

プログラミング言語論

演習5 解答と解説

演習5.1 解答

● 次の式は左辺値を持つか

- (1) (i) ○
- (2) i++ ×
- (3) i+j ×
- (4) a ○
- (5) a[i++] ○
- (6) a[i+j] ○

2

演習5.2 解答

● メインプログラムの実行結果

x = 6

y = 8

3

演習5.2 解説

- $x = f(2) + f(2);$
 - 1回目の関数 f は、 $u=2, v=1$ なので、 $v = v+u = 1+2$ で v が3となり、3をreturnする。
 - 2回目の関数 f を実行時、 v は再度割当てられ、1に初期化される。したがって、同じく3をreturn。
 - x は、 $3+3 = 6$ となる。

4

演習5.2 解説

- $y = g(2) + g(2)$
 - 1回目の g は、 f の場合と同じく、 $v = v+u = 1+2$ で、3をreturnする。
 - 2回目の g では、 v は存続し続けているので、1回目でセットされた3のまま。従って、 $v = v+u = 3+2 = 5$ をreturnする。
 - y は、 $3 + 5 = 8$ となる。

5

演習5.3 (1) 解答

- (1) 値呼出し
 $i = 2$ 、 $a[2] = 3$ 、 $a[3] = 4$

6

演習5.3 (1) 解説

- (1) 値呼出し

➤手続き内で仮引数の値を書き換え
ても、呼び出し側(実引数)には影
響しない



実引数の値はそのまま

7

演習5.3 (2) 解答

- (2) 参照呼出し

$i = 3$ 、 $a[2] = 2$ 、 $a[3] = 4$

8

演習5.3 (2) 解説

- (2) 参照呼出し

➤実引数のアドレス(左辺値)を仮引
数に割り当てる



$x \Leftrightarrow i$ 、 $y \Leftrightarrow a[2]$ として、これを交
換(swap)する

※ $y \Leftrightarrow a[2]$ の割当ては呼び出し時に行う
その後 i が変化しても変わらない

9
