


Arbeitsblatt: Bindings

Scope

Sie haben `where` und `let` Bindings kennengelernt. `Where` Bindings werden am Ende einer Funktionsklausel verwendet, um Ausdrücke an einen Namen zu binden. Diese Bindings sind innerhalb der Funktionsklausel sichtbar.

```
describeList :: [a] -> String
describeList []      = emptyListText
describeList (a:as) = nonemptyListText
  where emptyListText = "Empty List"
        nonemptyListText = "Non Empty List"
```



Kompiliert nicht, da `emptyListText` nur in der unteren Funktionsklausel sichtbar ist.

`Let` Bindings hingegen gelten nur innerhalb ihres `in`-Blocks. Zudem ist es möglich `let`-Bindings zu verschachteln. D.h. es können `let`-Bindings in `let`-Bindings vorkommen und zwar sowohl im `let`- als auch im `in`-Block:

```
letsNest :: Int -> Int
letsNest x = let
  c = let b = 6
      in x + b
  in
    let d = 4
    in c + d
```

Aufgabe 1

Was gibt obige Funktion aus bei folgendem Input? Bevor Sie die Funktion in `ghci` laufen lassen, überlegen Sie sich zuerst welchen Output Sie erwarten!

```
> letsNest 5
```

Erwartet:

Tatsächlich:

Zeichnen Sie oben ein, wie weit sich der Scope (die Sichtbarkeit) der Namen `b`, `c` und `d` erstreckt. Umkreisen Sie die Ausdrücke im jeweiligen Scope mit je einer anderen Farbe.

Shadowing

Wenn let-Bindings verschachtelt werden, ist es auch möglich, dass ein Binding einer inneren Stufe denselben Namen trägt wie ein Binding weiter aussen:

```
letsShadow :: Int -> Int
letsShadow x = let a = 5
                in
                  let a = 6
                  in a + x
```

Aufgabe 2

Was gibt obige Funktion aus bei folgendem Input? Bevor Sie die Funktion in ghci laufen lassen, überlegen Sie sich zuerst welchen Output Sie erwarten!

```
> letsShadow 2
```

Erwartet:

Tatsächlich:

Begründung:

Aufgabe: Cuboid

Implementieren Sie eine Funktion `cuboid :: Float -> Float -> Float -> Float` welche die Oberfläche eines Quaders berechnet. Wenn die Länge (l), Breite (w) und Höhe (h) gegeben sind, dann berechnet sich diese aus der Summe der doppelten Standfläche ($\text{baseArea} = l \cdot w$), der doppelten Seitenfläche ($\text{sideArea} = w \cdot h$) und der doppelten Frontfläche ($\text{frontArea} = l \cdot h$).

- Implementieren die cuboid-Funktion mit Hilfe der where-Klausel.
- Implementieren die cuboid-Funktion mit Hilfe von let-Bindings.

