

厚道 善良 守信 宽容 诚实 谦虚 正直 执着

首页 联系 管理

随笔- 82 文章- 363 评论- 140 阅读- 71万

《C# 程序设计》期末试卷

《C# 程序设计》期末试卷 <A卷>

班级: _____ 姓名: _____ 学号: _____ 成绩: _____

2022用.NET6

一、选择题 (15×1分=15分)

1、Visual Studio 2005运行在.Net Framework 平台下, 请选择需要的开发框架版本 (B)。

- A. .Net Framework 1.0 B. Net Framework 2.0
C. Net Framework 3.0 D. Net Framework 4.0

2、下列哪一个不是面向对象编程的特征 (C)。

- A. 继承 B. 封装 C. 接口 D. 多态

3、布尔类型中有TRUE和FALSE两种类型, 有如下代码

```
bool b1=true,b2 = false;
```

```
bool b3 =!( b1 && b2);
```

变量b3的值是 (B)。

- A. NULL B. True C. 未知 D. False

4、下列选项中, (C)是引用类型。

- A. enum类型 B. struct类型 C. string类型 D. int类型

5、C#中所有数据类型的基类是 (A)。

- A. Object B. int C. string D. Boolean

6、异常发生在什么时候 (C)。

- A. 编写程序时 B. 编译时 C. 运行时 D. 用户请求时

7、如果有一个类名为MyClass1, 则MyClass1的默认构造函数是下面哪一个 (B)。

- A. new MyClass1(); B. public MyClass1() {}
C. public class MyClass1 D. MyClass1();

8、C#中, 回车字符对应的转义字符串为 (A)。

- A. \r B. \f C. \n D. \a

9、C#中执行下列语句整型变量x和y的值是多少 (D)。

```
int x=100; int y=++x;
```

- A. x=100 y=100 B. x=101 y=100
C. x=100 y=101 D. x=101 y=101

10、在C#编制的财务程序中, 需要创建一个存储流动资金金额的临时变量, 则应使用下列哪条语句 (A)

- A. decimal theMoney; B. int theMoney;
C. string theMoney; D. dim theMoney as double

11、C#中, 新建一字符串变量str, 并将字符串“Tom's Living Room”保存到串中, 则应该使用下列哪条语句 (A)。

- A. string str = "Tom's Living Room";
B. string str = "string str = "Tom's Living Room";
C. string str("Tom's Living Room");
D. string str("Tom's Living Room");

12、在C#中, 表示一个字符串的变量应使用以下哪条语句定义。 (B)

- A. CString str; B. string str; C. Dim str as string D. char * str;

13、数组pins的定义如下: string[] pins = new string[4] {"a", "b", "c", "d"}; 执行下列语句后, 数组pins的值为 (B)。

E-mail:jhtchina@163.com 微信:Jhtwxk

昵称: jhtchina
园龄: 19年6个月
粉丝: 72
关注: 6
+加关注

搜索

找找看

积分与排名

积分 - 217403

排名 - 5119

文章分类 (357)

3G 手机开发(windows
Mobile,J2me,Android)(20)

Asp.Net(30)

Biztalk 2006+Work Flow+windows
service(16)

C#(42)

C++(24)

在C#编写的财务程序中, 存储流动资金金额的临时变量应选择一个适合处理金钱值的类型。通常, 处理货币数据时需要高精度和避免舍入误差, 因此 'decimal' 类型是最合适的选择。

因此, 正确答案是:

A. 'decimal theMoney';

这是因为:

- 'decimal' 类型是专门设计用于金融和货币计算的, 它提供更高的精度和更少的舍入误差问题。
- 'int' 类型用于存储整数, 不适合存储带小数的金额。
- 'string' 类型用于存储文本, 不适合进行数值计算。
- 'dim theMoney as double' 是VB (Visual Basic) 语法, 而不是C#语法。此外, 'double' 类型虽然可以存储小数, 但由于浮点精度问题, 不适合用于货币计算。

更多

文章档案 (363)

2024年6月(29)
2024年4月(1)
2024年1月(1)
2023年11月(2)
2023年10月(1)
2023年8月(1)
2023年6月(3)
2023年5月(9)
2023年4月(2)
2023年3月(5)
2023年2月(3)
2022年12月(5)
2022年8月(1)



