

Nachdenkzettel: Interfaces und Software-Architektur

.....
Vorname, Name, Matrikelnummer

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



Teile:
- Minuspol
- Pluspol
- Schutz
- Nicht leitend

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

2. Ist das a) eine korrekte Ableitung von der obigen Implementation? → Nein, wegen Schutzleiter
b) eine korrekte Implementation Ihres Interfaces → Keine separate Implementation, keine Leitung



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



3. Und das? Autor: somnusde,
wikimedia-commons, PD

→ Kein Schutzleiter
→ 4pin Zone
→ kann es verwenden

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

Beim Interface, da der Stecker es bis zu diesem Zeitpunkt erreichen muss, danach richtet sich das Material

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

— Absolut keinen

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: Name, Parameter (Type, Name, Reihenfolge), return Wert (type), public (Sichtbarkeit)
- static / public Attribute
- Klassenname, Overklasse, Package
- declared Exceptions
- final

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

→ Die Klassen, die es implementieren, müssen

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(); ??
x.foo();        ??
```

→ Nein! Sind verschiedene Typen

9. Ihr code enthält folgendes statement: `X xvar = new X();`

Was ist daran problematisch, wenn Sie eine Applikation für verschiedene Branchen/Kunden/Fälle bauen?

→ Das Problem ist, dass wenn der Kunde ein `xvar = x'()` will (also anders als `xvar = x()`), dann muss man den Code verändern und auf das neue anpassen

10. Von ArrayList ableiten oder eigene Klasse „Catalog“ oder ähnlich bauen und ArrayList<> verwenden? Sprich: soll man von Java Basisklassen ableiten? Beispiele: Vegetable, VegetableCatalog Task, TaskList, GameObject, GameObjectList etc.

→ Nein, falls man diese abändert