编程工具 产生编译的代码、被执行 公共语言运行库 基础类

调试器

网站开发技术 ASP。NET

BCL  
BASE CLASS Library基类库：（包含）

通用基础类-文件操作、字符串操作

集合类：列表、字典

线程和同步类-多线程程序

XML 创建、读取以及操作XML文档

编译过程解析：

.Net源文件包括C# VB等

编译器产生程序集（.dll 或者exe）

程序集中包括CIL（中共中间）

JIT（编译器）

本机代码

系统可以直接识别运行的代码

CLR概览：托管代码

程序集

内存管理、垃圾收集、类加载、异常处理、JIT

操作系统

、

1-5面向过程，

6-8面向对象

8-17其他

WinFoms

不讲 数据库

不讲ASP。NET

C#基础

这类函数解决问题

C#本质论

WinForm技术应用

先看一下视频文字介绍，这章讲了什么

一开始枯燥

案例+

函数+参数，慢慢有趣

分解步骤，场景引入

代码的成长

引入场景，问题，需求

数组和循环

视频可以暂停回退

方法和参数

类、继承 接口

委托、Lambda都是引入

多播委托 时间

迭代器，为什么需要迭代器

自定义集合、常用集合

扩展方法，查询运算符（系统提供的东西） 什么是，一步步怎么来的

个人价值判断 学习主线 ETC。编程爱好者

敲代码模仿

建议暂停视频，不要照抄

看到代码的成长，因此时间长

尽量用应用场景引出问题

什么是控制台来

目标：用c#写小黑床 asp.net格式

c#基础 c#基础

Visual studio

Csc。Exe

后期vs

Class HelloWorld

{

Static void Main()

{

String message ;

Message = “这是一个c#程序”；

System.Console.WriteLine(“Message”);

}

}

按钮是事件的拥有者

函数是将一堆

<https://www.runoob.com/csharp/csharp-tutorial.html>

Int a;

Checked{

a = int.MaxValue;

a=a+1;

}

Console.WriteLine();

checked{

int a =int.MaxValue;

a= a+1;

Console.WriteLine(a);

}

int a=1;

long b=1;

a=b;

Console.WriteLine(a);

var str1 = Console.ReadLine();

var str2 = Console.ReadLine();

Console.WriteLine(str1+str2);

Console.Read();

字符串相加 1+2=12

var str1 = Console.ReadLine();

var str2 = Console.ReadLine();

int a = int.Parse(str1);

int b = int.Parse(str2);

Console.WriteLine(a+b);

Console.Read();

int number =10;

//number++;

int result = 10+ --number;

Console.WriteLine(result);

Console.WriteLine(number);

先自减在参与运算

int number =10;

number += 100;

Console.WriteLine(number);

关系预算符

>= <= == !=

&& ||

False 与 true

Switch

If一种

If else 两种

If else if 多种

异常捕获

语法上没错误，但因为程序因某些原因程序出现了错误，不能在正常的运行

Console.WriteLine("请输入一个数字");

int number = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

异常捕获

我们在程序过程中经常会出现各种各样的异常，你如果想要你的程序变得坚强一些。

在你的代码中应该经常性的使用try-catch来进行异常捕获。

那行代码有可能出现异常，你就踹他一脚

语法

Try

{

可能会出现的异常的代码；

}

Catch

{

出现异常后执行的代码；

}、

执行的过程中try过程中的代码没有出现异常。，那么catch中的代码不会执行。

如果try中的代码出现了异常，那么异常后的代码之后一百航都不会执行了。

而是直接跳到catch中执行代码。

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他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，我们可以访问变量，超出这个范围不能使用这个变量。

让代码满足某些条件去执行，使用bool类型

int number = 0;

Bool b= true;

try

{

number = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(number\*2);

}

catch

{

Console.WriteLine("输入内容不能转化数字");

B=false

}

If{b=true}

{Console.WriteLine(number\*2);}

Try和catch之间必须紧密相连，不能有其他代码

李四的年终工作评定，如果定位A级，则工资涨500元，如果定位B级

则工资涨200元，如果定位c级，工资不变，如果定位D级，工资降200元

设李四的原工资为5000，请用户输入李四的评级，然后显示李四来年的工资

Int salary = 5000;

Switch

{

Case 1:

Salary +=500;

Case 2:

Salary +=200;

Case 3:

Salary =Salary;

Case 4:

Salary -=200;

Break;

}

double salary =5000;

Console.WriteLine("请输入对李四四年的登记评定");//a,b,c,d,e

string level = Console.ReadLine();

if(level == "A")

{

salary+=500;

}else

if(level == "B")

{

salary +=200;

}else

if(level == "C")

{

salary =salary;

}else

if(level == "D")

{

salary -=200;

}else

if(level == "E")

{

salary -=300;

}else

{

Console.WriteLine("李四来年的工资是0");

}

Switch - catch

用来处理多条件的的定值的判断

语法：

Switch(变量或者表达式的值)

{

Case 值：要执行的代码；

Break;

Case 值：要执行的代码；

Break;

Case值：要执行的代码；

Break;

Case值：要执行的代码；

Break;

}

If()else{}适合区间范围的判断；

double salary =5000;

Console.WriteLine("请输入对李四四年的登记评定");//a,b,c,d,e

string level = Console.ReadLine();

if(level == "A")

{

salary+=500;

}else

if(level == "B")

{

salary +=200;

}else

if(level == "C")

{

salary =salary;

}else

if(level == "D")

{

salary -=200;

}else

if(level == "E")

{

salary -=300;

}else

{

Console.WriteLine("李四来年的工资是0");

}

执行过程：程序执行到switch处，首先将括号中变量或者表达式的值计算出来，

然后拿这个值依次跟每个Case后面所带的值匹配，一旦匹配成功，则执行该Case所带的代码，一旦执行完成后，遇到break,跳出switch结构。

如果，跟每个case所带的值不匹配，就看当前这个switch-case结构中是否存在default，则执行default.如果没有default,则该switch-case结构中什么都不做。

Double salary = 5000;

Console.WriteLine(“请输入”);

#region 对区域代码块的注释。

double salary = 5000;

string level = Console.ReadLine();

switch(level)

{

case "A":salary+=500;

break;

case "B":salary+=200;

break;

case "C":

break;

case "D":salary-=200;

break;

case "E": salary-=300;

break;

}

double salary = 5000;

string level = Console.ReadLine();

switch(level)

{

case "A":salary+=500;

break;

case "B":salary+=200;

break;

case "C":salary=salary;

break;

case "D":salary-=200;

break;

case "E": salary-=300;

break;

default:

Console.WriteLine("输入有误");

}

Console.WriteLine(“李四明年的工资是{0}元”，salary);

}