- C. "a","b","c","d" D. "e","e","e","d"
- 14、在C#程序中，入口函数的正确声明为：(B)
- A.int main(){.....} B.static void Main(){.....}
- C.static void main(){.....} D.void main(){.....}
- 15、类的析构函数的作用是 (D)
- A.一般成员函数 B.类的初始化
- C.对象的初始化 D.删除对象创建的所有对象

```
using System;
namespace HelloWorldApplication
{
    class HelloWorld
    {
        static void Main(string[] args)
        {
            string[] str = new string[4] { "a", "b", "c", "d" };
            Console.WriteLine(str[0] + " " + str[1] + " " + str[2] + " " + str[3]);
        }
    }
}
```

好友的博客

LiTao's blogs
陈洪岗
范唯削
DuDu
宗瑞朋
xuzhong 的blogs

13、数组pins的定义如下:string[] pins = new string[4] {"a","b","c","d"};执行下列语句后,数组pins的值为 (B) .

```
string[] myArr = pins; myArr[3] = "e";
```

A. "a","b","c","e" B. "a","b","c","d"

C. "a","b","c","d" D. "e","e","e","d"

二、填空题 (15×1分=15分)

- 1、若double k=1234.567;且有程序片段: Console.WriteLine("{0,10:c4}",k),上面程序片段输出结果是 \$1,234.5670。
- 2、a=4>5?1:2,请问a的值为 2。
- 3、double d=-123.56; int i=(int)d; i的值现在是 -123。
- 4、如下代码:
- ```
String[] str1_array=new string[] { "C#", "C++", "Java", "PHP" }
```
- 其中str1\_array[2] 的值是 Java。
- 5、.NET框架的两个主要组件是 公共语言运行时 和 .NET Framework 类库。
- 6、C#中声明命名空间使用的关键字是 namespace，导入命名空间使用的关键字是 using。
- 7、当类实例化时首先执行的是 构造 函数；当实例从内存中销毁前最后执行的是 析构 函数。
- 8、请写出“-10<=x<=10”的C#语言表达式 ((x >= -10) && (x <= 10))。
- 9、C#中实现循环的语句主要有 for、while、foreach。
- 10、一个方法没有返回值，则其类型说明符应该是 void。

## 三、简答题 (4×5分=20分)

- 1、什么是静态方法和实例方法，请举例说明。是照一定的数据模型组织，长期存
- 静态方法使用static 修饰，非静态方法不使用static 修饰。

例如:

```
Public class A
{
 Public static int F1() {...}
 Public int F2() {...}
}
```

程序调用:

```
Int r1=A.F1();
A a=new A();
Int r2=a.F2();
```

- 2、什么是装箱，什么是拆箱请举例说明。

装箱：将值类型隐式转换为引用类型。

例如: int i=2008;object o=i;

拆箱：将引用类型隐式转换为值类型。

```
using System;
namespace HelloWorldApplication
{
 class HelloWorld
 {
 static void Main(string[] args)
 {
 double k = 1234.5;
 int b = 2;
 Console.WriteLine("{0, 10:c4} ||| {1, 10:f", k, b);
 }
 }
}
```

\$1,234.5000 ||| 2

## 推荐排行榜

1. 创建一个强名称密钥文件+ 如何在 Visual C# .NET 中将程序集安装到全局程序集缓存中(3)
2. 如何由XSD自动生成XML和实体类(引用)(1)
3. WinForm中Component Class、User Control及Custom Control的区别和使用(1)

## 最新评论

1. Re:使用XmlTextReader 读取XML  
哈哈,评论也很可爱.博主现在时架构师了么,博主写这篇文章的时候我还在小学,那时候电脑都没见过

--千山洞

2. Re:系统分析师学习计划(60天)  
你好,能把资料发我一份吗?

--habj

3. Re:XtraReports 打印控件的简单使用

推荐用StratoIO打印控件,浏览器和系统的兼容性都很好,而且不会崩溃.参考

--pop8989

4. Re:VB.Net +Access 一个简单的例子  
谢谢 对我有用

--拉马努金

5. Re:VB.Net +Access 一个简单的例子  
谢谢,了解了语句以及连接数据库的整体形式了。

--今朝的晚宴

3、什么是 托管代码 什么是 非托管代码 。

以公用语言运行库为目标的代码为“托管代码”。

不以公用语言运行库为目标的代码为“非托管代码”。

4、你对面向对象程序设计的理解。

继承 封装 多态，然后发挥

#### 四、根据程序给出运行结果（4×5分=20分）

1、使用out参数传递数据，并给出程序的运行结果？

```
Static void outcountsum(int i , int j, out int sum)
{
 j++;
 i++;
 sum =i+j;
}
static void Main(string[] args)
{
 int i=2101;
 int j=101;
 int sum;
 outcountsum(i, j, out sum);
 Console.WriteLine("i=" + i.ToString() + " j=" + j.ToString() + "
sum=" + sum.ToString());
 Console.Read();
}
```

