# C#个人总结5则范文

来源：网络 作者：风吟鸟唱 更新时间：2024-07-16

*第一篇：C#个人总结C#调用浏览器的功能实现浅析2024-11-13 来自：网上整理字体大小：【大中小】摘要：C#调用浏览器的功能实现使用了什么方法？C#调用浏览器的功能实现的步骤是什么呢？那么本文就向你介绍C#调用浏览器包括默认和非默...*

**第一篇：C#个人总结**

C#调用浏览器的功能实现浅析

2024-11-13 来自：网上整理

字体大小：【大中小】

摘要：C#调用浏览器的功能实现使用了什么方法？C#调用浏览器的功能实现的步骤是什么呢？那么本文就向你介绍C#调用浏览器包括默认和非默认浏览器的基本情况。

C#调用浏览器是如何实现的呢？我们在实际的开发过程中会遇到这样的需求，那么来实现C#调用浏览器会用到什么方法呢？这里向你介绍了调用IE以及默认浏览器的具体操作。C#调用浏览器之调用IE:

 System.Diagnostics.Process myProcess = new System.Diagnostics.Process(); myProcess.StartInfo.FileName = “iexplore.exe”; myProcess.StartInfo.Arguments = “ http://”;

 try  {  System.Diagnostics.Process.Start(target); }  catch ( System.ComponentModel.Win32Exception noBrowser) {  if(noBrowser.ErrorCode==-2147467259) MessageBox.Show(noBrowser.Message); }  catch(System.Exception other) {  MessageBox.Show(other.Message); } C#调用浏览器的操作就向你介绍到这里，希望对你了解和学习使用C#调用浏览器有所帮助。

F1.Show();与Application.Run(f1);的区别？/??? Application.Run()包含消息循环，直至窗口关闭才返回； Form.Show()将窗体显示出来，显示后就返回

在Main()里把默认的Application.Run(new Form1());换成new Form1().Show()就知道区别了。

private void OnClosing(object sender, FormClosingEventArgs e){ if(MessageBox.Show(“确实要退出吗”, “hello”, MessageBoxButtons.YesNo, MessageBoxIcon.Question)== DialogResult.No)e.Cancel = true;} }

写一个程序,窗口如同浏览器一样打开指定的网站怎么写

满意答案

热心问友 2024-04-24 在主窗体添加一个Textbox，Button，webBrowser,其中TextBox用来输入网址，Button来执行浏览网页操作，WebBrowser用来显示浏览的页面 一个按钮单击事件，一个对话框的KeyPress事件

namespace ie {

public partial class Form1 : Form {

public Form1(){

InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){

Uri address = new Uri(textBox1.Text);//创建一个Uri变量，用来存储浏览的网页地址

webBrowser1.Url = address;//在WebBrowser种显示制定的网页

} private void textBox1\_KeyPress(object sender, KeyPressEventArgs e){

if(e.KeyChar == 13)//按下Enter

{

if(textBox1.Text!= “\")

{

button1\_Click(sender,e);

}

} }

} }

有些没有完善，你自己完善一下，主要功能实现了

产生随机数：

Random.next(int32,int32)这个方法里面的两个参数分别写最小值和最大值，得到的随机数就是这两个参数范围内的值

这是我刚才试的代码 Random r = new Random();int a = r.Next(1, 10);int b = r.Next(1, 10);int c = r.Next(1, 10);int d = r.Next(1, 10);

结果并不是一样的，虽然有时会一样，随机数嘛

类型转换

C 货币 2.5.ToString(”C“)

￥2.50

D 十进制数 25.ToString(”D5“)00025

E科学型25000.ToString(”E“)

2.500000E+005

F固定点25.ToString(”F2“)

25.00

G 常规 2.5.ToString(”G“)

2.5

N 数字 2500000.ToString(”N“)

2,500,000.00 X 十六进制 255.ToString(”X“)

FF

http://www.feisuxsponentModel;using System.Data;using System.Drawing;using System.Linq;using System.Text;using System.Windows.Forms;using System.Threading;using System.IO;

namespace 速度赛跑 { publicpartialclassForm1 : Form

{ public Form1()

