

format メソッド

第9章の日付クラスでは、年月日をそれぞれ4桁・2桁・2桁で表した文字列（たとえば"2010年10月01日(金)"）を返却するメソッドを作成しました（p.318）。その際に利用したのが、`String` クラスに所属するクラスメソッド `format` です。

このメソッドは、`System.out.printf` の出力先を、コンソール画面から文字列に変えたものです。 **List 15-8** に示すプログラム例で理解を深めましょう。

List 15-8

Chap15/StringFormat1.java

```
// String.formatメソッドによる文字列の作成

class StringFormat1 {

    public static void main(String[] args) {
        String s1 = String.format("%5d", 123);
        String s2 = String.format("%9.3f", 123.45);

        System.out.println("s1 = " + s1);
        System.out.println("s2 = " + s2);
    }
}
```

実行結果

```
s1 =   123
s2 = 123.450
```

二つの文字列 `s1` と `s2` を作って表示するだけのプログラムです。`s1` は " `123`" となり、`s2` は "`123.450`" となります（はスペースです）。

`String.format` や `System.out.printf` は、C 言語の `printf` 関数を模して作られているため、`%d` や `%f` などの指定方法は、C 言語とおおむね同じです。

▶ C 言語の《関数》は、Java の《メソッド》に相当します。

しかし、まったく異なる点もあります。その一つが、C 言語の `printf` では書式化における“桁数”を可変値として指定できるのに対し、Java ではできないことです。C 言語で、桁数を可変値とした例を以下に示します。

```
/* 注：これはC言語のプログラムです。*/
int i;
for (i = 1; i <= 4; i++) {
    printf("%*d\n", i, 5); /* 整数値5をi桁で表示 */
}
```

```
5
 5
  5
   5
```

この `for` 文は、`i` の値を1から4まで増やします。その過程で行うことは、整数値5を少なくとも `i` 桁の幅で表示することです。書式文字列 `"%*d"` 中の `*` が桁数を表す引数 `i` に対応して、`d` が表示すべき整数値5に対応します。

Java の `printf` や `format` では、`*` は使えません。同等なことを実現するには、`"%1d\n"`、`"%2d\n"`、`"%3d\n"`、`"%4d\n"` という文字列を作成し、それを `printf` メソッドに渡す必要があります。そのように実現したプログラムを、以下に示します。

```

for (int i = 1; i <= 4; i++) {
    String f = String.format("%d%d\n", i); // "%id"を作る (iは数値)
    System.out.printf(f, 5);
}

```

`String.format` によって文字列を生成する様子を示したのが **Fig.15-6** です。`%` は `%` となり、黒網部の `%d` の部分に整数値が入ります。たとえば変数 `i` の値が 2 であれば、作られる文字列は `"%2d\n"` となります。

- ▶ `printf` による出力時には、連続する 2 個の `%` が、単一の `%` に変換されます (p.136)。`String.format` も同じです。

続く `printf` メソッドの呼出しでは、生成した文字列を利用して整数値 5 を表示します。

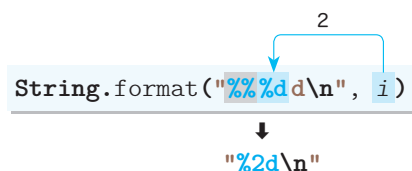


Fig.15-6 `String.format`による書式文字列の作成 (`i`が2のとき)

なお、生成した文字列を、わざわざ変数 `f` に代入することなく、直接 `printf` メソッドに渡すと、プログラムは簡潔になります。

そのように実現したのが、**List 15-9** に示すプログラムです。

List 15-9

Chap15/StringFormat2.java

```

// String.formatメソッドによる書式文字列の作成
class StringFormat2 {
    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            System.out.printf(String.format("%d%d\n", i), 5);
        }
    }
}

```

実行結果

```

5
5
5
5

```

演習 15-4

浮動小数点数値 `x` を、小数点以下の部分を `p` 桁で、全体を少なくとも `w` 桁で表示するメソッド `printDouble` を作成せよ。

```
printDouble(double x, int p, int w)
```