

情報処理概論

(Basic Theory of Information Processing)

第 8 回：配列を応用した Java プログラムの作成

-
- `System.out.printf()`
 - 素数の計算
 - 選挙結果の選択ソート

12 配列を応用した Java プログラムの作成

12.1 System.out.printf()

- System.out.printf() は、様々な値をフォーマットを指定して出力することができる。
- 引数の与え方は次の通りである。

System.out.printf(フォーマットを表す文字列, 値1, ..., 値N);

値の個数Nはフォーマットによって定まる。

- 基本的には最初の引数の「フォーマットを表す文字列」を表示する。
- その文字列の中に、引数の値を文字列に変換したものを埋め込んで表示する。
- 埋め込む内容は、2番め以降の引数として順番に与える。
- フォーマットを表す文字列の中で、%で始まる文字列で引数の値を埋め込むこと、および、その値を文字列に変換するためのフォーマットを表わす。その例として、以下のようなものがある。
 - %d は、整数を 10 進数表示で埋め込む。
 - %04x は、整数を 4 桁の 16 進数表示で埋め込む。引数の値が 0x0123 の場合は、0123 と表示される。
 - %f は、浮動小数点 (float や double 型) の値を固定小数点の数として埋め込む。
 - %e は、浮動小数点の値を固定小数点の数として埋め込む。
 - %c は、文字あるいは文字列を埋め込む。

- 上の例では，文字列として，"0x"，i を0つきの4桁の16進数で表した文字列，" が表わす文字は "，c が表す文字，" \n"(改行)をつなげて表示する。これを少し変えた例として，

```
System.out.printf("0x%04x が表わす文字は %c    %d    %f    %e\n",  
                  23665, '山', 200, 20.4, 0.00002 );
```

を実行すると，

```
0x5c71 が表わす文字は 山    200    20.400000    2.000000e-05  
が表示される。
```

```
public class Printf {  
    public static void main(String[] args) {  
        int i = 24, j = -24;  
        System.out.printf("I1: i = %d    j = %d\n", i, j);  
        System.out.printf("I2: i = %6d    j = %6d\n", i, j);  
        System.out.printf("I3: i = %06d    j = %06d\n", i, j);  
        System.out.printf("I4: i = %+6d    j = %+6d\n", i, j);    // + も付ける  
        System.out.printf("I5: i = %+06d    j = %+06d\n", i, j); // + も付ける  
        System.out.printf("I6: i = %12d    j = %12d\n", i, j);  
        System.out.printf("I7: i = %-12d    j = %-12d\n", i, j); // 左詰  
        System.out.printf("I8: i = %x    j = %x\n", i, j);    // 16進数で表す
```

```

System.out.printf("I9: i = %08x  j = %08x\n", i, j);
System.out.printf("IA: i = %010x  j = %010x\n", i, j);
System.out.printf("IB: i = %04x  j = %04x\n", i, j);
    // 上は, 実際より桁を短く設定すると無視される例
System.out.printf("IC: i = %o  j = %o\n", i, j);          // oは8進数
System.out.printf("ID: i = %011o  j = %011o\n", i, j);    // oは8進数

long l = 123456789012345L, m = -123456789012345L;
System.out.printf("L1: l = %d  m = %d\n", l, m);

double x = 1.2345e-3, y = -1.2345e+3;
System.out.printf("D1: x = %f  y = %f\n", x, y);
System.out.printf("D2: x = %12.5f  y = %12.5f\n", x, y);
System.out.printf("D3: x = %012.5f  y = %012.5f\n", x, y);
System.out.printf("D4: x = %+12.5f  y = %+12.5f\n", x, y);
System.out.printf("D5: x = %+012.5f  y = %+012.5f\n", x, y);
System.out.printf("D6: x = %6.5f  y = %6.5f\n", x, y);
System.out.printf("D7: x = %e  y = %e\n", x, y);
System.out.printf("D8: x = %12.5e  y = %12.5e\n", x, y);
System.out.printf("D9: x = %+014.5e  y = %+014.5e\n", x, y);

```

```

char yama ='山';
System.out.printf("C1: yama = %c\n", yama);

String str1 = "おーい", str2 = "伊藤園", str3 = "ベノア";
System.out.printf("S1: %s , お茶。お茶といったら%s。 \n",str1, str2);
System.out.printf("S2: %10s ,   お茶。お茶といったら%10s。 \n",str1, str3);
}
}