{

InitializeComponent();

CheckForIllegalCrossThreadCalls = false;

} privateThread thread1 = null;privateThread thread2 = null;privateThread thread3 = null;bool wanbi;privatevoid thread1Show()

{ while(true)

{

double n = 1;Random r = newRandom();int a = r.Next(1, 100);

progressBar1.Maximum = 1000;

progressBar1.Minimum = 0;

progressBar1.Value = 0;

progressBar1.Step = a;if(wanbi == true)

{ for(int m = 0;m -1 && idx -1 && idx > 关系操作符 = is 等于操作符 == 逻辑与

& 逻辑异或 ^ 逻辑或

| 条件与

&& 条件或

|| 条件操作符 ?: 赋值操作符 = \*= /= %= +=-= >= &= ^= |= 所有的二元操作符除赋值符外都是左联合的，即从左到右计算。

typeof()运算符可以从一个类名得到一个System.Type对象，而从System.Object对象继承来的GetType()方法则可从一个类实例来得到一个System.Type对象。例： Type t1 = typeof(Apple);//Apple是一个类名

Apple apple = new Apple();//apple是Apple类的一个实例 Type t2 = apple.GetType();//t1与t2是相同的 通过反射得到一个类的所有成员和方法： Type t = typeof(Apple);string className = t.ToString();//得到类名

MethodInfo[] methods = t.GetMethods();//得到所有方法 foreach(MethodInfo method in methods){ //用method.ToString()得到方法名 } MemberInfo[] members = t.GetMembers();//得到所有成员 foreach(MemberInfo member in members){ //用member.ToString()得到成员名 } sizeof()操作符用来计算值类型变量在内存中占用的字节数（Bytes），并且它只能在unsafe（非安全）

代码中使用。例：

static unsafe public void ShowSizes(){

int i, j;

j = sizeof(short);

j = sizeof(i);} 尽可能使用复合赋值操作符，它比不用复合赋值操作符的效率高。for语句的语法为：

for(initialization;Boolean-expression;step)

embedded-statement 在initialization和step部份还可以使用逗号操作符，例： for(int i = \'0\', j = 1;i 取成员

例：using System;class UnsafeApp {

public static unsafe void GetValues(int\* x, int\* y)

{

\*x = 6;

\*y = 42;

}

public static unsafe void Main()

{

int a = 1;

int b = 2;

GetValues(&a, &b);

} } fixed语法为：fixed(type\* ptr = expression)statements其中type也可以为非控类型，也可是void；expression是任何产生一个type指针的表达式；statements是应用的代码块。例： fixed(int\* f = &foo.x)//foo是Foo类的一个实例，x是Foo类的一个int属性 {

SetFooValue(f);//SetFooValue方法的定义为unsafe static void SetFooValue(int\* x)} 传统的COM组件可以通过互操作层(COM Interop)与.NET运行环境交互。互操作层处理在托管运行环境和非托管区域中的COM组件操作之间传递所有的消息。

要使COM组件能在.NET环境中使用,必须为COM组件生成元数据。.NET运行环境用元数据层业判断类型信息。在运行时刻使用类型信息，以便生成RCW(Runtime Callable Wrapper,运行时可调用包装)。当.NET应用程序与COM对象交互时，RCW处理对COM对象的装载和调用。RCW还完成许多其他的工作，如管理对象标识、对象生存周期以及接口缓冲区。对象生存周期管理十分关键，因为.NET GC把对象到处移动，并且当对象不再使用时，自动处理这些对象。RCW服务告诉.NET，应用程序正与托管.NET组件交互，同时又使非托管COM组件“觉得”COM对象是被传统的COM客户端调用的。

为了为COM组件生成元数据包装，必须使用tlbimp.exe(TypeLib Importer)工具： tlbimp some\_COM.tlb /out:som\_COM.dll

