Smart House

IT22020

In a smart house system, different devices like lights and smart speakers perform common action like turning on and off. To share this common behavior we use an abstract class called SmartDevice. Some device, like smart speakers, can also respond to voice commands, so we use a Java interface called VoiceControllable for that specific ability.

code:

```
interface VoiceControllable {
    void respondToVoice(String command);
}

abstract class SmartDevice{
    String name;

SmartDevice(String name){
    this.name = name; }
```

```
void turnOn(){
    System.out.println(name + " is ON.");
    }
void turnOff(){
    System.out.println(name + " is OFF.");
    }
abstract void deviceType();
}

class Light extends SmartDevice{
    Light(String name){
        super(name);
    }
void deviceType(){
    System.out.println(name + " is a Smart Light");
    }
}
```

```
class Speaker extends SmartDevice implements
                    VoiceControllable {
    Speaker (String name) {
        super(name)
    }
void deviceType() {
    System.out.println(name + " is a smart speaker.");
    }

public void responToVoice (String name) {
    System.out.println(" voice command : " + command);
    }
}

public class main {
    public Stroatic void main (String[] args) {
Light light = new Light ("Philips Light");
Speaker speaker = new Speaker("Amazon Echo");
    light.turnOn();
    light.deviceType();
```

```
        light.turnOff();
        System.out.println();
        speaker.turnOn();
        speaker.deviceType();
        speaker.respondToVoice("play music");
        speaker.turnOff();
    }
}
```

Abstract Class Vs Interface

Abstract class	Interface
can be extended using extends	can be implemented using implements
can have both abstract and non abstract methods	default and static methods can be implemented
can have instance variables	only public static final Constants

abstract class	Interface
Can have constructors	interface can not have constructors
support only single class inheritance	A class can implement multiple inheritance.
Can use privet, protected	All methods are public by default