

「プログラミング言語」2012年度中間試験問題 (その1)

答案用紙の1枚目の表に回答.

試験時間 8:45 ~ 10:15/ Jun. 13, 2012, 奥乃

持ち込み一切禁止

Answer each question in a designated area either in Japanese, English, or French. No inquiries are accepted. If you have any query, answer questions by specifying plausible assumptions.

問題ごとに指定したページに回答すること. 日本語, 英語, 仏語で回答してよい.

質問は一切受けない. 問題に不審のある場合にはその旨を明記し, 妥当な仮定を設定して回答すること.

Question 1 — Password-protected Bank Account (銀行口座) Page 1 (1枚目表) に回答

パスワードで保護された銀行口座を作成したい. ただし, 次の2つの要求条件を満たす必要がある.

要求条件 1 手続き 'make-account' は, 初期入金額に加えて, パスワードという引数を取る. つまり, 下記のように口座を作成する:

```
(define acc (make-account 100 'secret-password))
```

作成された口座オブジェクト (account object) は口座作成時に指定されたパスワードと同じパスワードが与えられた時にだけ, request の処理を行う.

```
((acc 'secret-password 'withdraw) 40)
```

```
60
```

```
((acc 'some-other-password 'deposit) 50)
```

```
"Incorrect password"
```

要求条件 2 手続き 'make-account' は, パスワードの間違があると, 下記のような処理を行う.

The above 'make-account' procedure should manage illegal accesses by adding another local state variable so that, if an account is accessed more than seven consecutive times with an incorrect password, it invokes the procedure 'call-the-cops'.

ただし, 正しいパスワードが入力された時の処理も忘れないこと.

以下の問いに答えなさい.

1-1) 次の手続きを完成させなさい. ただし, 完全なプログラム を解答すること.

```
(define (make-account balance password)
  (let ( (??) )
    let*  でもよい

    (define (withdraw amount)
      (??) )
    (define (deposit amount)
      (??) )
    (define (wrongpass m)
      (error "Incorrect password") )
    (define (call-the-cops m)
      (error "Call the cops") )

    (define (dispatch p m)
      (??) )

    dispatch )
```

1-2) balance という状態変数に相当する変数を引数ではなく global 変数として定義する. このとき, 複数の人がそれぞれ各自の銀行口座を作成するときに, どういう不都合が生じる可能性があるか, あなたの考えを述べなさい.

Question 2 — memo-proc Page 2 (1 枚目裏) に回答

‘cons-stream’ ではプロセスの遅延実行のために、2 つの重要な手続き、delay と force が使用される。このとき、下記の問に答えなさい。

- 2-1) ‘(delay <exp>)’ を、‘(memo-proc (lambda () <exp>))’ の syntax sugar として定義する。
‘memo-proc’ 手続きの説明を行い、それが必要な理由について述べなさい。
- 2-2) ‘memo-proc’ 手続きを書きなさい。
- 2-3) ‘force’ 手続きを書きなさい。

Question 3 — Calculate the value of π Page 3 (2 枚目表) に回答

π の具体的な値をストリームで求めるために、まず、手続き ‘partial-sums’ を定義する。この手続きは、引数としてストリーム S (この要素を S_0, S_1, S_2, \dots とする) をとり、次のような要素から構成されるストリームを返す:

$$S_0, S_0 + S_1, S_0 + S_1 + S_2, \dots$$

例: ‘(partial-sums integers)’ should be the stream 1, 3, 6, 10, 15,

では、漸化式で π の値を求める。まず、‘pi-summands’ ストリームを生成する。その要素は、奇数の逆数で、符号を反転したものである。すなわち、 $\frac{1}{1}, -\frac{1}{3}, \frac{1}{5}, -\frac{1}{7}, \dots$

次に、次式を用いて、

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

‘pi-stream’ ストリームの部分和のストリームを生成し (‘partial-sums’ 手続きを使用して)、得られたストリームの各要素を 4 倍して、‘pi-stream’ を求める。

- 3-1) ‘partial-sums’ の完全なプログラムを書きなさい。
- 3-2) ‘pi-summands’ と ‘pi-stream’ の完全なプログラムを書きなさい。

```
(define (partial-sums s)
  (??) )
```

```
(define (pi-summands n)
  (??) )
```

```
(define pi-stream
  (??) )
```

Question 4 — Mutual exclusion Page 4, (2 枚目裏) に回答

排他制御について、次の設問に答えなさい。

- 4-1) 排他制御がなぜ必要となるのか、銀行口座を例に取って説明しなさい。
- 4-2) 排他制御の具体的な方法について知っていることを書きなさい。

Enquete アンケート Page 4 (2 枚目裏) に回答

感想・意見・要望、とくに改善すべき点を書いて下さい。