

工学部専門科目
「プログラミング言語」
(その1: 導入)

五十嵐 淳

igarashi@kuis.kyoto-u.ac.jp

京都大学 大学院情報学研究科
通信情報システム専攻

April 10, 2013

本日のメニュー

- 講義導入
- SICP 第1~2章(「アルゴリズムとデータ構造入門」)の復習

事務的情報

- 名前: 五十嵐 淳 (いがらし あつし)
- 所属: 情報学研究科 通信情報システム専攻 計算機ソフトウェア分野
- オフィス: 総合研究7号館 224号室
 - ▶ オフィスアワー: 月曜 17:00 ~ 18:30
- よろず相談窓口: `pl13@fos.kuis....`
 - ▶ TAの井上 裕昭君, 花田 裕一郎君と五十嵐に連絡できます
- 講義 WWW ページ: `http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/class/pl/`

成績評価

- 宿題 40%
 - ▶ 宿題提出システムへの登録が必要 (後述)
- 期末試験 60%
- 随意課題によりさらに加点

講義内容

シラバスより

プログラミング言語についてコンピュータサイエンスの立場から論じる．使用するプログラミング言語は Scheme であり，高度なプログラミングの概念について学ぶとともに，実際にプログラミングに適用することを通じて，プログラミングの本質を習得する．教科書の前半（第 1 章および第 2 章）は「アルゴリズムとデータ構造入門」（奥乃先生，第 1 学年後期担当，91150）で取り上げ，本講義では第 3 章および第 4 章を取り上げる．

第3章: Modularity, Objects, and State

状態変化を伴う部品を使ったシステム記述について

- Modularity(モジュール性)...システム全体を, より小さく, 独立して開発/保守可能な部分システム(モジュール)に「自然に」分割可能であること
- Objects(オブジェクト)...(時間を経て変化する)内部状態を持つシステム部品
- State...状態

第4章: Metalinguistic Abstraction

超言語的抽象について

- プログラミング言語自体 (の意味) の変更による抽象化の実現
 - ▶ Scheme 処理系を Scheme 自身で記述
 - ▶ 処理系を変更して「Schemeの亜種」を作る, 特殊形式を追加する

SICP 第1 ~ 2章の復習

問題 1: 正整数 n に対し, 階乗 $n!$ を返す手続 `factorial` を定義せよ

```
(define (factorial n)
```

```
)
```

問題2: 問題1 で定義した factorial を使って $5!$ を計算する過程を示せ

(factorial 5)

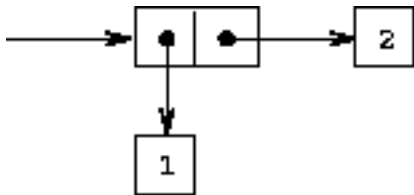
→

問題3: 以下の入力列を与えた時(最後の入力に対する) Scheme 処理系からの応答はどうなるか答えよ.

```
(define (foo x)
  (lambda (y) (* (+ 2 x) y)))
(define x 6)
(define bar (foo 5))
(define x 1)
(bar 4)
```

問題4: 以下の式を評価した結果作られる構造を図示せよ

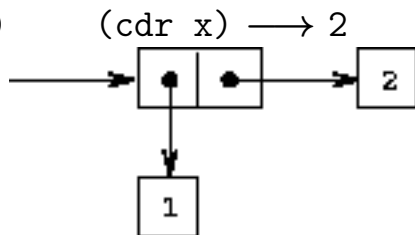
- 例: `(cons 1 2)`



- `(cons 1 nil)`
- `(list 1 2)`
- `(cons (cons 1 2) 3)`
- `(cons 1 (list 2 3))`
- `(list (list 2 3) 4)`

問題5: x を評価した結果, 以下が表示される時, x の指す構造を図示せよ.
また x に `car`, `cdr` を使って 2 を得る方法を示せ.

- 例: (1 . 2)



- (4 3 2)
- (7 (3 . 4) (5 2))
- (((2)))

問題6: 手続き `length` を以下のように定義する。この時、以下の式の値は何か

```
(define (length xs)
  (if (null? xs) 0
      (+ 1 (length (cdr xs)))))
```

- `(length '(4 8 2))`
- `(length '((1 2) 3 (4 5 (6))))`

重要: 宿題提出システムへの登録

以下の要領でメールを pl13@fos.kuis... へ送信

- Subject: 提出システム登録願
- 本文先頭4行に, 氏名, 氏名のふりがな, 学籍番号, メールアドレス, をこの順で明記
- 数日以内に受領の返事が届くはず

To: pl13@fos.kuis.kyoto-u.ac.jp

Subject: 提出システム登録願

五十嵐 淳

いがらし あつし

12345678

igarashi@kuis.kyoto-u.ac.jp

宿題：4/17 午前8時 締切

- 宿題提出システムへの登録
- 教科書 3.1 までを読み，以下の問いについて考えておこう
 - ▶ 「銀行口座」における状態とは何か？
 - ▶ 手続き `withdraw` が今までの手続きとは根本的に異なる挙動を示している，という説明があるが，それはどういうことか？
 - ▶ `withdraw` と `new-withdraw` の違いは何か？
 - ▶ (擬似) 乱数を状態を使って実現することのメリットは何か？
 - ▶ 破壊的代入 (`assignment`) による状態変化を導入することによって何が複雑になるといっているか？