

## ProInformatik II: Funktionale Programmierung

### 0. Übungsblatt

---

#### 1. Aufgabe

Haskell installieren (aus <http://www.haskell.org>).

#### 2. Aufgabe

Verwenden Sie Haskell als Taschenrechner und berechnen Sie folgende Ausdrücke:

-1	1/10	1 `div` 2	div 6 5
div 1 5	0 <= 0	4 /= 5	0.1 == 0.3/3
rem 5 2	mod 5 2	quot 6 5	quot 5 6
sin 0	pi	sin 2*pi	sin (2*p)
'\0'	'a' < 'A'	'9' < '0'	'\0' < '0'
"1+2"	"ba" < "aba"	"ay" < "y"	"\0abc"
2**1023	2^1023	2.0^1023	2^0
succ 4	pred 4	sqrt 3	sqrt (-1)
acos 3	abs (-7)	max (-2) 7	log 10
True < False	not True	True    False	False && True

Warum ist **(min -2 7)** kein gültiger Haskell-Ausdruck?

Warum ist der Ausdruck **(mod 1 0)** fehlerhaft?

Warum ist **(0.1 == 0.3/3)** gleich **False**?

#### 3. Aufgabe

Folgende Kommandos des GHCi-Compiler austesten.

:help	:?	:quit	:info sin
:browse Prelude	:!ls	:load <filename>	:show modules
:type 0	:type 'a'	:type '5'	:type "1"
:type 0.0	:reload	:usw.	

#### 4. Aufgabe

Was ist der **Wert** und der **Datentyp** folgender Ausdrücke? Versuchen Sie, zuerst die Lösungen zu schreiben, ohne in dem Haskell-Interpreter die Ausdrücke einzugeben.

(+) ((-) ((+) 4 5) 9) (-8)

2\*\*2 + (3^3)

log 1

```

(div) 1 2
mod 4 (-3)
(4 `mod` 5) == (4 `rem` 5)
(-4 `mod` 5) == (-4 `rem` 5)
(4 `mod` (-5)) == (4 `rem` (-5))
4 == (div 4 (-3))*(-3) + (mod 4 (-3))
True < False
succ 4 * 8
succ (4 * 8)
if (mod 5 2)==0 then "ja" else "nein"
(10**17)*((0.1)*3-(0.1)*2-(0.1))

```

## 5. Aufgabe

Was ist der Wert und der Datentyp folgender Ausdrücke? Finden Sie die Lösungen, ohne in dem Haskell-Interpreter die Ausdrücke einzugeben und begründen Sie Ihre Antworten.

```

True || undefined
True && not (undefined || False)
True && (not undefined || True)
3/0 ** 2
2 - 0/0

```