

課題 1

1029232337 杉本風斗

April 18, 2012

1 課題

実行例はすべてソースコード中にコメントアウトして載せた.

1.1 Ex 3.3

パスワードをたしかめるようにする.

```
1 (define (make-account balance password)
2   (define (withdraw amount)
3     (if (>= balance amount)
4       (begin (set! balance (- balance amount))
5             balance)
6       "Insufficient funds"))
7   (define (deposit amount)
8     (set! balance (+ balance amount))
9     balance)
10  (define (incorrect . a)
11    "Incorrect password")
12  (define (dispatch pass m)
13    (if (eq? password pass)
14        (cond ((eq? m 'withdraw) withdraw)
15              ((eq? m 'deposit) deposit)
16              (else (error "Unknown request"
17                            m))))
18    incorrect))
19  dispatch)
20
21 ; Usage:
22 ; (define acc (make-account 100 'hoge))
```

```

23 ; ((acc 'hoge 'withdraw) 40)
24 ; ((acc 'hoge 'deposit) 50)
25 ; ((acc 'fuga 'withdraw) 10)

```

1.2 Ex 3.4

7回以上パスワードを間違えるとおまわりさんと呼ぶようにする.

```

1 (define (make-account balance password)
2   (let ((incorrect-times 0))
3     (define (withdraw amount)
4       (if (>= balance amount)
5         (begin (set! balance (- balance amount))
6                 balance)
7         "Insufficient funds"))
8     (define (deposit amount)
9       (set! balance (+ balance amount))
10      balance)
11    (define (incorrect . a)
12      "Incorrect password")
13    (define (call-the-cops)
14      (error "Omawarisan, kotti desu!!!!"))
15
16    (define (dispatch pass m)
17      (if (eq? password pass)
18        (cond ((eq? m 'withdraw) withdraw)
19              ((eq? m 'deposit) deposit)
20              (else (error "Unknown request"
21                           m)))
22        (begin (set! incorrect-times (+ incorrect-times 1))
23                (if (> incorrect-times 7)
24                    (call-the-cops)
25                    incorrect))))
26    dispatch))
27
28 ; Usage:
29 ; (define acc (make-account 100 'hoge))
30 ; ((acc 'hoge 'withdraw) 40)
31 ; ((acc 'bar 'deposit) 50)
32 ; ((acc 'bar 'deposit) 50)
33 ; ((acc 'bar 'deposit) 50)
34 ; ((acc 'bar 'deposit) 50)

```

```

35 ; ((acc 'bar 'deposit) 50)
36 ; ((acc 'bar 'deposit) 50)
37 ; ((acc 'bar 'deposit) 50)
38 ; ((acc 'bar 'deposit) 50) ; おまわりさんこっちです！

```

1.3 Ex 3.7

共同預金口座を作れるようにする.
パスワードを認証して元のアカウントに処理を受け渡す.

```

1 (define (make-account balance password)
2   (define (withdraw amount)
3     (if (>= balance amount)
4       (begin (set! balance (- balance amount))
5             balance)
6       "Insufficient funds"))
7   (define (deposit amount)
8     (set! balance (+ balance amount))
9     balance)
10  (define (incorrect . a)
11    "Incorrect password")
12  (define (dispatch pass m)
13    (if (eq? password pass)
14        (cond ((eq? m 'withdraw) withdraw)
15              ((eq? m 'deposit) deposit)
16              ((eq? m 'pass) 'correct)
17              (else (error "Unknown request"
18                          m))))
19    incorrect))
20  dispatch)
21
22 (define (make-joint account password new-password)
23   (define (incorrect . a)
24     "Incorrect password")
25   (define (dispatch pass m)
26     (if (eq? new-password pass)
27         (account password m)
28         incorrect))
29
30   (if (eq? (account password 'pass) 'correct)
31       dispatch
32       (incorrect)))

```

```
33
34 ; Usage:
35 ; (define piyo-acc (make-account 100 'hoge))
36 ; (define poyo-acc (make-joint piyo-acc 'hoge 'fuga))
37 ; (define ugyo-acc (make-joint piyo-acc 'agya 'hunyo)) ; "Incorrect password"
38 ; ((piyo-acc 'hoge 'withdraw) 40)
39 ; ((poyo-acc 'fuga 'deposit) 30)
40 ; ((poyo-acc 'hoge 'withdraw) 20) ; "Incorrect password"
```