

## BÀI THỰC HÀNH 4

### Nội dung:

- Kỹ thuật thừa kế
- Tính đa hình

### Bài 4.1. Định nghĩa các lớp theo mô tả như sau:

- Lớp phương trình bậc nhất (PTB1) gồm có các thành phần:
  - Dữ liệu: hệ số  $b, c$  (phương trình  $bx + c = 0$ )
  - Phương thức:
    - Khởi tạo không tham số, có tham số;
    - Phương thức nhập, xuất dữ liệu cho PTB1;
    - Phương thức giải và biện luận PTB1.
- Lớp phương trình bậc hai (PTB2) kế thừa từ PTB1 và có thêm các thành phần:
  - Dữ liệu: hệ số  $a$  ( $ax^2 + bx + c = 0$ )
  - Phương thức:
    - Khởi tạo không tham số, có tham số;
    - Phương thức nhập, xuất dữ liệu cho PTB2;
    - Phương thức giải và biện luận PTB2.

Viết chương trình sử dụng các lớp định nghĩa ở trên để thực hiện:

- Tạo ra một đối tượng PTB1, một đối tượng PTB2;
- Xuất thông tin nghiệm của PTB1, PTB2 ra màn hình.

### Bài 4.2. Xây dựng các lớp theo mô tả sau:

- Phone (điện thoại) gồm có các thành phần dữ liệu: *mã số, nhãn hiệu* (Samsung, Iphone, Nokia,...), *giá nhập, năm sản xuất*; các phương thức: *thiết lập, nhập, xuất* thông tin điện thoại, *tính thuế nhập* dựa vào nhãn hiệu và giá nhập như sau:
  - Nếu nhãn hiệu là Iphone: thuế nhập = 10% giá nhập
  - Các nhãn hiệu khác: thuế nhập = 5% giá nhập
- SmartPhone (điện thoại thông minh) kế thừa lớp **Phone** và có thêm thành phần dữ liệu *dung lượng bộ nhớ* (GB); các phương thức: *thiết lập, nhập, xuất* thông tin SmartPhone, *tính giá bán* dựa vào dung lượng bộ nhớ như sau:
  - Dung lượng bộ nhớ  $\geq 128$ : giá bán = giá nhập + 20% giá nhập + thuế nhập.
  - $128 > \text{Dung lượng bộ nhớ} > 16$ : giá bán = giá nhập + 10% giá nhập + thuế nhập.
  - Còn lại: giá bán = giá nhập + 5% giá nhập + thuế nhập.

Viết chương trình định nghĩa các lớp và sử dụng để thực hiện các yêu cầu sau:

- Nhập danh sách n SmartPhone ( $2 \leq n \leq 30$ );
- Sắp xếp danh sách SmartPhone theo chiều giảm dần của giá nhập;
- Xuất ra danh sách SmartPhone.
- Đếm và in ra số lượng SmartPhone có năm sản xuất bằng x, với x được nhập vào từ bàn phím;

**Bài 4.3.** Cài đặt các lớp theo mô tả sau:

- Lớp hình học (Shape) gồm các thành phần: tên hình, phương thức khởi tạo, tính chu vi (virtual), tính diện tích (virtual), in thông tin hình (tên hình, chu vi, diện tích của hình).
- Lớp tam giác (Triangle) kế thừa từ lớp Shape có thêm các thành phần: 3 cạnh tam giác, phương thức khởi tạo, tính chu vi (override), tính diện tích tam giác (override)

(công thức Hero:  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , p là nửa chu vi).

- Lớp hình chữ nhật (Rectangle) kế thừa từ lớp Shape có thêm các thành phần: chiều dài, chiều rộng, phương thức khởi tạo, tính chu vi (override), tính diện tích (override).

Viết chương trình định nghĩa các lớp và sử dụng các lớp để thực hiện:

- Nhập thông tin cho n hình ( $2 \leq n \leq 20$ , cho phép chọn loại hình khi nhập)
- In ra thông tin của n hình vừa nhập;
- Tính và in ra diện tích trung bình của các hình chữ nhật.

