

PSE Blatt 08 - Gruppe 32

Marvin Hepp (3759031)& Nick Daiber (3728224)

December 13, 2024

1

a

Die Klasse ist nicht Semantisch korrekt, da sie nur mit dem Parameter `numberOfWheels` initialisiert wird. Der Wert `numberOfSeats` wird dabei nie initialisiert, soll aber in der Methode `getNumberOfSeats()` abgefragt werden können. Um die Klasse zu korrigieren, kann man den Konstruktor um einen Parameter vom Typ `int` erweitern, und die Konstante `numberOfSeats` damit initialisieren. Die Klasse gilt als unveränderlich (`immutable`), da alle Felder (`numberOfWheels` und `numberOfSeats`) `final` sind.

b

Die beiden Felder `name` und `birthDate` in Zeile 14 und 15 werden nicht als `final` deklariert. Daher sind sie veränderlich. Dafür gibt es in Zeile 54 die Methode `setName` welche man aufrufen kann um das Feld "name" einer Instanz der Klasse `Person` zu verändern. Außerdem gibt es in Zeile 67 die Methode `setBirthDate` mit der man das Feld "setBirthDate" ändern kann kann. Im Konstruktor wird zudem die Referenz auf ein Objekt der Klasse `Date` direkt übergeben und keine Kopie erstellt (Zeile 31). Da `Date` ein veränderliches Objekt ist, lässt sich dieses von außen verändern, nachdem es zum erstellen einer Instanz der Klasse `Person` genutzt wurde. Zum Beispiel kann mit der `getBirthDate` Methode die referenz zu dem Objekt der Klasse `Date` erhalten und dieses dann verändern.

2

a

Defensives Programmieren sollte wann möglich genutzt werden, da es zu skalierbaren und einfacher nutzbaren code führt. Hierbei ist besonders wichtig immer zu überprüfen, ob alle werte im programm an entsprechenden schnittstellen passen. Offensives Programmieren ist fehleranfälliger, aber schneller. Besonders bei "proof-of-concept" Fällen, wo es nicht darum geht "guten" code zu schreiben sondern schnell fertig zu sein ist Offensives Programmieren hilfreich

f

Man iteriert über alle Wände und formt eine Menge der vorhandenen Türen.
Dies ist zwar langsam, aber sehr robust und schnell zu implementieren.