Orientacion a Objetos 2

Dra Alejandra Garrido

Dr. Alejandro Fernandez

Dr. Gustavo Rossi

[garrido, alejandro.Fernandez, gustavo]@lifia.info.unlp.edu.ar

Laboratorio de Investigación y Formación en Informática Avanzada
Facultad de Informática, Universidad Nacional de La Plata
Calle 50 esq. 115 - Primer piso (1900) La Plata, Argentina
TE: (0221) 422 8252 http://www-lifia.info.unlp.edu.ar

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.

Descripcion de un patron

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

Descripcion….

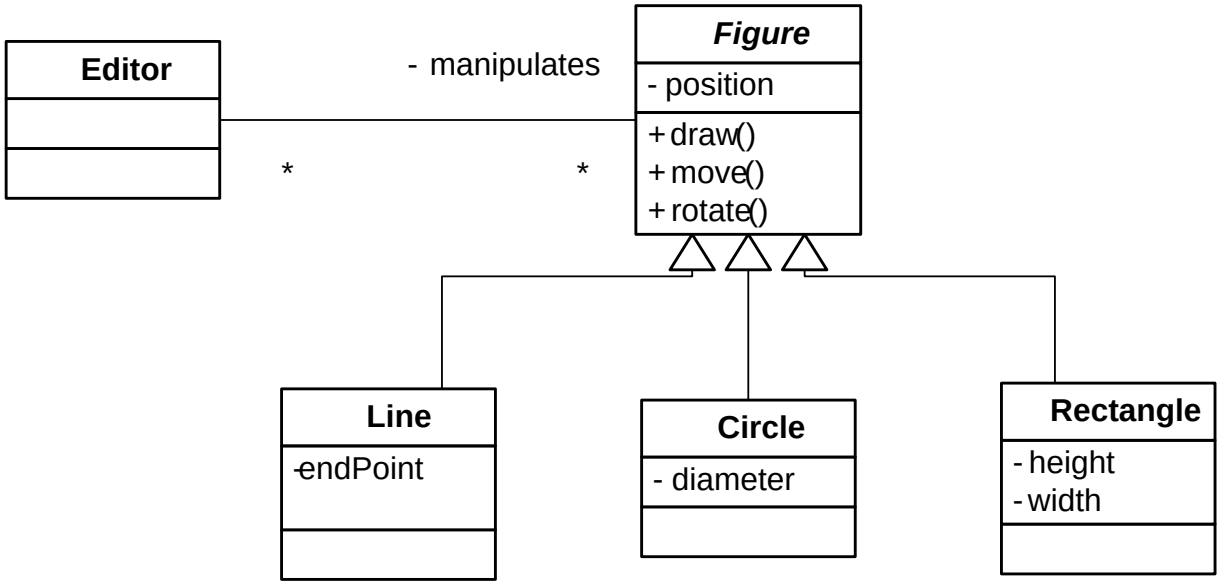
- ✓ *Consequences*
- ✓ *Implementatión*
- ✓ *Code*
- ✓ *Known Uses*
- ✓ *Related Patterns*

Como seguimos?

- ✓ Vamos a introducir varios Patrones
- ✓ Para cada uno de ellos, comenzamos con un ejemplo motivador y luego lo describimos con detalle

