

## Präsenzaufgaben 2

**27./28.03.2023**

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

### Grundlagen

Die umgekehrte polnische Notation (UPN) ist eine Notation für mathematische Operationen. Dabei werden zunächst die Operanden niedergeschrieben und anschließend der Operator.

Zum Beispiel steht der UPN-Ausdruck „3 5 +“ für „3+5“ in „normaler“ Notation. „2 3 4 + \*“ steht für „2\*(3+4)“.

Ausdrücke in UPN-Notation kann man besonders gut mit Hilfe eines Stacks bearbeiten. Die Regeln dazu sind:

- Der Ausdruck wird von links nach rechts abgearbeitet.
- Zahlen werden auf den Stack gelegt.
- Taucht ein Operator (für eine Grundrechenart) auf, werden die letzten beiden Werte vom Stack geholt, das Ergebnis damit berechnet und wieder auf den Stack gelegt.
- Am Ende der Auswertung liegt das Ergebnis des gesamten Ausdrucks, wenn kein Fehler aufgetreten ist, als letzter Wert noch auf dem Stapel.

Beispiel 1: Der Ausdruck „6 3 /“ bedeutet „6/3“. Das Aussehen des Stacks ist:

Stack	6	6	2
		3	
Ausdruck	6	3	/

Beispiel 2: Der Ausdruck „2 3 4 + \*“ wird wie folgt verarbeitet:

Stack	2	2	2	2	14
		3	3	7	
			4		
Ausdruck	2	3	4	+	*

### Aufgabe 1

Schreiben Sie einen Taschenrechner für die „Kommandozeile“, der sowohl Ausdrücke in umgekehrter polnischer Notation (UPN) berechnen als auch Variablen mit Wert definieren kann.

Zahlen und Operatoren sollen durch Leerzeichen getrennt sein. Berücksichtigen Sie als Operationen nur Addition (+), Subtraktion (-), Multiplikation (\*) und Division (/).

Zusätzlich kann der Taschenrechner auch mit Variablen arbeiten. Definiert und verwendet werden Variablen allgemein in der Form „VarName = UPN-Ausdruck“, also beispielsweise für „a“ bzw. „b“ so:

- „a = 1 2 +“
- „b = a 2 \*“

Speichern Sie die Variablen und ihre Werte in einer Hashtabelle. Variablennamen dürfen nur Kleinbuchstaben enthalten. Die Hashtabelle `mapVars` soll ein Attribut der Taschenrechner-Klasse sein. Nutzen Sie für den Stack eine geeignete Java-Klasse.

- a) Schreiben Sie eine Methode

```
private double eval(String exprUPN)
```

zur Auswertung eines UPN-Ausdrucks. Im Falle eines Fehlers geben Sie NaN zurück. Der Ausdruck darf auch Variablen enthalten, sofern sie in `mapVars` enthalten sind (also vorher definiert wurden).

Beispiele:

```
d = eval("1 2 +")           // d=3
d = eval("a 2 *")           // d=12 falls (a,6) in mapVars
```

- b) Schreiben Sie eine Methode

```
public String analyze(String expr)
```

die entweder einen übergebenen Ausdruck auswertet und das Ergebnis als String zurückgibt oder im Fall der Form „VarName = UPN-Ausdruck“ den Wert des Ausdrucks ermittelt und die Variable in `mapVars` mit diesem Wert anlegt bzw. ersetzt. Beispiele:

```
s = analyze("1 2 +")         // s="result: 3"
s = analyze("a = 1 2 +")     // s="result: a=3"
s = analyze("b = a 2 *")     // s="result: b=6"
```

- c) Verwenden und ergänzen Sie folgende Vorlage, um im Konsolen-Fenster von Eclipse direkt Ausdrücke eingeben und den Taschenrechner benutzen zu können:

```
Scanner in = new Scanner(System.in);

String z = "";
while (true) {
    z = in.nextLine();
    if (z.equals("quit")) {
        break;
    }
    if (z.equals("clear")) {
        // Loescht alle Variablen
        // zu implementieren
    }
    System.out.println("task '"+z+"' => "+analyze(z));
}
```