**浙江理工大学2021—2022学年第 2 学期**

**《 C#程序设计 》期末试卷（ A ）卷**

**（试题页）**

本人郑重承诺：本人已阅读并且透彻地理解《浙江理工大学考场规则》，愿意在考试中自觉遵守这些规定，保证按规定的程序和要求参加考试，如有违反，自愿按《浙江理工大学学生违纪处分规定》有关条款接受处理。

**承诺人签名： 学号： 班级：**

1. 选择题（共20分，每题1分）

|  |  |
| --- | --- |
|  | 已知有如下代码：  int a = 100;  void Func(ref int b){ }  则以下函数调用正确的是( c )。  A、Func(ref(10\*a)) B、Func(ref 10)  C、Func(ref a) D、Func(ref b) |
|  | 以下数据类型中不可以使用算术运算运算的是( a )  A. bool B. char  C. decimal D. sbyte |
|  | 下列数组初始化语句哪个是错误的? ( d )  A、int[] nums = new int[]{0,1,2,3,4};  B、int[] nums2 = {0,1,2,3,4,5};  C、int[][] num\_1 = {new int[]{0,1},new int[]{0,1,2},new int[]{0,1,2,3}};  D、int[][] num\_2 = {{0,1},{0,1,2},{0,1,2,3}}; |
|  | 有关对foreach(type identifier in name){statement;}语句描述不正确的是( B )  A、可以在foreach语句体中改变迭代变量的值。  B、foreach语句只能用来访问数组元素。  C、其中type和identifier分别表示迭代变量的类型和名称，name表示要迭代的数组。  D、foreach语句能够访问任何实现了IEnumerable接口的实例 |
|  | 关于委托的说法，不正确的是( D )。  A、委托属于引用类型 B、委托用于封装方法的引用  C、委托可以封装多个方法 D、委托不必实例化即可被调用 |
|  | 如果Employee从Person继承，Manager从Employee继承，下面哪条语句是有效的？A  A、Person alice = new Employee();  B、Employee bob = new Person();  C、Manager cindy = new Employee();  D、Manager dan = (Manager)(new Employee()); |
|  | 假设HouseBoat类隐式实现了IHouse接口,显式实现了IBoat接口,下面哪条语句是错误的?B  A、代码可用HouseBoat对象来访问其IHouse成员。  B、代码可用HouseBoat对象来访问其IBoat成员。  C、代码可将HouseBoat对象当作IHouse对象来访问IHouse成员。  D、代码可将HouseBoat对象当作IBoat对象来访问IBoat成员。 |
|  | 声明Action<Order>变量时,该变量代表下面的哪一项? ( C )  A、代表一类方法,不接收参数,返回一个Order对象。  B、代表一类方法,接收一个Order对象,返回一个Order对象。  C、代表一类方法,接收一个Order对象,无返回值。  D、Action类提供的方法,不接收参数,无返回值。 |
|  | 对于字符串实例string str = “Hello”，以下哪个操作是可行的?( A )  A、用foreach(var c in str)遍历所有字符  B、执行str.ToUpper();语句将str中字符转换为大写字母  C、执行str.InsertAt(1, ‘a’);语句将其变换为“Haello”  D、执行char end = str[5];获取其末尾字符 |
|  | 在WPF控件中，最常用的布局控件是( B )  A、GridBox B、Grid  C、Stack D、StackPanel |
|  | 下列关于“方法重载”的描述中，哪个选项是不正确的？( B )  A、方法ConsoleW(int \_value)是方法ConsoleW(string \_value)的重载  B、构造函数不可以重载  C、方法重载即“同样的方法名但传递的参数不同”  D、方法重载可以扩充现有类的功能 |
|  | WPF窗体中有一个年龄文本框 txtAge，下面( D )代码可以获得文本框中的年龄值。  A、int age = txtAge; B、int age = txtAge.Text;  C、int age = Convert.ToInt32(txtAge); D、int age = int.Parse(txtAge.Text); |
|  | 以下关于泛型的叙述中错误的是( D )。  A、泛型是通过参数化类型来实现在同一份代码上操作多种数据类型  B、泛型编程是一种编程范式，其特点是参数化类型  C、泛型类型和普通类型的区别在于泛型类型与一组类型参数或类型变量关联  D、以上都不对 |
|  | 当你需要使用一种数据类型表达同一类事物的不同状态，比如：男人、女人、小孩、老人或者春夏秋冬时，C#中最可靠和直观的解决方案是什么？( B )  A、使用struct结构  B、使用enum枚举  C、使用int类型，用不同的数值{0，1，2……}表示  D、C#中无法通过一种数据类型实现同一事务的不同状态。 |
|  | 在C#程序中，定义如下方法，下面选项中( A )错误的实现对该方法的方法重载。  public string Do(int value, string s){//省略代码}  A、public int Do(int value, string s){//省略代码}  B、public string Do(string s, int value){//省略代码}  C、public void Do(string s, int value) {//省略代码}  D、public void Do(){{//省略代码}} |
|  | 以下关于结构的说法，正确的是( B )  A、结构不可以通过ref或out形参以引用方式传递给函数成员  B、结构是值类型，类是引用类型  C、结构和类一样，均支持继承  D．结构由于没有引用作为中介，总是比类更高效 |
|  | 关于base关键字，下列使用方法错误的是：( C )  A、在子类中，base可以调用父类的构造函数  B、在子类中，base关键字可以访问父类的公共属性  C、在子类中，base关键字不可以调用父类的protected成员  D、在子类中，base关键字不可以调用父类的private成员 |
|  | 对于WPF中XAML文件的功能，说法不正确的是：( A )  A、C#程序能做的事，XAML文件都能做  B、XAML文件中每个控件标签都对应一个对象  C、使用XAML文件是为了将图形界面的设计与事件处理代码区分开  D、C#程序能访问到XAML文件中以x:Name命名的对象 |
|  | 关于LINQ的说法正确的是：( C )  A、创建LINQ查询后，马上执行该查询  B、SQL语句能做的，LINQ查询也能做到  C、可用匿名类型定制查询结果  D、可用array.Where(x>0)搜索出数组中大于0的值 |
|  | 关于集合类型，不正确的是：( A )  A、Set是“集”类，其中的元素按序排列  B、Dictionary是“字典”类，其中存储了键值对  C、对Dictionary中已有的键，执行Add()方法，将抛出异常  D、List对象的空间能够自动扩展 |