**第三篇：C#实习总结**

现在的社会日新月异，科技飞速发展，但是随之而来的环境问题也日益严重，当然这也有很多其他诸如人口、能源等因素，不过这些让人们越来越重视绿色产业，而软件就是首当其冲的一个。我积极参加了学校组织的软件实习，主要是对.Net方面的实训和实战，一方面想要继续深造之前在大学里面学习的软件方面的知识，另一方面，也是为了自己的职业规划，说直接点，就是以后找份好工作。虽然实训中时间安排有点紧，但很充实，学到了很多之前在大学课堂上没有学到的知识，可以说我从这次实训中的收获很大。

对于计算机编程，我可以说从高中就开始接触，大学从专科开始一直学习的都是软件开发的专业，学过C#、Java，学过开发，也学过测试。但是这次实训让我体会了和之前完全不同的学习感受，不仅是时间安排、或者说效率还是教学授课的方式上，都是和大学课堂完全不同的，不是一节课讲一点，也不是一点一点教给你，这个实训中更多的是培养我们独立动手解决实际问题的能力，分析考虑问题的方法思路，让我们真正成长起来。

学校为我们提供了良好的学习的平台，提供了好工作的面试机会。我们觉得我们更应该充分利用资源，珍惜机会，努力学习，不断地给自己“充电”，这样才能不断的开拓进取，勇于创新，才不至于被社会淘汰。

这次实习中，我们主要有S0、S1、S2三个阶段，S0阶段主要是自学C#基础知识，S1阶段是在老师的指导下学习C#基础编程、C#桌面应用开发、C#数据库应用开发、C#网络应用开发、软件测试和软件工程几个方面的技术方面的课程，S2阶段则是进入项目实战，做实际项目来进行巩固和总结。整个过程中，还有穿插了职业素养课和英语口语课，学技术的同时，我们也注重职业素养的培养和英语口语的锻炼。

到目前为止，我们实习的主要成果有（按时间顺序）：Point24(24点游戏求解小程序)、HandsUp（举手游戏模拟小程序）、SimpleCalculator（简单计算器）、GottaU（捕捉鼠标小游戏）、TypeGame（打字小游戏）、WebBrowser（简易网络浏览器）、Checker（单机对战跳棋小游戏）、MDITextEditor（多文档界面文本编辑器）、FileManager（文件管理器）、SimpleMediaPlayer（简单媒体播放器）等，这些小程序或是项目，虽然和市面的那些软件比起来，在功能强大、界面美观、Bug稀少等方面都有欠缺，但是都是实打实的，我们自己动脑、动手做出来的，当然在做的过程中遇到了不少困难、挫折，但是正因为这些困难和挫折，我们才真正学到了知识和技术。更值得注意的则是，在老师带领我们做项目的过程中，我们从老师的言传身教中学到了很多很多编程思想、解决问题的方法思路等各种“无形”的知识。

众多收获之中，让我感触颇深的，则是老师给我们介绍的“10-90”原则和“Quality-Client-Cost”原则。“10-90”原则的意思是，我们通常都是花10%的代价就可以完成一件事或者一个项目的90%，而剩下的10%，却需要我们花费90%的金钱、精力等各种代价，能不能做到这最后的10%，往往是衡量一个人或者一家公司很重要的一个原则。而且这个原则还是可以嵌套的，最后10%的部分中，也可以再分成90%和10%，同样是开始的90%需要10%的代价，最后的10%需要90%的代价。这个原则不仅适用于做软件、做项目、做公司，其实人生中做人也是如此，很多时候最后的10%是很重要的一把标尺。虽然不能说这个原则放之四海而皆准，但是确实是适用于很多地方、很多方面，让我获益良多。“Quality-Client-Cost”原则讲的则是做软件或是做软件公司的一个原则，就是质量一定要放在第一位，客户在第二位。我自己原本心目中对这三项的排序是Client-Quality-Cost”，但是老师的讲解点醒了我，最具有说服力的例子，就是像微软这样的大公司，都是Quality第一的，他们宁可花费多一些，价格贵一些，但是一定把产品尽善尽美。这个原则同样可以用在人生中，Quality就是人品、品德、素质，Client是朋友、客户、亲人，我们当然应该把个人的品格放在第一位。这些都属于这次实习中，我在技术以外的收获。