运行结果：

i=2101 j=101 sum=2204

2、字符串常用处理，根据给出的

```
string oldstr1 = "This is a string";
string newstr1 = oldstr1.Replace("string", "STRING");
string newstr2 = oldstr1.Insert(0, "Insert string ");
string newstr3 = oldstr1.Remove(4, 3);
int index = oldstr1.IndexOf("string");
Console.WriteLine(oldstr1);
Console.WriteLine(newstr1);
Console.WriteLine(newstr2);
Console.WriteLine(newstr3);
Console.WriteLine(index);
Console.Read();
```

This is a string  
This is a STRING  
Insert string This is a ..  
This a string  
10

运行结果：

3、虚函数和重写函数。

```
{
 static void Main(string[] args)
 {
 B b = new B();
 A a = b;
 a.F1();
 b.F1();
 a.F2();
 b.F2();
 Console.Read();
 }
}

public class A
{
 public virtual void F1()
 {
 Console.WriteLine("A.F1");
 }
 public void F2()
 {
 Console.WriteLine("A.F2");
 }
}

public class B : A
{
 public override void F1()
 {
 Console.WriteLine("B.F1");
 }
 public void F2()
 {
 Console.WriteLine("B.F2");
 }
}
```

B.F1;  
B.F2

1 2

A.F2  
B.F2

运行结果:

B.F1

B.F1

A.F2

B.F2

A.F2  
B.F2

#### 4、日期时间处理:

```
DayOfWeek week = DateTime.Now.DayOfWeek;
```

变量week的值是: <考试那天的日期 Monday>

```
int dayCount = DateTime.DaysInMonth(2010, 2);
```

变量dayCount的值是: 28



变量dt的值是：2010-07-05

考试的当前日期

```
DateTime adddt = DateTime.Now.AddDays(2);
```

变量adddt的值是：2010-07-07 &lt;考试的当前日期加2天&gt;

```
TimeSpan ts = new TimeSpan(1, 0, 0, 0, 0);
DateTime dt1 = DateTime.Now.Add(ts);
```

变量dt1的值是：

&lt;考试的当前日期加1天，时间部分在考试之间之内&gt;

6/14/2024 10:15:22 AM

## 五、程序设计（3分×10=30分）

1、把10进制整数转化为2进制数。（例如：10进制数15的2进制数是1111）

要求：

1 写出算法 及10进制数15转换为2进制数1111的过程。

2 写方法static string ConverIntToBinary(int n)

3 主程序main()调用静态方法

```
static string ConvertIntToBinary(int n)
{
 string binary = string.Empty; ///保存二进制字符串
 int i = n;
 int m = 0;
 while(i > 1)
 {
 i = n / 2; ///整除
 m = n % 2; ///取模
 binary = m.ToString() + binary; ///添加到binary字符串中
 n = i;
 }
 if(i > 0) binary = "1" + binary; ///如果余数为1，则在binary最前面添加1
 ///
 return binary;
}
```

```
Console.WriteLine(ConvertIntToBinary(15));
```

2、将下面的文本分割出来，统计出现Hello单词的次数。

文本内容如下：

The C# preprocessor is a macro processor that the C# compiler automatically uses to transform your program before actual compilation. The C# preprocessor allows you to define macros, which are abbreviations for longer constructs.

要求：

1 使用split方法，对句子进行分割，并保存单词到数组里面

2 统计出现“C#”单词的次数，并显示出来。

```
string str = "The C# preprocessor is a macro processor that the C# compiler automatically uses to transform your program before actual compilation. The C# preprocessor allows you to define macros, which are brief abbreviations for longer constructs.";
string[] array_str = str.Split(' ');
```

