Bridge Pattern Extension Report

Name: Wang Qingyun

Date: 2025-04-29

Introduction

The Bridge pattern is a structural design pattern that decouples an abstraction from its implementation, allowing the two to vary independently. It is particularly useful when both the abstraction and implementation may evolve separately over time. In the original implementation, the system had devices (TV and Radio) implementing a Device interface and remotes (BasicRemote and AdvancedRemote) implementing a Remote interface. The remotes interacted with the devices through composition, following the Bridge pattern principles.

New Functionality

To extend the system, two new classes were introduced: SmartTv and SmartRemote. The SmartTv class extends the basic functionality of TV by introducing a new feature browseInternet(), allowing users to simulate browsing the internet. The SmartRemote class extends AdvancedRemote and introduces voiceControl(String command) functionality, which allows controlling the device through voice commands such as turning on/off, changing volume, setting channels, and browsing the internet.

Implementation

- SmartTv.java:

```
package devices;

public class SmartTv implements Device {
   private boolean on = false;
   private int volume = 30;
   private int channel = 1;

   @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; }

    @Override
    public void printStatus() {
        System.out.println("------------------------------------");
        System.out.println("| I'm Smart TV.");
        System.out.println("| I'm " + (on ? "enabled" : "disabled"));
        System.out.println("| Current volume is " + volume + "%");
        System.out.println("| Current channel is " + channel);
        System.out.println("------------------------------------
");
    }

    public void browseInternet() {
        if (on) {
            System.out.println("Browsing the internet on Smart TV...");
        } else {
            System.out.println("Cannot browse internet. Smart TV is turned off.");
        }
    }
}
```

- SmartRemote.java:

```
package remotes;

import devices.Device;
import devices.SmartTv;

public class SmartRemote extends AdvancedRemote {

    public SmartRemote(Device device) {
        super(device);
    }

    public void voiceControl(String command) {
        System.out.println("Voice Control Activated: " + command);
        if (command.equalsIgnoreCase("turn on")) {
            device.enable();
        } else if (command.equalsIgnoreCase("turn off")) {
            device.disable();
        } else if (command.equalsIgnoreCase("volume up")) {
            device.setVolume(device.getVolume() + 10);
        } else if (command.equalsIgnoreCase("volume down")) {
            device.setVolume(device.getVolume() - 10);
        } else if (command.startsWith("set channel ")) {
            try {
                int channel = Integer.parseInt(command.substring(12));
                device.setChannel(channel);
            } catch (NumberFormatException e) {
                System.out.println("Invalid channel number.");
            }
        } else if (device instanceof SmartTv && command.equalsIgnoreCase("browse internet"))
{
            ((SmartTv) device).browseInternet();
        } else {
            System.out.println("Unknown voice command.");
        }
    }
}
```

Verification

The SmartTv and SmartRemote functionalities were verified by updating the Demo.java class to include testing of the new classes. The test included turning on the SmartTv, browsing the internet, adjusting the volume, changing the channel, and turning off the device using voice commands. The expected outputs matched the functionalities implemented.

Conclusion

This extension successfully demonstrated how the Bridge pattern can be expanded by introducing new devices and remotes without modifying existing code. The SmartTv and SmartRemote added meaningful new features while adhering to the open/closed principle.

Alternative designs, such as adding more specialized remotes or more smart device features, were considered but kept for future improvements.