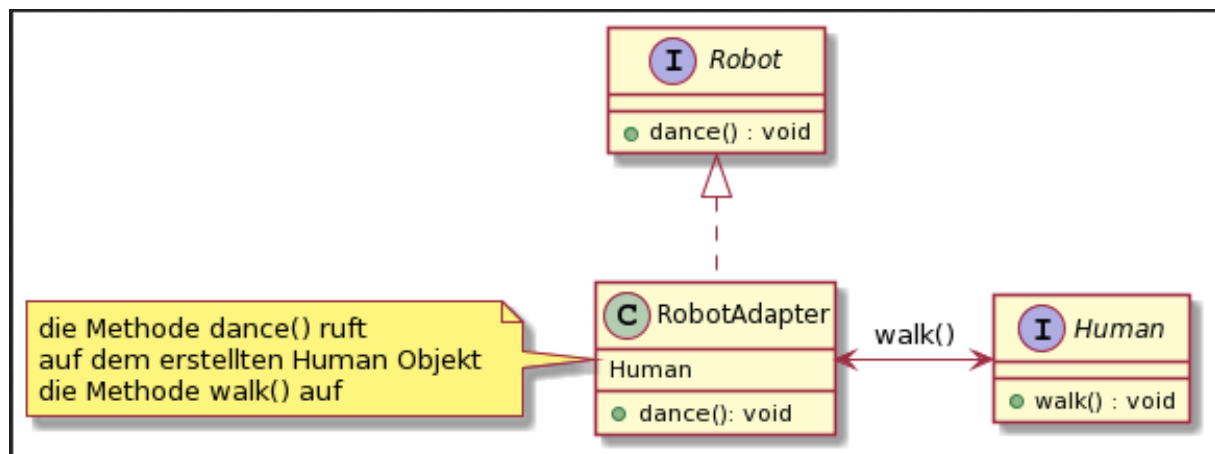


Adapter Pattern

Zweck:

Das Adapter Pattern ist dazu da um, zwei eigentlich inkompatible Klassen, kompatibel zu machen, ohne die Klassen selbst zu bearbeiten. Der Adapter konvertiert dabei ein Interface zu dem Ziel Interface.

UML:



Beschreibung:

Adapter Klassen kommen ins Spiel, wenn das Interface der Client Klasse nicht kompatibel zum Interface der Target Klasse ist. Um die Interfaces oder Klassen nicht direkt bearbeiten zu müssen werden Adapter Klassen genutzt. In dieser Adapter Klasse wird das Interface des Clients implementiert aber ein Objekt des inkompatiblen Targets erstellt. Durch den Adapter kann dann die Client Klasse die ihr bekannten Methoden aufrufen, diese werden aber im Adapter entsprechend geschachtelt, um die Methoden des nicht kompatiblen Targets aufzurufen.

Beispiel in Java:

```
public interface Robot {
    public void dance();
}
```

```
public class Droid implements Robot{

    @Override
    public void dance() {
        System.out.println("Up, Down, Up, Down..");
    }

}
```

```
public interface Human {  
    public void walk();  
    public void talk();  
}
```

```
public class Man implements Human{  
  
    @Override  
    public void walk() {  
        System.out.println("Left, Right, Left, Right..");  
    }  
  
    @Override  
    public void talk() {  
        System.out.println("well hello there");  
    }  
}
```

```
public class HumanAdapter implements Robot{  
  
    Human human;  
  
    public HumanAdapter(Human human){  
        this.human = human;  
    }  
    @Override  
    public void dance() {  
        human.walk();  
    }  
}
```

```
public class App {  
    public static void main(String[] args) throws Exception {  
        Man luke = new Man();  
        Robot r2d2 = new Droid();  
  
        System.out.println("Man..");  
        luke.walk();  
        luke.talk();  
  
        System.out.println("\nDroid..");  
        r2d2.dance();  
        //wrapping a human in a HumanAdapter so it can call Robot Methods  
        Robot ki = new HumanAdapter(luke);  
  
        System.out.println("\nHumanAdapter ");  
        System.out.println("calling Robot Method dance()...");  
        ki.dance();  
    }  
}
```

```
Man..  
Left, Right, Left, Right..  
well hello there  
  
Droid..  
Up, Down, Up, Down..  
  
HumanAdapter  
calling Robot Method dance()...  
Left, Right, Left, Right..
```