

## BÀI THỰC HÀNH 3

### Nội dung:

- Định nghĩa lớp, sử dụng đối tượng
- Thành phần tĩnh
- Cài đặt mối quan hệ giữa các lớp

**Bài 3.1.** Định nghĩa lớp Circle theo sơ đồ lớp và mô tả sau:

Circle
- radius: float
+ Circle() + Circle(float r) + SetRadius(float): void + GetPerimeter(): double + GetArea(): double + Info(): void + operator+(Circle, Circle):double

- radius bán kính đường tròn
- Circle() phương thức khởi tạo không tham số, gán bán kính bằng 1.
- Circle(float r) phương thức khởi tạo có tham số, gán bán kính bằng r.
- SetRadius(f) gán giá trị bán kính đường tròn bằng một giá trị đưa vào.
- GetPerimeter() tính và trả về chu vi đường tròn.
- GetArea() tính và trả về diện tích đường tròn.
- Info(): in ra màn hình thông tin: radius, Perimeter, Area.
- operator+(Circle, Circle): toán tử cộng hai đối tượng Circle với ý nghĩa cộng diện tích của hai đối tượng với nhau.

*Sử dụng lớp Circle thực hiện các yêu cầu:*

- Nhập vào 2 đối tượng hình tròn bằng hai phương thức khởi tạo khác nhau.
- Xuất thông tin 2 đối tượng ra màn hình;
- Cập nhật bán kính của đối tượng tạo ra bằng khởi tạo không tham số;
- Tính và in ra tổng diện tích của 2 hình tròn dùng toán tử cộng;
- Nhập vào n hình tròn ( $2 < n < 30$ );
- In ra màn hình thông tin n hình tròn vừa nhập.
- Tính và in ra màn hình tổng chu vi của n hình tròn.
- Tìm và in ra thông tin hình tròn có diện tích lớn nhất.

**Bài 3.2.** Xây dựng các lớp theo mô tả sau:

- Học phần gồm:
  - Các thuộc tính: mã học phần, tên học phần, số tín chỉ (sotc), số tín chỉ thực hành (tcth), học phí 1 tín chỉ dùng chung cho tất cả các đối tượng với giá trị khởi tạo là 350.
  - Các phương thức:
    - Khởi tạo không tham số, có tham số;
    - Nhập thông tin học phần;
    - Tính tiền học phí:  $tcth * \text{học phí 1 tín chỉ} * 1.5 + (sotc - tcth) * \text{học phí 1 tín chỉ}$ .
    - Xuất thông tin học phần (mã học phần, tên học phần, số tín chỉ, tiền học phí)
  - Các toán tử:

- Toán tử cộng một số nguyên với một đối tượng, ý nghĩa cộng một số với số tín chỉ thực hành của đối tượng;
- Hóa đơn học phí gồm:
  - Các thuộc tính: mã số sinh viên, họ tên, danh sách các học phần đăng ký học trong một học kỳ của sinh viên, biết mỗi sinh viên đăng ký 2-9 học phần/học kỳ.
  - Các phương thức:
    - Khởi tạo không tham số, có tham số;
    - Nhập thông tin hóa đơn học phí;
    - Xuất thông tin hóa đơn học phí (mssv, họ tên, danh sách học phần, tổng tiền học phí);
    - Tính tổng số tín chỉ thực hành trong kỳ của sinh viên.

Viết chương trình cài đặt và sử dụng các lớp ở trên để thực hiện:

- Nhập vào 1 hóa đơn học phí;
- In ra thông tin hóa đơn vừa nhập;
- In ra tổng số tín chỉ thực hành (sử dụng toán tử cộng)

### **Bài 3.3.** Xây dựng các lớp theo mô tả sau:

- Phiếu lưu trú khách sạn gồm:
  - Các thuộc tính: số phòng, họ tên khách, ngày đến, ngày đi, loại phòng (1, 2, 3), giá sàn dùng chung cho tất cả các đối tượng có giá trị là 300.
  - Các phương thức:
    - Khởi tạo không tham số, có tham số;
    - Nhập thông tin phiếu lưu trú;
    - Tính tiền phòng: số ngày ở \* giá phòng theo loại phòng, biết:  
Số ngày ở = ngày đi – ngày đến;  
Giá theo loại phòng: nếu loại phòng là 1 thì giá phòng là 1.5\*giá sàn; nếu loại phòng là 2 thì giá phòng là 1,3\*giá sàn, ngược lại giá phòng bằng giá sàn;
    - Xuất thông tin phiếu lưu trú (số phòng, họ tên khách, số ngày ở, giá phòng, tiền phòng);
  - Các toán tử:
    - Toán tử so sánh < với ý nghĩa so sánh số ngày ở của hai phiếu lưu trú;
    - Toán tử cộng một số với một đối tượng, ý nghĩa cộng một số với tiền phòng của đối tượng;

