## 责任链（Chain Of Responsibility）

### Intent

使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链发送该请求，直到有一个对象处理它为止。

### Class Diagram

* Handler：定义处理请求的接口，并且实现后继链（successor）

### Implementation

public abstract class Handler {  
  
 protected Handler successor;  
  
  
 public Handler(Handler successor) {  
 this.successor = successor;  
 }  
  
  
 protected abstract void handleRequest(Request request);  
}

public class ConcreteHandler1 extends Handler {  
  
 public ConcreteHandler1(Handler successor) {  
 super(successor);  
 }  
  
  
 @Override  
 protected void handleRequest(Request request) {  
 if (request.getType() == RequestType.TYPE1) {  
 System.out.println(request.getName() + " is handle by ConcreteHandler1");  
 return;  
 }  
 if (successor != null) {  
 successor.handleRequest(request);  
 }  
 }  
}

public class ConcreteHandler2 extends Handler {  
  
 public ConcreteHandler2(Handler successor) {  
 super(successor);  
 }  
  
  
 @Override  
 protected void handleRequest(Request request) {  
 if (request.getType() == RequestType.TYPE2) {  
 System.out.println(request.getName() + " is handle by ConcreteHandler2");  
 return;  
 }  
 if (successor != null) {  
 successor.handleRequest(request);  
 }  
 }  
}

public class Request {  
  
 private RequestType type;  
 private String name;  
  
  
 public Request(RequestType type, String name) {  
 this.type = type;  
 this.name = name;  
 }  
  
  
 public RequestType getType() {  
 return type;  
 }  
  
  
 public String getName() {  
 return name;  
 }  
}

public enum RequestType {  
 TYPE1, TYPE2  
}

public class Client {  
  
 public static void main(String[] args) {  
  
 Handler handler1 = new ConcreteHandler1(null);  
 Handler handler2 = new ConcreteHandler2(handler1);  
  
 Request request1 = new Request(RequestType.TYPE1, "request1");  
 handler2.handleRequest(request1);  
  
 Request request2 = new Request(RequestType.TYPE2, "request2");  
 handler2.handleRequest(request2);  
 }  
}

request1 is handle by ConcreteHandler1  
request2 is handle by ConcreteHandler2

### JDK

* [java.util.logging.Logger#log()](http://docs.oracle.com/javase/8/docs/api/java/util/logging/Logger.html#log%28java.util.logging.Level,%20java.lang.String%29)
* [Apache Commons Chain](https://commons.apache.org/proper/commons-chain/index.html)
* [javax.servlet.Filter#doFilter()](http://docs.oracle.com/javaee/7/api/javax/servlet/Filter.html#doFilter-javax.servlet.ServletRequest-javax.servlet.ServletResponse-javax.servlet.FilterChain-)