using System;

```
class HelloWorld
{
 static string ConvertIntToBinary
 (int n)
 {
 string binary = string.Empty
 ;
 while (n != 0)
 {
 binary += (n % 2).
 ToString();
 n /= 2;
 }
 return binary;
 }

 static void Pro2()
 {
 string str = "The C#
 preprocessor is a macro processor
 that the C# compiler automatically
 uses to transform your\r\nprogram
 before actual compilation. The C#
 preprocessor allows you to define
 macros, which are brief\r\n
 abbreviations for longer constructs
 .";

 string[] words = str.Split('
 ');

 int cnt = 0;
 for(int i = 0; i < words.
 Length; i++)
 {
 Console.WriteLine(words
 [i]);

 if (words[i] == "C#")
 {
 cnt++;
 }
 }
 Console.WriteLine("出现了 "
 + cnt.ToString() + " 次 C#");
 }

 static void Main(string[] args)
 {
 Pro2();
 }
}
```

```
for (int i = 0; i < array_str.Length; i++)
{
 if (array_str[i] == str1)
 {
 i_Count++;
 }
}
Console.WriteLine("单词C# 一共出现了:"+i_Count.ToString()+"次");
```

3、按照给出的类图，完成代码设计与实现。（12分）

要求：

- 1 根据类图给出的Person、Teacher和Student类完成设计的设计与编码
- 2 根据类图提供的字段实现3个类。
- 3 根据类图提供的属性实现3个类。
- 4根据类图提供的方法ShowInfo()在3个类中完成。
- 5 在主程序Main()中实例化教师Teacher类和学生Student类。

```
/// <summary>
/// Person类，包含姓名、年龄和课程列表等信息，并显示这些信息
/// </summary>
public class Person
{
 protected string name;
 protected int age;
 protected string[] courses;

 /// <summary>
 /// 姓名
 /// </summary>
 public string Name
 {
 get
 {
 return name;
 }
 set
 {
 name = value;
 }
 }

 /// <summary>
 /// 年龄
 /// </summary>
 public int Age
 {
 get
 {
 return age;
 }
 set
 {
 age = value;
 }
 }

 /// <summary>
 /// 初始化姓名和年龄
 /// </summary>
```

```
public Person(string name,int age)
{
 this.name = name;
 this.age = age;
}

/// <summary>
/// 初始化姓名、年龄和课程列表
/// </summary>
/// <param name="name"></param>
/// <param name="age"></param>
/// <param name="courses"></param>
public Person(string name,int age,string[] courses)
{
 this.name = name;
 this.age = age;

 if(courses == null || courses.Length <= 0)
 {
 throw new Exception("参数错误");
 }
 this.courses = new string[courses.Length];

 Array.Copy(courses,this.courses,courses.Length);
}

/// <summary>
/// 显示姓名、年龄和课程列表
/// </summary>
public virtual void ShowInfo()
{
 Console.WriteLine(string.Format("姓名: {0}",name));
 Console.WriteLine(string.Format("年龄: {0}",age));
 Console.WriteLine("课程如下: ");
 ///显示课程
 if(courses == null)return;
 for(int i = 0; i < courses.Length; i++)
 {
 if(i < courses.Length - 1)
 {
 Console.Write(string.Format("{0},",courses[i]));
 }
 else
 {
 Console.WriteLine(string.Format("{0}",courses[i]));
 }
 }
 Console.WriteLine();
}
}

/// <summary>
/// Person类, 包含姓名、年龄、职务和课程列表等信息, 并显示这些信息
/// </summary>
public class Teacher:Person
{
 private string duty;

 /// <summary>
 /// 职务
 /// </summary>
 public string Duty
 {
 get
 {
 return duty;
 }
 set
 {
 duty = value;
 }
 }
}
```

```
/// 初始化姓名、年龄和职务
/// </summary>
/// <param name="name"></param>
/// <param name="age"></param>
/// <param name="duty"></param>
public Teacher(string name,int age,string duty)
 : base(name,age)
{
 this.duty = duty;
}

/// <summary>
/// 初始化姓名、年龄、职务和课程列表
/// </summary>
/// <param name="name"></param>
/// <param name="age"></param>
/// <param name="courses"></param>
/// <param name="duty"></param>
public Teacher(string name,int age,string[] courses,string duty)
 : base(name,age,courses)
{
 this.duty = duty;
}

/// <summary>
/// 显示姓名、年龄、职务和课程列表
/// </summary>
public override void ShowInfo()
{
 Console.WriteLine(string.Format("姓名: {0}",name));
 Console.WriteLine(string.Format("年龄: {0}",age));
 Console.WriteLine(string.Format("职务: {0}",duty));
 Console.WriteLine("课程如下: ");
 ///显示课程
 if(courses == null) return;
 for(int i = 0; i < courses.Length; i++)
 {
 if(i < courses.Length - 1)
 {
 Console.Write(string.Format("{0},",courses[i]));
 }
 else
 {
 Console.WriteLine(string.Format("{0}",courses[i]));
 }
 }
 Console.WriteLine();
}

}

/// <summary>
/// Person类, 包含姓名、年龄、班级和课程列表等信息, 并显示这些信息
/// </summary>
public class Student:Person
{
 private string className;

 /// <summary>
 /// 班级
 /// </summary>
 public string ClassName
 {
 get
 {
 return className;
 }
 set
 {
 className = value;
 }
 }
}
```



```
/// 初始化姓名、年龄和班级
/// </summary>
/// <param name="name"></param>
/// <param name="age"></param>
/// <param name="className"></param>
public Student(string name,int age,string className)
: base(name,age)
{
 this.className = className;
}

/// <summary>
/// 初始化姓名、年龄、班级和课程列表
/// </summary>
/// <param name="name"></param>
/// <param name="age"></param>
/// <param name="courses"></param>
/// <param name="className"></param>
public Student(string name,int age,string[] courses,string className)
: base(name,age,courses)
{
 this.className = className;
}

/// <summary>
/// 显示姓名、年龄、班级和课程列表
/// </summary>
public override void ShowInfo()
{
 Console.WriteLine(string.Format("姓名: {0}",name));
 Console.WriteLine(string.Format("年龄: {0}",age));
 Console.WriteLine(string.Format("班级: {0}",className));
 Console.WriteLine("课程如下: ");
 ///显示课程
 if(courses == null) return;
 for(int i = 0; i < courses.Length; i++)
 {
 if(i < courses.Length - 1)
 {
 Console.Write(string.Format("{0},",courses[i]));
 }
 else
 {
 Console.WriteLine(string.Format("{0}",courses[i]));
 }
 }
 Console.WriteLine();
}
}
```

