Vorgehensweise und Entscheidungsfindung

Wir haben uns dazu entschieden, die Grundrechenarten über eine abstrakte Klasse zu implementieren. Alle Rechenarten benötigen die Methode execute zum ausführen der Rechenoperationen, weiterhin benötigen alle Klassen die Methode getReverse um für die redo & undo Implementation die inverse Rechenoperation zu bekommen.

Weiterhin wurde in der abstrakten Klasse die toString-Methode überschrieben. Die toString Methode gibt uns das derzeitige tatsächliche Ergebnis auf der Konsole aus inklusive der Rechenart (getClass()) .

Die Klasse Calculator implementiert sogenannte Stacks, initialisiert mit der abstrakten Klasse. Wir implementieren hier zwei Stacks, weil wir einen Stack für die History brauchen und einen Stack für die undo-History. Wenn wir nur die History hätten und wir würden etwas rückgängig machen, dann könnten wir später nicht mehr auf diese Stelle verweisen (redo).

Das derzeit aktuelle Ergebnis wird in currentValue gespeichert. Die Methode run führt sämtliche logischen Operationen, sowie die Ein- und Ausgabe zusammen. In CurrentValue wird bei Programmstart der erste Operand gespeichert, auf dem im Methodenablauf eine der einzelnen Rechenoperationen ausgeführt wird.

Die do-while Schleife in der Klasse Calculator wird solange ausgeführt, bis der Programmablauf durch ein Drücken der Taste 'e' beendet wird. Im Grunde wird hier das Menü ausgegeben und die vom User gewünschte Aktion eingelesen.

In dem ersten If-then-else Block wird abgefragt ob eine der Funktionen „Rückgängig, vorwärts, wiederholt oder neustarten“ ausgeführt werden soll. Die

Die Klasse Stack hat u.a. drei Methoden implementiert, welche einfache Operationen zur Umsetzung eines LIFO Prinzipes beinhalten:

- peek(); pop(); push();

Wir haben im Übrigen angefangen die Anwendung mit if-then-else Blöcken zu implementieren und waren im Nachhinein zu lethargisch, die Umstände an einen Switch Case anzupassen.

Im zweiten Teil des großen If-then-else Konstruktes, man stelle es sich als Switch vor, teilen wir den eingegebenen String in einen mathematischen

Operator und einen Operanden und speichern im folgenden 'case' die entsprechende Rechenanweisung im Attribut 'command'.

Mit der letzten If-Anweisung wird überprüft, ob ein gültiges Kommando eingegeben wurde. Ist dies der Fall, wird das entsprechende Kommando ausgeführt. Wurde kein gültiges Kommando eingegeben, werden einmal die Stacks ausgegeben und der Schleifendurchlauf der while Schleife von vorne begangen.

Die Klasse CalculatorIO hat sämtliche Methoden im Bauch, welche Ein- und Ausgaben abwickeln.

Die Klasse RunApp enthält die Main Methode.

TODO: if-else refaktorisieren.

Switch Statement
Sun Microsystems, Inc.: Switch Statment
http://java.sun.com/docs/books/tutorial/java/nutsandbolts/switch.html
(abgerufen am 2.6.2010)

Martin Slowikowski
Matrikelnummer: 199 91 66

Jan-Tristan Rudat
Matrikelnummer: 200 78 52

Teamname: Bernie und Ert

Quellenangaben

Stack Klasse
Sun Microsystems, Inc.; Class Stack
URL: http://java.sun.com/j2se/1.4.2/docs/api/java/util/Stack.html
(abgerufen am 30.05.2010)