

Nachdenkzettel: Interfaces und Software-Architektur

Sara Tietze (40800), Pia Schilling(40905), Merve Özdemir (41709)

1. Spezifizieren Sie das Interface „Stecker“ für diese Implementation.



copyright Aunkrig, [CC-BY-SA-4.0](https://creativecommons.org/licenses/by-sa/4.0/)

Stecker muss folgende Eigenschaften erfüllen:

- muss DIN-Norm erfüllen (DIN-Norm ist das Interface)

2. Ist das

a) eine korrekte Ableitung von der obigen Implementation?

Nein, der obere Stecker hat kein zusätzliches Loch und passt deswegen nicht in die Steckdose von unten.

b) eine korrekte Implementation Ihres Interfaces

Jain, manche Stecker passen rein, wenn sie oben ein Loch haben. Für andere bräuchte man einen Adapter. Die DIN-Norm schreibt vor wie Stecker und Steckdose aussehen sollen.



copyright hic et nunc, Cc-by-sa-3.0-migrated

3. Und das?

- Ja, es ist eine korrekte Implementation, welche die DIN-Norm erfüllt



Autor: somnusde, wikimedia-commons, PD

4. Wie sieht es mit 220 V aus? Interface oder Implementation? Und das Material des Schukosteckers?

220 V ist eine Implementation und das Material ebenfalls. Das Interface ist die DIN Norm.

5. Wieviel Spass hätten wir ohne die DIN Norm für Schukostecker oder Eurostecker?

Gar keinen, weil dann jeder Stecker unterschiedlich wäre. Man müsste für jeden einen Adapter oder ähnliches kaufen.

6. Was gehört alles zum „Interface einer Klasse“ in Java? (Anders formuliert für UX-Leute:

wenn ich von jemandem eine Klasse in meinem Code benutze: was ärgert mich, wenn es geändert wird?)

- Methoden, die jede Klasse implementieren muss
- Konstanten
- Name des Interfaces
- geschweifte Klammern ! :)

7. „Class B implements X“. Jetzt fügen Sie eine neue Methode in Interface X ein. Was passiert?

Java gibt eine Fehlermeldung aus, da die Klasse B die neue Methode noch nicht implementiert. Die Methode B muss jedoch zwingend alle Methoden des Interfaces implementieren.

8. Zwei Interfaces sind nicht voneinander abgeleitet, haben aber zufällig die gleiche Methode. Können Sie Implementationen dieser Interfaces polymorph behandeln?

```
Interface X {                Interface Y {                class B implements Y { ...}
public void foo();
}
```

```
X x = new B(); ??    ->    Y y = new B();
x.foo(); ??
```

Nein, da es sich um zwei verschiedene Objekte handelt. Es müsste statt X x, Y y heißen. Polymorphie ist nicht Möglich, da ein X kein Y ist, nur weil sie dieselbe Methode haben.

9. Ihr code enthält folgendes statement: X xvar = new X(); Was ist daran problematisch, wenn Sie eine Applikation für verschiedene Branchen/Kunden/Fälle bauen?

Es wird immer ein X instanziiert. Es wäre besser, je nach Anwendungsfall zu entscheiden, welches Objekt erstellt wird, beispielsweise mit einer Factory-Klasse/Methode.