Adapter Design Pattern

Structural Design Patterns

Structural patterns are concerned with how classes and objects are composed to form larger structures.

⏩ Adapter	structural *class* pattern
Composite	structural *object* pattern
Decorator	structural *object* pattern

Adapter Pattern

Intent

Convert the interface of a class into another interface that clients expect.

Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

Also Known As

Wrapper

Adapter Pattern

Motivation

Sometimes a toolkit class that is designed for reuse is not reusable only because its interface *does not* match the domain-specific interface an application requires.

Adapter Pattern

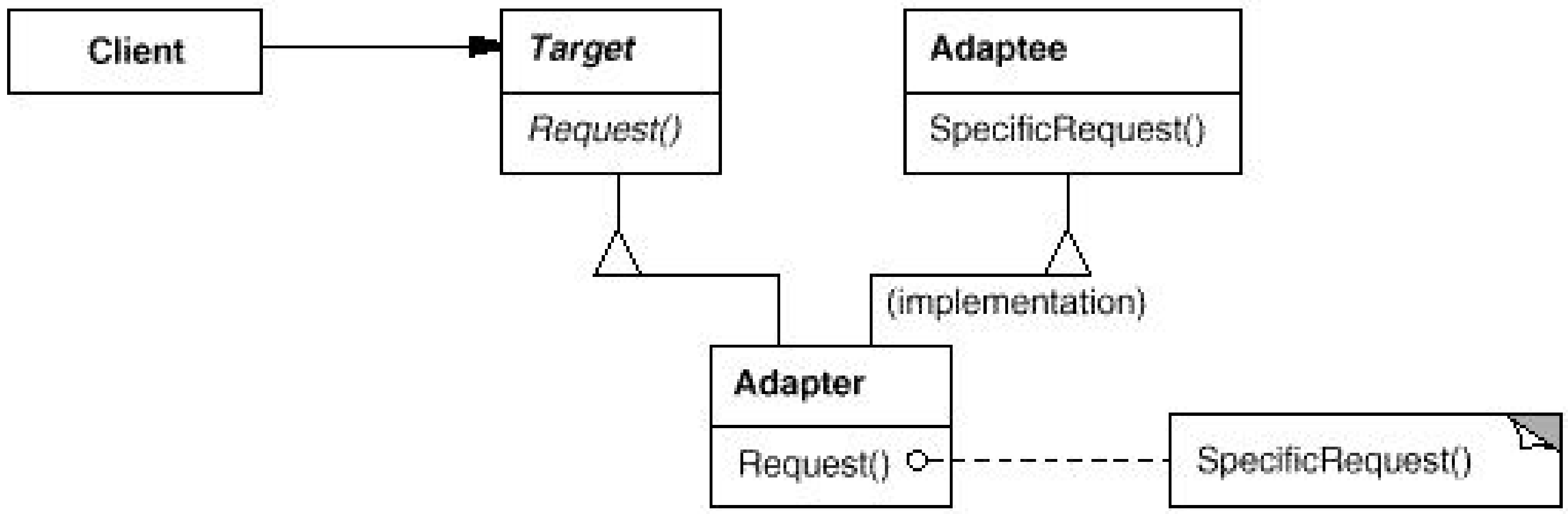
Applicability

Use the Adapter pattern when

> you want to use an existing class, and its interface does not match the one you need.
>
> (object adapter only)
> you need to use several existing subclasses. An object adapter can adapt the interface of its parent class.

Adapter Pattern

Structure



Adapter Pattern

Participants

Target (DescriptorTable)

defines the domain-specific interface that Client uses.

Client (FileSystem)

collaborates with objects conforming to the Target interface.

Adaptee (ArrayList)

defines an existing interface that needs adapting.

Adapter (FileDescriptorTable)

adapts the interface of Adaptee to the Target interface.

Adapter Pattern

Collaborations

Clients call operations on an Adapter instance. In turn, the adapter calls Adaptee operations that carry out the request.

Adapter Pattern

Implementation

Sample Code

Take a look at the simple emulation of UNIX file descriptor table interface, in the *src* directory.