

ADAP CW#09

Projektdaten

Projekt name: 3D-Printing
Projekt repository: <https://github.com/ModischFabrications/wahlzeit>
This week's tag: adap-hw08 on master
Homework diff: <https://github.com/ModischFabrications/wahlzeit/compare/adap-hw07...adap-hw08>
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)

Location, PrintPhoto,

PrintPhotoTest

(Rest war bereits abgesichert)

Erklärung

Die aktuelle Hausaufgabe befasst sich mit Errorhandling, daher habe ich zuerst die schon vorhandenen Prüfungen angeschaut. Viele der Klassen enthalten bereits precondition-checks, beispielsweise eine einfache Überprüfung auf null-Werte, daher mussten diese nicht ergänzt werden.

Dann habe ich die von mir verwendeten `IllegalArgumentException`s zur Methodensignatur hinzugefügt, um Nutzer der Implementation auf diesen Fehlerfall hinzuweisen.

Um die von mir entwickelten Preconditions auch zu prüfen habe ich dann einen weiteren Testfall hinzugefügt, der so eine Exception provoziert.

Eine explizite Fehlerbehandlung war in Fall der Koordinaten nicht notwendig, da die von mir erstellen Klassen keine Verwendung in anderen Modulen haben. Im Fall der Photoklasse reichen die Abhängigkeiten tief in den Code des Frameworks, außerhalb der von mir erstellten Klassen, weshalb hier ebenfalls keine Änderungen vorgenommen wurden.

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

Fragen

What type of exceptions did you use, where, and why?

Ich habe überwiegend `IllegalArgumentException`s für die Preconditions genutzt, da mit diese die semantische Aussage des falschen Parameters auf eine genormte Weise ausdrückt. Eine häufig verwendete Alternative wären selbst definierte Exceptions gewesen, hierfür waren die bereits existierenden Exceptions aber aussagekräftig genug. Weitere Exception-Definitionen hätten den Programmcode nur aufgebläht, ohne weiteren Informationsgehalt zu liefern.

Wenn es realistisch ist würde ich immer checked-Exceptions verwenden, da der Client explizit auf die Möglichkeit eines Fehlers hingewiesen werden soll, um auch die dafür sinnvolle Fehlerbehandlung zu implementieren. Die verwendeten `IllegalArgumentException`s sind leider unchecked, die semantische Bedeutung war mir aber wichtiger als das Erzwingen des Typs. Als Hilfestellung habe ich diese Exception dennoch in die Methodensignatur aufgenommen.