

Activitat 2.5. CORBA Model d'herència i model de delegació

A partir de l'exemple vist a l'activitat 2.4 corresponent a la implementació RMI/IIOP amb POA del servidor de xat, explica breument:

- Els dos models de desenvolupament que trobem a CORBA: model d'herència i el de delegació.

Model d'herència. Un cop definit l'IDL i generat els stubs y skeletons, s'ha de realitzar la implementació amb una classe que hereti de l'skeleton. Aquest model inclou l'ús de POA

- POA. D'una interfície `Abc` definida a `Abc.idl` es genera `AbcPOA.java`, la implementació haurà d'heretar de `AbcPOA` (skeleton que hereda de `org.omg.PortableServer.Servant`).
- `ImplBase`. Donat una interfície `Abc` definida a `Abc.idl` es genera `_AbcImplBase.java`. La implementació ha d'heretar de `_AbcImplBase`. Aquest model es obsolet, només esta suportat per compatibilitat.

Model de delegació (o Tie model). En aquest model la implementació no ha d'heredar obligatoriament de cap classe, d'aquesta forma ens queda lliure la possibilitat d'heretar la nostra implementació d'una altre classe que ens convingui (donat que a Java no existeix la herència múltiple). S'utilitzaran dos classes:

- La classe `Tie` generada automàticament des del IDL. La seva funció es delegar totes les crides a la classe que conte la implementació.
- Classe amb la implementació de les operacions definides a la interfície del IDL.

- Que és una interface TIE i per a que la fem servir? Explica-ho a partir de l'exemple de XAT.

La classe `Tie` es utilitzada per processar les crides entrants i enviar-les a la classe que conté la implementació real.

En el cas del servidor de Xat, tenim la implementació real del servidor al objecte `server`:

```
ChatServer server = new ChatServer();
```

Y creem una classe `Tie` lligada a aquest objecte:

```
_ChatServer_Tie tie = (_ChatServer_Tie)Util.getTie( server );
```

Al POA registrem la classe `Tie`, així rebrà les peticions remotes que després les redirigirà cap a l'objecte `server` fent com una mena de pont.

```
String nomId = "Xat";  
byte[] id = nomId.getBytes();  
tPOA.activate_object_with_id( id, tie );
```

- Es podria fer la mateixa aplicació usant únicament el model d'herència?

No es pot utilitzar el model d'herència quan usem RMI sobre IIOP com en aquesta aplicació, perquè necessitem que els objectes remots heretin de `PortableRemoteObject`:

```
public class ChatServer extends PortableRemoteObject implements IChatServer {...}
```

No podem fer ús del model d'herència ja que apart d'heredar de `PortableRemoteObject` hauríem d'heredar del skeleton generat des del IDL, i Java no suporta herència múltiple.

Alumne: Sergio Blanco Cuaresma.