

Khoá học: Game Developer

Bài 06: Tính Đa hình (Polymorphism)



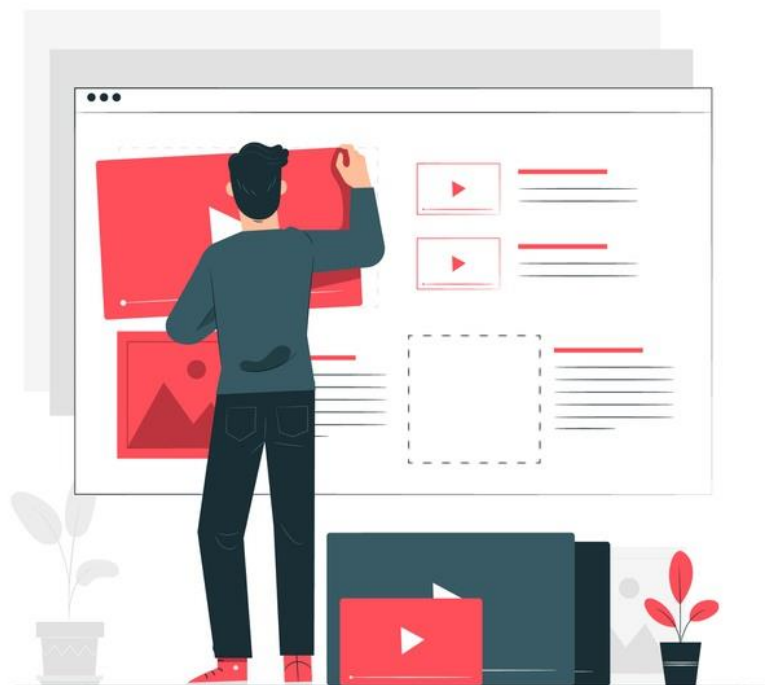


Mục tiêu bài học

Kết thúc bài học này, bạn có thể

- Hiểu khái niệm về sự đa hình
- Hiểu khái niệm về phương thức ảo
- Hiểu cơ chế ghi đè phương thức.
- Phân biệt 02 cơ chế Overload và Override

Nội dung



1. Khái niệm Kế thừa
2. Cơ chế ghi đè và che giấu của phương thức
3. Khái niệm Đa hình
4. Cơ chế nạp chồng

Tham khảo: <https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>

Tính đa hình

Polymorphism

- **Đa hình** tuân theo một nguyên tắc sinh học mà một giống loài/tổ chức có thể thực hiện cùng hành vi nhưng hình thức thể hiện khác nhau.
- **Poly** means “many” and **morph** means “form.” Tính đa hình đề cập đến việc có thể sử dụng nhiều dạng của một loại mà không cần quan tâm đến các chi tiết.
- Trong lập trình, tính đa hình được thực hiện thông qua **cơ chế ghi đè (overriding mechanism)**.
- Các lớp con có thể tự định nghĩa lại các hành vi và thừa hưởng một vài chức năng từ lớp cha.



Cơ chế ghi đè

Overriding Mechanism

- **Mục đích của việc ghi đè** là để sửa lại nội dung phương thức của lớp cha trong lớp con.
- **Cơ chế ghi đè** là việc khai báo các phương thức trùng tên, cùng danh sách tham số và kiểu trả về giữa lớp cha và lớp con.

```
Animal a = new Duck();  
a.Speak();
```

Kết quả: Quack! Quack

```
public class Animal {  
    public virtual void Speak()  
    {  
        Console.WriteLine("Cannot speak");  
    }  
}
```

```
public class Duck : Animal {  
    public override void Speak()  
    {  
        Console.WriteLine("Quack! Quack");  
    }  
}
```

Phương thức ảo

Virtual Method

- Từ khoá **virtual** để tạo phương thức ảo trong lớp cha.
- Từ khoá **override** được sử dụng khi lớp con cần ghi đè lên nội dung của lớp cha.

```
Animal a = new Duck();  
a.Speak();
```

Kết quả: Quack! Quack

```
public class Animal {  
    public virtual void Speak()  
    {  
        Console.WriteLine("Cannot speak");  
    }  
}
```

```
public class Duck : Animal {  
    public override void Speak()  
    {  
        Console.WriteLine("Quack! Quack");  
    }  
}
```

Phương thức ghi đè

Override Method

- Chỉ được phép override phần hiện thực hàm (implementation)
- Không được phép override phần khai báo hàm (method name and signature)
- Tức là, không được khai báo thêm tham số truyền vào khi thừa kế từ 1 hàm không có tham số truyền vào đó.

```
Animal a = new Duck();  
a.Speak();
```

Kết quả: Quack! Quack

```
public class Animal {  
    public virtual void Speak()  
    {  
        Console.WriteLine("Cannot speak");  
    }  
}
```

```
public class Duck : Animal {  
    public override void Speak(String name)  
    {  
        Console.WriteLine("Quack! Quack");  
    }  
}
```

Phương thức che giấu

Hiding Method

- Từ khoá **new** được sử dụng khi đối tượng muốn che giấu các phương thức thừa kế.

```
Animal a = new Duck();  
Duck d = new Duck();  
a.Speak();  
d.Speak();
```

Kết quả: Cannot speak
 Quack! Quack

```
public class Animal {  
    public void Speak()  
    {  
        Console.WriteLine("Cannot speak");  
    }  
}
```

```
public class Duck : Animal {  
    public new void Speak()  
    {  
        Console.WriteLine("Quack! Quack");  
    }  
}
```


Cơ chế nạp chồng và ghi đè

Overloading and Overriding Mechanism

Overloading

- Cơ chế cho phép khai báo các phương thức trùng tên nhưng khác danh sách tham số.
 - Số lượng tham số khác nhau
 - Hoặc cùng số lượng nhưng:
 - Khác kiểu dữ liệu
 - Trật tự tham số khác nhau.

Overriding

- Cơ chế cho phép ghi đè các phương thức trùng tên và trùng chữ ký (signature).
 - Phương thức được thừa kế phải khai báo virtual/abstract