

## 实验二 类与对象编程练习

### 【实验目的】

1. 理解 C#语言是如何体现面向对象编程基本思想;
2. 掌握类对象的定义;
3. 了解类的封装方法, 以及如何创建类和对象;
4. 了解成员变量和成员方法的特性;
5. 掌握静态成员的用法;
6. 掌握构造函数和析构函数的含义与作用、定义方式和实现, 能够根据要求正确定义和重载构造函数。能够根据给定的要求定义类并实现类的成员函数;
7. 理解类的成员的访问控制的含义, 公有、私有和保护成员的区别;
8. 掌握参数传递的用法;
9. 掌握属性的作用和使用。

### 【实验要求】

1. 编写一个包含类和类方法的程序;
2. 编写一个创建对象和使用对象的方法程序;
3. 编写不同成员变量、方法修饰方法的程序;。
4. 编写含有构造方法的类的程序;
5. 编写重载构造方法的程序;
6. 编写类含有静态变量的程序;
7. 编写通过 ref、out 修饰符传递参数的类程序;
8. 编写含有属性的类程序。

### 【实验内容】

#### 一、定义一个圆类, 计算圆的面积和周长

```
public class circle
{

    public static void Main()
    {

        double radium, delimeter, square;
        const double pai = 3.1415926;
```

```

        radium = Convert.ToInt32(Console.ReadLine());
        delimiter = 2 * pai * radium;
        square = pai * pai * radium;
        Console.WriteLine("delimiter={0},square={1}", delimiter, square);
        Console.ReadLine();
    }

}

```

或者:

```

public class circle
{
    double  delimiter, square;
    const double pai = 3.1415926;
    public void calculate(double rad)
    {
        delimiter = 2 * pai * rad;
        square = pai * pai * rad;
        Console.WriteLine("delimiter={0},square={1}", delimiter, square);
    }

    public static void Main()
    {
        double radium;
        circle cir = new circle();
        radium = Convert.ToInt32(Console.ReadLine());
        cir.calculate(radium);
        Console.ReadLine();
    }
}

```

请比较以上两个程序，看起来后一个程序把问题复杂化了，是不是不如第一个程序好，它从设计思想上有什么优势么？

```

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
6
delimiter=37.699111200000004,square=59.21762438624856

```

第二个程序更好，因为第二个程序使用到了面向对象的设计方法，可以将特定的功能模块化。

## 二、实现一个学生类

程序要求如下：

其中有 3 个数据成员有学号、姓名、年龄，以及若干成员函数。同时编写主函数使用这

个类，实现对学生数据的赋值和输出。要求：使用成员函数实现对数据的输出；使用构造函数实现对数据的输入。

参考代码如下：

```
public class students
{
    string id,name;
    int age;
    public students(string id,string name,int age )
    {
        this.id = id;
        this.name = name;
        this.age = age;
    }
    public void Display()
    {
        Console.WriteLine("id={0},name={1},age={2}", id, name, age);
    }

    public static void Main()
    {
        //string id, name;
        //int age;
        students stu = new students("0001","zhangsan",16);
        stu.Display();
        Console.ReadLine();
    }
}
```

以上程序使用了构造方法，请回答关键字 `this` 有何作用，你能将成员函数 `Display` 修改成别的代码也实现响应的功能么？

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
id=0001,name=zhangsan,age=16
```

`this` 关键字表示当前的对象，使用另一种代码也能实现响应

```

1 reference
public static void Display2(students student){
    Console.WriteLine($"id={student.id},name={student.name},age={student.age}");
}

0 references
public static void Main()
{
    //string id, name;
    //int age;
    students stu = new students("0001", "zhangsan", 16);
    //stu.Display();
    students.Display2(stu);
    Console.ReadLine();
}

```

### 三、实现一个日期类型

程序要求如下：

定义日期类型 Date。要求有以下成员：年、月、日变量，重载的构造方法，一个实现年、月、日单独的输入，一个实现从系统时间里读出年月日，并实现打印方法成员，该方法要按照“XXXX 年 XX 月 XX 日”格式输出日期。

参考代码如下：

```

public class Date
{
    private int Year, Month, Day;
    public Date(int Year, int Month, int Day)
    {
        this.Year=Year;
        this.Month=Month;
        this.Day=Day;
    }
    public Date(System.DateTime dt)
    {
        Year = dt.Year;
        Month = dt.Month;
        Day = dt.Day;
    }
    public void DisplayDate()
    {
        Console.WriteLine("{0}年{1}月{2}日", Year, Month, Day);
    }
}

public class Tester
{
    public static void Main()

```

```

    {
        System.DateTime currentTime=System.DateTime.Now;

