

Bridge Assignment

Design Patterns

1. Introduction

The bridge design pattern let's you separate a class or a set of classes into two separate hierarchies to enable developing the parts separately. I chose the recommended implementation (<https://refactoring.guru/design-patterns/bridge/java/example#example-0--remotes>). to extend. In this implementation remotes have been separated from devices to allow for independent development of each and to increase interoperability.

2. New functionality

I added a SmartTV and a SmartRemote to the implementation. The SmartTV has Netflix-functionality and the remote can toggle Netflix on/off.

3. Implementation

SmartTV:

```
public class SmartTv extends Tv{
    private boolean netflix = false;

    @Override
    public void toggleNetflix(){
        netflix = !netflix;
    }
    @Override
    public void printStatus(){
        super.printStatus();
        System.out.println("| Smart features:");
        System.out.println("| Netflix is " + (netflix ? "on" : "off"));
        System.out.println("-----\n");
    }
}
```

SmartRemote:

```
public class SmartRemote extends AdvancedRemote{
    public SmartRemote(Device device) {
        super(device);
    }
    public void toggleNetflix(){
        System.out.println("Remote: toggling netflix");
        device.toggleNetflix();
    }
}
```

I also had to add a toggleNetflix() function to the Device interface.

4. Verification

I verified the code works as intended by adding tests for SmartRemotes in the testDevice()-function and by running the testDevice()-function with a SmartTv as a parameter.

```
Tests with smart remote
Remote: toggling netflix
Radio: Sorry, no netflix on this device
-----
| I'm radio.
| I'm disabled
| Current volume is 0%
| Current channel is 1
-----
```

Image 1 Output of the code wit a new Radio()

```
Tests with advanced remote.
Remote: power toggle
Remote: mute
-----
| I'm TV set.
| I'm disabled
| Current volume is 0%
| Current channel is 1
-----

| This tv is smart
| Smart features:
| Netflix is off
-----

Tests with smart remote
Remote: toggling netflix
-----
| I'm TV set.
| I'm disabled
| Current volume is 0%
| Current channel is 1
-----

| This tv is smart
| Smart features:
| Netflix is on
-----
```

Image 2 Output of the code with a new SmartTv()

5. Conclusion

New SmartTv device was added and new SmartRemote remote was added. They are both implemented by extending Tv and AdvancedRemote respectively. They both could have just implemented their respective interfaces but as a SmartTv and SmartRemote should have all the existing functionality of their non-smart counterparts it seemed counterproductive at this time.