

錬成問題

- 'A'のように単一引用符'で文字を囲んだ式を、と呼ぶ。その型は型である。
- 演算子 *=, /=, %=, += のように、演算子と代入という二つの働きをもった演算子の総称は、である。
- 演算子 ++ には2種類がある。++aのようにオペランドの左側に置いて利用する演算子の名称が、増分演算子であり、a++のようにオペランドの右側に置いて利用する演算子の名称が、増分演算子である。
- 繰返しのことをカタカナ3文字でと呼び、入れ子構造となっている繰返しのことを と呼ぶ。
- 2項演算子による演算では、まず左オペランドがされて、それから右オペランドがされて、最後に演算が行われる。
- 以下に示す各プログラム部分の実行結果を示せ。

```
int a = 135;
System.out.println("a *= 10 : " + (a *= 10));
System.out.println("a /= 10 : " + (a /= 10));
System.out.println("a %= 10 : " + (a %= 10));
System.out.println("a += 10 : " + (a += 10));
System.out.println("a -= 10 : " + (a -= 10));
```

```
a *= 10 : 
a /= 10 : 
a %= 10 : 
a += 10 : 
a -= 10 : 
(9)
```

```
int a = 1, b = 3, c = 5;
System.out.println("a++ : " + (a++));
System.out.println("++a : " + (++a));
System.out.println("b-- : " + (b--));
System.out.println("--b : " + (--b));
System.out.println("c : " + (c = a++ + --b));
```

```
a++ : 
++a : 
b-- : 
--b : 
c : 
(10)
```

- 以下に示すのは、変数 n に対して、キーボードから偶数のみを読み込むためのプログラムである（偶数でない値が入力された場合は再入力させる）。
※変数 `stdin` は、本文 p.26 で解説したとおりに宣言されているものとする（次章以降の錬成問題でも同様である）。

```
int n;
 {
    System.out.println("偶数 : ");
    n = stdin.nextInt();
}  (
```

- 以下に示すのは、1 から n までの値（たとえば n が 5 であれば、1, 2, 3, 4, 5）を、1 行に 1 個ずつ表示するプログラムである。

```
int i = 1;
(14) (i <= n) {
    System.out.println((15));
    (16)++;
}
```

```
(17) (int i = 1; i <= n; (16)++)
    System.out.println((15));
```

- 以下に示すのは、 n 以下のすべての奇数（たとえば n が 8 であれば、1, 3, 5, 7）を小さいほうから順に 1 行に 1 個ずつ表示するプログラムである。

```
int i = 1;
(14) (i <= n) {
    System.out.println((18));
    i (19) 2;
}
```

- 以下に示すのは、 n 以下の 10 の倍数（たとえば n が 45 であれば、10, 20, 30, 40）を小さいほうから順にすべて 1 行に 1 個ずつ表示するプログラムである。

```
(20) (int i = (21); i <= n / 10; i (22))
    System.out.println((23));
```

- 以下に示すのは、3 桁の整数を当てさせる数当てゲームである。正解するまでキーボードから整数を読み込み、何回で当たったのかを最後に表示する。

```
Random rand = new Random();
Scanner stdIn = new Scanner((24));

int no = (25) + rand.nextInt((26)); // 当てるべき数 (100~999)

System.out.println("数当てゲーム開始!!");
System.out.println("100~999の数を当ててください。");

int x;           // プレーヤーが入力した値
int count = 0;  // 入力回数

(27) {
    System.out.print("いくつかね：");
    x = stdIn.nextInt();
    count (28);

    if (x > no)
        System.out.println("もっと小さな数だよ。");
    else if (x < no)
        System.out.println("もっと大きな数だよ。");
} (29) (x != no);

System.out.println("正解です。");
System.out.println((30) + "回で当たりましたね。");
```

▪ **if** 文と **switch** 文の総称は (31) である。また、**do** 文と **while** 文と **for** 文の総称は (32) である。

▪ **do** 文では、ループ本体は (33)。**while** 文では、ループ本体は (34)。**for** 文では、ループ本体は (35)。

▶ 共通の選択肢：(a)必ず一度は実行される (b)一度も実行されないことがある

▪ 以下に示すのは、変数 n に対して、キーボードから3桁の奇数のみを読み込むためのプログラムである（そうでない値が入力された場合は再入力させる）。

```
int n;
do {
    System.out.println("3桁の奇数：");
    n = stdin.nextInt();
} while (!( (36) ));
```

