

# Design Pattern Adaptator

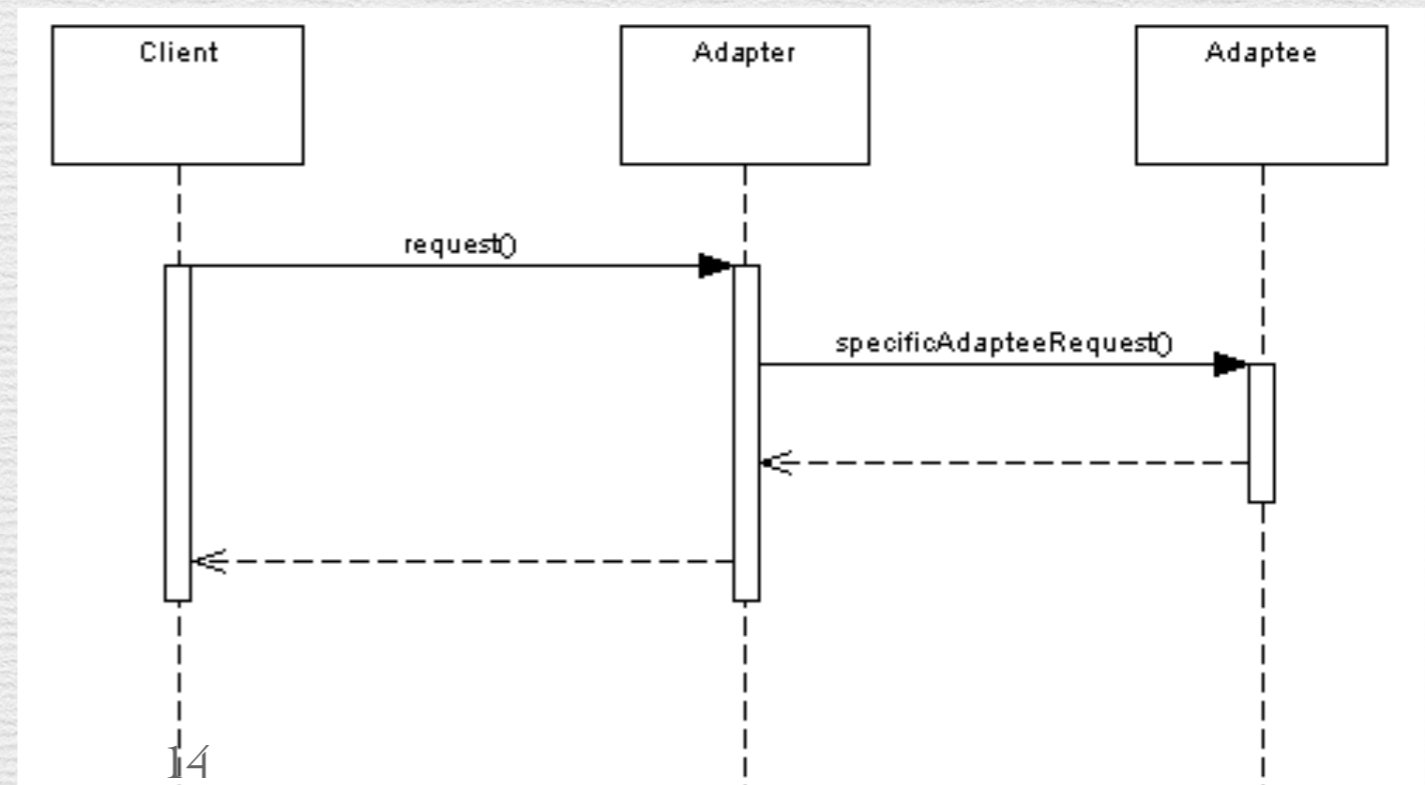
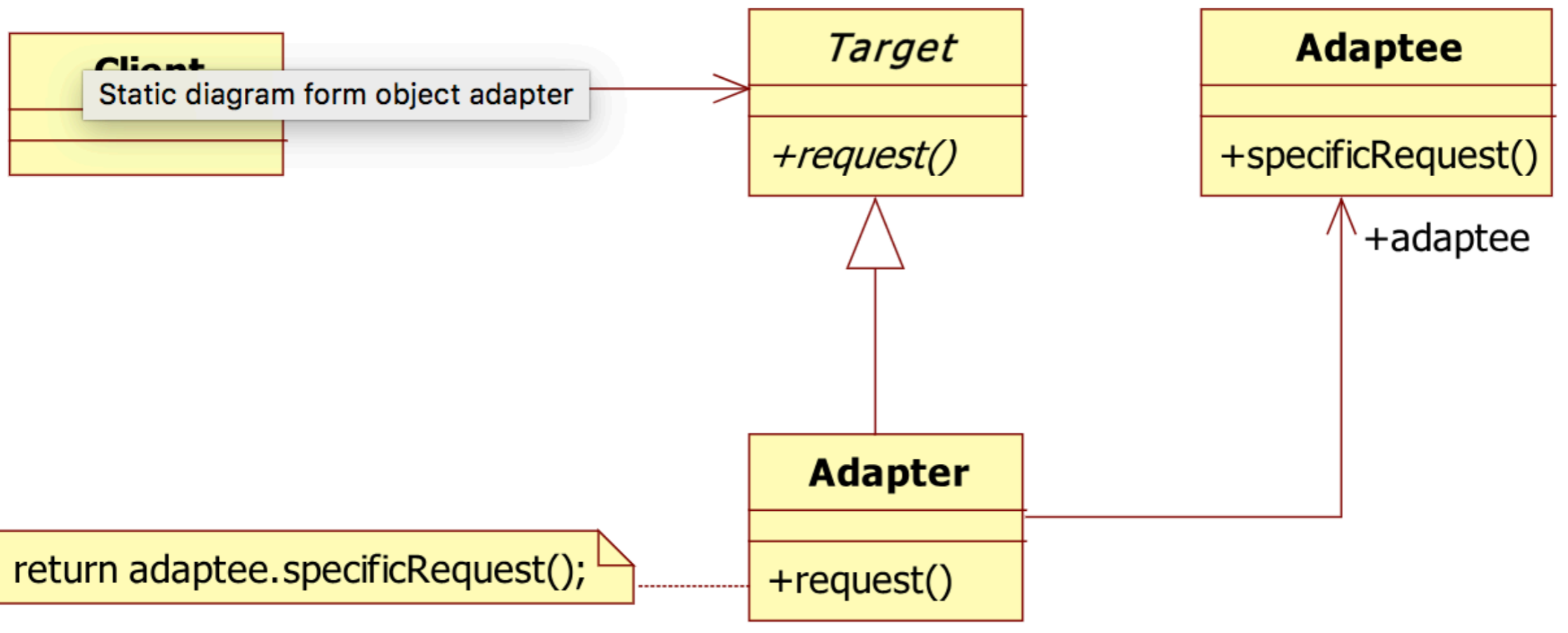
# Patron Adapter : le problème

