

錬成問題

- 以下に示すのは、整数型の各型で表現できる値の範囲とビット数である。

型	表現できる値の範囲	ビット数
char	(1) ~ (2)	(3)
byte	(4) ~ (5)	(6)
short	(7) ~ (8)	(9)
int	(10) ~ (11)	(12)
long	-9223372036854775808 ~ 9223372036854775807	(13)

- 整数リテラルには3種類の基数表記がある。基数の小さいほうから順に、(14) 進整数リテラル、(15) 進整数リテラル、(16) 進整数リテラルである。

0は(17) 進整数リテラル、01は(18) 進整数リテラル、10は(19) 進整数リテラル、010は(20) 進整数リテラル、0x1は(21) 進整数リテラルである。

- 整数リテラルの型は基本的には **int** 型である。**long** 型とするためには、(22) または (23) の整数接尾語を末尾に付ける。

- 浮動小数点型には **double** 型と **float** 型とがある。これらの型の表現範囲は、大きさと (24) が異なる。なお、前者は(25) ビットで、後者は(26) ビットである。

- 浮動小数点リテラルは基本的には **double** 型である。**double** 型であることを明示するために、(27) または (28) の浮動小数点接尾語を末尾に付けてもよい。なお、**float** 型とするためには、(29) または (30) の浮動小数点接尾語を末尾に付ける。

- 2項の算術演算では、オペランドに対して(31) と呼ばれる型変換が自動的に適用される。

- 基本型の(32) 変換では、定数による初期化・定数の代入は例外ではあるものの、原則として明示的な型変換が必要であるのに対し、基本型の(33) 変換では、型変換が自動的に行われる。

- 文字\で始まる文字の並びによって、単一の文字を表す表記が(34) である。各文字を表す(34) は以下のようになっている。

書式送り … (35)	後退 … (36)
復帰 … (37)	水平タブ … (38)
改行 … (39)	逆斜線 … (40)
単一引用符 … (41)	二重引用符 … (42)

- 以下に示す各プログラム部分の実行結果を示せ。

```
System.out.println( 15);
System.out.println( 015);
System.out.println(0x15);
```

(43)

```
System.out.println(15 / 2 );
System.out.println(15.0 / 2.0);
System.out.println(15.0 / 2 );
System.out.println(15 / 2.0);
```

(44)

```
System.out.println((double)15 / 2);
System.out.println(15 / (double)2);
```

(45)

```
System.out.println((double)0);
System.out.println((int)3.14);
```

(46)

```
System.out.println(true == true);
System.out.println(true == false);
```

(47)

```
System.out.printf("%o\n", 111);
System.out.printf("%d\n", 111);
System.out.printf("%x\n", 111);
System.out.printf("%X\n", 111);
```

(48)

```
System.out.printf("%d\n", 12345);
System.out.printf("%3d\n", 12345);
System.out.printf("%7d\n", 12345);
System.out.printf("%5d\n", 123);
System.out.printf("%05d\n", 123);
```

(49)

```
System.out.printf("%8.1f\n", 5.4321);
System.out.printf("%8.2f\n", 5.4321);
System.out.printf("%8.3f\n", 5.4321);
System.out.printf("%8.4f\n", 5.4321);
```

(50)

```
System.out.print( "\\//%/\n");
System.out.printf("\\//%/\n");
```

(51)

```
System.out.print("ABCDEFGH");
System.out.print("\r");
System.out.print("12345");
```

(52)

```
System.out.print("ABCDEFGH");
System.out.print("\b");
System.out.print("12345");
```

(53)

▪ 8進整数リテラルの先頭は (54) で始まり、必ず (55) 桁以上で表記する。16進整数リテラルの先頭は (56) または (57) で始まり、桁数は任意である。

▪ 整数リテラルは (58) の基数表現ができ、浮動小数点リテラルは (59) の基数表現ができる。

- ▶ 共通の選択肢: (a) 8進数と10進数と16進数 (b) 10進数と16進数
(c) 2進数と8進数と10進数と16進数

▪ 以下に示すリテラルの型を示せ。

5 ... (60) 5L ... (61) 5D ... (62) 5F ... (63)

▪ 演算子 () は、オペランドの値を、任意の型で表現した値に変換する演算子であり、その名称は (64) 演算子である。

▪ 二項数値昇格とは、二項の算術演算に適用される、以下に示す型変換である。

- 一方のオペランドが (65) 型ならば、他方を (65) 型に変換する。
- そうでなく、一方のオペランドが (66) 型ならば、他方を (66) 型に変換する。
- そうでなく、一方のオペランドが (67) 型ならば、他方を (67) 型に変換する。
- そうでなければ、両オペランドを (68) 型に変換する。

▪ 以下に示すプログラムについて、コンパイルエラーとなる行には×を、エラーとならない行には○を記入せよ。

(69)	<code>int a = 1;</code>
(70)	<code>int b = 1L;</code>
(71)	<code>int c = 3.14;</code>
(72)	<code>short d = 1;</code>
(73)	<code>short e = a;</code>
(74)	<code>float d = 1L;</code>
(75)	<code>float e = 3.14;</code>
(76)	<code>double x = 1;</code>
(77)	<code>double x = 3.14;</code>
(78)	<code>double x = 3.14D;</code>
(79)	<code>double x = 3.14F;</code>

▪ 以下に示すのは、『"ABC"』と表示するプログラムである (『』は表示しないが、"は表示する: 以下同様)。

```
System.out.print((80));
```

▪ 以下に示すのは、単一引用符を n 個連続して表示するプログラムである。

```
for (int i = 0; i < (81); i++)
    System.out.print(' (82)');
```

- 以下に示すのは、キーボードから読み込んだ文字列 `***` を、『あなたは `***` と入力しましたね。』と表示するプログラムである。

```
String s = stdin.next();
System.out.println(" (83) + s + (84) );
```

```
String s = stdin.next();
System.out.printf(" (85) , s);
```

- 以下に示すのは、パーセント記号 `%` を n 個連続して表示するプログラムである。

```
for (int i = 1; i <= (86); i++)
    System.out.print(" (87) ");
```

```
for (int i = 0; i < (88); i++)
    System.out.printf(" (89) ");
```

- 以下に示すのは、逆斜線文字を並べて、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 < (90); i++) {
    for (int j = 0; j < (91); j++)
        System.out.print(" (92) ");
    System.out.println();
}
```

```
\
 \\
  \ \
   \ \ \
    \ \ \ \
```

- 以下に示すのは、`int` 型変数 x を y で割った商と剰余を少なくとも 4 桁の幅で、計算式を含めて表示するプログラムである（実行例に示すのは x が 15 で y が 2 の場合の出力であり、実行例の `□` は空白文字）。

```
System.out.printf(" (93) = (94) ", x / y);
System.out.printf(" (95) = (96) ", x % y);
```

```
x / y = □□□7
x % y = □□□1
```

- 以下に示すのは、`int` 型変数 x の値を 8 進数、10 進数、16 進数で表示するプログラムである（実行例に示すのは x が 63 の出力）。

```
System.out.printf(" (97) ", x);
System.out.printf(" (98) ", x);
System.out.printf(" (99) ", x);
```

```
63
77
3F
```

- 以下に示すのは、`double` 型変数 x の値を、全体を少なくとも 10 桁、小数点以下を 5 桁で表示するプログラムである（実行例に示すのは x が 3.14159265 の出力）。

```
System.out.printf(" (100) ", x);
```

```
□□□3.14159
```