▪ 以下に示すのは、 n から1までの値（たとえば n が5であれば、5, 4, 3, 2, 1）を、1行に1個ずつ表示するプログラムである。

```
int i = 1;
while (i <= n) {
    System.out.println( (37) );
    (38) ++;
}
```

```
for (int i = 1; i <= n; (38) ++)
    System.out.println( (37) );
```

```
int i = 0;
while (i < n) {
    System.out.println( (39) );
    (40) ++;
}
```

```
for (int i = 0; i < n; (40) ++)
    System.out.println( (39) );
```

```
int i = n;
while ( (41) ) {
    System.out.println(i);
    (42) --;
}
```

```
for (int i = n; (41); (42) --)
    System.out.println(i);
```

▪ 以下に示すのは、2の n 乗（たとえば n が3であれば8）を求めて表示するプログラムである。

```
int x = (43);
while ( (44) > 0)
    x (45) 2;
System.out.println("2の" + n + "乗は" + x + "です。");
```

```
int x = (46);
for (int i = 1; i <= (47); i++)
    x (48) 2;
System.out.println("2の" + n + "乗は" + x + "です。");
```

- 以下に示すのは、記号文字 '-' を n 個連続して表示するプログラムである。

```
int i = 0;
(49) {
    System.out.print('-');
    i++;
} while ((50));
```

```
int i = 1;
(49) {
    System.out.print('-');
    i++;
} while ((51));
```

```
int i = 0;
while ((52)) {
    System.out.print('-');
    i++;
}
```

```
int i = 1;
while ((53)) {
    System.out.print('-');
    i++;
}
```

```
while ((54) >= 1)
    System.out.print('-');
```

```
while ((55) > 0)
    System.out.print('-');
```

```
for (int i = 0; (56); i++)
    System.out.print('-');
```

```
for (int i = 1; (57); i++)
    System.out.print('-');
```

- 以下に示すのは、0 と 1 を交互に合計 n 個連続して表示する（たとえば n が 5 であれば『01010』と表示する）プログラムである。

```
for (int i = 0; i < (58); i++)
    System.out.print((59));
```

```
for (int i = 1; i <= (60); i++)
    System.out.print((61));
```

- 以下に示すのは、1234567890...の数字を循環的に合計 n 個連続して表示する（たとえば n が 15 であれば『123456789012345』と表示する）プログラムである。

```
for (int i = 1; i <= n; i++)
    System.out.print((62));
```

- 以下に示すのは、0123456789...の数字を循環的に合計 n 個連続して表示する（たとえば n が 15 であれば『012345678901234』と表示する）プログラムである。

```
for (int i = 0; i < n; i++)
    System.out.print((63));
```

```
for (int i = 1; i <= n; i++)
    System.out.print((64));
```

▪ do文が行うのは (65) 判定繰返しであり、while文とfor文が行うのは (66) 判定繰返しである。

▪ 以下に示すプログラム部分の実行結果を示せ。

```
for (int i = 1; i < 3; i++);
    System.out.print("こんにちは。");
```

(67)

▪ 以下に示すのは、正の整数値 n の全桁の合計を変数 sum に求めるプログラムである (たとえば n が 513 であれば、 sum の値は $5 + 1 + 3$ すなわち 9 となる)。

```
sum = 0;
while (n > (68)) {
    sum += (69);
    n /= (70);
}
```

▪ 以下に示すのは、1 から n までの全整数の積を $fact$ に求めるとともに、その計算式と結果を表示するプログラムである (たとえば n が 5 であれば、 $fact$ の値を 120 にするとともに、『 $1 * 2 * 3 * 4 * 5 = 120$ 』と表示する)。なお、 n は 1 以上の正の整数値が与えられているものとする。

```
int fact = (71);
for (int i = 1; i < (72); i++) {
    System.out.print((73));
    fact (74) i;
}
System.out.print((75));
fact (76) n;
System.out.println((77));
```

▪ 以下に示すのは、九九の表を出力するプログラムである。

```
System.out.print(" |");
for (int i = 1; i <= 9; i++)
    System.out.print((78));
System.out.print("\n---");
for (int i = 1; i <= (79); i++)
    System.out.print('-');
System.out.println();
for (int i = 1; i <= (80); i++) {
    System.out.print((81));
    for (int j = 1; j <= (82); j++) {
        int t = (83);
        if (t < 10)
            System.out.print((84));
        else
            System.out.print((85));
        System.out.print(t);
    }
    System.out.println();
}
```

