

## BÀI 8. INTERFACE

### Bài 1. Cơ bản về Interface – Interface của một lớp

Một quyển sách (Book) chứa một số thông tin: **mã sách, tên sách, giá sách, giảm giá.**

Khai báo interface **IBook** cho một quyển sách. Trong interface **IBook** có:

- Các properties của quyển sách
- Phương thức tính giá một quyển sách

Xây dựng lớp **Book** hiện thực (cài đặt) interface **IBook** trên

Chú ý: Phương thức tính giá bán của 1 quyển sách được tính như sau:

$$\boxed{\text{Giá bán một quyển sách} = \text{Giá sách} - \text{Giảm giá}}$$

### Bài 2. Cơ bản về Interface – Nhiều lớp thực hiện cùng 1 interface và thực hiện tính đa hình

Lớp **Rectangle** (hình chữ nhật) có: chiều rộng và chiều cao

Lớp **Circle** (hình tròn) có: bán kính

- Xây dựng interface **IShape** có phương thức tính diện tích của hình
- Xây dựng lớp **Rectangle** và lớp **Circle** cài đặt interface **IShape** trên

### Bài 3. Tính lương nhân viên

Công ty phần mềm TinySoft đang cần quản lý các nhân viên của mình. Nhân viên công ty thuộc một trong hai bộ phận: **Programmer** và **Tester**. Thông tin cơ bản của nhân viên gồm: mã nhân viên, họ tên nhân viên

Lương hàng tháng của mỗi bộ phận như sau:

- **Programmer:** Mỗi lập trình viên có lương hàng tháng riêng. Nếu trong tháng số lỗi lập trình < 20 thì được tăng thêm 10% lương.
- **Tester:** Mỗi nhân viên kiểm thử phần mềm ngoài mức lương cơ bản hàng tháng, nhân viên còn nhận được tiền overtime (gọi tắt là OT: tiền làm qua đêm). Số tiền OT bằng số giờ OT nhân với 200.000 đồng

***Yêu cầu:***

- Xây dựng **interface** chứa phương thức tính lương, lớp **abstract** chứa các thông tin cơ bản của Programmer và Tester
- Xây dựng lớp **Programmer, Tester**
- Hãy viết chương trình nhập n nhân viên và tính lương của n nhân viên này.

**File Employee.cs**

```
using System;

namespace ThuchanhOOP
{
    class Employee
    {
        string maNV, hoten;
        double luongcb;

        public string MaNV { get => maNV; set => maNV = value; }
        public string Hoten { get => hoten; set => hoten = value; }
        public double Luongcb { get => luongcb; set => luongcb = value; }

        public Employee(string _maNV, string _hoten, double _luongcb)
        {
            MaNV = _maNV;
            Hoten = _hoten;
            Luongcb = _luongcb;
        }

        public Employee()
        {
            MaNV = "";
            Hoten = "";
            Luongcb = 0;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 8

```
public virtual void Input()
{
    Console.Write("Ma nhan vien: ");
    MaNV = Console.ReadLine();

    Console.Write("Ho ten: ");
    Hoten = Console.ReadLine();

    Console.Write("Luong can ban: ");
    Luongcb = Convert.ToDouble(Console.ReadLine());
}

public virtual double TinhLuong()
{
    return Luongcb;
}

public override string ToString()
{
    return $"Ma nhan vien: {MaNV} \nHo ten: {Hoten} \nLuong can ban:
                                                {Luongcb:0.00}";
}
}
```

### File Programmer.cs

```
using System;

namespace ThuchanhOOP
{
    class Programmer:Employee
    {
        int soloi;

        public int Soloi { get => soloi; set => soloi = value; }

        public Programmer():base()
        {
            Soloi = 0;
        }

        public Programmer(string _maNV, string _hoten, long _luongcb, int _soloi) :
            base(_maNV, _hoten, _luongcb)
        {
            Soloi = _soloi;
        }

        public override double TinhLuong()
        {
            double phantram = 0;
            if (Soloi < 20)
                phantram = 0.1; // 10%
            else
                phantram = 0;

            return base.TinhLuong() + base.TinhLuong()*phantram;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 8

```
public override void Input()
{
    base.Input();
    Console.Write("So loi trong thang: ");
    Soloi = Convert.ToInt32(Console.ReadLine());
}
public override string ToString()
{
    return base.ToString() + $"\\nSo loi: {Soloi} \\nThuc lanh:
                                                                    {TinhLuong():0.00}";
}
}
}
```

