

Primer pattern

"Diseñar una máquina de reconocer monedas"



ES DE 20, 10, 5, 1 ... ?

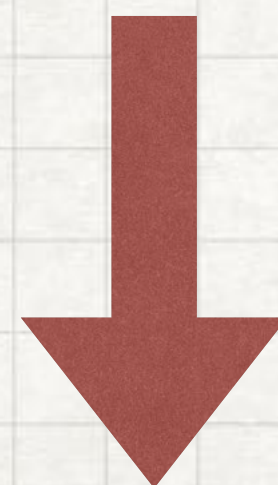
Valor XX



aceptada



rechazada



ES DE 20, 10, 5, 1 ... ?

- Estructura monolítica
- Difícil reusar partes
- Complicado mantenimiento:
 - ¿Añadir una moneda?
 - ¿Quitar una moneda?



¿y si diseñamos que reconozca UNA moneda?



ES DE X ?

valor X



aceptada



rechazada





ES DE 10?

valor 5
→
aceptada
→



rechazada

ES DE 5?

valor 5
→
aceptada
→



rechazada

• • •



rechazada

- Sencillo
- Mantenable
- Reusable

¿Cual es la esencia de esta solución?

¿Es aplicable a otros problemas?



ES DE 5?

valor 5

aceptada



¿Cual es la esencia de esta solución?

TAREA



PUEDO
REALIZARLA?

resultado



delega

PUEDO
REALIZARLA?

resultado



...

¿Es aplicable a otros problemas?

- Una incidencia llega a un empleado, ¿la puede tratar él o delega en un superior?
- Un paciente llega a un centro de atención primaria. ¿Lo tratan allí o lo llevan al hospital?
- Un servicio web SOAP permite registrar "handlers". Cada uno de ellos puede indicar si es capaz de procesar un mensaje concreto o no
- JDBC debe conectarse a una base de datos y tiene una lista de *drivers* candidatos

Chain of responsibility

“

Intent

Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it.

— *GoF*

”

Creational	Structural	Behavioral
<ul style="list-style-type: none">• Abstract factory pattern• Builder pattern• Factory method pattern• Prototype pattern• Singleton pattern	<ul style="list-style-type: none">• Adapter• Bridge• Composite• Decorator• Facade• Flyweight• Proxy	<ul style="list-style-type: none">• Chain of responsibility• Command• Interpreter• Iterator• Mediator• Memento• Observer• State• Strategy• Template method• Visitor