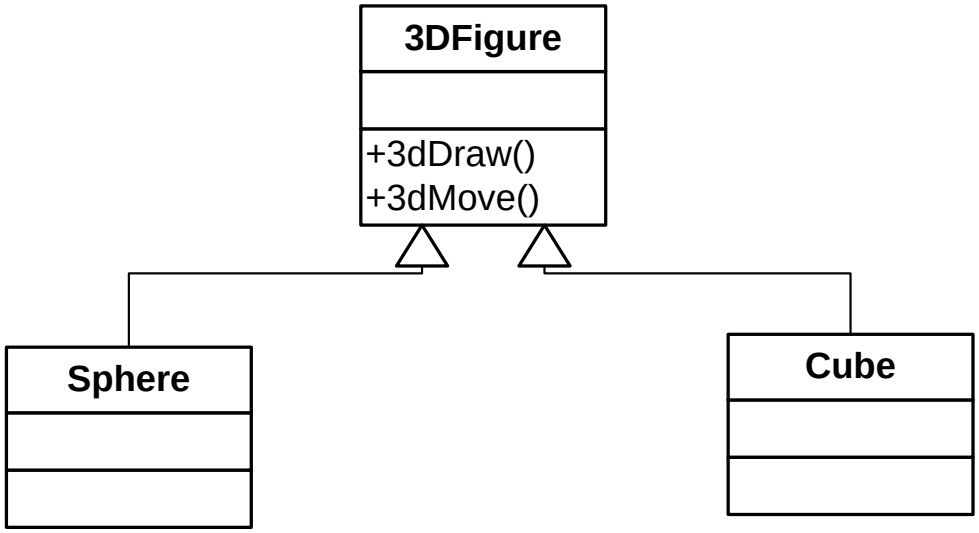
Ejemplo

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



Ejemplo..

