

# NSY102 Session de juin 2015-durée : 2 heures

## Une idée...

```
public class ContentBasedRouterImpl implements ContentBasedRouterI{
    private String name;
    private Map<ReceiverI,FilterI> receivers;
    private ExceptionListener listener = null;

    public ContentBasedRouterImpl(String name){
        this.name = name;
        this.receivers = new HashMap<ReceiverI, FilterI>();
    }

    public void installReceiver(ReceiverI receiver){
        receivers.put(receiver, FilterI.ALL_MESSAGES);
        receiver.addRouter(this);
    }

    public void setFilter(ReceiverI receiver, FilterI filter) throws ReceiverNotInstalledException{
        if(receivers.get(receiver) == null) throw new ReceiverNotInstalledException(receiver.toString());
        receivers.put(receiver, filter);
    }

    public int sendMessage(MessageI msg){
        int number = 0;
        Iterator<ReceiverI> it = receivers.keySet().iterator();
        while(it.hasNext()){
            final ReceiverI receiver = it.next();
            FilterI filter = receivers.get(receiver);
            try{
                if(filter.accept(msg)){
                    receiver.receive(msg, this);
                    number++;
                }
            }catch(final Exception e){
                if(listener!=null){
                    new Thread(
                        new Runnable(){
                            public void run(){
                                synchronized(listener){
                                    listener.onException(ContentBasedRouterImpl.this,receiver,e);
                                }
                            }
                        }
                    ).start();
                }
            }
        }
        return number;
    }

    public void addExceptionListener(ExceptionListener el){
        this.listener = el;
    }

    public Set<ReceiverI> getReceivers(){
        return new HashSet<ReceiverI>(receivers.keySet());
    }

    public String getName(){return this.name;}
    public String toString(){return getName();}
}

public class Receiver implements ReceiverI{
    private String name;
    private MessageI received;
    private Set<ContentBasedRouterI> routers;

    public Receiver(String name){
        this.name = name;
        this.routers = new HashSet<ContentBasedRouterI>();
    }

    public Receiver(String name, ContentBasedRouterI router){
        this(name);
    }
}
```

```

    this.routers.add(router);
}

public void addRouter(ContentBasedRouterI router){
    this.routers.add(router);
}

public void receive(MessageI msg, ContentBasedRouterI router) throws Exception{
    this.received = msg;
    System.out.println(name + " received : " + msg + " from : " + router);
}

public Set<ContentBasedRouterI> getRouters(){
    return new HashSet<ContentBasedRouterI>(routers);
}

public String getName(){
    return this.name;
}

public String toString(){
    return getName();
}
}

import javax.management.NotificationBroadcasterSupport;
import javax.management.Notification;

public final class ContentBasedRouter extends NotificationBroadcasterSupport implements ContentBasedRouterMBean{
    private ContentBasedRouterI router;

    public ContentBasedRouter(ContentBasedRouterI router){
        this.router = router;
    }

    public void installReceiver(ReceiverI receiver){
        router.installReceiver(receiver);
    }

    public void setFilter(ReceiverI receiver, FilterI filter) throws ReceiverNotInstalledException{
        router.setFilter(receiver, filter);
    }

    public int sendMessage(MessageI msg){
        return router.sendMessage(msg);
    }

    public void addExceptionHandler(ExceptionListener nl){
        router.addExceptionHandler(new OnException(nl));
    }

    private static long sequenceNumber = 0L;

    private class OnException
        implements ContentBasedRouterI.ExceptionListener{
        private ContentBasedRouterI.ExceptionListener nl;
        public OnException(ContentBasedRouterI.ExceptionListener nl){
            this.nl = nl;
        }
        public void onException(ContentBasedRouterI router, ReceiverI r, Exception e){
            String msg = e.getMessage() + " levée par : " + r + " depuis : " + router;
            try{
                nl.onException(router, r, e);
            }finally{
                sendNotification(new Notification("onException",ContentBasedRouter.this,sequenceNumber++,msg));
            }
        }
    }

    public Set<ReceiverI> getReceivers(){
        return router.getReceivers();
    }

    public String getName(){
        return router.getName();
    }
}

```

}