Summary

Project Name

Patterns

Project Repository

https://github.com/Lukeekul/wahlzeit

Project CI

https://travis-ci.org/Lukeekul/wahlzeit

Current Tag

adap-hw07 on master

Diff to last tag

https://github.com/Lukeekul/wahlzeit/compare/adap-hw06...Lukeekul:adap-hw07

Overview

- Design-by-Contract implemented for Coordiante Classes
- Test added to check if specific Exception are thrown

Details

Construction

- All assert-methods are declared in the *AbstractCoordinate* Class and also declared *protected* since they are not accessible from without these classes
- All leaf classes of Coordinate shared the declared assert methods, the implementation of those methods can vary and is done within leaf class

Implementation

- *assertIsValidInput*: abstract in *AbstractCoordinate*, implemented in leaf classes, called in constructors to ensure the given parameter values are within correct limits. Throws *IllegalArgumentException*.

- *assertIsValidCoordinate*: abstract in *AbstractCoordinate*, implemented in leaf classes. Called to ensure the member variables are within correct limits. In *SphericCoordinate* the parameter values for the constructor

are in degrees, the member variables in radiants, therefore are different implementation is needed. In *CartesianCoordinate* the methods just calles *assertIsValidInputs* with member variables as parameters. Throws *IllegalArgumentException*.

- *assertIsNonNullArgument*: Checks if parameter is *null*, throws *NullPointerException*.

- *assertClassInvariants*: Checks minimal requirement for classes to be true at all time, being valid Coordinate. Calls *assertIsValidCoordinate*, throws *IllegalStateException*.

- *assertIsValidValue*: Checks if a double parameter is within given limits (positive, less than inf). Throws *IllegalStateException*.

- *assertIsValidAngle*: Checks if a double parameter is within given limits (positive, less than 360 degrees). Throws *IllegalStateException*.

Test

- new test is added to check for specific Exception if called with illegal Arguments by using junit as harness.
- old tests ensure no unintentional violations occure.