Bài Học

Kỹ thuật viết Kế thừa từ Interface





- √ Nó không phải là giao diện người dùng.
- ✓ Để tránh đi tính phức tạp của đa thừa kế (multi-inheritance) trong lập trình hướng đối tượng, Java thay thế bằng interface.
- ✓ Một lớp có thể kế thừa từ nhiều interface với các lớp khác để thừa hưởng thêm vùng dữ liệu và phương thức của các interface này.
- ✓ Interface được khai báo như một lớp, với các thuộc tính là các hằng và các phương thức là các phương thức trừu tượng.
- ✓ Methods trừu tượng là các methods chỉ có phần header, không có phần body.
- ✓ Trong các lớp có cài đặt các interface ta phải tiến hành cài đặt cụ thể các phương thức này.
- ✓ Các class có thể implements các interface.

```
interface DienTich {
  public abstract void sHinhVuong(float a);
  public abstract void sHinhChuNhat(float a, float b);
interface ChuVi {
  public abstract void cVHinhVuong(float a);
  public abstract void cVHinhChuNhat(float a, float b);
interface CongThuc extends DienTich, ChuVi {
```

```
class CongThucTinh implements CongThuc {
  @Override
  public void sHinhVuong(float a) {
    System.out.println("Diện tích hình vuông là: " + a * a);
  @Override
  public void sHinhChuNhat(float a, float b) {
  System.out.println("Diện tích hình chữ nhật là: " + a * b);
  @Override
 public void cVHinhVuong(float a) {
    System.out.println("Chu vi hình vuông là: " + 4 * a);
  @Override
  public void cVHinhChuNhat(float a, float b) {
 System.out.println("Chu vi hình chữ nhật là: " + 2 * (a + b));
```

```
public class JavaAndroidVn {
   public static void main(String[] args) {
      CongThucTinh x = new CongThucTinh();
      x.sHinhVuong(5.0f);
      x.cVHinhVuong(5.0f);
      x.sHinhChuNhat(5.5f, 10.0f);
      x.cVHinhChuNhat(5.5f, 10.0f);
   }
}
```





Hey! **Coding** is easy!