

Chương 5. Tính đa hình

Nội dung

#2

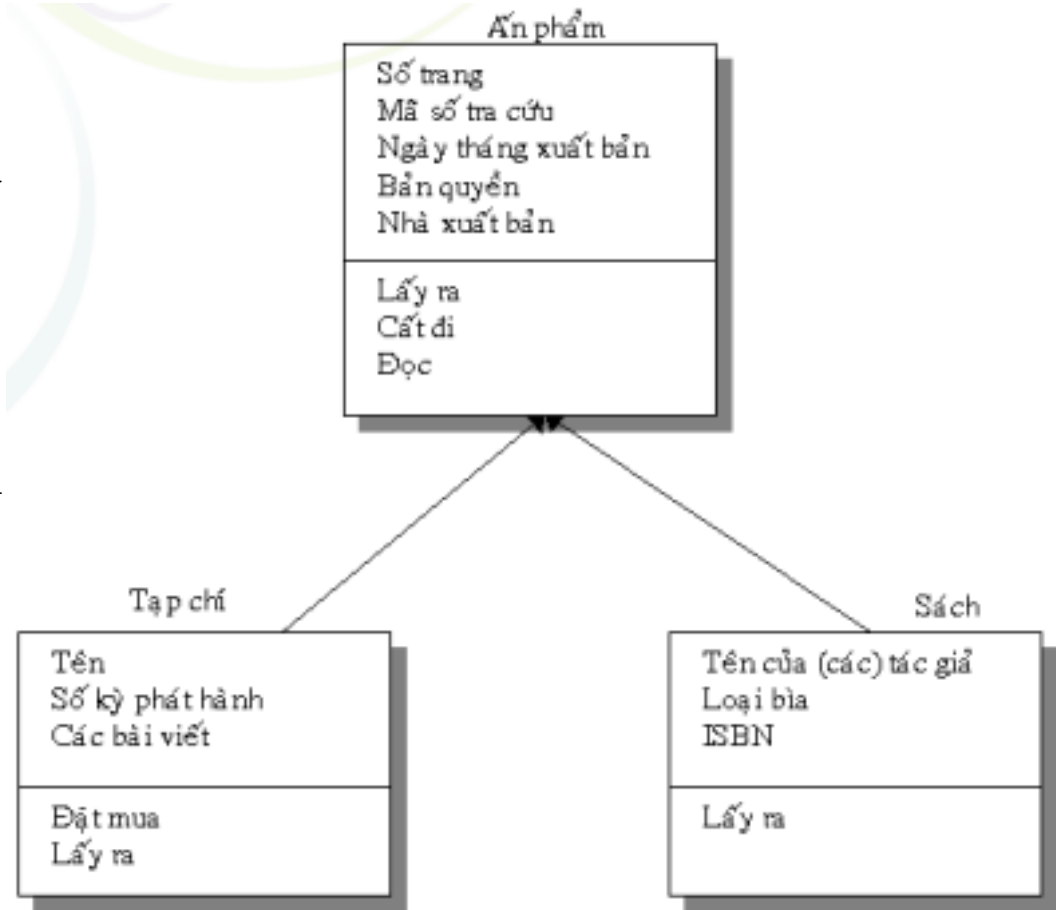
- Khái niệm về tính đa hình
- Thiết kế lớp trừu tượng ▪

Các ví dụ minh họa

Đặt vấn đề

#3

Làm thế nào lưu
danh sách (mảng)
2 loại ấn phẩm
cùng lúc & thực
thi đúng hành
động “LayRa” của
loại ấn phẩm đó ?



Khái niệm tính đa hình

#4

- Tính đa hình là khả năng để cho một thông điệp có thể thực hiện bằng nhiều cách khác nhau tùy thuộc vào đối tượng cụ thể nhận thông điệp.
- Khi một lớp dẫn xuất được tạo ra, nó có thể thay đổi cách thực hiện các phương thức nào đó mà nó thừa hưởng từ lớp cơ sở.

Trừu tượng hóa

#5

- Trừu tượng hóa là khả năng mô tả khái quát các thao tác chung của các lớp đối tượng.
- Đặc tính này giúp cho việc thiết kế lớp mang tính đa hình

Ví dụ

#6

- Nhận xét đoạn code sau

```
static void Main()  
{  
    AnPham a = new AnPham();  
    a.LayRa();  
  
    TapChi t = new TapChi();  
    t.LayRa();  
}
```

```
    a = t;  
    a.LayRa();  
}
```

Ví

dụ2 #7

- Nhận xét đoạn code sau

```
static void Main()  
{  
    AnPham[] ds = new AnPham[100];  
    for(int i=0;i<100;i++)  
    {  
        if(Nhập tạp chí ?)
```

```
        {  
            ds[i] = new TapChi();  
        }  
    else  
    {  
        ds[i] = new Sach();  
    }  
}  
}
```

Lớp trừu tượng

#8

Phương thức trừu tượng là phương thức chỉ có tên thôi và nó phải được cài đặt lại ở tất cả các lớp kế thừa. Lớp trừu tượng chỉ thiết lập một cơ sở cho các lớp kế thừa mà nó không thể có bất kỳ một thể hiện nào tồn tại

```

abstract class COSO
{
    protected kiểu data1;
    protected kiểu data2;
    public abstract void
Method1(); public abstract void
Method2(); }
    class DANXUAT : COSO
    {

```

```

        private kiểu data3;
        public override void
Method1() {}
        public override void
Method2() {}
    }

```

Lớp trừu tượng

#9

```

abstract class Window
{

```

```

    protected int top, left;
    public Window(int top, int left)

```



```
{ this.top = top; this.left = left; }  
abstract public void DrawWindow( );  
}  
class ListBox : Window  
{  
    private string listBoxContents;  
    public ListBox(int top, int left, string contents) : base(top,  
left) { listBoxContents = contents; }  
    public override void DrawWindow( )  
    {  
        Console.WriteLine("Writing string to the listbox: {0}",  
listBoxContents); }  
}
```

Lớp trừu tượng

#10

```
public class Button : Window  
{  
    public Button( int top, int left): base(top, left)
```

```
{ }  
public override void DrawWindow( )  
    { Console.WriteLine("Drawing a button at {0}, {1}\n", top, left); } }  
public class Tester  
{  
    static void Main( )  
    {  
        Window[] winArray = new Window[3];  
        winArray[0] = new ListBox(1,2,"First List Box");  
        winArray[1] = new ListBox(3,4,"Second List Box");  
        winArray[2] = new Button(5,6);  
        for (int i = 0; i < 3; i++)  
            { winArray[i].DrawWindow( ); }  
    }  
}
```

Ví dụ

FAQs

