

プログラミング言語論

演習1 解答と解説

演習1.1 解答

●(1) $7 \div 5 \times 2$
 $= (7 \div 5) \times 2$ ← この2項の商
 $= 1.4 \times 2$
 $= 2.8$

●(2) $(10 - 3) \times (25 \div 5) \times 2$
 $= (10 - 3) \times (25 \div 5) \times 2$ ← 2項の積
 $= 7 \times 5 \times 2$
 $= 70$

2

演習1.2 解答

●(1) $x - y \times z$
 $= x - yz$

●(2) $b \times b - 4 \times a \times c$
 $= b^2 - 4ac$

3

演習1.2 解説

- (1) $x - y * z$
 - 全体は、項 x と項 $y * z$ の差
 - 項 $y * z$ は、 $* y z$

$$\begin{array}{c} \Downarrow \\ - x * y z \end{array}$$

4

演習1.2 解説 (続き)

- (2) $b * b - 4 * a * c$
 - 全体は、項 $b * b$ と項 $4 * a * c$ の差
 - 項 $b * b$ は $\Rightarrow * b b$
 - 項 $4 * a * c$ は、項 $4 * a$ ($\Rightarrow * 4 a$) と項 c の積 $\Rightarrow * * 4 a c$

$$\begin{array}{c} \Downarrow \\ - * b b * * 4 a c \end{array}$$

5

演習1.3 解答

- (1) $x - y * z$
 $x y z * -$
- (2) $b * b - 4 * a * c$
 $b b * 4 a * c * -$

6

演習1.2、1.3 解説

- (2) の $b * b - 4 * a * c$ は、
 - $* b b * 4 * a c$ (前置)
 - $b b * 4 a c * * -$ (後置)
 としない。
- 四則演算は左結合、つまり
 - $4 * a * c$ は $(4 * a) * c$ であり、
 - $4 * (a * c)$ ではない。

7

演習1.2、1.3 解説 (続き)

- 前置記法 \Leftrightarrow 中置記法 \Leftrightarrow 後置記法
の変換を行っても、項(変数)の順序
は変わらない
- $+ a b \Leftrightarrow a + b \Leftrightarrow a b +$
- 項と演算子の位置関係が変わる
だけである

8

演習1.4 解答

- 解答例(1) -- Java

```
public static int fib (int n)
{
    if (n <= 0)
        return 0;           // fib(0)=0
    else if (n == 1)
        return 1;          // fib(1)=1
    else
        return fib(n-1) + fib(n-2);
}
```

9

演習1.4 解答 (続き)

- 解答例(2) -- C言語

```
int fib (int n)
{
    if (n <= 0)
        return 0;        // fib(0)=0
    else if (n == 1)
        return 1;        // fib(1)=1
    else
        return fib(n-1) + fib(n-2);
}
```

10

演習1.4 解説

- 再帰を使うのは、演習のため
 - 実際には、フィボナッチ数の計算では、再帰を使わない方がよい
 - 再帰が適しているデータ構造もある



Webサイトの資料「木構造」参照

11

演習1.4 解説 (続き)

- フィボナッチ数を求めるプログラムの非再帰版を、Webサイトに掲載しておいた
 - 再帰版の時間計算量は $O(\text{fib}(n))$ であるが、非再帰版では $O(n)$ である

12

演習1.5 解答

- (1) -1

1111 1111 1111 1111

(0xFFFF)
- (2) 10

0000 0000 0000 1010

(0xA)
- (3) -10

1111 1111 1111 0110

(0xFFF6)

13

演習1.5 解答 (続き)

- (4) 255

0000 0000 1111 1111

(0x00FF)
- (5) -255

1111 1111 0000 0001

(0xFF01)

14

演習1.6 解説

- 1バイト系文字コード
 - ASCII (アメリカの規格、7ビット)
 - 英数字、記号、制御記号
 - JISコード (JIS X0201)
 - 英数字、記号、カタカナ、制御記号
 - ASCIIをほぼそのまま含む
 - EBCDIC (IBM)
 - メインフレームで使用

15

演習1.6 解説 (続き)

- 多バイト系文字コード (漢字コード)
 - EUCコード
 - UNIXシステムで使用
 - Unicode
 - 世界中の文字を統一的に扱うため提案された
 - UTF-7、UTF-8、UTF-16等の符号化方式がある

16

演習1.6 解説 (続き)

- JIS漢字コード (JIS X0208)
 - 第1水準、第2水準、補助漢字などがある
- シフトJISコード
 - Windowsなどで使用されている

17

演習1.7 解答

- 関係「 \geq 」は、
 - 反射的である
 - $a \geq a$ は、常に成り立つ
 - 推移的である
 - $a \geq b$ 、 $b \geq c$ ならば $a \geq c$
 - 対称的ではない
 - $a \geq b$ でも、 $b \geq a$ とは限らない

18
