

```
1: package com.delta.abstraction;
2:
3: import android.app.Activity;
4: import android.os.Bundle;
5: import android.view.Menu;
6: import android.view.MenuItem;
7:
8: public class AbstractionActivity extends Activity {
9:
10:     @Override
11:     protected void onCreate(Bundle savedInstanceState) {
12:         super.onCreate(savedInstanceState);
13:         setContentView(R.layout.activity_abstraction);
14:         //
15:         // Start here
16:         //
17:
18:         Fan myFan = new Fan();
19:         Light myLight = new Light();
20:
21:         myFan.breakDevice();
22:         myLight.breakDevice();
23:
24:
25:         Switch mySwitch = new Switch();
26:
27:         mySwitch.wireUp(myFan);
28:         mySwitch.wireUp(myLight);
29:
30:         mySwitch.flipSwitchUp();
31:         mySwitch.flipSwitchDown();
32:
33:         mySwitch.unWire(myFan);
34:
35:         mySwitch.flipSwitchUp();
36:         mySwitch.flipSwitchDown();
37:
38:
39:     }
40:
41:
42:     @Override
43:     public boolean onCreateOptionsMenu(Menu menu) {
44:
45:         // Inflate the menu; this adds items to the action bar if it is present.
46:         getMenuInflater().inflate(R.menu.abstraction, menu);
47:         return true;
48:     }
```

```
49:
50:     @Override
51:     public boolean onOptionsItemSelected(MenuItem item) {
52:         // Handle action bar item clicks here. The action bar will
53:         // automatically handle clicks on the Home/Up button, so long
54:         // as you specify a parent activity in AndroidManifest.xml.
55:         int id = item.getItemId();
56:         if (id == R.id.action_settings) {
57:             return true;
58:         }
59:         return super.onOptionsItemSelected(item);
60:     }
61:
62: }
```

```
1: package com.delta.abstraction;
2:
3: /**
4:  * Created by learnovate on 2/20/14.
5:  */
6: public abstract class Device {
7:
8:     protected String deviceName;
9:     protected boolean isOn;
10:
11:     public abstract void breakDevice();
12:
13:     public String getDeviceName() {
14:         return deviceName;
15:     }
16:
17:     public void setDeviceName(String deviceName) {
18:         this.deviceName = deviceName;
19:     }
20:
21:     public boolean isOn() {
22:         return isOn;
23:     }
24:
25:     public void setOn(boolean isOn) {
26:         this.isOn = isOn;
27:     }
28: }
```

```
1: package com.delta.abstraction;
2:
3: import android.util.Log;
4:
5: /**
6:  * Created by learnovate on 2/20/14.
7:  */
8: public class Fan extends Device implements Switchable{
9:
10:
11:     public Fan(){
12:         setDeviceName("FAN");
13:     }
14:
15:     public void breakDevice(){
16:         Log.e(getDeviceName(), "BANG, CRASH!.....oops.");
17:     }
18:
19:     public void turnOn(){
20:         Log.e(getDeviceName(), "whooooooooooooooooooooooooooooosh");
21:     }
22:
23:     public void turnOff(){
24:         Log.e(getDeviceName(), "darn, now its hot and stuffy in here");
25:     }
26: }
```

```
1: package com.delta.abstraction;
2:
3: import android.util.Log;
4:
5: /**
6:  * Created by learnovate on 2/20/14.
7:  */
8: public class Light extends Device implements Switchable{
9:
10:     public Light(){
11:         setDeviceName("LIGHT");
12:     }
13:     @Override
14:     public void breakDevice() {
15:         Log.e(getDeviceName(), "Glass Everywhere!.... i guess its not that bad");
16:     }
17:
18:     public void turnOn(){
19:         Log.e(getDeviceName(), "Now I can see. There are my keys!");
20:     }
21:
22:     public void turnOff(){
23:         Log.e(getDeviceName(), "Ouch, now it's so dark I stubbed my toe!");
24:     }
25: }
```

```
1: package com.delta.abstraction;
2:
3: import java.util.ArrayList;
4: import java.util.List;
5:
6: /**
7:  * Created by learnovate on 2/20/14.
8:  */
9: public class Switch {
10:
11:
12:     private List<Switchable> switchableItems = new ArrayList<Switchable>();
13:
14:
15:     public void wireUp(Switchable thing){
16:         switchableItems.add(thing);
17:     }
18:
19:     public void unWire(Switchable thing){
20:         switchableItems.remove(thing);
21:     }
22:
23:     public void flipSwitchUp(){
24:         for(Switchable thing:switchableItems){
25:             thing.turnOn();
26:         }
27:     }
28:
29:
30:     public void flipSwitchDown(){
31:         for(Switchable thing:switchableItems){
32:             thing.turnOff();
33:         }
34:     }
35: }
36:
37: }
```

```
1: package com.delta.abstraction;
2:
3: /**
4:  * Created by learnovate on 2/20/14.
5:  */
6: public interface Switchable {
7:
8:     public void turnOn();
9:     public void turnOff();
10:
11: }
```

1	abstraction/src/main/java/com/delta/abstraction/AbstractionActivity.java	2	pages	62	lines	14/09/18	15:58:39
2	abstraction/src/main/java/com/delta/abstraction/Device.java	1	pages	28	lines	14/09/18	15:58:39
3	abstraction/src/main/java/com/delta/abstraction/Fan.java	1	pages	26	lines	14/09/18	15:58:39
4	abstraction/src/main/java/com/delta/abstraction/Light.java	1	pages	25	lines	14/09/18	15:58:39
5	abstraction/src/main/java/com/delta/abstraction/Switch.java	1	pages	37	lines	14/09/18	15:58:39
6	abstraction/src/main/java/com/delta/abstraction/Switchable.java	1	pages	11	lines	14/09/18	15:58:39