



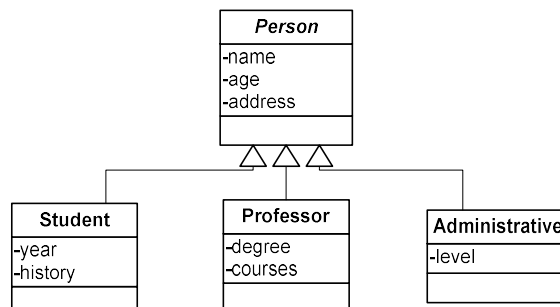
La importancia de la Experiencia de Diseño

- ✓ Diseñar Software es difícil (aun usando buenas técnicas)
- ✓ La experiencia es insustituible
- ✓ Como la capturamos, transmitimos y usamos?



Ejercicio

- ✓ Supongamos una aplicación universitaria para gestionar los alumnos, docentes y administrativos?
- ✓ Solucion ingenua:





Problemas

- ✓ Puede una persona ser estudiante y administrativo al mismo tiempo?
- ✓ Puede una persona “evolucionar” en el tiempo; pasar de estudiante a profesor?
- ✓ Puede la misma persona ser estudiante “2 veces”? En dos carreras diferentes?



Design Patterns

- ✓ Originados en la Arquitectura (Alexander)
- ✓ Diferentes tipos de patrones: de diseño, programación?, arquitectura de software, testeo, interfaz gráfica, Web



Patterns-Definiciones

✓ Según Alexander

“A pattern describes a problem that occurs once and again in a context, and describe the core of the solution in such a way that it can be used millions of times without doing the same twice”



Patterns-Definiciones

- ✓ Un patron es un par problema-solucion
- ✓ Los patrones tratan con problemas recurrentes y buenas soluciones a esos problemas
- ✓ Los patrones deben incluir una regla:
Cuando aplico el patron?



Patrones de Diseño

Definicion (GOF):

A design pattern names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design. The design pattern identifies the participating classes and instances, their roles and collaborations, and the distribution of responsibilities. Each design pattern focuses on a particular object-oriented design problem or issue. It describes when it applies, whether it can be applied in view of other design constraints, and the consequences and trade-offs of its use.

Gamma, Helms, Johnson, Vlissides: Design Patterns. Elements of Reusable Object-Oriented Software.



Clasificacion de Patrones

✓ De acuerdo a la actividad

- ✓ Patrones de Soft
- ✓ Patrones de Testing
- ✓ Patrones de Proceso
- ✓ Patrones pedagogicos
- ✓
- ✓



Clasificacion de Patrones

- ✓ De acuerdo al nivel de abstraccion
 - ✓ Patrones de Analisis
 - ✓ Patrones de Arquitectura
 - ✓ Patrones de Diseño
 - ✓ Idioms



Clasificacion..

- ✓ De acuerdo al dominio de aplicacion:
 - ✓ Financiero
 - ✓ Comercio electronico
 - ✓ Salud
 - ✓ Tiempo Real
 - ✓ ..
 - ✓ ..



Clasificacion

- ✓ **De acuerdo al ambiente/paradigma:**
 - ✓ Patrones Web
 - ✓ Patrones de Modelos de datos
 - ✓ XML
 - ✓ Interaccion



Clasificacion de Patrones de Diseño

- ✓ **De acuerdo al proposito:**
 - ✓ Creationales
 - ✓ Estructurales
 - ✓ De Comportamiento



Descripción de un patrón

- ✓ Según GoF:
 - ✓ *Name*
 - ✓ *Intent*
 - ✓ *Motivation (Problem)*
 - ✓ *Aplicability*
 - ✓ *Structure*
 - ✓ *Participants*
 - ✓ *Collaborations*



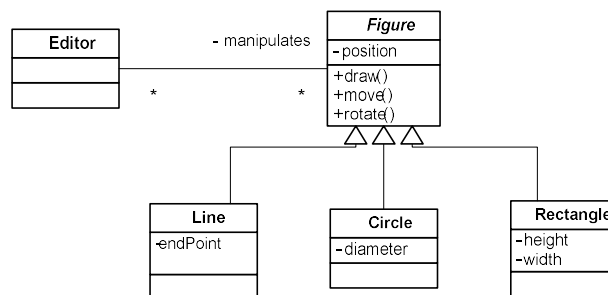
Descripción....