        Date dt=new Date(2008, 7, 18);
        dt.DisplayDate();
        Date dt2 = new Date(currentTime);
        dt2.DisplayDate();
        Console.ReadLine();
    }
}

```

请练习System.Datetime类型及其各属性的应用。列出System.Datetime主要包含哪些属性，并思考重载构造方法的作用。

```

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
2008年7月18日
2020年10月14日

```

Year: 年, Month: 月, Day: 日, DayOfWeek: 星期几, Date: 截取日期部分, TimeOfDay: 截取时间部分。

重载构造方法可以用多种参数表来初始化对象。

#### 四、实现一个包含类属性方法的简单加法程序

程序要求如下：

建立一个类，使其可以进行简单的加法运算。该程序要包含类、属性、方法等面向对象的基本元素。

```

public class PropertyTest
{
    double xVal, yVal;
    double zVal, sVal;
    public PropertyTest(double x)
    {
        xVal = x;
    }
    public double YVAL
    {
        get
        {
            return yVal;
        }
        set
        {
            yVal = value;
        }
    }
}

```

```

    }

    public void Display()
    {
        //Console.WriteLine("xVal={0}, yVal={1}", xVal, yVal);
        zVal = Math.Sqrt(Math.Pow(xVal, 2) - Math.Pow(yVal, 3));
        sVal = Math.Log(yVal);
        // zVal = xVal * xVal - yVal;
        // sVal = yVal + 5;

        Console.WriteLine("xVal={0}, yVal={1}, zVal={2}, sVal={3}", xVal, yVal, zVal, sVal);
    }
}

public class Tester
{
    public static void Main()
    {
        PropertyTest pt = new PropertyTest(4);
        pt.YVAL = 2;
        pt.Display();
        Console.ReadLine();
    }
}

```

作以下修改分析程序输出的原因：

将“xVal={0}, yVal={1}, zVal={2}, sVal={3}”

改成“xVal={0}, yVal={1}, zVal={0}, sVal={1}”查看区别，写出原因；

了解 Math 类下主要的静态数学方法，掌握求幂，求平方和对数等静态方法的使用规则。

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
xVal=4,yVal=2,zVal=2.8284271247461903,sVal=0.6931471805599453
```

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
xVal=4,yVal=2,zVal=4,sVal=2
```

## 五、通过类程序说明静态变量/方法与实例变量/方法的区别

static 声明的成员变量/方法被视为类的成员变量/方法，而不把它当作实例对象的成员变量/方法。换句话说，静态变量/方法是类固有的，可以直接引用，其它成员变量/方法仅仅被声明，生成实例对象后才存在，才可以被引用。基于这样的事实，也把静态变量/方法称为类变量/方法，非静态变量称为实例变量/方法。

### 1. 程序功能要求

编写帐户类，对每一账号赋值帐户并设置初始化存款为 0.00 元，设计一变量统计账号生成的数目。

```

public class BankAccount
{
    static int totalAccountNumber=0;
    string BankAccountId;
    double initialDepositAmount = 0.00;
    public BankAccount(string myId)
    {
        this.BankAccountId = myId;
        this.initialDepositAmount = 0.00;
        totalAccountNumber++;
    }
    public void displayid()
    {

Console.WriteLine("mbaid={0}, initialDepositAmount={1}", this.BankAccountId, this.
initialDepositAmount);
    }
    public static void display()
    {
        Console.WriteLine("totalAccountNumber={0}", totalAccountNumber);
    }
}

public class Tester
{
    public static void Main()
    {
        BankAccount mba = new BankAccount("37000001");
        BankAccount mba2 = new BankAccount("37000002");
        BankAccount mba3 = new BankAccount("");
        BankAccount mba4 = new BankAccount("37000004");
        // Console.WriteLine("mba2ID={0}", mba2.BankAccountId);
        mba2.displayid();
        BankAccount.display();

        Console.ReadLine();
    }
}

```

请回答问题：

- (1) 按你自己的算法修改以上程序，比如可只输出生成的账户数。
- (2) 把注释去掉后会怎样，为什么？
- (3) 为什么 display 用类名直接引用，可以用对象来引用么？尝试输出结果。
- (4) 类的静态变量和非静态变量的引用区别。判断一下语句的正确性：

静态方法只能使用静态变量，不能使用实例变量。因为对象实例化之前，实例变量不可用。

这个观点真确么？（）

类的静态变量只有一个版本，所有实例对象引用的都是同一个版本。（）

对象实例化后，每个实例变量都被制作了一个副本，它们之间互不影响。（）

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
mbaid=3700002,initialDepositAmount=0
totalAccountNumber=4
```

