
Abgabeprotokoll

Python TDD

Softwareentwicklung
4BHIT 2015/16, GruppeB

Philip Vonderlind

Note:
Betreuer: Prof. Raffener & Prof. Dolezal8

Version 0.1
Begonnen am -
Beendet am 23.10.2016

Inhaltsverzeichnis

1	Einführung	1
1.1	Aufgabenstellung	1
2	Ergebnisse	2
2.1	Tests Allgemein	2
2.2	Vergleichs Teste	3
2.3	String und Iteration	3
2.4	Addition und Subtraktion	4
2.5	Multiplikation und Division	5
2.6	Code Coverage	6

1 Einführung

Schreiben Sie zu die Klasse Bruch in einem Modul bruch

Nutzen Sie die Testklassen in PyCharm.

Ziel: Coverage > 95%

1.1 Aufgabenstellung

Empfohlene Vorgehensweise:

Projekt in PyCharm erstellen Modul bruch erstellen Klasse Bruch erstellen Test-Ordner erstellen
Unit-Tests entpacken und lauffähig machen

Abgabe:

Protokoll mit Testreports (inkl. Coverage) und Dokumentation (html)

Abgabe des Python-files

Achtung: Vergessen Sie nicht auf eine ausführliche Dokumentation mittels sphinx

2 Ergebnisse

2.1 Tests Allgemein

Zu den allgemeinen Tests zählten hier Konversionen der Brüche in Integer, Float und so weiter. Dazu kann man sich, wie bei fast allen anderen Testfällen auch, die Parameterüberschreibung von Python zunutze machen. Konkret heißt dass, es werden Funktionen wie zum Beispiel `__float__` überschrieben. Auch die verwendeten Operatoren wie `+`, `-`, usw. können über die zugehörigen Methoden überschrieben werden. Um Fehler bei jeweiligen Konversionen und Operationen aufzudecken, werden Exceptions mittels 'raise' geworfen. Auch interessant ist der im Konstruktor verwendete Standardwert für `denom`, da man sonst keine Brüche mit nur einem Zähler als Input erzeugen könnte. Aus Gründen des Platzes ist es sinnlos den Code hier anzuzeigen, da dieser durch Kommentare sehr bloated wurde, daher zeige ich hier die Testergebnisse.

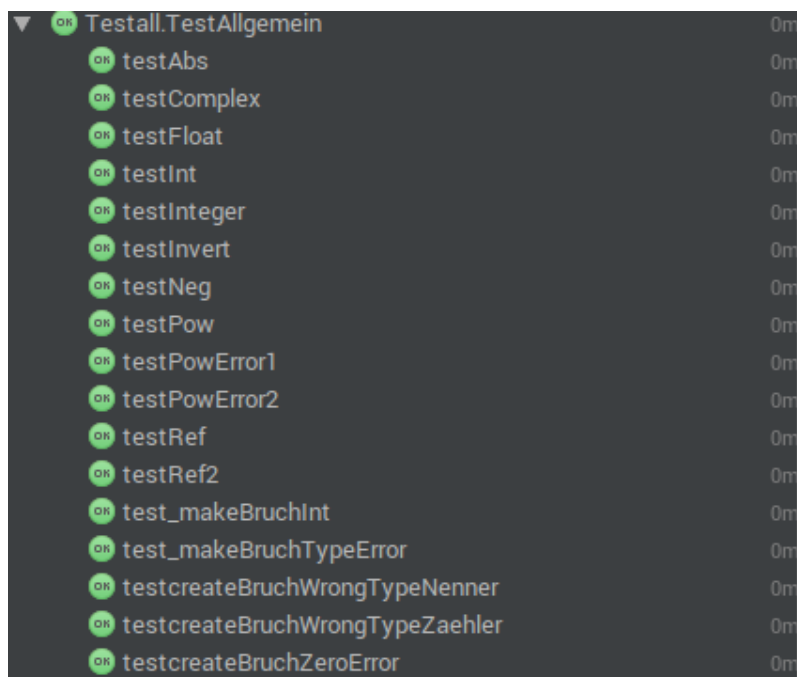


Abbildung 1: Test Allgemein

2.2 Vergleichs Teste

Der Vorgang ist auch hier derselbe wie bei dem vorherigen Testgebiet, also werde ich keine lange, sinnlose Erklärung anhängen.

