DESIGN PATTERNS

ADAPTER PATTERN

By
YASHWANTH G

DESIGN PATTERNS

A design patterns are well-proved solution for solving the specific problem/task.

1) Creational Design Patterns.
2) Structural Design Patterns.
3) Behavioral Design Patterns.

STRUCTURAL DESIGN PATTERNS

Structural design patterns are concerned with how classes and objects can be composed, to form larger structures.

The structural design patterns simplifies the structure by identifying the relationships.

These patterns focus on, how the classes inherit from each other and how they are composed from other classes.

ADAPTER PATTERNS

An Adapter Pattern says that just "converts the interface of a class into another interface that a client wants".

The Adapter Pattern is also known as “**Wrapper**”.

ADAPTER PATTERNS

To use an adapter:

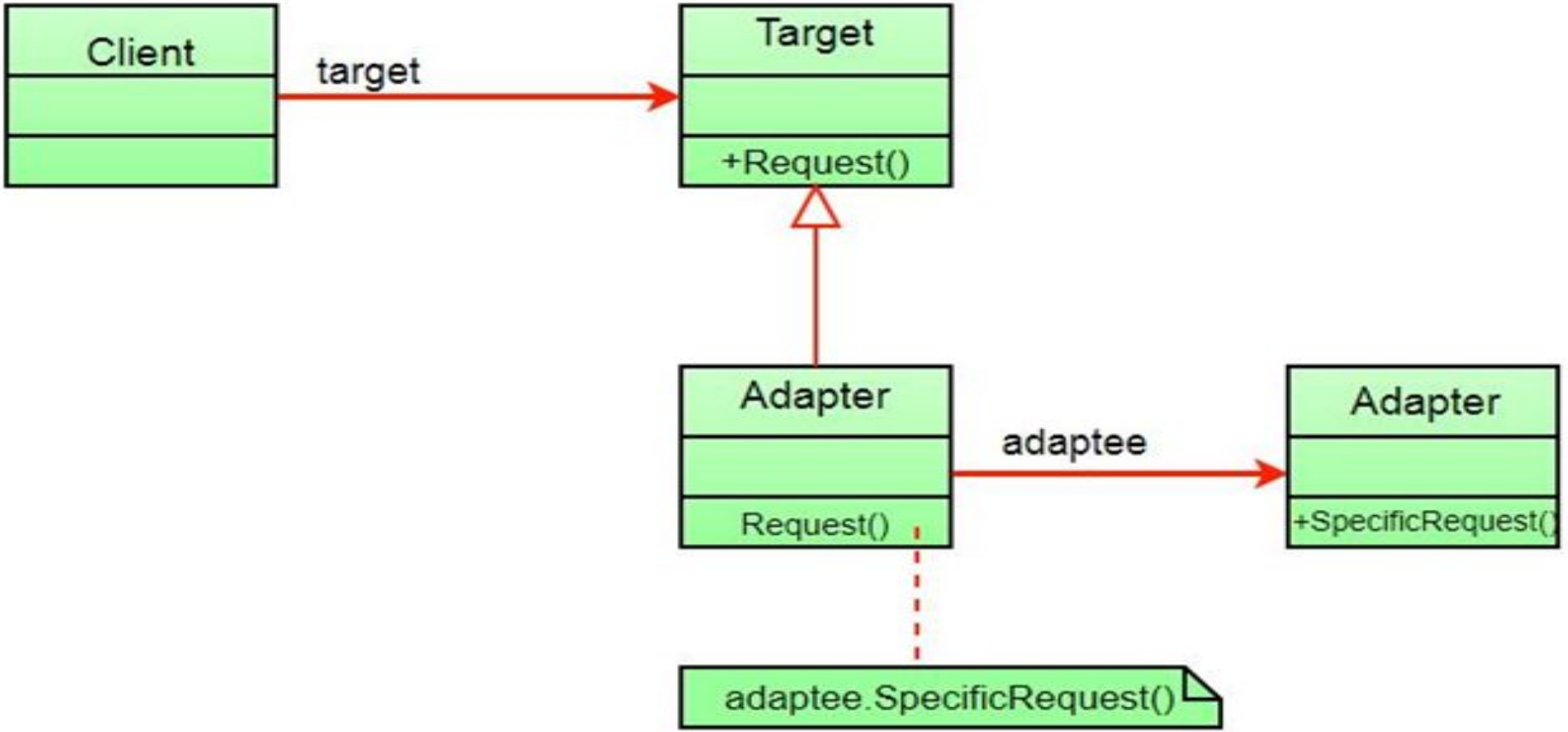
1. The client makes a request to the adapter by calling a method on it using the target interface.
2. The adapter translates that request on the adaptee using the adaptee interface.
3. Client receive the results of the call and is unaware of adapter's presence.

ADVANTAGES

1) It allows two or more previously incompatible objects to interact.

2) It allows reusability of existing functionality.

CLASS DIAGRAM



EXAMPLE

Suppose you have a Bird class with fly(), and makeSound() methods. And also a ToyDuck class with squeak() method. Let's assume that you are short on ToyDuck objects and you would like to use Bird objects in their place. Birds have some similar functionality but implement a different interface, so we can't use them directly. So we will use adapter pattern. Here our client would be ToyDuck and adaptee would be Bird.