

Nachdenkzettel: Interfaces

1. `class B implements X`. Jetzt fügen Sie eine neue Methode in Interface `X` ein. Was passiert? Was bedeutet dies für die "Robustheit" der Interface-Definition?

2. Ihr Code enthält folgendes Statement

```
X myX = new X();  
myX.doSomething();
```

Was ist daran problematisch, wenn Sie eine Applikation für verschiedene Branchen/Kunden/Fälle bauen? Wie schaffen Sie Abhilfe? Welche Rolle spielen dabei Factories?

3. Was gehört zum "Interface" einer Java-Klasse? Heißt: Was ist für außenstehende Objekte ersichtlich?

4. *Optionale Aufgabe*: Zwei Interfaces sind nicht voneinander abgeleitet, haben aber zufällig die gleiche Methode. Können Sie Implementationen dieser Interfaces polymorph behandeln?

```
interface X{  
    void foo();  
}  
  
interface Y{  
    void foo();  
}  
  
class B implements Y {...}  
  
X x = new B(); // möglich?  
x.foo(); // möglich?
```