- ✓ Queremos extender el editor a figuras 3D

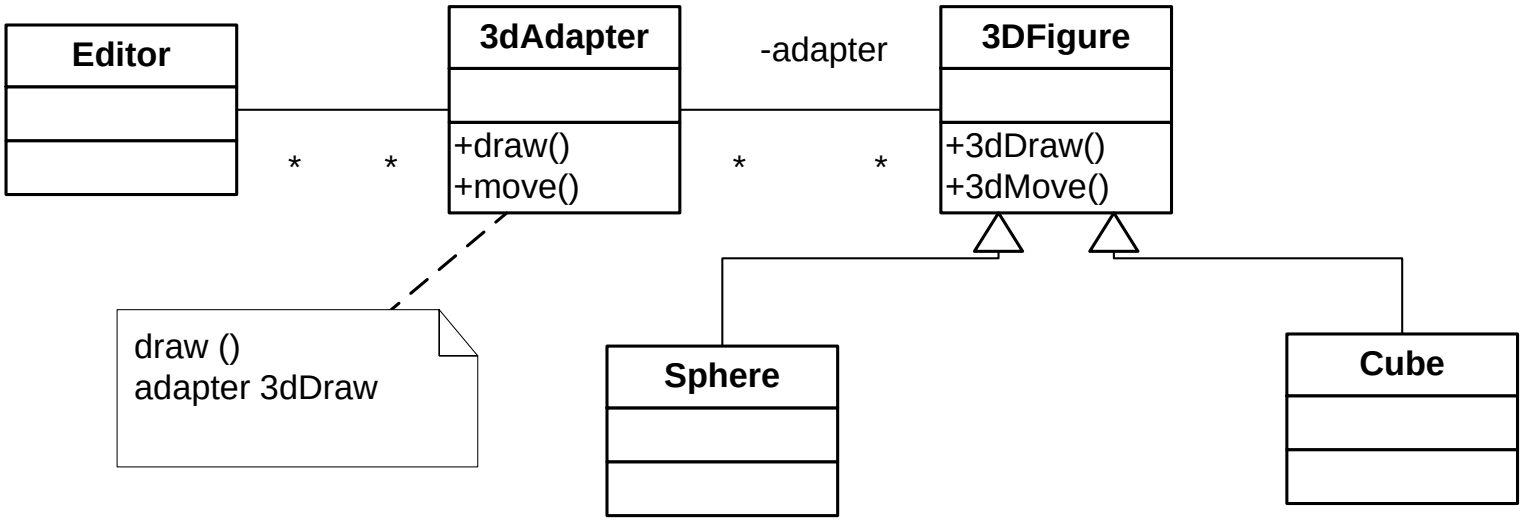


Problema

- ✓ 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

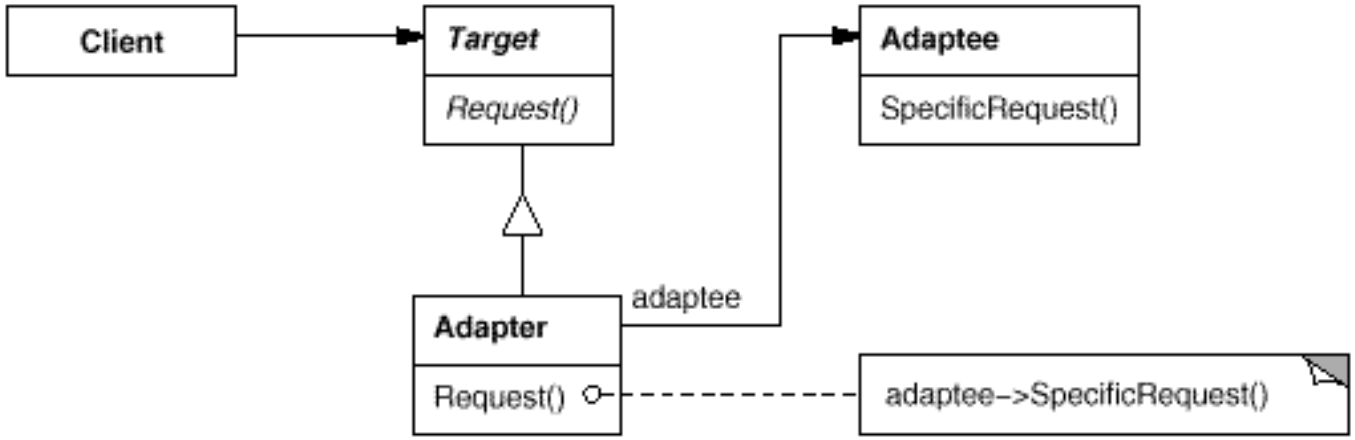
Applicabilidad:

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.

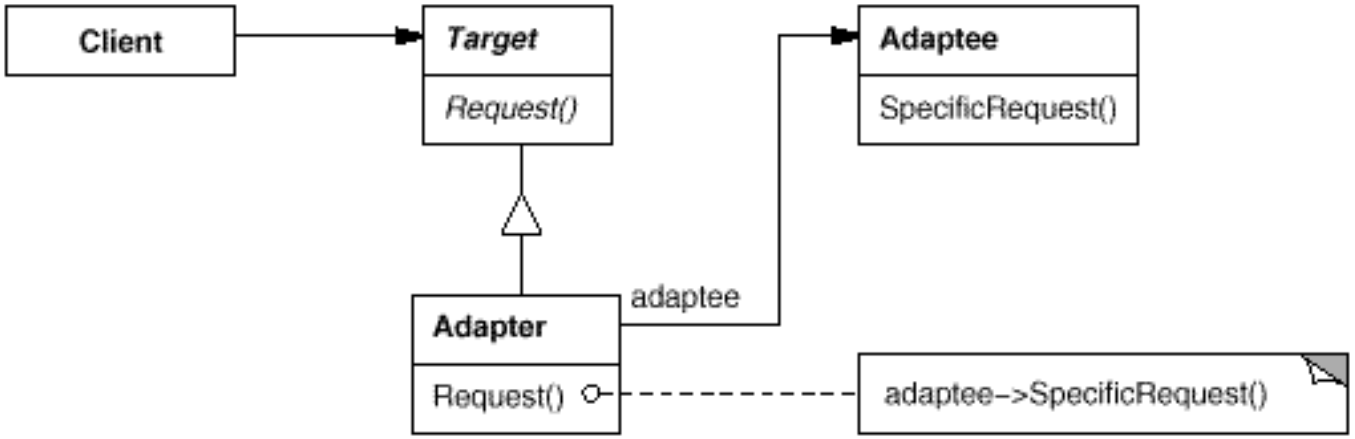
Descubriendo Patrones

- ✓ 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?