		1	2	3	4	5	6	7	8	9
1		1	2	3	4	5	6	7	8	9
2		2	4	6	8	10	12	14	16	18
3		3	6	9	12	15	18	21	24	27
4		4	8	12	16	20	24	28	32	36
5		5	10	15	20	25	30	35	40	45
6		6	12	18	24	30	36	42	48	54
7		7	14	21	28	35	42	49	56	63
8		8	16	24	32	40	48	56	64	72
9		9	18	27	36	45	54	63	72	81

- 以下に示すのは、記号文字 '*' を並べて、 $height$ 行 $width$ 列の長方形を表示するプログラムである（実行例に示すのは $height$ が 4 で $width$ が 9 の場合の出力）。

```
for (int i = 0; i < (86); i++) {
    for (int j = 0; j < (87); j++)
        System.out.print('*');
    System.out.println();
}
```

```
*****
*****
*****
*****
```

- 以下に示すのは、記号文字 '*' を並べて、左下側が直角で一辺の長さが n である二等辺三角形を表示するプログラムである（実行例に示すのは n が 4 の場合の出力）。

```
for (int i = 0; i < (88); i++) {
    for (int j = 0; j < (89); j++)
        System.out.print('*');
    System.out.println();
}
```

```
*
**
***
****
```

- 以下に示すのは、記号文字 '*' を並べて、左上側が直角で一辺の長さが n である二等辺三角形を表示するプログラムである（実行例に示すのは n が 4 の場合の出力）。

```
for (int i = (90); i >= 0; i--) {
    for (int j = 0; j < (91); j++)
        System.out.print('*');
    System.out.println();
}
```

```
****
***
**
*
```

- 以下に示すのは、記号文字 '*' を並べて、右下側が直角で一辺の長さが n である二等辺三角形を表示するプログラムである（実行例に示すのは n が 5 の場合の出力）。

```
for (int i = 0; i < (92); i++) {
    for (int j = 0; j < (93); j++)
        System.out.print(' ');
    for (int j = 0; j < (94); j++)
        System.out.print('*');
    System.out.println();
}
```

```
*
**
***
****
*****
```

- 以下に示すのは、記号文字 '*' を並べて、右上側が直角で一辺の長さが n である二等辺三角形を表示するプログラムである（実行例に示すのは n が 5 の場合の出力）。

```
for (int i = (95); i >= 0; i--) {
    for (int j = 0; j < (96); j++)
        System.out.print(' ');
    for (int j = 0; j < (97); j++)
        System.out.print('*');
    System.out.println();
}
```

```
*****
****
***
**
*
```

■ 以下に示すのは、記号文字 '+' を、 w 個ごとに改行しながら、計 n 個表示するプログラムである (n と w は正の整数値が与えられるものとする)。最後の '+' の出力の後に改行文字を 1 個だけ出力するものとし、2 個以上出力してはならない (実行例に示すのは n が 15 で w が 6 の場合の出力)。

```
for (int i = 1; i <= n; i++) {
    System.out.print('+');
    if ( (100) )
        System.out.println();
}
if ( (101) )
    System.out.println();
```

```
++++++
++++++
+++
```

■ 以下に示すのは、int 型整数 n の値を 15 桁の幅で右よせに表示するプログラムである。(実行例に示すのは n が 1,234,567,890 の場合と、 n が -1,234,567,890 の場合である。なお、□ は空白文字である)。

```
int d = ( (102) ) ? 1 : 0;
int x = ( (103) ) ? n : -n;
while (x > 0) {
    d (104);
    x (105) 10;
}
for (int i = 0; i < (106); i++)
    System.out.print(' ');
System.out.print( (107) );
```

```
1234567890
```

```
-1234567890
```

■ 以下に示すのは、記号文字 '+' を、 w 個ごとに改行しながら、計 n 個表示するプログラムである。表示においては、3 行おきに空の行を出力する。最後の '+' の出力の後に改行文字を 1 個だけ出力するものとし、2 個以上出力してはならない (n と w は正の整数値が与えられるものとする)。実行例に示すのは n が 45 で w が 6 の場合の出力)。

```
int p = (108);
int q = (109);
for (int i = 1; i <= (110); i++) {
    for (int j = 1; j <= w; j++)
        System.out.print('+');
    System.out.println();
    if (i % 3 == 0 && (111))
        System.out.println();
}
for (int i = 1; i <= (112); i++)
    System.out.print('+');
if (q != 0) System.out.println();
```

```
++++++
++++++
++++++

++++++
++++++
++++++

++++++
+++
```