另外在实习还有职业素养课中，也学到了很多其他在今后的工作中，要注意的东西，首先要有团队合作精神，现金的大中型软件的编写，分工越来越细，这样在开发、测试的过程中，团队的合作、成员间的交流就变得尤为重要，是决定效率的重要因素；要学会独立解决问题，工作不比在学校学习，遇到的各种实际问题不会有老师专门帮助解决，自己独立分析、解决问题的能力就变得极其重要；要有创新精神，在工作和学习中，如果只是一味的模仿、学习，那就永远不会真正得学得得心应手、融会贯通，必须自己去实践，在实践中创新，这样才能把学来的，变成自己的；要有耐心，学会自我规划和管控，耐心自是不必多说，有耐心才能把事情仔细做好，而公司的管理制度和学校根本上的不同，使得自我规划和管控就成了能否按时圆满完成任务的很重要的条件。

通过这次实习，我在个人素质方面有较大的提高，不仅是在C#的技术方面，还包括面对、分析和处理问题的思路、能力，思维的创造性和全面性，交流和沟通能力，英语口语水平，同时也克服了一些自己的缺点，获得很大进步。

总的来说，这次实习对我有很大意义，不仅巩固和实践了之前在大学课堂上学到的各种知识，扩充了自己对软件技术方面的知识储备，同时也给了我很大的成就感，增强了我的自信。并不是说因为我做成的很难的程序，或者做出来的软件很强很完善，但都是自己认真完成的，过程中有遇到各种困难和挫折，但是经

过网上查资料、小组讨论、向老师请教等几种途径，最终都克服了，当最终自己完成时，总会有一些成就感。

此外，我还人知到在以后的工作和学习中，不仅要努力学习和锻炼专业技能知识，包括C#技术、算法技巧、MVC模式框架的开发、中间技术等等，而且也要注意让自己兴趣广泛起来，拓宽自己的知识面，多积累各种知识，这不仅对以后的软件开发工作有一定好处，对自身的个人修养的提升也是大有裨益。

社会的竞争是激烈的，我想我们应该好好把握住大学学习的时间，充实、完善自我，掌握更多的专业知识，加强实践和设计能力，同时也注意全面发展，这样更有利于将来的发展，在自己的专业领域有所作为。

**第四篇：C#期末总结**

C#期末总结

接触C#已经有半学期时间，对C#感触颇多。C#作为逻辑性极强，功能强大的程序编程语言，仅仅学习半学期是远远不够的，希望自己能在这条路上不断前行，能够活到老学到老。

跟绝大部分的初学者一样，当看见密密麻麻的代码，就觉得这会是一门枯燥无味的课程。但在平时生活中使用各种功能强大的app，应用程序等时，我又会对程序背后的代码感到好奇。就是在这样矛盾的心情中，我一点点初步掌握了C#。

C#中最难理解也是最有意思的莫过于类的使用了。总的来说类是属性相同的方法或对象的集合。编程语言中的类较为抽象，难以理解。编程语言中类的思想在现实生活有许多体现。比如我们通常会把马路上所有的四轮交通工具归为一类----汽车，在生物学中也使用界、门、纲、目、科、属、种把所有生物分级分类。似乎这样想，类很好理解。但在具体编程中会遇到许多问题，比如类成员中各个成员变量的访问权限问题以及类与类之间的逻辑关系问题，等等。解决这些问题，需要对事物和目标有准确的理解和把握，当然扎实的C#功底基础也是必不可缺的。如果能够充分使用类，那么代码的编写会变得井井有条，至少代码会看上去富有生命力。