1. 判断题（共5分，每题1分）

|  |  |  |
| --- | --- | --- |
|  | C#中对象构建时，先执行字段赋值再执行构造方法。 | ( T ) |
|  | 方法能做到的事，属性也能做到。 | ( F ) |
|  | ADO.NET Entity Framework提供了对象与数据库表的映射。 | ( T ) |
|  | 构造方法应返回true或false以表明对象是否成功构造。 | ( F ) |
|  | WPF中的控件由模板构建，因此其外观可利用模板技术重新定义。 | ( T ) |

1. 程序设计题（75分）

1. 写一个方法string RemoveDuplicates(string str)，它能够将字符串中的重复字符删去，也就是说，它返回的字符串中每个字符都是不一样的（大写字符与小写字符视为不同字符）。同时原字符串中的先后关系应得到保留。举例：“bananas”🡪“bans”, “lalalamama”🡪“lam”。（8分）

2. 定义并实现三角形类：（35分）

首先定义Point3d类，代表一个三维的点，提供：（10分）

* X、Y、Z属性；
* double DistanceTo(Point3d p)方法，计算两个点之间的距离。
* bool Equals(object obj)方法，在两个坐标点距离小于1.0e-6时认为两个点相等；

接着定义Triangle类，代表三维空间中的三角形，包含三个三维点。并提供：（25分）

* 一个无参构造方法和一个有参构造方法；
* 成员方法，输出三条边的长度；
* 只读属性，判断是否合法(valid)；
* 只读属性，计算面积(area), 计算方法：p = (a+b+c)/2, area = ；
* 成员方法，判断是否与另一三角形相邻（至少有一顶点重合，connected）
* 利用IComparable或IComparer接口（可以是泛型接口）对三角形对象实现任意一种比较。

3. 写一个程序，读取《三国演义》文本，并进行统计：（32分）

《三国演义》共120回，每一回都被存储在一个文本文件中，如“第一回 宴桃园豪杰三结义 斩黄巾英雄首立功”存储在”001.txt”，“第二十回 曹阿瞒许田打围 董国舅内阁受诏”存储在”020.txt”，所有文件存储在“Novel”文件夹中。要求：

1）用Lambda表达式为一个Predicate<string>变量isHead赋值，用于判断字符串是否为每回标题。

2）读取“Novel”文件夹中所有文件（001.txt, 002.txt, …120.txt），统计每“回”中刘备的出场次数（出场名称可以是"刘备", "玄德", "先主", "皇叔"等），每行只计一次，但在标题中出现时则不计入。用Dictionary<string, int>记录每回的出场次数。

3）用LINQ查询，统计刘备出场次数超过5次的章回数量，并以“第一回 : 12次”的格式输出。针对Dictionary中的每个元素item，可访问其item.Key得到章回名称，item.Value得到出场次数。

提示：Directory.GetFiles()方法以字符串数组返回目录中的文件名；string的StartsWith()方法判断是否以某个字符或字符串开头，Contains()方法判断是否包含某个字符串，LastIndexOf()方法从后向前搜索获取某个字符的位置，SubString(int startIndex, int length)获取某个子串。

