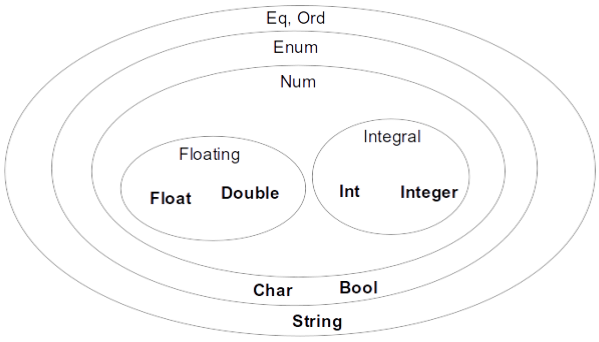
Haskell

Die funktionale Programmiersprache Haskell erschien 1990 und ist nach dem Mathematiker Haskell Brooks Curry benannt. Funktionale Programmiersprachen basieren auf dem Lambda-Kalkül, einer formalen Sprache zur Beschreibung und Auswertung von Funktionsdefinitionen.

Der Haskell-Interpeter GHCi (Glasgow Haskell Compiler interactive) ist auf den Schulcomputern installiert und wird im Unterricht in der Konsole für die Umsetzung von Haskell-Programmen genutzt werden.

Datentypen und Typklassen in Haskell

Haskell: Hierarchie ausgewählter Datentypen und Typklassen



Typklassen: Eq, Ord, Num, Floating, Integral
Datentypen: Float, Double, Int, Integer, Char, String, Bool

Recherchieren Sie, was man unter Typklassen versteht. Grenzen Sie dann die hier aufgeführten Typklassen voneinander ab.

Einfache Ausdrücke und Datentypen

1. Geben Sie folgende Ausdrücke im Haskell-Interpreter ein und notieren Sie jeweils das Ergebnis und den Typ. Den Typ kann man sich mit `:type` oder `:t` anzeigen lassen, z. B. `:type 2+9`

```
a) 3+7        b) 4-8        c) 4*8        d) 5/8
e) 3^2        f) mod 28 3   g) div 28 3   h) pi
```

2. Lassen Sie die folgenden logischen Ausdrücke im Haskell-Interpreter auswerten und notieren Sie das Ergebnis:

```
a) True                b) not True            c) 5==7    d) 4==4
e) (5==7)&&(4==4)      f) (5==7)||(4==4)      g) 5/=7    h) False||True
i) False&&True
```

3. Laden Sie die Zeichenbibliothek mit `import Data.Char`, indem Sie eine Haskell-Datei `02.hs` erstellen und darin nur die Zeile `import Data.Char` speichern. Laden Sie die Datei dann im Interpreter und lassen Sie die Auswirkungen der folgenden Ausdrücke auswerten. Notieren Sie die Ergebnisse und Vermutungen über das Zustandekommen.

```
a) 'H'                 b) head "Hallo"          c) ord 'a'
d) toUpper 'a'         e) digitToInt 'F'        f) chr 65
g) tail "Hallo"        h) head (tail "Hallo")   i) "Hallo"++"du"
j) length "Hallo"      k) reverse "Haskell"
```