- ❧ “ Permettre l’adaptation d’une classe à une autre interface qui est attendue par le client : **autoriser ainsi des classes ayant des interfaces incompatibles à collaborer.** ”
- ❧ Synonyme : wrapper

## motivation

Parfois, une bibliothèque de code ne peut pas être utilisée car son interface est incompatible avec l'interface requise par une application. Nous ne pouvons pas changer l'interface de la bibliothèque, puisque nous ne pouvons pas accéder ou modifier le code source .... De même il ne serait pas raisonnable d'avoir autant de versions d'une bibliothèque que de ses usages.

# Patron Adapter : solution



- On a défini un jeu dans lequel des avatars peuvent avancer, être affecté à un joueur, ou se battre.

```
public interface Avatar {  
    //Renvoie un nombre de Points  
    int attaquer();  
    void avancer();  
    void affecterJoueur();  
}
```

- On voudrait pouvoir utiliser des objets présents dans un autre jeu.

```
public class Tank {  
    //Force du tir  
    public int tirer() {... }  
  
    public void rouler(int vitesse, Direction d) {...  
    }  
}
```

```
public class Robot {  
    private String maitre;  
    public void frapper() {...}  
    public void marcher() {...}  
    public void reagirAHumain(String maitre) {  
        this.maitre = maitre; ..  
    }  
}
```