(1) :

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
totalAccountNumber=4
```

(2) :

```
public static void Main()
{
    BankAccount mba = new BankAccount("37000001");
    BankAccount mba2 = new BankAccount("37000002");
    BankAccount mba3 = new BankAccount("");
    BankAccount mba4 = new BankAccount("37000004");
    Console.WriteLine("mba2ID={0}", mba2.BankAccountId);
    mba2.displayid();
    BankAccount.display();

    Console.ReadLine();
}
```

会报编译器错误，原因是该字段为私有属性。

(3) : 不可以

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
Program.cs(35,13): error CS0176: 无法使用实例引用来访问成员“BankAccount.display()”: 请改用类型名来限定它 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2\exp2.csproj]

生成失败。请修复生成错误并重新运行。
```

(4) : 类的静态成员只能由类名访问，类的动态成员只能由类实例访问。

## 六、方法中参数传递的练习

在其它语言中，函数调用或过程调用时参数有传值调用和传地址调用之分。在C# 中，方法中的参数传递可以分为传值调用或对象方法调用等方式。传值调用即传递的参数是基本数据类型，调用方法时在方法中将不能改变参数的值，这意味着只能使用它们。对象调用是指先调用对象，再调用对象的方法，这种方式可以修改允许存取的成员变量。所以，如果不想改变参数的值，可以采用传值调用的方法。如果想改变参数的值，可采用对象调用的方法，间接修改参数的值。

### 1. 编写一个传值调用的程序

程序功能要求：程序首先给整型变量x 和y 赋初值3, 5，然后使用传值调用方式调用方法对x 和y 做乘方并及输出x 和y 的乘方值，最后输出x和y得值。再将此方法给为对象调用加ref修饰查看输出结果差异。参考代码如下：



```

public class Power
{
    // public void MyPower(ref int x, ref int y)
    public void MyPower(int x, int y)
    {
        x = 1; y = 2;
        Console.WriteLine("x={0},y={1}", x, y);
        Console.WriteLine("x*x={0},y*y={1}", x*x, y*y);
    }
}

public class Tester
{
    public static void Main()
    {
        int x, y;
        x = 3; y = 5;
        Power mp = new Power();
        // mp.MyPower(ref x, ref y);
        mp.MyPower(x, y);
        Console.WriteLine("x={0},y={1}", x, y);
        Console.ReadLine();
    }
}

```

思考：（1）将响应的注释修改再调试查看结果，分析原因。

（2）将 **Main** 中 **x** 和 **y** 赋初值去掉，结果会怎样？如果 **Main** 中加 **ref**，类 **Power** 的方法中参数前不加 **ref** 又会有何变化？说明了什么？

3）如果不想对 **x** 作无用的初始化，直接作参数传递，怎么实现？

（1）：

```

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
x=1,y=2
x*x=1,y*y=4
x=3,y=5

```

```

PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
Program.cs(22,28): error CS1615: 参数 1 不可与关键字“ref”一起传递 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2\exp2.csproj]
Program.cs(22,34): error CS1615: 参数 2 不可与关键字“ref”一起传递 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2\exp2.csproj]

```

生成失败。请修复生成错误并重新运行。

**ref** 传参要求函数声明带有 **ref**

（2）：

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
x=3,y=5
x*x=9,y*y=25
x=3,y=5
█
```

```
PS C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2> dotnet run
Program.cs(23,24): error CS1620: 参数 1 必须与关键字“ref”一起传递 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2\exp2.csproj]
Program.cs(23,26): error CS1620: 参数 2 必须与关键字“ref”一起传递 [C:\Users\cht\Program\homework.gallery\computer science\csharp\exp2\exp2.csproj]
生成失败。请修复生成错误并重新运行。
```

说明 ref 传参时声明和调用都必须同时用 ref 修饰。

(3): 声明和调用的参数都加上 ref 关键字。

### 【思考题】

1. 方法的参数传递有哪些方式？区别时什么？

方法的参数传递有值传参，引用传参，out传参三种方式。

值传参将传递参数的值，具体的，当传递的参数为引用类型时，传递的是引用地址，传递的参数为值类型时，传递的是值的副本。

引用传参将传递参数的引用

out传参是引用传参的一个特殊形式，其限定了只能从内部向外部传递参数的引用。

2. 什么是构造方法。

构造方法用于构造一个类，其没有返回值。类的创建必须调用构造方法才能创建。