编程的过程自然枯燥无味，但当看见自己编写的程序运行成功，那种满足感不言而喻。例如本学期的贪吃蛇游戏编程过程，就让我感触颇深。（首先必须承认的一点就是贪吃蛇游戏的主体代码是我借鉴网上大神，光是看懂捋清大神的代码就花了我一星期的时间，接下来就是添加完善代码）首先想要在游戏开始之前添加一个难度选择功能界面，这就必须解决两个问题：一个是怎样控制两个界面（难度选择界面和游戏界面）出现先后次序问题，另一个是如何保存已经选好的难度选择并在游戏中体现出来。对于第一个问题我的解决方案如下：先在speed.cs（难度选择代码区）文件中加入”this.Close();”语句；再在program.cs（程序的主入口）文件中加入”Application.Run(new Speed());”和”Application.Run(new form());”语句。加入”this.Close();”语句的目的是当难度选择完成后让难度选择界面自行关闭；加入”Application.Run(new Speed());”和”Application.Run(new form())”的目的是控制两个界面先后出现的次序。这个问题的解决过程还算顺利，它主要涉及函数调用。第二个问题的解决就比较难，大概耗费我两星期的时间。第二个问题属于在类与类之间调用变量问题。这一类问题在各种编程中会常常遇到。通过查各种资料和不断试验，我的解决方案如下：先添加一个新类PublicClass，然后在PublicClass定义一个静态变量z，再在主代码区调用z。添加新类的目的是使代码简洁明了并方便后期修改，定义变量z的目的是将难度选择值赋给z。这两个问题解决使得贪吃蛇程序能够顺利运行。但顺利运行并不代表这是一个合格的程序，同学反映说这个程序看上去并不是很美观，我意识到还需要在用户体验方面下工夫，没想到这给我带来不小的麻烦。我的计划是用图片代替方块来显示蛇身，但我错误以为只要将定义方块的函数修改成绘画图片的函数就可以了，导致的结果就是程序运行崩溃。我不断翻看所有的代码，一遍又一遍试验，但都找不到解决的办法。就这样过了三四天的时间，我甚至有了放弃的打算。但或许是印证了那句话“山重水复疑无路，柳暗花明又一村”，主代码中有段注释引起了我的注意------”//画出方块并显示在画板上”,这时我才恍然大悟：原来不管是方块还是图片都要定义并使用draw()才能在界面上显示出来，定义方块的函数只不过定义了最基本的方块属性（形状，颜色，大小等）。如果需要更换蛇身图片，只需要在”draw()”中更改就可以了（因为图片是从网上下载，图片的各个属性已固定无需再定义）。最终成功的给蛇换上了一件”新衣服”，程序的用户体验性大大提升。

完成贪吃蛇的编程后，我深深体会到C#是一款面向对象的编程语言。把解决问题的各个因素分门别类，实现分工合作以达到高效率的解决问题。就像现实中的社会分工一样，每个人都在各自的岗位上辛勤劳动，无论工作大小，国家社会离不开每个人的奉献付出。C#中还有其他有趣的控件，比如timer，picturebox，textbox等。灵活组合这些控件，能够编写出很多有意思的小程序。曾经看见网上有C#计算器，C#计时器的教程，觉得很有意思并尝试了一下，但发现很难实现程序，究其原因是因为作者的思维与我的思维有着很大的不同，这说明每个人解决问题的方式不同，而程序便是程序员思维的体现。每个自己编写的程序都是与自己的思维相通的，这么理解的话可以把程序看成自己思维在其他事物上的拓展，其乐无穷，乐此不疲。或许这就是为什么世界上有那么多的天才编程员愿意夜以继日的编写程序，或许这也就是为什么好莱坞每每拍出人工智能统治世界的科幻片总能席卷全球票房。名侦探柯南剧场版《贝克街的亡灵》中的超级人工智能诺亚方舟在天才计算机少年泽田弘树死后代替其意志继续在日本存活下去。想想确实可怕，人工智能究竟会改变人多少。计算机注定改变每个人的生活。因此掌握计算机无疑就是掌握了未来。

