Nachdenkzettel: Interfaces und Software-Architektur

..................................................................

Vorname, Name, Matrikelnummer

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



Interface EUStecker {

Public int anzahlPins(Integer anzahl){

Return anzahl;

}

1. Anzahl Pins
2. Material
3. Zulässige Stromstärke
4. Schutzverkleidung

(boolean?)

copyright Aunkrig, [CC-BY-SA-4.0](https://commons.wikimedia.org/wiki/Category:CC-BY-SA-4.0)

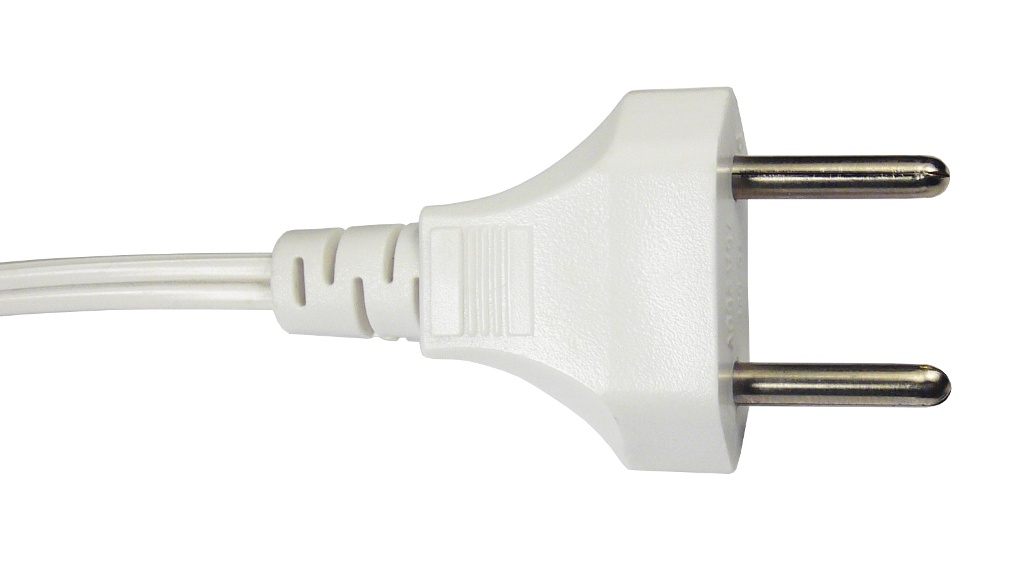
1. Ist das a) eine korrekte Ableitung von der obigen Implementation?

b) eine korrekte Implementation Ihres Interfaces



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

1. Nein, da der untere Stecker einen Pin mehr besitzt als der obige Stecker.
2. Nein, da unser Interface nur für EU Stecker implementiert werden soll.



1. Und das?

Dieser Stecker könnte das EUStecker-Interface implementieren.

Autor: somnusde, wikimedia-commons, PD

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

Die 220V sind ein genormter Wert der Spannung in DE, daher würde dieser Wert im Interface festgelegt werden. Da das Material des Schukosteckers auch genormt ist, gehört dies auch in das Interface.

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

Keinen, weil ohne Normen für jeden einzelnen Stecker die jeweiligen Eigenschaften und Kompatibilitäten mit den Steckern überprüft werden müsste.

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?)

-Name des Interfaces

-Methodennamen, deren Rückgabetyp und Access Modifier

-

-

-

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

Die Klasse B muss die neu hinzugefügte Methode im Interface X nun 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(); public void foo();

} }

X x = new B(); ??

x.foo(); ??

Die Klasse B implementiert das Interface X nicht und in der Klasse B wird kein Objekt des Interfaces Y erstellt, daher kann die foo()-Methode nicht aufgerufen werden und polymorph behandelt 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?

Es ist problematisch, da für viele verschiedene Branchen/Kunden/Fälle immer erneut ein Objekt vom Interface X erstellt werden muss.