- ✓ *Consequences*
- ✓ *Implementación*
- ✓ *Code*
- ✓ *Known Uses*
- ✓ *Related Patterns*



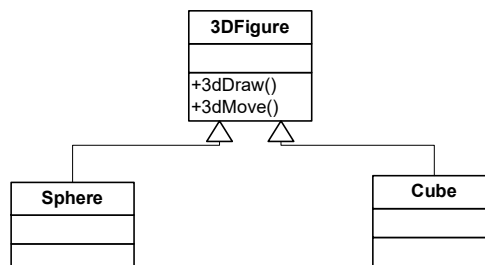
Ejemplo

- ✓ Editor grafico para manejar figuras geometricas:
- ✓ Observen polimorfismo en la relacion Editor-Figure



Ejemplo..

- ✓ Queremos extender el editor a figuras 3D





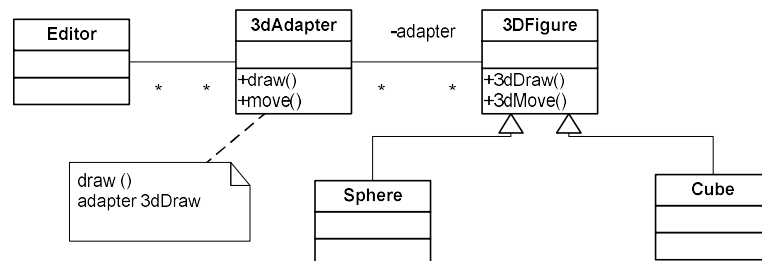
Problema

- ✓ Como integramos esta jerarquia?
- ✓ Que problemas tenemos?
- ✓ Sacrificamos polimorfismo?
- ✓ Editamos el codigo de la nueva jerarquia?



Solution

- ✓ Cuando tratamos con interfases incompatibles, intentar adaptarlas.



3dAdapter is sub-clase de?



Adapter

✓ Intencion:

“Convertir” la interfaz de una clase en otra que el cliente espera EL Adapter permite que ciertas clases trabajen en conjunto cuando no podrian por tener interfaces incompatibles

✓ Aplicabilidad:

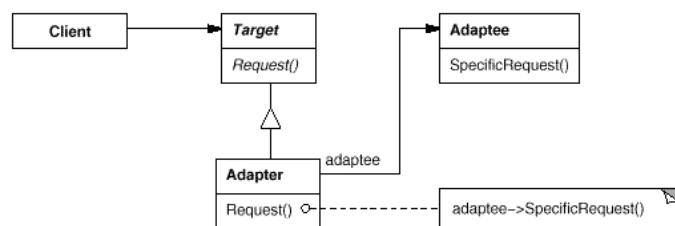
Use el adapter cuando:

- ✓ Ud quiere usar una clase existente y su interfaz no es compatible con lo que precisa



Adapter

✓ Estructura





Adapter. Participants

✓ Participantes:

✓ **Target** (Figure)

- ✓ defines the domain-specific interface that Client uses.

✓ **Client** (Editor)

- ✓ collaborates with objects conforming to the Target interface.

✓ **Adaptee** (3DFigure)

- ✓ defines an existing interface that needs adapting.

✓ **Adapter** (3DAdapter)

- ✓ adapts the interface of Adaptee to the Target interface.



Descubriendo Patrones

- ✓ Como es el proceso de descubrimiento?
- ✓ Que tipo de observacion/abstraccion realizamos?

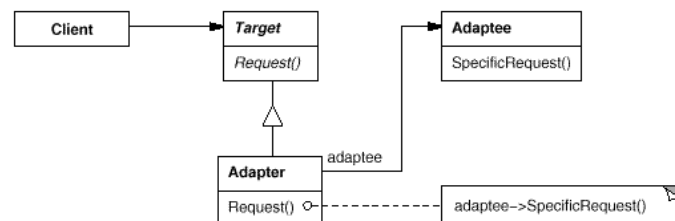


Usando Patrones

- ✓ Supongamos que conocemos patrones(e.g. Adapter).
- ✓ Como mapeamos un patron a un diseño especifico?
- ✓ Como aplicamos el principio de Alexander (“use the patterns millions of times without doing the same thing twice”)?



Adapter



Como usamos esta informacion? Es suficiente?
Que mas necesitamos?