

# ADAP CW#10

## Projektdaten

Projekt name: 3D-Printing  
Projekt repository: <https://github.com/ModischFabrications/wahlzeit>  
This week's tag: adap-hw09 on master  
Homework diff: <https://github.com/ModischFabrications/wahlzeit/compare/adap-hw08...adap-hw09>  
CI: <https://travis-ci.org/ModischFabrications/wahlzeit>  
Docker Hub Repo: <https://hub.docker.com/r/modischfabrications/wahlzeit>

## Hausaufgaben

### Implementation

**Neue Klassen (falls vorhanden)**

**Veränderte Klassen (falls vorhanden)**

AbstractCoordinate, CartesianCoordinate, SphericCoordinate

LocationTest, CoordinateTest, CartesianCoordinateTest, SphericCoordinateTest

### Erklärung

Anfangen habe ich diese Woche mit eine Review der schon existierenden Klassen, dabei fiel mir auf, dass die Koordinate bereits immutable ist, dieser Teil der Aufgabe ist also bereits erfüllt. Hierfür relevant ist, dass Werte final sind und keine Veränderung der Attribute möglich ist. Mit shared Instances könnte hier wieder nur auf die Identität statt auf die Werte geprüft werden, darauf wurde in diesem Kontext aber im Sinne der Rückwärtskompatibilität und fehlenden Notwendigkeit verzichtet.

Dann habe ich das Instanz-Sharing durch statische Maps implementiert, die bei einer neu eingeführten Factory-Methode das Sharing gewährleisten. Rein logisch hätte auch ein Set gereicht, da diese aber keine Getter bereitstellen wird hier der Umweg über eine selbstgeschlüsselte Map gewählt.

Die neu hinzugefügte Implementation von der Factory-Methode wurde dann in die Testmethoden aufgenommen und es wurde je Koordinatenklasse ein "assertSame" Test eingeführt, der auf eine geteilte Instanz prüft.

Zuletzt habe ich alle Änderungen gepusht und den letzten Commit mit dem oben genannten Tag für die Abgabe markiert.

## Fragen

### **What benefits or drawbacks does the Value Object pattern have in this context?**

Der große Vorteil ist die Fehlersicherheit, da Objekte nicht versehentlich verändert werden können. Außerdem wird so die Nutzung als austauschbarer Container für abstrakte Koordinatinformationen vereinfacht. Falls Cocorreny eingeführt wird kann auch das Synchronisieren minimiert werden.

### **How did you handle the mixing of Coordinate objects of different classes and why?**

Ich habe keine Änderung vorgenommen an den letzten Aufgaben, die aktuelle Lösung ist damit immer noch die Konvertierung zu einer „Normklasse“, in der mit einer Unschärfe verglichen wird.

Die Map, die das Sharing bereitstellt, wurde dabei bewusst in den Kindklassen gehalten, um ein implizites Überschneiden der Mengen durch das geteilte equals zu vermeiden.