

Lab4 面向对象

学习目标:

- 1.掌握属性的定义方法
- 2.了解索引器的定义方法

一、属性（要求重点掌握）

在 C#中，为保证类中内部数据的安全，可以使用属性来封装字段，首先需要将字段访问级别设为 `private`，再通过属性的 `get` 和 `set` 访问器来对字段进行读写操作，具体语法格式如下：

```
[访问修饰符] 数据类型 属性名
{
    get{获得属性的代码;}
    set{设置属性的代码;}
}
```

如果设置读写属性，需要同时使用 `get` 和 `set` 访问器；如果设置只读属性，只需要使用 `get` 访问器，一般用于构造方法中给属性赋值，在程序运行过程中不能修改该属性的值；如果设置只写属性，只需要使用 `set` 访问器，在程序运行过程中只能写入值而不能读取值；如果设置自动属性，则不需要书写任何属性的代码，也就是在 `get` 和 `set` 访问器后面不加大括号，直接加 “;” 即可。

【实例】定义 `ComputerStudent` 类的自动属性、只读属性和读写属性。

程序代码如下：

```
namespace Property
{
    class Program
    {
        static void Main(string[] args)
        {
            ComputerStudent s= new ComputerStudent();
            s.Age = 19;
            s.Speak();
            Console.ReadLine();
        }
    }

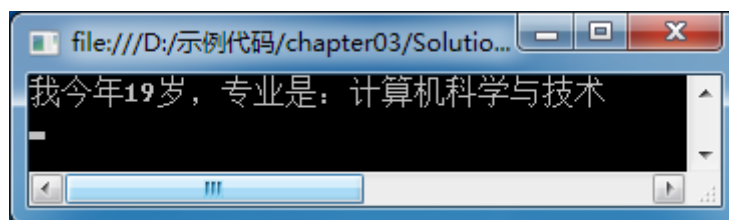
    public class ComputerStudent
    {
        //自动属性
        public string Name {get; set;}
        //只读属性
        private string _specialtyName="计算机科学与技术";
        public string SpecialtyName
        {
            get { return _specialtyName; }
        }
    }
}
```

```

//读写属性
private int _age;
public int Age
{
    get
    {
        return _age;
    }
    set    //检查参数合法性
    {
        if (value <= 0)
        {
            Console.WriteLine("年龄输入不合法!");
        }
        else
        {
            _age = value;
        }
    }
}

public void Speak()
{
    Console.WriteLine("我今年"+_age+"岁，专业是："+_specialtyName);
}
}
}

```



【分析】

在本实例中，通过字段来存储数据，而通过属性来完成对字段的访问，通过方法来对数据进行操作，将操作得到的结果交给调用方，数据和操作融为一体，这就是封装。

【注意】从 C#3.0 开始出现了一种新的简洁的属性定义方式，即自动属性。在自动属性中，无须定义一个相应的私有字段，也不必写任何 `return` 和 `value` 语句。自动属性虽然简洁，但不能通过属性来完成任何更多的复杂逻辑。

二、索引器（要求理解）

索引器可以使得对象能用下标来得到一个值。前面我们已经使用过索引器，例如 `str[0]` 表示字符串中首个字符。

索引器的定义方法：与定义属性类似

修饰符 类型名 `this`[参数列表]

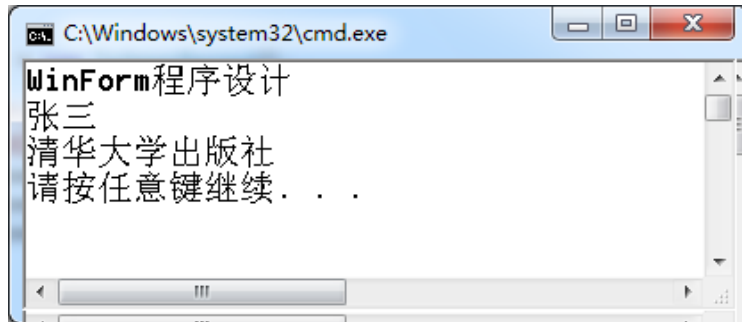
```
{
```

```

        set{}
        get{}
    }

```

实例：使用索引器来表示一本书的记录，在这里定义两个索引器，一个用整数作参数，一个用关键字作参数。



```

class IndexerRecord
{
    private string[] data = new string[3];
    private string[] keys = { "Author", "Publisher", "Title" };
    public string this[int idx]
    {
        set
        {
            if (idx >= 0 && idx < data.Length)
                data[idx] = value;
        }
        get
        {
            if (idx >= 0 && idx < data.Length)
                return data[idx];
            return null;
        }
    }
    private int FindKey(string key)
    {
        for (int i = 0; i < keys.Length; i++)
            if (keys[i] == key) return i;
        return -1;
    }
    public string this[string key]
    {
        set
        {
            int idx = FindKey(key);
            this[idx] = value;
        }
    }
}

```

```

        get
        {
            return this[FindKey (key)];
        }
    }
}

static void Main()
{
    IndexerRecord r = new IndexerRecord();
    r[0] = "张三";
    r[1] = "清华大学出版社";
    r[2] = "WinForm程序设计";
    Console.WriteLine(r["Title"]);
    Console.WriteLine(r["Author"]);
    Console.WriteLine(r["Publisher"]);
}
}

```

三、练习项目

需求：创建一个 **Employee** 类，该类的属性有姓名、年龄、性别和工资，并定义输出信息的方法。创建一个窗体，用于添加员工信息，当点击“确定”按钮时，输出员工信息。

步骤 1：新建一个窗体项目 Lab4——在项目中创建一个类 **Employee**，分别定义字段、属性、构造方法以及 **Display** 方法。

```

class Employee
{
    //定义属性：姓名、年龄、性别、工资
    private string name;
    public string Name
    {
        get
        {
            return name;
        }
    }
}

```

```

    }
    set
    {
        name = value;
    }
}

private int age;
public int Age
{
    get { return age; }
    set { age = value; }
}

private string sex;
public string Sex
{
    get { return sex; }
    set { sex = value; }
}

private double salary;
public double Salary
{
    get { return salary; }
    set { salary = value; }
}

//添加不带参数的构造方法
public Employee()
{
    name = "";
    age = 0;
    sex = "";
    salary = 0;
}

//添加带参数的构造方法
public Employee(string _name, int _age, string _sex, int _salary)
{
    this.name = _name;
    this.age = _age;
    this.sex = _sex;
    this.salary = _salary;
}

//添加Display方法
public void Display()
{
    string msg = "姓名" + Name + "\n年龄" + Age + "\n性别" + Sex + "\n

```

```

    工资" + Salary;
    MessageBox.Show(msg); //需要引入命名空间using System.Windows.Forms;
}
}

```

步骤 2：设计窗体。

Name 属性	类型	Text 属性	其它属性
Form1		添加员工信息	
Label1	Label	姓名	
Label2	Label	年龄	
Label3	Label	性别	
Label4	Label	工资	
txtName	TextBox		
txtAge	TextBox		
rbMale	RadioButton	男	Checked 为 True
rbFemale	RadioButton	女	
txtSal	TextBox		
btnOk	Button	确定	

步骤 3：为确定按钮添加事件处理程序。

项目完成后思考：封装 **Employee** 类的好处？？