由于半学期的学习时间实在太少，想要深刻理解并灵活运用C#是需要多年的实践学习的。光是上课时间认真听讲是远远不够的，还需要课下不断的练习钻研。学习c#语言不能停留在学习它的语法规则，而是利用学到的知识编写c#语言程序，解决实际问题。即把c#语言作为工具，描述解决实际问题的步骤，由计算机帮助我们解题。只有通过上机才能检验自己是否掌握c#语言、自己编写的程序是否能够正确地解题。语法是学习编程语言的关键，光靠背语法，是远远不够的（其他语言学习也是如此）。记笔记是上课时候必须要做的一件事，但不能认为做了笔记就不用上课认真听讲了。C#作为一名理科学科是需要不断深入理解实践才能掌握的。有几点学习心得想分享一下：.不要妄自菲薄。不能因为遇到困难就不相信自己，每个人都是从零开始。2.坚持不懈。学习C#是件漫长的过程，需要不断激励自己鼓励自己。

3.多和他人交流。一个人的力量是渺小的，多个人相互交流不断切磋才能不断进步。4.不要偷懒，经常思考。C#中有许多思想是和现实生活相通的，不断对生活感悟对生活进行反思，从生活中寻找灵感，这样能提高对C#的兴趣以及学习的主动性和趣味性。

5.经常上机实验，动手操作。实践出真知，在实践中理解那些深奥晦涩难懂的专业知识。6.在编程时除了要认真谨慎之外，更重要的是掌握基本的知识，比如if语句和while语句的区别。初学者常常会被这两个语句搞得晕头转向，在编程过程也不会太在意两个语句的使用范围。再比如C#中命名空间，这是常常忽略的方面，初学者一开始编程时会发现定义一个变量或者函数常常会遇到系统提醒错误，开始认真检查下来，却怎么也找不到错在哪，这很有可能就是命名空间错误。

大学之前从未接触编程语言，对于C#的学习也是在一时兴起和怀疑自己中不断转换。平时会去图书馆借些C#书籍，但总有种意犹未尽的感觉。这种感觉或许来自对编程语言的兴趣，希望自己能学到更多关于编程语言的知识。

**第五篇：c#基础总结**

【1】面向对象程序设计语言的三大特点是什么？答：【23】Main函数特点？答：1）Main方法是C#程序的限定，默认的为private【2】三种命名法则的特点？答：匈牙利命名法：在变限定符，返回类型为void或int类型，Main方法必须是静态方法；3)一个类或结构只能有一个有效的入驼表示法：一个标示符用若干个有意义的英语单词或口点函数；4）main方法必须定义在某一个类中。缩写组成，第一个单词首字母小写，后边的首字母大【24】多态（重写、隐藏）

写；Pascal表示法：与骆驼表示法类似，但是第一个 单词的首字母也需要大写。【3】C#注释的三种形式特点？答1）单行注释：// 2）class A /// {public virtual void F()【4】引用类型和值类型的数据类型？ {Console.WriteLine(“A.F”);}} abstract class B:A（1）int valOne = 0;{public abstract override void F();}int valTwo = valOne;答：abstract override 是不可以一起修饰 int valThree = 333;例：在下面的例子里 valTwo = 333;TestValueRefRef1 = new TestValueRef();class A TestValueRefRef2 = Ref1;{public A(){PrintFields();} Ref2.value = 444;public virtual void PrintFields(){} } Console.WriteLine(“values:{0},{1}”, Ref1.value, class B:A Ref2.value);{int x=1;int y;public B(){y=-1;} Console.WriteLine(“values:{0}, {1},{2}”,valOne, public override void valTwo,valThree);PrintFields(){Console.WriteLine(“x={0},y={1}”,答：输出结果：values:444,444 x,y);} 当使用new B()创建B的实例时，产生什么输出？（2）public class EnumTest答：x=1,y=0 { enum Days {Sat=1, Sun, Mon, Tue, Wed, Thu, Fri};分析：执行new B()时，由于B继承自A，所以会调用static void Main()A的构造函数，并执行其中的PrintFields()方法，由{int x=(int)Days.Sun;inty=(int)Days.Fri;于该方法为虚方法，在继承类中被重写，所以，将执Console.WriteLine(“Sun = {0}”, x);行B中的PrintFields()方法。此时，将在DOS界面Console.WriteLine(“Fri = {0}”, y);}} 上输出“x=1,y=0”，然后，在运行B的构造函数中的答：输出结果：Sun = 2Fri = 7 y=-1。（建议同学们将此程序放入到代码中，设置断点【5】枚举类型的字段和关联值？枚举类型有三个要看执行过程。）

