



# プログラム言語論 課題 1

氏名 (学籍番号) : 春名航亨

(201811528 )

課題 1-7 (a) (define (unko! 1)

(if (null? 1) '() (cons (set-car! 1 "unko") (unko! (cdr 1)))))

(b) いつ実行しても同じ引数には同じ結果を返す性質である参照透明性を担保している、外部の変数を用いず、引数と値のみで完結しているような式。

外部の変数に破壊的な影響を及ぼさないもの。また append や pop が可能なミュータブルな配列を用いず、イミュータブルなリストを用いる式。

(c) 副作用を持たない関数は思わぬ影響を他の変数や環境に及ぼしたり、また環境ごとに違った結果になったりしないため、バグやエラーの温床になりにくいといえる。

また環境に依存する、使い捨てのプログラムではなく、再生しやすいプログラムに持ち込んでも使える関数となる。

課題 1-8 (a) S 式という expression (式) の記述方式を用いて記述される言語だから。

(b) 関数 or は短絡評価なので、左の引数から順にみていき、一つでも真ならそこで値を返し処理を終了するため、後者の式だと #t が評価された時点で評価が終了し関数 diverge は評価されないため。

(c) ((lambda (n) (\* 3 (\* n n))) x)

課題 1-9 (a) 2 個

課題 1-10 (a) 異なるプログラムのソースコード中での類似または一致した部分。

「コピーとペースト」によるプログラミングや、意図的に同一処理を繰り返して書くことにより、プログラムテキスト中に作りこまれる。

(92 words)

参考: <https://sel.ist.osaka-u.ac.jp/research/clone/index.html.ja>

(b) プログラムのソースコードの複雑度を下げたり無駄な処理を省いたりすることでコードを整理して見やすくし、保守性を高めるという目的で、プログラムの動作を変えないまま、ソースコードの内部構造を整理すること。(99 words)

参考: <https://wa3.i-3-i.info/word1494.html>

<https://ja.wikipedia.org/>

wiki/リファクタリング (プログラミング)