```

実行結果

```

I1: i = 24   j = -24
I2: i =      24   j =      -24
I3: i = 000024   j = -00024
I4: i =      +24   j =      -24
I5: i = +00024   j = -00024
I6: i =          24   j =          -24
I7: i = 24          j = -24
I8: i = 18   j = ffffffff8
I9: i = 00000018   j = ffffffff8

```

IA: i = 0000000018 j = 00ffffffe8
IB: i = 0018 j = fffffffe8
IC: i = 30 j = 37777777750
ID: i = 00000000030 j = 37777777750
L1: l = 123456789012345 m = -123456789012345
D1: x = 0.001235 y = -1234.500000
D2: x = 0.00123 y = -1234.50000
D3: x = 000000.00123 y = -01234.50000
D4: x = +0.00123 y = -1234.50000
D5: x = +00000.00123 y = -01234.50000
D6: x = 0.00123 y = -1234.50000
D7: x = 1.234500e-03 y = -1.234500e+03
D8: x = 1.23450e-03 y = -1.23450e+03
D9: x = +001.23450e-03 y = -001.23450e+03
C1: yama = 山
S1: おーい, お茶。お茶といったら伊藤園。
S2: おーい, お茶。お茶といったら ベノア。

12.2 System.out.print()

引数の文字列をそのまま表示する。System.out.println()との違いは、表示したあとに改行しないことである。

```
public class Print {  
    public static void main(String[] args) {  
        System.out.println("こんにちは、");  
        System.out.println("世界。");  
        System.out.print("こんにちは、");  
        System.out.println("赤ちゃん。");  
    }  
}
```

の表示は、以下ようになる。

```
こんにちは、  
世界。  
こんにちは、 赤ちゃん。
```

12.3 配列を使って，素数を求めるプログラム

- 今まで求めた素数を配列に格納し，配列から取り出して割り切れないことを確かめる。
- 上の考えに基づいて，小さいものから順番に，10000個素数を表示するプログラムを作成せよ。

```
public class PrimeArray {  
    public static void main(String[] args) {  
        int N = 10000, nPrime = 0, primeCand = 2, k;  
        int primeArray[] = new int[N];
```

(素数を求めるプログラムを書く)

```
        for (k = 0 ; k < nPrime ; ++k) {  
            System.out.println((k + 1) + "番目の素数は，"  
                                + primeArray[k] + "です。");  
        }  
    }  
}
```

- 素数かどうか調べる数の平方根以下の素数で割りきれないことを確かめれば，判定できるので，もう少し計算量を下げることができる。この考えに基づいて，10000個素数を表示するプログラムを作成せよ。
- このとき，doubleの平方根を計算する関数`Math.sqrt()`を使っても良いが，使わなくても類似のことはできる。
- 適当な数の素数を計算するプログラムを作成し，両者の実行時間の差を調べよ。このとき，表示は最後の素数だけを表示するようにすること。

12.3.1 選択ソートのプログラム

- 整数を降順(大きいものが先)に並び替える(授業では昇順に並び替えた)。

```
public class SortAsc {  
    public static void main(String[] args) {  
        int data[] = {1, 3, 5, -2, -3, 6, -9};  
  
        for (int k = 0 ; k < data.length ; ++k) {  
            System.out.println((k + 1) + "番目に大きい値は ," + data[k] + "です。");  
        }  
    }  
}
```

- 次の配列で表された選挙結果を，得点の大きい順番に降順に並び替え，名前と年齢と共に表示する **プログラムを作成する**。

```
String name[] = {"柏木由紀  ", "小嶋陽菜  ", "指原莉乃  ", "島崎遥香  ",  
                "須田亜香里", "高橋みなみ", "松井珠理奈", "松井玲奈  ",  
                "山本彩      ", "渡辺麻友  "};  
int age[]      = {22, 26, 21, 20, 22,  
                  23, 17, 22, 20, 20};  
int point[] = {104364, 62899, 141954, 67591, 48182,  
               57388, 90910, 69790, 67916, 159854};
```

- 次に年齢が若い人，年齢が同じ場合は，得票が多い方が先になるように並べるプログラムを作成する。

12.4 今日の終わりに

- 家に帰って再度プログラミングして見よう。