

Đối tượng và lớp trong Java

Trong chương này, chúng ta sẽ tìm hiểu về lớp và đối tượng trong Java. Trong kỹ thuật lập trình hướng đối tượng, chúng ta thiết kế một chương trình bởi sử dụng các lớp và các đối tượng.

- **Object** - Đối tượng là thực thể mang tính vật lý cũng như mang tính logic, trong khi lớp chỉ là thực thể logic. Đối tượng có các trạng thái và các hành vi. Ví dụ: Một dog có trạng thái là color, name, breed (dòng dõi) và cũng có các hành vi: Wag (vẫy đuôi), bark (sủa), eat (ăn). Một đối tượng là một instance (ví dụ, trường hợp) của một lớp.
- **Class** - Một lớp là một nhóm các đối tượng mà có các thuộc tính chung. Lớp là một Template hoặc bản thiết kế từ đó đối tượng được tạo.

Đối tượng trong Java

Đó là một thực thể có trạng thái và hành vi, ví dụ như bàn, ghế, xe con, mèo, ... Nó có thể mang tính vật lý hoặc logic. Ví dụ về logic đó là Banking system.

Một đối tượng có ba đặc trưng sau:

- **Trạng thái:** biểu diễn dữ liệu (giá trị) của một đối tượng.
- **Hành vi:** biểu diễn hành vi (tính năng) của một đối tượng như gửi tiền vào, rút tiền ra, ...
- **Nhận diện:** việc nhận diện đối tượng được triển khai thông qua một ID duy nhất. Giá trị của ID là không thể nhìn thấy với người dùng bên ngoài. Nhưng nó được sử dụng nội tại bởi JVM để nhận diện mỗi đối tượng một cách duy nhất.

Ví dụ: Bút là một đối tượng. Nó có tên là Thiên Long, có màu trắng, ... được xem như là trạng thái của nó. Nó được sử dụng để viết, do đó viết là hành vi của nó.

Đối tượng là sự thể hiện (Instance) của một lớp. Lớp là một Template hoặc bản thiết kế từ đó đối tượng được tạo. Vì thế đối tượng là Instance (kết quả) của một lớp.

Lớp trong Java

Một lớp là một nhóm các đối tượng mà có các thuộc tính chung. Lớp là một Template hoặc bản thiết kế từ đó đối tượng được tạo. Một lớp trong Java có thể bao gồm:

- Thành viên dữ liệu

- Phương thức
- Constructor
- Block
- Lớp và Interface

Cú pháp để khai báo một lớp

```
class ten_lop{  
    thanh_vien_du_lieu;  
    phuong_thuc;  
}
```

Ví dụ đơn giản về Lớp và Đối tượng trong Java

Trong ví dụ này, chúng ta tạo một lớp Student có hai thành viên dữ liệu là id và name. Chúng ta đang tạo đối tượng của lớp Student bởi từ khóa new và in giá trị đối tượng.

```
class Student1{  
    int id; //thanh vien du lieu (cung la bien instance)  
    String name; //thanh vien du lieu (cung la bien instance)  
  
    public static void main(String args[]){  
        Student1 s1=new Student1(); //tao mot doi tuong Student  
        System.out.println(s1.id);  
        System.out.println(s1.name);  
    }  
}
```

Một lớp có thể chứa bất kỳ loại biến sau:

- **Biến Local:** Các biến được định nghĩa bên trong các phương thức, constructor hoặc block code được gọi là biến Local. Biến này sẽ được khai báo và khởi tạo bên trong phương thức và biến này sẽ bị hủy khi phương thức đã hoàn thành.
- **Biến Instance:** Các biến instance là các biến trong một lớp nhưng ở bên ngoài bất kỳ phương thức nào. Những biến này được khởi tạo khi lớp này được tải. Các biến instance

có thể được truy cập từ bên trong bất kỳ phương thức, constructor hoặc khối nào của lớp cụ thể đó.