**Ghi chú:**

- *typeof* lấy một tên kiểu dữ liệu được chỉ định tại thời gian biên dịch.
- *GetType* lấy tên kiểu dữ liệu tại thời gian chạy của một thể hiện.

```
if (ds[i].GetType () == typeof(Rectangle))
{
    sum +=ds[i].Are();
    count++;
}
return sum/count;
```

**Bài 4.4\*.** Định nghĩa các lớp theo mô tả như sau:

- Lớp Pixel có các thành phần:
  - Dữ liệu: hoành độ (x), tung độ (y)
  - Phương thức:
    - Khởi tạo không tham số, có tham số;
    - Xuất ra ký tự 'o' tại vị trí x,y.
- Lớp Colored\_Pixel kế thừa từ lớp Pixel và có thêm các thành phần:

- Dữ liệu: màu sắc (c)
- Phương thức:
  - Khởi tạo không tham số, có tham số;
  - Xuất ra ký tự 'o' tại vị trí x, y với màu chữ là c. (sử dụng cơ chế định nghĩa lại phương thức)

Viết chương trình sử dụng các lớp định nghĩa ở trên để thực hiện:

- Tạo ra một đối tượng Pixel, một đối tượng Colored\_Pixel;
- Xuất thông tin Pixel, Colored\_Pixel ra màn hình.

**Bài 4.5\*.** Định nghĩa các lớp theo mô tả như sau:

- Lớp *Xe* gồm các thành phần dữ liệu: tên, màu xe, giá xe; Các phương thức: thiết lập, nhập thông tin xe, tính lệ phí trước bạ (abstract), xuất thông tin (tên, màu xe, giá xe, lệ phí trước bạ).
- Định nghĩa lớp *Xe máy* kế thừa từ lớp *Xe* có thêm thành phần dữ liệu đăng ký lần đầu (đúng hay không); Các phương thức: thiết lập, nhập thông tin cho xe máy, tính lệ phí trước bạ (override): nếu đăng ký lần đầu là 5% giá xe, lần sau là 1% giá xe.
- Định nghĩa lớp *Xe hơi* kế thừa từ lớp *Xe* có thêm thành phần dữ liệu loại xe (1: xe dưới 9 chỗ chạy xăng, 2: xe chạy điện, ..); Các phương thức: thiết lập, nhập thông tin xe hơi, tính lệ phí trước bạ (override): Nếu loại 1 thì lệ phí là 10% giá xe, loại 2 là 0% giá xe, các loại còn lại là 2% giá xe.

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

- Nhập vào danh sách gồm n chiếc xe, với  $2 \leq n \leq 20$  (cho phép chọn loại xe khi nhập)
- Xuất ra danh sách n xe;
- Đếm và xuất ra số lượng xe máy trong danh sách n xe;
- Tính và in ra lệ phí trước bạ trung bình của các xe hơi trong danh sách.

**Bài 4.6\*** Cho định nghĩa lớp PTB1 và PTB2 như dưới đây

```
class PTB1
{
    protected int a, b;
    public PTB1(int a1=2, int b1 = 4)
    {
        a = a1; b = b1;
    }
    public void Xuat()
    {
        Console.WriteLine($"{a}\t{b}");
    }
}
class PTB2:PTB1{
    int c;
    public PTB2(int a=3, int b=5, int c = 7):base(a,b)
    {
        this.c = c;
    }
}
```

```

    }
    public new void Xuat()
    {
        base.Xuat();
        Console.WriteLine($"{t}{c}");
    }
}

```

Hãy cho biết kết quả chạy các đoạn chương trình sau:

a. `class Program`

```

{
    static void Main(string[] args)    {
        PTB2 p = new PTB2();
        p.Xuat();
    }
}

```

b. `class Program`

```

{
    static void Main(string[] args)
    {
        PTB1 p = new PTB1();
        p.Xuat();
    }
}

```

c. `class Program`

```

{
    static void Main(string[] args)    {
        PTB1 p = new PTB1(1);
        p.Xuat();
    }
}

```

d. `class Program`

```

{
    static void Main(string[] args)    {
        PTB2 p = new PTB2(1);
        p.Xuat();
    }
}

```

e. `class Program`

```

{
    static void Main(string[] args)
    {
        PTB2 p = new PTB2(1,2);
        p.Xuat();
    }
}

```