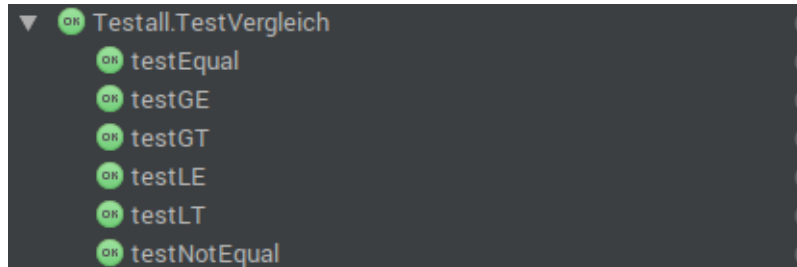


Abbildung 2: Test Allgemein

2.3 String und Iteration

Da diese beiden jeweils nur eine Methode umfassen, schreibe ich sie hier gemeinsam an. Die Iteration ist relativ selbsterklärend. Die String Methoden sind anzusehen wie toString in Java. Ich habe mir hier einfach String Templates zunutze gemacht um die Werte in String einzufüllen und diese zu returnen. Man muss hierbei aber auch wieder auf Sonderfälle wie ganze Brüche achten. Diese werden nämlich in ihrer Integer Form dargestellt.

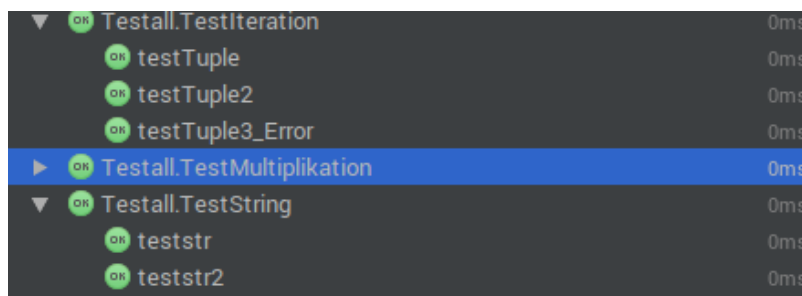


Abbildung 3: Test Iteration und Strings

2.4 Addition und Subtraktion

Da diese dasselbe Prinzip verfolge, werden sie zusammengefasst. Wieder mittels der Überschreibung von passenden Funktionen, kann man die Addition und Subtraktion von Brüchen mit Integern, sowie mit anderen Brüchen ermöglichen. Hierbei muss aber immer das kleinste gemeinsame Vielfache als Wert des Nenners für die neuen Brüche beachtet werden. Damit die Subtraktion nicht extra implementiert werden musste, habe ich einfach die Additionsfunktion hier auch benutzt um mit einem negativen Wert zu addieren.

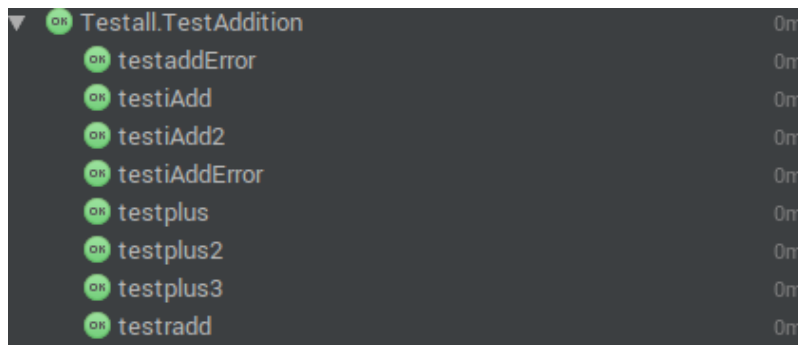


Abbildung 4: Test Addition

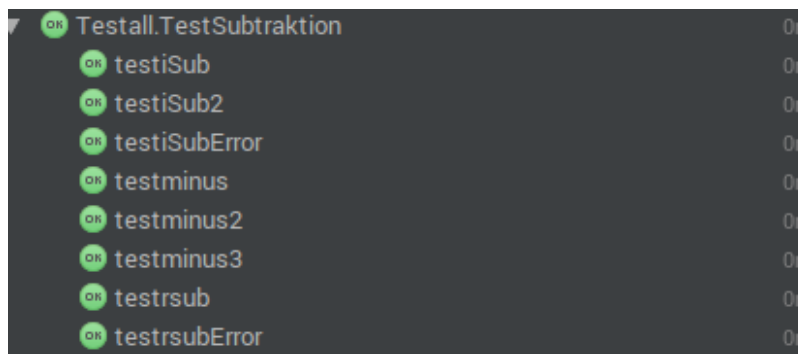


Abbildung 5: Test Subtraktion

2.5 Multiplikation und Division

Verfolgen das selbe Prinzip wie der Code von Addition und Subtraktion, daher sind diese auch gleich aufgebaut, nur mit den jeweils passenden Überschreibungen versehen. Man muss hier aber aufpassen, dass einem keine Division durch 0 untergerät, daher werden hier häufiger Checks durchgeführt. Auch beim Kreuzprodukt bei der Multiplikation und bei dem Doppelbruch der Division muss man auf jeweilige Sonderfälle aufpassen.

