## 桥接（Bridge）

### Intent

将抽象与实现分离开来，使它们可以独立变化。

### Class Diagram

* Abstraction：定义抽象类的接口
* Implementor：定义实现类接口

### Implementation

RemoteControl 表示遥控器，指代 Abstraction。

TV 表示电视，指代 Implementor。

桥接模式将遥控器和电视分离开来，从而可以独立改变遥控器或者电视的实现。

public abstract class TV {
 public abstract void on();

 public abstract void off();

 public abstract void tuneChannel();
}

public class Sony extends TV {
 @Override
 public void on() {
 System.out.println("Sony.on()");
 }

 @Override
 public void off() {
 System.out.println("Sony.off()");
 }

 @Override
 public void tuneChannel() {
 System.out.println("Sony.tuneChannel()");
 }
}

public class RCA extends TV {
 @Override
 public void on() {
 System.out.println("RCA.on()");
 }

 @Override
 public void off() {
 System.out.println("RCA.off()");
 }

 @Override
 public void tuneChannel() {
 System.out.println("RCA.tuneChannel()");
 }
}

public abstract class RemoteControl {
 protected TV tv;

 public RemoteControl(TV tv) {
 this.tv = tv;
 }

 public abstract void on();

 public abstract void off();

 public abstract void tuneChannel();
}

public class ConcreteRemoteControl1 extends RemoteControl {
 public ConcreteRemoteControl1(TV tv) {
 super(tv);
 }

 @Override
 public void on() {
 System.out.println("ConcreteRemoteControl1.on()");
 tv.on();
 }

 @Override
 public void off() {
 System.out.println("ConcreteRemoteControl1.off()");
 tv.off();
 }

 @Override
 public void tuneChannel() {
 System.out.println("ConcreteRemoteControl1.tuneChannel()");
 tv.tuneChannel();
 }
}

public class ConcreteRemoteControl2 extends RemoteControl {
 public ConcreteRemoteControl2(TV tv) {
 super(tv);
 }

 @Override
 public void on() {
 System.out.println("ConcreteRemoteControl2.on()");
 tv.on();
 }

 @Override
 public void off() {
 System.out.println("ConcreteRemoteControl2.off()");
 tv.off();
 }

 @Override
 public void tuneChannel() {
 System.out.println("ConcreteRemoteControl2.tuneChannel()");
 tv.tuneChannel();
 }
}

public class Client {
 public static void main(String[] args) {
 RemoteControl remoteControl1 = new ConcreteRemoteControl1(new RCA());
 remoteControl1.on();
 remoteControl1.off();
 remoteControl1.tuneChannel();
 RemoteControl remoteControl2 = new ConcreteRemoteControl2(new Sony());
 remoteControl2.on();
 remoteControl2.off();
 remoteControl2.tuneChannel();
 }
}

### JDK

* AWT (It provides an abstraction layer which maps onto the native OS the windowing support.)
* JDBC