主程序如下:

```
static void Main(string[] args)
{
 string[] tcourses = new string[] {
 "C#语言","数据结构";
 };
 Teacher teacher = new Teacher("张老师",35,tcourses,"系主任");
 string[] s1courses = new string[] {
 "C#语言","数据结构","离散数学";
 };
 Student s1 = new Student("学生1",20,s1courses,"200701班");
 string[] s2courses = new string[] {
 "C#语言","数据结构","汇编语言";
 };
 Student s2 = new Student("学生2",21,s2courses,"200702班");
 teacher.ShowInfo(); ///显示教师的信息
 s1.ShowInfo(); ///显示学生1的信息
 s2.ShowInfo(); ///显示学生2的信息
 Console.ReadLine();
}
```



博客园cnblogs.com

会员

众包

新闻

博文

AI培训

云市场

好文要顶

关注我

收藏该文

微信分享



jhtchina

粉丝 - 72 关注 - 6

10

10

+加关注





升级成为会员

posted @ 2011-02-20 15:45 jhtchina 阅读(4511) 评论(0) 编辑 收藏 举报


刷新评论 刷新页面 返回顶部

发表评论 升级成为园子VIP会员

编辑 预览

B    

支持 Markdown

 自动补全

提交评论 退出 订阅评论 我的博客

[Ctrl+Enter快捷键提交]

- 【推荐】这场阿里云开发者社区有奖征文活动，期待您出文相助
- 【推荐】三生石上：ASP.NET Core中运行WebForms业务代码
- 【推荐】「指间灵动，快码加编」：通义灵码，再次降临博客园
- 【推荐】凡泰极客：跨越技术“鸿”沟，小程序一键生成鸿蒙App
- 【推荐】阿里云创新加速季，5亿补贴享不停，上云礼包抢先领



编辑推荐：

- 如何安全地访问互联网
- 还在拼冗长的 WhereIf 吗？100行代码解放这个操作
- C#异步编程是怎么回事（番外）
- [WPF] Dispatcher 与消息循环
- 网关限流功能性能优化

阅读排行：

- 「指间灵动，快码加编」：阿里云通义灵码，再次降临博客园
- 面试官：你讲下接口防重放如何处理？
- 如何安全地访问互联网
- 推荐十个优秀的ASP.NET Core第三方中间件，你用过几个？
- .NET借助虚拟网卡实现一个简单异地组网工具



试用水印