结果类型必须跟表达式3的结果类型一致，并且也要跟整个三元表达式的结果类型一致。

复杂数据类型

枚举

语法：
[public] enum 枚举名
{

值1,
值2,
值3,
.....
}
public:访问修饰符。公开的公共的，哪都可以访问。
enum：关键字，声明枚举的关键字
枚举名：要符合Pascal命名规范。
将枚举声明到命名空间的下面，类的外面，表示这个命名空间下，所有的类都可以使用这个枚举。
枚举就是一个变量类型 ，int，double，string，decimal.
只是枚举声明、赋值、使用的方式跟那些普通的变量类型不一样。我们可以将一个枚举类型的变量跟int类型和string类型互相转换。
枚举类型默认是跟int类型相互兼容的，所以可以通过强制类型转换的语法互相转换。
当转换一个枚举中没有的值的时候，不会抛异常，而是直接将数字显示出来。
枚举同样也可以跟string类型互相转换，如果将枚举类型转换成string类型，则直接调用ToString().
如果将字符串转换成枚举类型则需要下面这样一行代码：
(要转换的枚举类型)Enum.Parse(typeof(要转换的枚举类型),"要转换的字符串");
如果转换的字符串是数字，则就算枚举中没有，也会不会抛异常。
如果转换的字符串是文本，如果枚举中没有，则会抛出异常。

ToString()

所有的类型都能够转换成string类型，调用ToString()。

结构

可以帮助我们一次性声明多个不同类型的变量。
语法：
public struct 结构名
{
成员;//字段
}
变量在程序运行期间只能存储一个值，而字段可以存储多个值。

排序

Array.Sort(); //对数组升序排列
Array.Reverse(); //对数组翻转方法

参数 out、ref、params1

out参数。
如果你在一个方法中，返回多个相同类型的值的时候，可以考虑返回一个数组。
但是，如果返回多个不同类型的值的时候，返回数组就不行了，那么这个时候，
我们可以考虑使用out参数。
out参数就侧重于在一个方法中可以返回多个不同类型的值。
int i ;
int b;

```
int j;
string s = “null”;
bool b = function( s , out i , out b ,out j);
方法：
public static bool function(string s , out int a,out int b ,out int c){
a = 10;
b = 20;
c = 25;
return false；
}
```

2、ref参数

能够将一个变量带入一个方法中进行改变，改变完成后，再讲改变后的值带出方法。

ref参数要求在方法外必须为其赋值，而方法内可以不赋值。

```
int a = 200;
function2(ref a);
//这样输出的a就是300了。
方法：
public static void function2( ref int a){
a = a + 100
}
```

3、params可变参数

将实参列表中跟可变参数数组类型一致的元素都当做数组的元素去处理。

params可变参数必须是形参列表中的最后一个元素。

```
int[] nums = { 1, 2, 3, 4, 5 };
int sum = GetSum(8,9);
Console.WriteLine(sum);
Console.ReadKey();
}
public static int GetSum(params int[] n) {
int sum = 0;
for (int i = 0; i < n.Length; i++) {
sum += n[i];
}
return sum;
}
```

字符串的长度string s = “123456789”;

```
int a = s.length;
Console.WriteLine(a);
```

a==9

C# csharp

暂无评论

发送评论

评论内容

昵称

邮箱 / QQ 号

Markdown

发送

上一篇

VS2017中的快捷键

下一篇

使用git上传文件到github