- **Biến Class:** Các biến class là các biến được khai báo với một lớp, bên ngoài bất kỳ phương thức nào, với từ khóa static.

Phương thức trong Java

Trong Java, một phương thức là khá giống hàm, được sử dụng để trưng bày hành vi của một đối tượng. Phương thức giúp code tăng tính tái sử dụng và tối ưu hóa code.

Từ khóa new được sử dụng để cấp phát bộ nhớ tại runtime.

Constructor trong Java:

Khi bàn luận về các lớp, một trong những chủ đề quan trọng là các constructor. Mỗi lớp có một constructor. Nếu chúng ta không viết một constructor một cách rõ ràng cho một lớp thì bộ biên dịch Java xây dựng một constructor mặc định cho lớp đó.

Mỗi khi một đối tượng mới được tạo ra, ít nhất một constructor sẽ được gọi. Quy tắc chính của các constructor là chúng có cùng tên như lớp đó. Một lớp có thể có nhiều hơn một constructor.

Sau đây là ví dụ về một constructor:

```
public class Xecon{
    public Xecon(){
    }

    public Xecon(String ten){
        // Constructor nay co mot tham so la ten.
    }
}
```

Java cũng hỗ trợ **Lớp Singleton trong Java**, ở đây bạn sẽ có thể tạo chỉ một instance của một lớp.

Tạo một đối tượng trong Java:

Như đã đề cập trước đó, một lớp cung cấp bản thiết kế cho các đối tượng. Vì thế, về cơ bản, một đối tượng được tạo từ một lớp. Trong Java, từ khóa new được sử dụng để tạo một đối tượng mới.

Có ba bước khi tạo một đối tượng từ một lớp:

- **Khai báo:** Một khai báo biến với một tên biến với một loại đối tượng.
- **Cài đặt:** Từ khóa new được sử dụng để tạo đối tượng
- **Khởi tạo:** Từ khóa new được theo sau bởi một lời gọi một constructor. Gọi hàm này khởi tạo đối tượng mới.

Dưới đây là ví dụ về tạo một đối tượng:

```
public class Xecon{

    public Xecon(String ten){

        // Constructor nay co mot tham so la ten.

        System.out.println("Ten xe la : " + ten );

    }

    public static void main(String []args){

        // Lenh sau se tao mot doi tuong la Xecuatoi

        Xecon Xecuatoi = new Xecon( "Toyota" );

    }

}
```

Nếu chúng ta biên dịch và chạy chương trình, nó sẽ cho kết quả sau:

```
Ten xe la :Toyota
```

Truy cập các biến instance và các phương thức trong Java

Các biến instance và các phương thức được truy cập thông qua các đối tượng được tạo. Để truy cập một biến instance, path sẽ là như sau:

```
/* Dau tien, ban tao mot doi tuong */
Doituongthamchieu = new Constructor();

/* Sau do ban goi mot bien nhu sau */
Doituongthamchieu.TenBien;
```

```
/* Bay gio ban co the goi mot phuong thuc lop nhu sau */  
Doituongthamchieu.TenPhuongThuc();
```

Ví dụ:

Ví dụ này giải thích cách để truy cập các biến instance và các phương thức của một lớp:

```
public class Xecon{  
  
    int Giaxe;  
  
    public Xecon(String ten){  
        // Constructor nay co mot tham so la ten.  
        System.out.println("Ten xe la : " + ten );  
    }  
    public void setGia( int gia ){  
        Giaxe = gia;  
    }  
  
    public int getGia( ){  
        System.out.println("Gia mua xe la : " + Giaxe );  
        return Giaxe;  
    }  
    public static void main(String []args){  
        /* Tao doi tuong */  
        Xecon Xecuatoi = new Xecon( "Toyota" );  
  
        /* Goi mot phuong thuc lop de thiet lap gia xe */  
        Xecuatoi.setGia( 100000000 );  
  
        /* Goi mot phuong thuc lop khac de lay gia xe */  
        Xecuatoi.getGia( );  
  
        /* Ban cung co the truy cap bien instance nhu sau */  
    }  
}
```

```
        System.out.println("Gia tri bien :" + Xecuatoi.Giaxe );
    }
}
```

