

## BÀI THỰC HÀNH 3

### Nội dung:

- Định nghĩa lớp, sử dụng đối tượng của lớp
- Phương thức thiết lập
- Toán tử trên lớp

### Bài 3.1. Định nghĩa phân số gồm có:

- Các thuộc tính: tử số, mẫu số
- Các phương thức:
  - Thiết lập không tham số (tử số là 0, mẫu số là 1), có tham số, sao chép.
  - Cộng, Trừ, Nhân, Chia hai phân số.
  - Rút gọn phân số về phân số tối giản
  - Xuất phân số dạng tử số/mẫu số
  - Định nghĩa các toán tử +, -, \*, / hai phân số

Viết chương trình cài đặt và sử dụng lớp phân số với các yêu cầu:

- Tạo hai phân số dùng các phương thức thiết lập.
- Thực hiện Cộng/Trừ/Nhân/Chia hai phân số (sử dụng phương thức, toán tử)
- Xuất các kết quả rút gọn phân số sau Cộng/Trừ/Nhân/Chia.

### Bài 3.2. Định nghĩa lớp vé máy bay gồm các thành phần sau:

- Dữ liệu: mã vé, họ tên, ngày mua (12/03/2023,...), ngày khởi hành (25/03/2023,...), loại vé (First, Business, Premium,...), giá sàn dùng chung cho tất cả các vé máy bay và có giá trị 500.
- Phương thức:
  - Khởi tạo không tham số, có tham số;
  - Nhập thông tin vé máy bay;
  - Tính tiền vé: Tỷ lệ giảm giá \* giá theo loại vé + 10% VAT trên tổng số tiền, biết:
    - ✓ Loại First giá bằng 3 lần giá sàn, loại Business bằng 2 lần giá sàn, loại Premium bằng 1.5 lần giá sàn, các loại vé còn lại bằng giá sàn;
    - ✓ Tỷ lệ giảm giá:  $0.5\% \times \text{số ngày mua trước (ngày khởi hành - ngày mua)}$ .
  - Xuất thông tin vé gồm: mã vé, họ tên, ngày khởi hành, loại vé, tiền vé;
  - Định nghĩa toán tử cộng với ý nghĩa cộng tiền vé của vé máy bay với một số;
  - Định nghĩa toán tử so sánh < với ý nghĩa so sánh tiền vé của hai vé máy bay;

Sử dụng lớp vé máy bay ở trên để thực hiện:

- Nhập n vé máy bay của một khách hàng ( $2 \leq n \leq 20$ );

- In ra n vé máy bay được sắp xếp theo chiều giảm dần của tiền vé (sử dụng toán tử so sánh);
- Tính và in ra tiền vé trung bình của n vé máy bay;
- Đếm và in ra số lượng vé máy bay loại Business;

### **Bài 3.3. 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 250.
  - 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)
  - Toán tử:
    - Cộng với ý nghĩa cộng tiền học phí với một số;
- 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.

**Bài 3.4.** Bổ sung vào định nghĩa lớp Time trong Bài tập 2.6 các toán tử (**phần chữ đỏ**):

Time	- hour: phần giờ, minute: phút, second: giây
- hour: int - minute: int - second: int	- Time() phương thức khởi tạo không tham số, gán hour, minute, second bằng 0. - Time(int h, int m, int s) phương thức khởi tạo có tham số, gán hour bằng h, minute bằng m, second bằng s.
+ Time() + Time(int h, int m, int s) + <b>operator++(): Time</b> + <b>operator--(): Time</b> + <b>operator+(int s): Time</b> + Show(): void	- <b>operator++()</b> tính và trả về mốc thời gian sau 1 giây. - <b>operator--()</b> tính và trả về mốc thời gian trước 1 giây. - <b>operator+(int s)</b> tính và trả về mốc thời gian được cộng thêm s giây. - Show() hiện thông tin đối tượng ra màn hình thời gian theo định dạng h:m:s

Sử dụng lớp Time thực hiện các yêu cầu:

- Tạo một đối tượng thời gian (t1) dùng phương thức thiết lập không tham số;
- Tính mốc thời gian sau 1 giây (dùng toán tử ++), in đối tượng ra màn hình;
- Tạo một đối tượng thời gian (t2) dùng phương thức thiết lập có tham số.
- Tính mốc thời gian trước 1 giây (dùng toán tử --), in đối tượng ra màn hình;
- Tính mốc thời gian t1 + s, với s được nhập vào từ bàn phím, in đối tượng ra màn hình.

**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  $50 \geq \text{số tiết học} > 30$  thì tiền giảm 7% (số tiết học \* học phí 1 tiết);  
+ Còn lại không được giảm.
  - Xuất thông học viên gồm: tên học viên, lớp học, số tiết, tiền học phí;
  - 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;
- 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 ();
    }
}
```