*Sử dụng lớp được định nghĩa ở trên để thực hiện các yêu cầu:*

- Nhập vào n phiếu lưu trú ( $1 < n < 30$ );
- Xuất thông tin n phiếu ra màn hình theo thứ tự giảm dần của số ngày ở;
- Đếm và xuất ra màn hình số lượng phiếu lưu trú trong quý 1 năm 2024;
- Tính và in ra màn hình tiền phòng trung bình của n phiếu lưu trú.

**Bài 3.4\*.** Sử dụng định nghĩa lớp Adult trong bài 2.7\* để thực hiện các yêu cầu sau:

- Nhập thông tin sức khỏe của n thanh niên được triệu tập thực thi nghĩa vụ quân sự của phường ABC ( $2 < n < 100$ )
- Hiện thị danh sách thanh niên có sức khỏe tốt.

**Bài 3.5\*.** Định nghĩa lớp học viên theo mô tả sau:

- Các thuộc tính: mã học viên, họ tên học viên, lớp học (A, B, C), số tiết học, học phí 1 tiết học dùng chung cho tất cả học viên và có giá trị 100.
- Các phương thức:
  - Khởi tạo không tham số, khởi tạo có tham số;
  - Nhập thông tin cho học viên;
  - Tính tiền học phí phải đóng theo công thức:  
*Số tiết học \* học phí 1 tiết - tiền giảm.* Biết tiền giảm:  
+ Nếu số tiết học  $> 50$  thì tiền giảm 10% (số tiết học \* học phí 1 tiết);  
+ Nếu  $50 \geq$  số tiết  $> 30$  thì tiền giảm 7% (số tiết học \* học phí 1 tiết);  
+ Còn lại không được giảm.
  - Xuất thông tin học viên gồm: tên học viên, lớp học, số tiết, tiền học phí;
- Các toán tử:
  - Định nghĩa toán tử so sánh  $>$  với ý nghĩa so sánh số tiết học của 2 đối tượng học viên;
  - Định nghĩa toán tử cộng với ý nghĩa cộng tiền học phí học viên với một số.

*Sử dụng lớp được định nghĩa ở trên để thực hiện các yêu cầu:*

- Nhập vào n học viên của một khóa học ( $2 < n < 30$ );
- Xuất danh sách học viên ra màn hình theo thứ tự tăng dần của số tiết học (sử dụng toán tử so sánh để so sánh 2 đối tượng học viên);
- Dùng toán tử cộng để tính và xuất ra tổng số tiền học phí của n học viên;

**Bài 3.6\*.** Cho lớp Bill được định nghĩa như sau:

```
class Bill
{
    string name;
    byte num;
    public static float price=10.5f;
    public Bill()
    {
        name = "Dell Latitude E7440";
        num = 1;
        price = 9.5f;
    }
    public Bill(string na, byte n, float p)
    {
        name = na;
        num = n;
        price = p;
    }
    public float Cal_Bill()
    {
        return num * price;
    }
}
```

```

    }
    public void Print()
    {
        Console.WriteLine($"{name}\t{num}\t{Cal_Bill()}");
    }
}

```

Hãy cho biết kết quả chạy các đoạn chương trình sử dụng lớp Bill ở trên:

a. `class Program`

```

{
    static void Main(string[] args)
    {
        Bill b = new Bill();
        Console.WriteLine($"{b.Cal_Bill()}");
    }
}

```

b. `class Program`

```

{
    static void Main(string[] args)
    {
        Bill b = new Bill();
        Console.WriteLine($"{b.price}");
    }
}

```

c. `class Program`

```

{
    static void Main(string[] args)
    {
        Bill b = new Bill("Acer", 1, 7.5f);
        b.Print ();
    }
}

```

d. `class Program`

```

{
    static void Main(string[] args)
    {
        Bill b1 = new Bill();
        Bill b2 = new Bill("Acer", 1, 7.5f);
        b1.Print ();
    }
}

```

----- hết -----