

Introduction

Javan Bridge-suunnittelumalli on rakenteellinen (Structural) suunnittelumalli. Sen idea on irrottaa abstraktio sen toteutuksesta, niin että kumpikin puoli voi kehittyä itsenäisesti. Eli toisinsanoen ei sido tiettyyn toteutukseen, vaan siinä on viittaus rajapintaan, joita konkreetiset toteutukset tarjoavat.

New Functionality

Esimerkissäni olen käyttänyt Refactoring.guru:n suunnittelumallin esimerkkiä. Olen lisännyt esimerkkiin SmartTv-luokan, joka implementoi Device-rajapintaa. Sen lisäksi luokassa on setApp()- sekä getApp()-metodit.

Kaukosäätimenä toimii SmartRemote-luokka, joka laajentaa Remote-rajapintaa toteuttavaa BasicRemote-luokkaa.

Kaukosäätimellä voi avata applikaatioita älytelevisiossa, kuten nykyisin kaikissa moderneissa televisioissa. Tämä uuden toiminnallisuuden idea tuli arjesta – nykyään TV:ssä käytetään enemmän sovelluksia kuin perinteisiä kanavia.

Linkki suunnittelumallin esimerkkiin:

<https://refactoring.guru/design-patterns/bridge/java/example>

Implementation

```
SmartTv.java > SmartTv > getVolume()
public class SmartTv implements Device{
    private boolean on = false;
    private int volume = 30;
    private int channel = 0;
    private String app = "Youtube";

    @Override
    public boolean isEnabled(){
        return on;
    }

    @Override
    public void enable(){
        on = true;
    }

    @Override
    public void disable(){
        on = false;
    }

    @Override
    public int getVolume(){
        return volume;
    }

    @Override
    public void setVolume(int volume){
        if (volume > 100){
            this.volume = 100;
        } else if (volume < 0) {
            this.volume = 0;
        } else {
            this.volume = volume;
        }
    }

    @Override
    public int getChannel(){
        return channel;
    }

    @Override
    public void setChannel(int channel){
        this.channel = channel;
    }

    public void setApp(String name){
        this.app = name;
    }

    public String getApp(){
        return app;
    }

    @Override
    public void printStatus(){
        System.out.println(x:"-----");
        System.out.println(x:"| I'm SmartTV set.");
        System.out.println("I'm " + (on ? "enabled" : "disabled"));
        System.out.println("Current volume is " + volume + "%");
        System.out.println("Current app open: " + app);
        System.out.println(x:"-----\n");
    }
}

src > SmartRemote.java > ...
1
2 public class SmartRemote extends BasicRemote{
3     public SmartRemote(Device device){
4         super.device = device;
5     }
6
7     public void openApp(String appName){
8
9         if (device instanceof SmartTv) {
10             ((SmartTv) device).setApp(appName);
11             System.out.println("SmartRemote: Opening app " + appName);
12         } else {
13             System.out.println("SmartRemote: Device does not support " + appName);
14         }
15     }
16 }
17
```

Verification

Testasin uuden toiminnallisuuden luomalla SmartTv-olion ja ohjaamalla sitä SmartRemote-kaukosäätimellä. Alla esimerkkikoodi ja tuloste, joka vahvistaa applikaation vaihdon.

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
        testDevice(new Tv());  
        testDevice(new Radio());  
        testDevice(new SmartTv());  
    }  
  
    public static void testDevice(Device device) {  
        System.out.println(x:"Tests with basic remote.");  
        BasicRemote basicRemote = new BasicRemote(device);  
        basicRemote.power();  
        device.printStatus();  
  
        System.out.println(x:"Tests with advanced remote.");  
        AdvancedRemote advancedRemote = new AdvancedRemote(device);  
        advancedRemote.power();  
        advancedRemote.mute();  
        device.printStatus();  
  
        System.out.println(x:"Test with Smart remote");  
        SmartRemote smartRemote = new SmartRemote(device);  
        smartRemote.power();  
        device.printStatus();  
        smartRemote.openApp(appName:"Spotify");  
        device.printStatus();  
    }  
}
```

```
Test with Smart remote  
Remote: power toggle  
-----  
| I'm SmartTV set.  
| I'm enabled  
| Current volume is 30%  
| Current app open: Youtube  
-----  
  
SmartRemote: Opening app Spotify  
-----  
| I'm SmartTV set.  
| I'm enabled  
| Current volume is 30%  
| Current app open: Spotify  
-----
```

Conclusion

Tässä tehtävässä laajensin Bridge-suunnittelumallin alkuperäistä esimerkkiä lisäämällä SmartTV- sekä SmartRemote luokkia. Uuden toiminnallisuuden myötä on mahdollista simuloida modernia käyttötilannetta.

Kokonaisuudessaan uusi toiminnallisuus osoittaa, kuinka Bridge-malli mahdollistaa helposti laajennettavan ja ylläpidettävän rakenteen, johon voidaan lisätä uusia laitteita ja uusia ohjaimia ilman merkittäviä muutoksia koodissa.