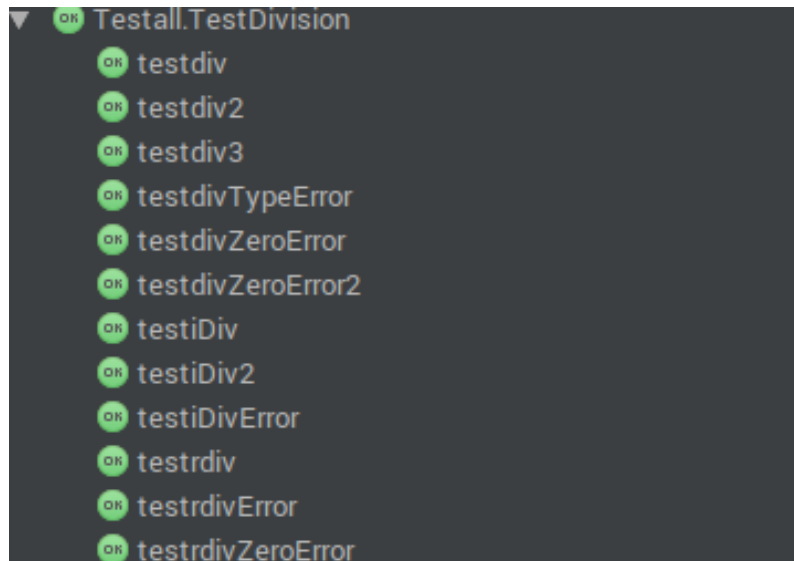


Abbildung 6: Test Divison

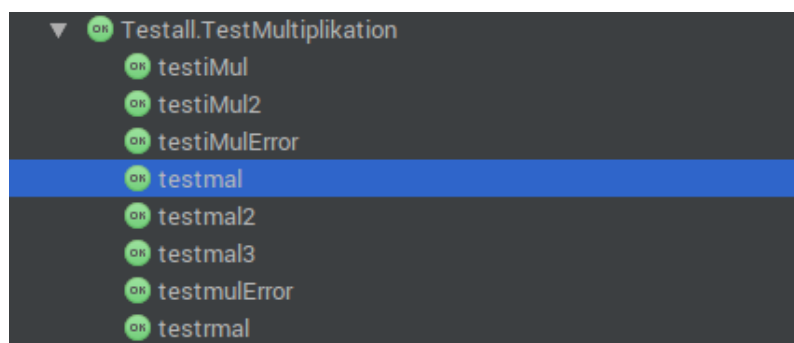


Abbildung 7: Test Multiplikation

2.6 Code Coverage

Anhand der Testreports im doku Ordner wird erkennt man gut, das die erforderlichen 95% erreicht wurden. Hier anbei aber noch einmal ein Screenshot der PyCharm Ausgabe.

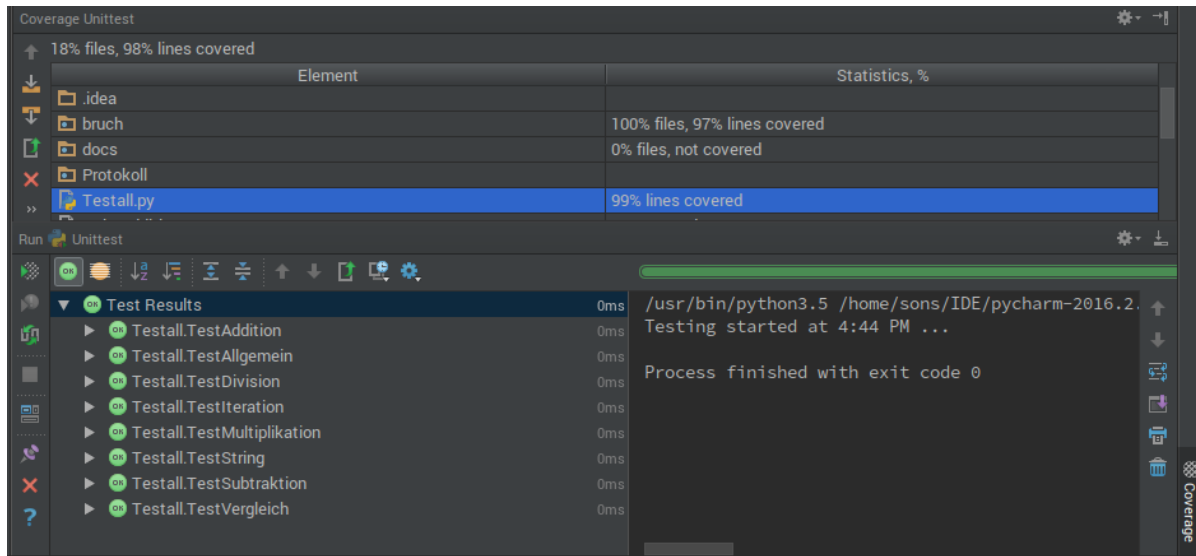


Abbildung 8: Test Coverage

Tabellenverzeichnis

Listings

Abbildungsverzeichnis

1	Test Allgemein	2
2	Test Allgemein	3
3	Test Iteration und Strings	3
4	Test Addition	4
5	Test Subtraktion	4
6	Test Divison	5
7	Test Multiplikation	5
8	Test Coverage	6