

Nachdenkzettel: Interfaces und Software-Architektur

.....
Vorname, Name, Matrikelnummer

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



metallplatte in der mitte
größe und anzahl der pins
größe allgemein
erdung

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

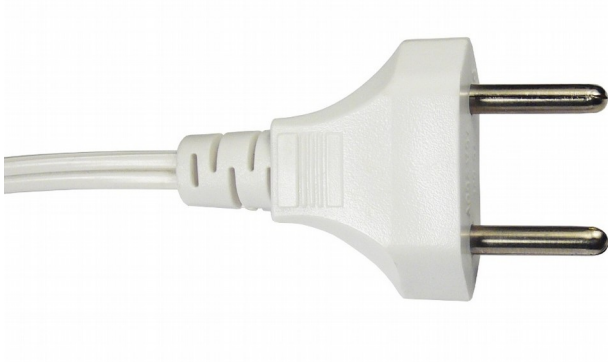
2. Ist das a) eine korrekte Ableitung von der obigen Implementation?
b) eine korrekte Implementation Ihres Interfaces

falsche erdung

nein



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



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

ja

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

damit es nicht zu Überspannung kommt, muss es genormt sein. Grenzbereich, denn eigentlich eher Attribut.
Material ist nicht zwingend egal und muss im Interface als "nicht leitend" klassifiziert sein.

->Grenzfälle

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

weniger, weil nichts zusammen passt und man immer Adapter bräuchte.

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?)

- setter und getter für wichtige Attribute, die jede Klasse zwingend braucht
- public Methoden(return Type, attribute ändern)
- Name der Klasse
- Konstruktor
- exception declaration

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

muss auch in B implementiert werden

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();        ??
```

implementiert nur Y und nicht X, x kann nicht als B implementiert werden

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

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

dass X immer abgeändert werden muss. Es sollte ein Interface mit Factory als Platzhalter verwendet werden um verschiedene Factories einbauen zu können.