

7.5 おまけ

家に帰ってゴロ寝を決め込んでいる人には余計なお世話だろうけど、任意桁数 (←ちょっとウソ) の円周率を計算するコードである。

programming list [Pivalue.java]

```

1: import java.text.DecimalFormat;
2:
3: public class Pivalue {
4:     static int    uSUP, vSUP, ARR;
5:     static int[]  p, s, t;
6:
7:     public static void main(String[] args) {
8:         ARR = (Integer.parseInt(args[0]) + 3) / 4 + 2;
9:         p = new int[ARR+1];
10:        s = new int[ARR+1];
11:        t = new int[ARR+1];
12:        uSUP = (int)((Integer.parseInt(args[0]) / Math.log10( 5) + 1) / 2 + 1);
13:        vSUP = (int)((Integer.parseInt(args[0]) / Math.log10(239) + 1) / 2 + 1);
14:
15:        init(p, 0,    ARR);
16:
17:        init(s, 16*5, ARR);
18:        for(int n = 1; n <= uSUP; n++) {
19:            div(s, 5*5,  ARR);
20:            for(int i = 0; i < ARR; i++) {t[i] = s[i];}
21:            div(t, 2*n-1, ARR);
22:            if(n % 2 == 0) {
23:                sub(p, t, ARR);
24:            } else {
25:                add(p, t, ARR);
26:            }
27:        }
28:
29:        init(s, 4*239, ARR);
30:        for(int n = 1; n <= vSUP; n++) {
31:            div(s, 239*239, ARR);
32:            for(int i = 0; i < ARR; i++) {t[i] = s[i];}
33:            div(t, 2*n-1,  ARR);
34:            if(n % 2 == 0) {
35:                add(p, t, ARR);
36:            } else {
37:                sub(p, t, ARR);
38:            }
39:        }

```

```

40:     disp(p, ARR);
41: }
42:
43: public static void init(int[] p, int atop, int array) {
44:     for(int i = 0; i < array-1; i++) {
45:         p[i] = 0;
46:     }
47:     p[array-1] = atop;
48: }
49:
50: public static void div(int[] p, int d, int array) {
51:     for(int i = array-1; i > 0; i--) {
52:         p[i-1] += (p[i] % d) * 10000;
53:         p[i ] = p[i] / d;
54:     }
55:     p[0] = p[0] / d;
56: }
57:
58: public static void add(int[] p, int[] q, int array) {
59:     for(int i = 0; i <= array-1; i++) {
60:         p[i+1] += (p[i] + q[i]) / 10000;
61:         p[i ] = (p[i] + q[i]) % 10000;
62:     }
63: }
64:
65: public static void sub(int[] p, int[] q, int array) {
66:     for(int i = 0; i <= array-1; i++) {
67:         p[i+1] += -1 + (10000 + p[i] - q[i]) / 10000;
68:         p[i ] = (10000 + p[i] - q[i]) % 10000;
69:     }
70: }
71:
72: public static void disp(int[] p, int array) {
73:     System.out.println(p[--array] + ".");
74:
75:     DecimalFormat df = new DecimalFormat("0000");
76:     while(--array > 0) {
77:         System.out.print(df.format(p[array]));
78:     }
79:     System.out.println();
80: }
81: }

```

プログラムは引数を桁数としてとる。% javac Pivalue.java で実行ファイルを作ったら、% java Pivalue 100 のように引数をとって実行してみよう。この場合なら 100 桁分の円周率が表示される。

EX. 73:, 76:行目に `--array` の表記がある。これは `array--` とは動作が異なる。違いを調べよう。

TRY! 10 万桁程度の計算なら、おそらく数分以内に結果が表示されるだろう。プログラムには改良の余地がまだまだあるので、好きにしてくれたまえ。