

S03 - Calculator

TGM 4AHIT

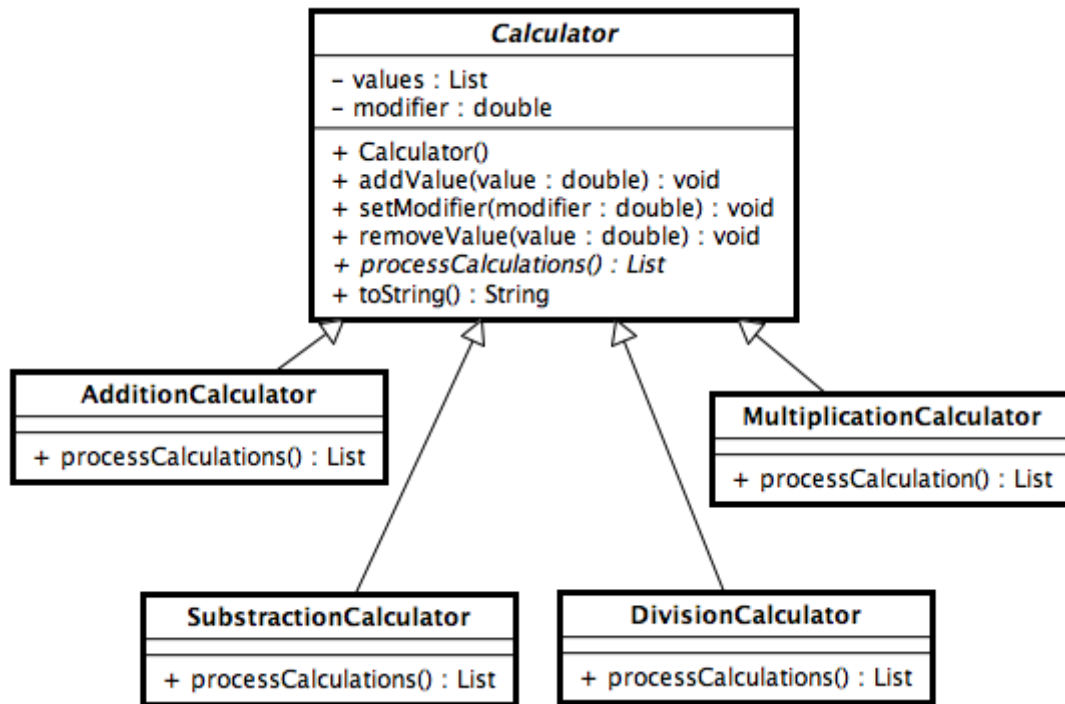
November 12, 2014
Verfasst von: Thomas Taschner

Inhalt

Aufgabenstellung	2
Designüberlegung	3
Zeiteinteilung	3
Arbeitsdurchführung/Lessons Learned	3
Testbericht	4

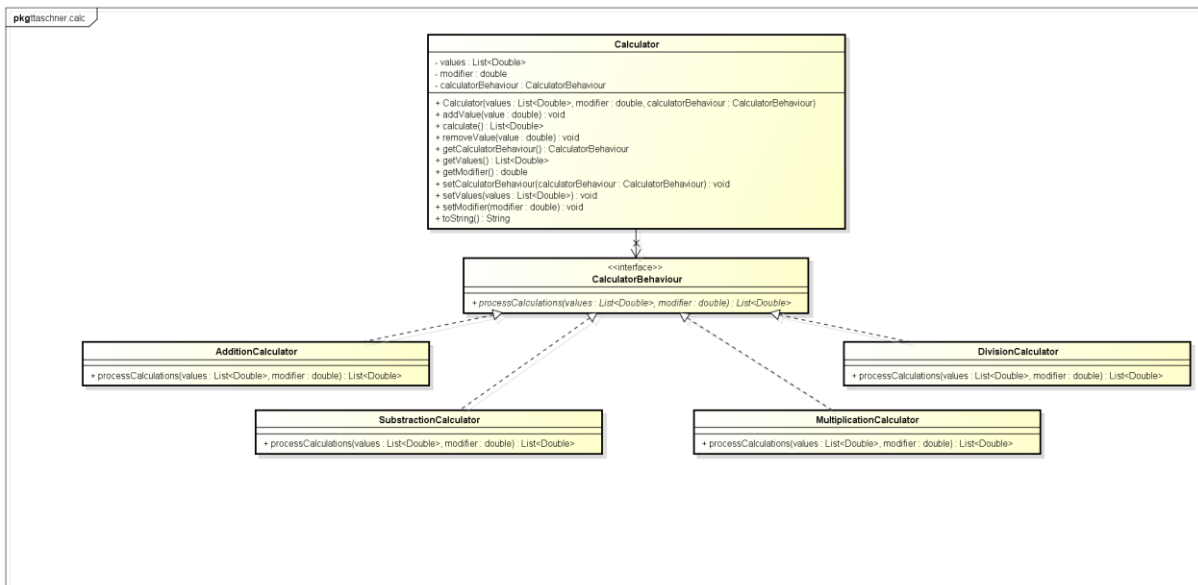
Aufgabenstellung

Ändere folgendes UML-Diagramm so um, dass es dem Strategy-Pattern entspricht und implementiere dann dieses.



Die abstrakte Klasse Calculator hat die Aufgabe, Werte aus einer Liste mit einem modifier zu verändern und das Ergebnis als neue Liste zurück zu geben. Dazu dient die abstrakte Methode `processCalculations`, die in den konkreten Subklassen so überschrieben wurde, dass sie je nach Klasse die Werte aus der Liste mit dem modifier addiert, subtrahiert, multipliziert oder dividiert.

Designüberlegung



Damit das UML-Diagramm dem Strategy-Pattern entspricht, wurde ein Interface hinzugefügt, welches von den 4 Calculator* Klassen implementiert wird. In der Klasse Calculator wurde ebenfalls noch CalculatorBehaviour Objekt mit dem Namen calculatorbehaviour hinzugefügt. Außerdem sind bei der processCalculations Methode noch 2 Parameter hinzugekommen, um mit diesem veränderten Softwaredesign arbeiten zu können. Der Konstruktor setzt jetzt auch die Werte für modifier, values und calcbehaviour, die er nun mitbekommt. Abschließend wurden noch Getter- und Setter-Methoden hinzugefügt, mehr Änderungen gibt es nicht.

Zeiteinteilung

Aktion	Geplant	Tatsächlich
Erstellen des UMLs	1h	0.5h
Implementieren des UMLs	2h	0.75h
Testen des Codes	1h	1.5h
Dokumentieren	0.5h	1h

Arbeitsdurchführung/Lessons Learned

Das Strategy-Pattern kann einem viel Arbeit ersparen, da der Code bei Änderungen nur an wenigen Stellen geändert werden muss. Ein bereits vorhandenes UML-Diagramm ist durchaus praktisch, da man einen Überblick über die Aufgabe hat und das Code Grundgerüst auch von Astah automatisch generiert werden kann.

Testbericht

Die Tests haben ohne Probleme funktioniert.