Biên dịch và thực thi chương trình sẽ cho kết quả sau:

```
Ten xe la :Toyota
Gia mua xe la :1000000000
Gia tri bien :1000000000
```

Ví dụ đối tượng và lớp mà duy trì bản ghi các sinh viên

Trong ví dụ này, chúng ta tạo hai đối tượng của lớp Student và khởi tạo giá trị của các đối tượng này bằng việc triệu hồi phương thức insertRecord trên nó. Ở đây, chúng ta đang hiển thị trạng thái (dữ liệu) của các đối tượng bằng việc triệu hồi phương thức displayInformation.

```
class Student2{
    int rollno;
    String name;

    void insertRecord(int r, String n){ //phuong thuc
        rollno=r;
        name=n;
    }

    void displayInformation(){System.out.println(rollno+" "+name);} //phuong thuc

    public static void main(String args[]){
        Student2 s1=new Student2();
        Student2 s2=new Student2();

        s1.insertRecord(111,"HoangThanh");
        s2.insertRecord(222,"ThanhHuong");

        s1.displayInformation();
        s2.displayInformation();
    }
}
```

```
}  
}
```

Ví dụ khác về lớp và đối tượng trong Java

Ví dụ khác duy trì các bản ghi của lớp Rectangle. Phần giải thích tương tự như trên:

```
class Rectangle{  
    int length;  
    int width;  
  
    void insert(int l,int w){  
        length=l;  
        width=w;  
    }  
  
    void calculateArea(){System.out.println(length*width);}  
  
    public static void main(String args[]){  
        Rectangle r1=new Rectangle();  
        Rectangle r2=new Rectangle();  
  
        r1.insert(11,5);  
        r2.insert(3,15);  
  
        r1.calculateArea();  
        r2.calculateArea();  
    }  
}
```

Các cách khác nhau để tạo đối tượng trong Java?

Có nhiều cách để tạo một đối tượng trong Java. Đó là:

- Bằng từ khóa new

- Bằng phương thức newInstance()
- Bằng phương thức clone(),
- Bằng phương thức factory, ...

Đối tượng vô danh (anonymous) trong Java

Vô danh hiểu đơn giản là không có tên. Một đối tượng mà không có tham chiếu thì được xem như là đối tượng vô danh. Nếu bạn phải sử dụng một đối tượng chỉ một lần, thì đối tượng vô danh là một hướng tiếp cận tốt.

```
class Calculation{

    void fact(int n){
        int fact=1;
        for(int i=1;i<=n;i++){
            fact=fact*i;
        }
        System.out.println("factorial is "+fact);
    }

    public static void main(String args[]){
        new Calculation().fact(5); //Goi phuong thuc voi doi tuong vo danh (anonymous)
    }
}
```

Tạo nhiều đối tượng bởi chỉ một kiểu

Chúng ta có thể tạo nhiều đối tượng bởi chỉ một kiểu như khi chúng ta thực hiện trong các kiểu gốc. Ví dụ:

```
Rectangle r1=new Rectangle(),r2=new Rectangle(); //Tao hai doi tuong
```

Ví dụ:

```
class Rectangle{
    int length;
    int width;
```



```
void insert(int l,int w){
    length=l;
    width=w;
}

void calculateArea(){System.out.println(length*width);}

public static void main(String args[]){
    Rectangle r1=new Rectangle(),r2=new Rectangle(); //Tao hai doi tuong

    r1.insert(11,5);
    r2.insert(3,15);

    r1.calculateArea();
    r2.calculateArea();
}
}
```

Kết quả là:

```
Output:55
        45
```