Interfaces und Software-Architektur

1. Spezifizieren Sie das Interface „Stecker“ für diese Implementation.
Buchse (weibliche Plug)

- 2 runde Löcher
- rund, mit abgeflachten Seiten
- zwei Auskerbungen links und rechts

Stecker (männliche Plug):

- Zwei Stifte
- Länge
- Material (leitfähig)
- Dicke
- Runde Form
- Einkerbung Kontakte

2. Ist das
a) eine korrekte Ableitung von der obigen Implementation?
Nein, Einkerbungen für die Kontakte fehlen. Obere Schwarze Stecker passt in diese Buchse nicht herein.
Zusammenhang Liskovsche Prinzip:
Alle abgeleitete Klassen müssen mindestens so viel können wie die Basis-Klassen

b) eine korrekte Implementation Ihres Interfaces
Nein, da die Schutzkontakte wieder fehlen und - alle abgeleiteten Klassen müssen mindestens so viel können wie die Basis-Klasse -

3. Und das?
Nein, da Verstoß gegen Liskov. Kein Schutz.

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

- 220 V ist ein Interface für alle elektrischen Geräte

- Das Material könnte man jetzt aus verschiedenen Perspektiven beachten, da es natürlich nicht leitfähig sein darf aber es nicht unbedingt aus genau diesem Kunststoff-Material bestehen muss wie oben gezeigt.

5. Wieviel Spass hätten wir ohne die DIN Norm für Schukostecker oder Eurostecker?
Jeder hätte andere Stecker, Voltangaben, etc. Also gäbe es keine Einheitlichkeit. Das würde heißen: noch mehr Adapter für Apple-User!

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?)

- besondere Form einer Klasse
- enthält abstrakte Methoden und Konstanten
- anstatt class wird int
  erface deklariert
- alle Methoden sind implizit abstrakt und öffentlich
- neben Methoden kann Interface Konstanten enthalten
- Definition von Konstruktoren ist allerdings erlaubt
- Return Typen der Methoden
- Reihenfolge der Methodenparameter

7. „Class B implements X“. Jetzt fügen Sie eine neue Methode in Interface X ein. Was passiert?
Fehlermeldung in Klasse B - Die Klasse muss alle Methoden besitzen, die im Interface hinterlegt sind. Eine abstrakte Klasse in dem Interface erstellen, damit alle Klassen die Methode von der abgeänderten abstrakten Klasse erhalten.

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();
      public void foo();
      } }
X x = new B(); ??
x.foo(); ??
```

Nein, da das System durch die fehlende Ableitung keine Verbindung zwischen den Interfaces sieht und somit kein polymorphes Behandeln zulässt.

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

Der Variablennamen ist leider schlecht gewählt. Es ist somit nicht nachvollziehbar was der doofe Entwickler hier ausdrücken will … Vermutlich zu lange im Keller gesessen. Er benötigt mal wieder frische Luft.
Ausserdem wird immer genau der Typ X erstellt, der nicht Kundenspezifisch geändert werden kann.