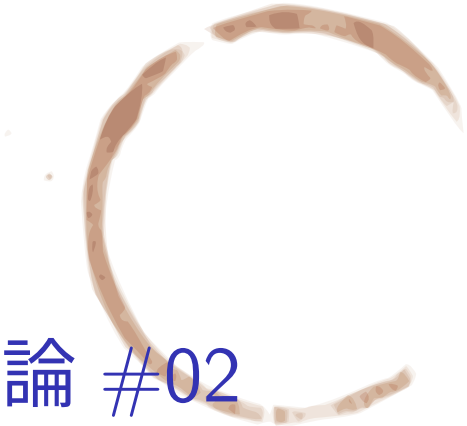

プログラミング言語論 #02

Haskell

2018-04-16

変数定義が `int *****x ;`
関数定義が `xxxx f(int **  );` ならば
関数呼出しは `f(  ***x);`

関数定義が `xxxx f(int ** );`
関数呼出しが `f(  x);` ならば
変数定義は `int **x;`

変数定義が `int *x ;`
関数定義が `xxxx f(int **  );` ならば
恒等式 $x = *\&x$ より変数定義を `int **&x ;` と変形して
関数呼出しは `f(  &x);`

すなわち

変数定義の*数
$=$ **関数定義の*数** $+$ **関数呼出の*数**

この講義では関数の型について考える．

それはプロトタイプ宣言についてあれこれ考えることと同じ．

しかし C 言語のプロトタイプ宣言はわかりにくい．

例えば引数が 2 引数関数へのポインタの配列型で返値が 1 引数関数へのポインタ型の関数 f のプロトタイプ宣言

別の言語を使ったほうがマシである．

そこで Haskell を導入する．

Haskell 言語

https://www.haskell.org/

- 純関数型プログラミング言語
- Haskell 言語の利用
  - コンパイラ： ghc
  - インタプリタ： ghci
  - オンライン IDE[1]： https://repl.it/languages/Haskell

[1] IDE=Integrated Development Environment, 統合開発環境

プログラム例の作成と実行

編集：

Main.hs （中央の編集パネルに表示されているファイルを使う）

```haskell
module Main where    -- repl.itでは必ず必要なおまじない

main :: IO ()        -- repl.itでは必ず必要なおまじない
main = return ()     -- repl.itでは必ず必要なおまじない

x :: Int
x = 10
```

実行：

1. 実行ボタンを押すと編集中のファイルが右側のパネル（これをインタプリタという）にロードされる
2. main 関数が自動実行されるが，その後も入力待ちになるので Haskell の式を実行できる

実行ボタンを押さなくても入力可能だがファイルがロードされていない

変数に関する構文

変数に関する型宣言文と定義文

```
1  -- comment
2  xxx:: YYY
3  xxx= vvv
```

1. コメント文（--で始まる）
2. 型宣言文 (行 2)
   - 変数名（必ず小文字で始まる．自分で定義する時もこのルール）
   - ::
   - 型名（必ず大文字で始まる．自分で定義する時もこのルール）
3. 定義文 (行 3)
   - 変数名
   - =
   - 値（型ごとに見かけ上の特徴がある）

講義で使う型 1

Int:64bit 整数

- (+), (-), (*) : 加減乗
- (^) : n 乗

注意：負数は括弧でくくるのが安全：「1 - -1」よりは「1 - (-1)」

実数は都合により途中まで無視（都合により存在していないことにします）

String：文字列型

C 言語と同じくダブルクォートで囲む

- (++) 左右の文字列を連結する

講義で使う型 2

Bool：真値=True と偽値= False をまとめた型[2]

Bool 型を引数に取る関数・演算子

- (||), (&&)： C 言語と同じ
- not ： C 言語での (!)

返値[3]が Bool 型になる演算子

- (==),(<), (>), (<=), (>=) ： C 言語と同じ
- (/=) ： C 言語での (!=)

() : C 言語の void にほぼ相当

[2] C 言語にはない型

[3] 返り値は Google で 18 万件，返値は 2 万件ヒット．同じもの．

型と関数・演算子の対応

Int 型

- 加減乗, div, mod

String 型

- (++)

Bool 型

- (||), (&&), not

以上の型で共通に利用可能なもの

- (==), (/=), (<), (<=), (>), (>=)

確認事項

1. 大文字小文字の区別
2. C 言語との型名の違い，存在しないもの
3. プロトタイプ宣言と類似したものの存在
4. 定義と宣言の違い
5. コメントの書き方

プログラミング言語論専門用語

1

declaration, 宣言

そういうものが存在していることにする．型を決める．

definition, 定義

メモリ上に場所を確保し，名前と番地の対応を確定する．

プログラミング言語論専門用語

2

function, 関数

名前の右に全ての引数が並ぶという構文を持つもの.

operator, 演算子

関数のルールに沿わないもの.

- 例えば自分の左右に引数が並ぶ.
- 多くの場合，記号文字で表すがそれは定義ではない.