**浙江理工大学2021—2022学年第 2 学期**

**《 C#程序设计 》期末试卷（ A ）卷**

**（答题纸）**

本人郑重承诺：本人已阅读并且透彻地理解《浙江理工大学考场规则》，愿意在考试中自觉遵守这些规定，保证按规定的程序和要求参加考试，如有违反，自愿按《浙江理工大学学生违纪处分规定》有关条款接受处理。

**承诺人签名： 学号： 班级：**

一．选择题（共20分，每题1分）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第1题 | 第2题 | 第3题 | 第4题 | 第5题 |
|  |  |  |  |  |
| 第6题 | 第7题 | 第8题 | 第9题 | 第10题 |
|  |  |  |  |  |
| 第11题 | 第12题 | 第13题 | 第14题 | 第15题 |
|  |  |  |  |  |
| 第16题 | 第17题 | 第18题 | 第19题 | 第20题 |
|  |  |  |  |  |

二．判断题（共5分，每题1分）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第1题 | 第2题 | 第3题 | 第4题 | 第5题 |
|  |  |  |  |  |

（程序设计题在另页上答题）

**浙江理工大学2021—2022学年第 2 学期**

**《 C#程序设计》期末试卷（ A ）卷标准答案和评分标准**

一、选择题

1. CADBD

6 ABCAB

11 BDDBA

16 BCACA

二、判断题

√×√×√

三、编程题

1、

static string RemoveDuplicates(string str) {

string newString = "";

bool duplicate = false;

for (int i = 0;i < str.Length; i++)

{

if (newString.IndexOf(str[i]) >= 0)

continue;

newString += str[i];

//for (int j = 0; j < newString.Length; j++)

//{

// if (str[i] == newString[j])

// {

// duplicate = true;

// break;

// }

//}

//if (!duplicate)

// newString += str[i];

//duplicate = false;

}

return newString;

}

2、

class Point3d //1{

public double X { get; set; }//2

public double Y { get; set; }

public double Z { get; set; }

public override bool Equals(object? obj)//2 {

if (obj == null || !(obj is Point3d))//1

return false;

Point3d other = obj as Point3d;//1

return DistanceTo(other) < 1.0e-6;//1

}

public double DistanceTo(Point3d p2)//1

{

if(p2 == null) throw new ArgumentNullException();

return Math.Sqrt((X - p2.X) \* (X - p2.X) +

(Y - p2.Y) \* (Y - p2.Y) + (Z - p2.Z) \* (Z - p2.Z));//1

}

}

class Triangle : IComparable<Triangle>//1

{

private Point3d[] v;//1

public Triangle()//2

{

v = new Point3d[3] {

new Point3d { X = 1, Y = 0, Z = 0 },

new Point3d { X = 0, Y = 1, Z = 0 },

new Point3d { X = 0, Y = 0, Z = 1 },

};

}

public Triangle(Point3d[] vs)//2

{

if (vs == null || vs.Length != 3)

throw new ArgumentException();

v = vs;

}

public void GetLengths(out double a, out double b, out double c)//3

{

a = v[0].DistanceTo(v[1]);

b = v[1].DistanceTo(v[2]);

c = v[0].DistanceTo(v[2]);

}

public bool Valid//4

{

get

{

GetLengths(out double a, out double b, out double c);

if (a + b < c || a + c < b || b + c < a)

return false;

else

return true;

}

}

public double Area//4

{

get

{

GetLengths(out double a, out double b, out double c);

double p = (a + b + c) / 2;

return Math.Sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

}

public bool IsConnectedTo(Triangle t2)//4

{

for (int i = 0; i < 3; i++) {

Point3d p1 = v[i];

for (int j = 0; j < 3; j++)

{

Point3d p2 = t2.v[j];

if (p1.Equals(p2))

return true;

}

}

return false;

}

public int CompareTo(Triangle? other)//4 {

if (other == null)

throw new ArgumentNullException();

return Area.CompareTo(other.Area);

}

}

3、

Predicate<string> isHead = (str) => str.StartsWith("第") && str.Contains('回');//4

string dir = @"D:\三国演义\";

string[] files = Directory.GetFiles(dir);//1

Dictionary<string, int> counts = new Dictionary<string, int>();//2

int times = 0;

foreach (string file in files)//1

{

string title = file.Substring(file.LastIndexOf('\\') + 1, 3);//2

StreamReader sr = new StreamReader(file);//1

times = 0;

while(!sr.EndOfStream)//1 {

string line = sr.ReadLine();//1

string[] subject = { "刘备", "玄德", "先主", "皇叔" };//1

if (isHead(line))//3

continue;

for (int i = 0; i < subject.Length; i++)//3

{

if (line.Contains(subject[i]))

times++;

}

}

sr.Close();//1

counts[title] = times;//3

}

var query = from c in counts//5

where c.Value > 5

select $"第{c.Key}回 : {c.Value}次";

Console.WriteLine($"超过5次的章回共有：{query.Count()}个");//1

foreach (var item in query)//2

{

Console.WriteLine(item);

}

}