

Aufgabenblatt 3

Schreiben Sie ein Programm, das von der Konsole in einer Schleife jeweils eine ganze Zeile einliest, diese Zeile als arithmetischen Ausdruck auswertet und seinen Wert ausgibt.

Ein arithmetischer Ausdruck besteht aus den Operationen $+$, $*$ und $^$ (Potenzoperator; z.B. ist $2^{10} = 1024$). Der Potenzoperator hat die höchste und $+$ wie üblich die niedrigste Präzedenz. Alle Operationen sind linksassoziativ.

Freiwillig: Sie können den Potenzoperator $^$ auch rechtsassoziativ realisieren, so wie es beispielsweise in Python der Fall ist. Dort wird der Potenzoperator mit „ $**$ “ geschrieben ($2^{**2^{**3}} = 256$).

Implementieren Sie den in der Vorlesung besprochenen Shift-Reduce-Parser, der Ausdrücke mit Hilfe eines Kellers auswertet. Setzen Sie einen Keller ein, der Elemente vom Typ Object speichern kann.

Auf der Web-Seite ist bereits eine Klasse `Tokenizer` vorhanden, um einen String in eine Folge von Tokens zu zerlegen. Versuchen Sie diese Klasse zu verstehen und testen Sie sie.

Auf der Web-Seite finden Sie eine weitere rudimentäre Klasse `Evaluator`. Ergänzen Sie die Methoden in der Klasse geeignet. Fehlerhafte Ausdrücke führen zur einer Fehlerausgabe und werden ignoriert.

Das folgende Beispiel zeigt eine Folge von Konsoleneingaben in schwarz mit entsprechender Konsolenausgabe in blau.

```
>> 2*3.1+1.2  
7.4
```

```
>> 1.1+2*3  
7.1
```

```
>> 5 + 2^10  
1029
```

```
>> 1**2  
!!! error
```

```
>> (2+3*2)^2  
64.0
```

```
>> 2^3^2  
64.0
```

```
>> end  
bye!
```

Hinweis: Testen Sie Ihr Programm mit dem Debugger! Setzen Sie einen Breakpoint in den Rumpf der while-Schleife in der eval-Methode. Gehen Sie dann Ihr Programm schrittweise durch und beobachten Sie, wie sich der Stack und das aktuelle Token ändern.