### File Tester.cs

```
using System;

namespace ThuchanhOOP
{
    class Tester : Employee
    {
        double sogioOT;

        public double SogioOT { get => sogioOT; set => sogioOT = value; }

        public Tester() : base()
        {
            sogioOT = 0;
        }

        public Tester(string _maNV, string _hoten, double _luongcb, double _sogioOT) :
                                                                    base(_maNV, _hoten, _luongcb)
        {
            SogioOT = _sogioOT;
        }

        public override void Input()
        {
            base.Input();
            Console.Write("So gio lam them trong thang: ");
            SogioOT = Convert.ToDouble(Console.ReadLine());
        }

        public override double TinhLuong()
        {
            return base.TinhLuong() + SogioOT * 200000;
        }

        public override string ToString()
        {
            return base.ToString() + $"\\nSo gio lam them: {SogioOT} \\nThuc lanh:
                                                                    {TinhLuong():0.00}";
        }
    }
}
```

### File Program.cs

```
using System;
using System.Collections.Generic;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            List<Employee> employees = new List<Employee>();
            char ans = new char();
            do
            {
                Console.WriteLine("Nhap thong tin Nhan vien (P:Programmer, T:Tester or Q:quit): ");

                ans = Convert.ToChar(Console.ReadLine());
                if (ans.Equals('p') || ans.Equals('P'))
                {
                    Programmer p = new Programmer();
                    p.Input();
                    employees.Add(p);
                }
                if (ans.Equals('t') || ans.Equals('T'))
                {
                    Tester t = new Tester();
                    t.Input();
                    employees.Add(t);
                }
            } while (!ans.Equals('q') && !ans.Equals('Q'));

            foreach (var emp in employees)
                Console.WriteLine(emp.ToString());

            Console.ReadKey();
        }
    }
}
```

### **Bài 1.** Tính điểm thí sinh

Phân tích, thiết kế và hiện thực theo hướng đối tượng chương trình tính điểm thi cho các thí sinh trong một cuộc thi tin học, thông tin **Cuộc thi**: danh sách thí sinh.

Cuộc thi giành cho hai đối tượng **thí sinh**:

**Chuyên** (giành cho những thí sinh chưa có giải trước đây)

**Siêu cúp** (giành cho những thí sinh đã đoạt giải trước đây).

Biết rằng các thí sinh phải làm 3 bài thi lập trình, riêng đối tượng **Chuyên** làm thêm bài thi tiếng Anh (để cộng điểm thêm), đối tượng **Siêu cúp** phải làm thêm bài cơ sở dữ liệu (CSDL).

- Thông tin chung của các thí sinh: sbd (số báo danh), hoten (họ tên), bai1, bai2, bai3 (điểm của bài 1, 2, 3), tongdiem (tổng điểm cuối cùng của từng thí sinh)
- Thông tin riêng cho đối tượng **Chuyên**: tiengAnh (điểm tiếng Anh)
- Thông tin riêng cho đối tượng **Siêu cúp**: csdl (điểm CSDL)

Kết quả được xét như sau:

- Thí sinh Chuyên: Tổng 3 bài thi lập trình + điểm thưởng tiếng Anh được xét như sau:
  - $7 \leq \text{tiengAnh} \leq 8$ : cộng 1 điểm
  - $9 \leq \text{tiengAnh} \leq 10$ : cộng 2 điểm
- Thí sinh Siêu cúp: Tổng điểm của 4 bài thi

**Yêu cầu:** Viết chương trình nhập thông tin cuộc thi và xuất tổng điểm thi của từng thí sinh.

*(Sinh viên tự thực hiện)*