【25】什么是多态对象？答：用基类类名声明，但是特性 修饰符 enum 枚举名 ： 基础类型 {枚举成员声明，枚举成员声明，„„，枚举成员声明}默认的基础函数来实例化的对象。这类对象的主要用途是引发多类型为int；关联值：如果没有被声明，默认为0。态，为了将它们和一般的对象（声明和创建都使用同【6】强制类型转换（例：若有double f=2.7;int 一个类型名的对象）加以区别、揭示它们的特点和用2）将源类型的对象途，这种形态的对象称为多态对象。转换成为目的类型的对象 【26】接口的特点。答：接口只能包含抽象方法，不【7】运算符&和&&的区别？答：条件“与”运算符(&&)没有访问修饰符，接口成员必须是方法属性事件或者时才计算第二个操作数。而&需要计算所有操作数，索引器不能包含常数字段运算符也不能有静态成员。并且优先级高于&& 【27】委托和事件，【8】装箱和拆箱的概念？答：装箱就是把一个值类型委托的定义修饰符 delegate 返回类型 委托类型名(参数列表)； 【9】for循环和if语句联合使用的程序分析，for(;;)eg: public delegate int DelegateClass(stringinfo);

和continue的区别？答：break跳出循委托的创建(实例化)委托对象 ＝ new 委托名（关联方法）； 【11】命名空间的特点答：它提供一种命名机制，是eg: DelegateClass obj=new DelegateClass(MethodA);

合方式无关，不能表明源文件的存取方式，命名空间DelegateClass obj=MethodA;//隐式创建和初是按层次组织的。始化(不用new)【12】数组元素的的复制和读值 例：分析下列语句： int[3]{5,6,2},new int[5]{6,9,7,8,3},new Hello(string target);} int[2]{3,2}};myArray3[2][2]的值是（D）A)9；B)2；该语句的作用是：在TestCS 命名空间中定义了了一C)6；D)越界 个名为Hello 的委托类型；

【13】类和对象的关系？答：类是对象的抽象，对象【28】Windows窗体中Button按钮触发的事件是什【14】关键字this和base的区别？答：base指代基【29】Windows窗体中控件的标识符如何修改？答：【15】关键字new、using的多种用法？答：new修饰【30】如何修改Windows窗体的启动窗体？答：修改被重写，但new修饰符可终止这一特性；向下传播； 实例化一个对象。Using：导入命名空间；自动释放【31】要使用SQL Server需要使用哪两个命名空间？ Using代码框里的资源。【16】索引器的特点？答：索引器允许重载；字符串Using System.Date.SqlClient： 【32】什么是DataSet、DataAdapter？两者联系？答：过签名标识；通过元素访问来访问；必须为实例成员；索引器的get访问器具有与索引器相同的形参表；除DataAdapter：数据适配器，数据库与DataSet间的桥value参数外，索引器的set访问器还具有与索引器梁，把数据库中数据下载到DataSet或回传回去。相同的形参表。【33】用户登录和密码修改（带数据库）【17】静态数据成员特点？答：为所有类所共享，区用户登录 【18】构造函数的特点？答：（1）构造函数名与类名UserName=\'“ + txtUsername.Text.Trim().ToLower()+ ”\' and UserPwd=\'“ + txtPassword.Text.Trim()+ 【19】析构函数的特点？答：（1）析构函数名是在类”\'“;if(OperateDB.ExecuteReader(sql))型(默认为空)和修饰符；（3）析构函数不能被重载。{username = txtUsername.Text.Trim().ToLower();【20】什么是方法的重载？重载的特点是什么？答： frmMain frm = new frmMain();frm.Show();this.Hide();} 定义一组方法。重载的特点：1）位于同一类中；2）else

