Making a Light Switch

Make a reusable Switch class that can be used with ANY object, not just the Fan and Light class. When writing your Switch class, create a Switchable interface to make your code simpler and easier.

Setup

Download DeltaAbstraction.zip and open it in Android Studio.

The Device

Create a Device class that is abstract with the following properties:

- Has a name with a Setter and Getter
- Has a method called breakDevice with no implementation

The Fan

Create a Fan class that is concrete with the following properties:

- Extends from Device
- Has a method turnOn that uses its name as the Log tag and writes something a fan would do when turned on to the log
- Has a method turnOff that uses its name as the Log tag and writes something a fan would do when turned off to the log
- Implement the breakDevice() method that uses its name as the Log tag and writes something a fan would do when it was broken or smashed to the log

The Light

Create a Light class that is concrete with the following properties:

- Extends from Device
- Has a method turnOn that uses its name as the Log tag and writes something a light would do when turned on to the log
- Has a method turnOff that uses its name as the Log tag and writes something a light would do when turned off to the log

- Implement the breakDevice() method that uses its name as the Log tag and writes something a light would do when it was broken or smashed to the log

The Switch

Create a Switch and a separate Switchable interface to solve the following constraints:

- Has methods that will add or remove an object to a List (like an ArrayList)
- has a flipSwitchUp method that will call turnOn() on every object in that list
- has a flipSwitchDown method that will call turnOff() on every object in that list

Putting it all together

Make any adjustments in the rest of your code to perform the following in the onCreate method of AbstractionActivity:

- Instantiate a Switch object
- Instantiate a Fan object
- Instantiate a Light object
- Wire both of them to the switch and flip the switch on and off using the methods you created.
- Verify you have the right responses in the log for the expected behavior
- Congratulations, you just graduated to being a stronger Java developer!