浙江理工大学 2017—2018 学年第 2 学期

《C#程序设计》期末试卷（B）卷

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、单选题（在每小题的四个备选答案中选出一个正确答案，并将正确答案的序号填入题后的括号内。每小题 2 分，共 30 分。）

(1) 以下关于结构的说法，正确的是（ ）
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. 结构允许声明无形参的实例构造函数

(2) 下列类型中, 哪些不属于引用类型?（ ）。
A．String B． int C．Class D．Delegate

(3) 下列关于多态的说法中，哪个选项是正确的（ ）。
A．重写虚方法时可以为虚方法指定别称
B．抽象类中不可以包含虚方法
C．虚方法是实现多态的唯一手段
D．多态性是指以相似的手段来处理各不相同的派生类。

(4) .NET 框架是.NET 战略的基础，是一种新的便捷的开发平台，它具有两个主要的组件，分别是( )和类库。
A．公共语言运行库 B．Web 服务
C．命名空间 D．Main（）函数

(5) 下列的( )不是构造函数的特征。
A．构造函数的函数名和类名相同
B．构造函数可以重载
C．构造函数可以带有参数
D．可以指定构造函数的返回值

(6) 指定操作系统读取文件方式中的 FileMode .Create 的含义是（ ）。
A．打开现有文件
B．指定操作系统应创建文件，如果文件存在，将出现异常
C．打开现有文件，若文件不存在，出现异常
D．指定操作系统应创建文件，如果文件存在，将被改写

(7) 调用方法时，如果想给方法传递任意个数的参数时，应选用（　　）关键字。
A．ref　　B．out　　C．params　　D．无特殊要求

(8) 要使一个对象能够通过序列化输出，其所属的类必须用（　　）特性修饰。
A．Serializable　　B．Comparable　　C．List　　D．Disposable

(9) 在 C#的类结构中，class 关键字前面的关键字是表示访问级别，下面哪个关键字的访问级别是表示只有在同一个程序集内，且内部类型或成员才是可访问的?（　　）。
A．public　　B．private　　C．protected　　D．internal

(10) 下列数组初始化语句哪个是不正确的?
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}};

(11) WPF 中最灵活最常用的布局控件是(　　)。
A. Canvas　　B. DockPanel　　C.Grid　　D. StackPanel

(12) 多态是指两个或多个属于不同对象，对于同一个消息作出不同响应的方式。C#中的多态不能通过(　　)实现。
A．接口　　B．抽象类　　C．虚方法　　D．密封类

(13) 关于静态方法的说法，错误的是（　　）。
A. 静态方法不对特定实例进行操作，不与实例相关联
B. 使用静态方法的语法格式： 类名.静态方法()
C. 静态方法能访问类中的静态成员，不能访问非静态成员
D. 静态方法不能访问类中的静态字段

(14) 在 C#中，以下关于属性的描述正确的是（　　）。
A．属性是以 public 关键字修饰的字段，以 private 修饰的字段不可称为属性
B．属性是访问字段的一种灵活机制，属性更好地实现了数据的封闭和隐藏
C．要定义只读属性只需在属性名前加上 readonly 关键字
D．在 C#中类中不能自定义属性

(15) 以下 C#程序的执行情况是（　　）。

```
namespace aaa
{    delegate void delep(int i);
     class Program
     {    static void Main()
          {    funb(new delep(funa));      }
          public static void funa(int t)
          {    funb(21);      }
          public static void funb(int i)
          {    Console.WriteLine(i.ToString());    }
     } }
```

A．代码中存在错误，delegate void delep(int i);不能定义在名称空间或者类之外

B．代码中存在错误，代码行 funb(new delep(funa));使用委托错误

C．程序正常运行，输出为 0

D．程序正常运行，输出为 21

(16) 以下关于泛型的叙述中错误的是（　　）。

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

二、程序设计题（本题共 68 分）

1．应用委托和泛型集合类知识，完成以下程序。（本题 8 分）

程序框架：

```
class Delegates
{
   //创建委托类型
   public delegate int NumberCalculation( int number1, int number2 );

   static void Main( string[] args )
   {
      int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
      //生成委托实例
      NumberCalculation add = Add;
      //计算总和
      int sum = MapCalculation(add, numbers);
      //描述并输出
      Console.WriteLine("数组所有元素总和为: {0}", sum);
   }

   //数据中的值两两操作，得到最终结果
   private int MapCalculation ( NumberCalculation cal, params int[] intArray)
   {
```

```
        …code1…
    }

    //两个整数相加
    private static int Add( int number1, int number2 )
    {
        …code2…
    }
}
```

2．定义以下结构与类：[42 分]

1）Point 结构，包含：[8 分]

两个 double 类型实例变量 x 和 y，代表点的二维坐标；

一个 Point 类型静态常量 Orign，代表原点；

带两个 double 参数的构造方法；

重写 ToString 方法，输出格式 Point(x: 1.0, y:1.0)

2）Shape 抽象类，包含：[8 分]

一个 double 类型属性 Area，可读取面积；

一个 double 类型属性 Perimeter，可读取周长；

一个方法 Contains(double x, double y)，判断参数所组成的点是否在图形内。

以上均为抽象成员。

2）Rect 类，继承 Shape 抽象类并实现所有抽象成员，并包含：[8 分]

一个 Point2D 类型的属性 TopLeft，代表矩形的左上角点；

两个 double 类型属性 Width 和 Height，代表矩形长和宽；

重写 ToString 方法，输出格式 Rect[TopLeft: Point(x: 1.0, y:1.0), Width:1.0, Height: 1.0]

3）Circle 类，继承 Shape 抽象类并实现所有抽象成员，并包含：[12 分]

一个 Point2D 类型的属性 Center，代表圆心坐标；

一个 double 类型属性 Radius，可读写圆半径；

一个构造方法，接收指定坐标和半径；

一个无参构造方法，调用上个构造方法，创建一个默认圆，圆心在原点，半径为 1。

重写 ToString 方法，输出格式 Cirle[Center: Point(x: 1.0, y:1.0), Radius: 1.0, Area: 3.14259]

4）在 Main 方法中创建矩形与圆，声明为 Shape 变量，测试其成员。[6 分]

3．利用 LINQ 技术查询 Racer 对象[18 分]

已有 Racer 类和冠军车手数据，进行以下查询，并打印结果：

（1） 查询来自英国的所有世界冠军，并按胜利场数降序排列。[3 分]

输出结果时，可用 Console.WriteLine("{0:A}", racer)。使用方法语法重写这个查询。

（2） 首先建立查询 query2，查出所有胜利场数超过，用 foreach 循环并打印；然后将所有英国（UK）车手的国名改为“United Kingdom”；再次用 foreach 循环并打印。观赛同一查询的不同执行结果。[5 分]

（3） 查出首发场数超过 100，并且胜利场数超过 20 的车手的姓名（包含名和姓）。[3 分]

（4） 基于现有数据，统计德国人的胜利总场数。[4 分]

（5） 进行投影查询，输出人名与首发胜率，人名包含名和姓，首发胜率为 Wins 除以 Starts。[4 分]

Racer类部分定义如下：

```
    public class Racer : IComparable<Racer>, IFormattable
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Country { get; set; }
        public int Wins { get; set; }//夺冠场数
        public int Starts { get; set; }//首发场数
        public string[] Cars { get; private set; }//赛车手获得冠军那一年使用的所有车型
        public int[] Years { get; private set; }//赛车手获得冠军的年份
    }
```