方法名相同；3）方法参考列表不同，包括参数个数不{MessageBox.Show(”用户名或密码错误“, ”出错了“, 同和参数类型不同；4）与方法返回值和修饰符没关系。MessageBoxButtons.OK, MessageBoxIcon.Error);} 【21】虚函数的特点？答：1)虚方法前不允许有修改密码： 修饰符；2)虚方法不能是私有的，因此不能使用private修饰符； where UserName=\'” + frmLogin.username + “\' and 【22】抽象类和抽象方法的主要特点？答：抽象类：UserPwd=\'” + txtOldPwd.Text.Trim()+ “\'”;（或者if(OperateDB.ExecuteReader(sqlCheckPwd))说，不能产生对象。但是，它可以有构造函数。（2）{string sql = “update UserInfo set UserPwd=\'” 设计abstract类的目的是为了被继承。抽象方法：是+ txtNewPwd.Text.Trim()+ “\' where UserName=\'” + 不完整的，不能执行的。frmLogin.username + “\'”;

if(OperateDB.ExecuteNonQuery(sql)== 1)

{MessageBox.Show(“密码修改成功！”);}else

{ MessageBox.Show(“密码修改失败！”);}}

else{MessageBox.Show(“旧密码不正确！”);}

【34】抽象类定义和继承使用

特点：1.没有被完整定义，因而它不能用来实例化，或者说，不能产生对象。(但是，它可以有构造函数。)2.设计abstract类的目的是为了被继承。public abstract class Employee{public virtual void Pay(){ }

public abstract void CalculatePay();} public class HourlyEmployee: Employee

{public override void Pay(){CalculatePay();}public override void CalculatePay(){ }} 【35】接口及继承类的使用

特定功能的抽象成员的集合。一个类可以继承多个接口，从而获得多个行为的描述，将它们组合成新的功能并在类中实现。继承类中必须实现接口中的所有抽象成员。

定义接口的格式：修饰符 interface 接口名：基接口列表 {接口体} 其中，接口体的声明可以包括：接口方法声明；接口属性声明；接口事件声明；接口索引器声明

public delegate void

StringListEvent(IStringList sender);public interface IStringList{ void Add(string s);//方法int Count{get;}//属性event StringListEvent Changed;//事件string this[int index]{get;set;}//索引器} 【编程题例题】

定义一MobilePhone类，包括属性成员——网络类型(NetworkType)，字段成员——屏幕尺寸(screenSize)、手机品牌(brand)，手机型号

(brandModel)，公共方法成员——Open、Close。其中screenSize为单位是英寸的双精度数，brand为字符串，NetworkType只能是“GSM”或“CDMA”字符串。要求：（1）在此类中包含构造函数，构造函数用于对数据（屏幕尺寸、手机品牌和手机型号）进行初始化。（2）公共成员方法中输出相应提示信息（参见（3）中的输出结果格式）。（3）写一测试类，在类中实例化一MobilePhone对象，最后能在DOS界面下显示如下结果：诺基亚N81（屏幕尺寸2.0英寸），是一款GSM手机。手机关机了。using System;

public enum NetworkType {GSM,CDMA,}

public class MobilePhone {public double screenSize;public string brand;

public string brandModel;

public NetworkType networkType;public NetworkType NetworkType{get { return networkType;}}

public MobilePhone(double ss, string bra, string bm, NetworkType nt){screenSize = ss;brand = bra;brandModel = bm;networkType = nt;}public void Open()

{Console.WriteLine(“{0}{1}(屏幕尺寸{2}英寸),是一款{3}手机.”,brand,brandModel,screenSize.ToString(“.0”), networkType);}

public void Close()

{Console.WriteLine(“手机关机了。”);} }

public class Test

{public static void Main()

{MobilePhone mp = new MobilePhone(2.0, “诺基亚”, “N81”, NetworkType.GSM);mp.Open();mp.Close();

System.Console.ReadLine();} }

【例】写一名为Desk的类，包含两个字段Length（双精度类型）、Height（双精度类型）。再写一继承类ComputerDesk类。ComputerDesk类除了有Length和Height外，还有KeyboardTray（字符串类型）。Public class desk {double length;double height;}

Public class computerdesk:desk {string keyboardtray}

本文档由站牛网zhann.net收集整理，更多优质范文文档请移步zhann.net站内查找