

2.2 指数関数

$f(x) = x^2$ と $f(x) = 2^x$ では、2 と x の位置が逆になっているだけなのに格段の差がある。 $x = 5$ のとき、 5^2 であれば 5×5 で済むが、 2^5 であれば $2 \times 2 \times 2 \times 2 \times 2$ と計算する。誰だって地道に掛け算を繰り返すのだ。 x^2 はいつだって 2 回の掛け算で済むのに、 2^x は x の値によっては何回も何回も何回も ... 掛け算をしなくてはならない。計算機があれば楽なのになあ、と思った君、その考えは甘い。実はコンピュータだって何回も何回も何回も ... 掛け算をしているんだな。

指数関数を相手にしてみよう。

programming list [XpowOf2.java]

```

1: import java.util.Scanner;
2:
3: public class XpowOf2 {
4:
5:     public static void main(String[] args) {
6:         Scanner s = new Scanner(System.in);
7:         System.out.print("input 'x' of 2^x: ");
8:
9:         int n = s.nextInt();
10:
11:         System.out.println("2^" + n + " = " + powf(n));
12:     }
13:
14:     public static int powf(int x) {
15:         int p = 1;
16:         while(x-- > 0) {
17:             p = p * 2;
18:         }
19:         return p;
20:     }
21: }

```

指数を扱うので指数計算をするメソッド名は `powf` とした。`powf()` メソッドは整数値を受け取り整数値を返すので、14:行目にあるように `int powf(int x)` としてある。

あれれ? [`Xsquare.java`] に較べて中身が急に増えてしまったぞ。なんと 14:-20:行目までの 7 行を占領している。何回も何回も何回も ... 掛け算をしなくてはならないので当然だ。

`powf()` メソッドの仕事は大体伝わっているだろう。15:行目で `p = 1` にしておけば、17:行目で 2 を掛けるたびに `p` の値が 2 倍になる。必要な回数だけ掛けて、19:行目の `return;` 文でそのときの

p の値を返せば、 2^x が求められる寸法だ。

`powf()` メソッドには見慣れない書き方が一つされている。16:行目の `while(x-- > 0)` のことである。for 構文で `i++` が、`i` の値を 1 ずつ増やす意味で使われるように、`x--` は `x` を 1 ずつ減らすことである。で、これが不等号と組み合わせるとどうなるんだろう。

これは、`x > 0` の判断を 1 回するたびに、`x` が 1 ずつ減っていく仕組みになっている。いつか `x > 0` の判定が偽となるので無事 `p` の値が返ってくるのである。

`main()` メソッドは [`Xsquare.java`] と同じ形式だ。違いは 11:行目で、`sqf()` が `powf()` になったことだけである。メソッドの差し替えだけで別の機能を持つプログラムにできるのが、メソッドの強みでもある。

EX. “input 'x' of 2^x : ” の要求に対し、負の数を入力するとんでもないことになる。なぜか？

TRY! 負の数の入力での変な振る舞いを減らすなら `while` 構文より `for` 構文の方が適している。
`for` 構文に書き換えてみよ。

プログラム [`Xsquare.java`] と [`XpowerOf2.java`] でメソッドが登場した。見ての通りメソッドの中身は、ごく簡単なものにも複雑なものにもできる。また、メソッドを何種類作ってもかまわない。こう言うと、メソッドの作り方に何かルールがないかと考えてしまうだろう。実はルールはあってないようなものだ。私はこの豆旅を介して、生産的なソフトウェアを作ったり効率的なプログラムを書いているわけではない。だからメソッドの作り方もいい加減なものだ。ただ習慣や慣例によって、メソッドの作り方のガイドラインは存在している。そして私には無関係なだけなのだ。

しかしメソッドの作り方の基本は、一つのメソッドに多くの機能を持たせないことだ。一つのメソッドには一つの機能が望ましい。

一つの関数に多くの機能を持たせると、トラブルが発生したとき、メソッド内のどこが問題であるかを特定しづらい。もし、二つのメソッドに分けてプログラムが組めるなら是非そうしよう。なぜなら、トラブルの原因を特定しやすいからだ。メソッドには勢いよく機能を詰めがちだが、自分のためにもいくつかの小さなメソッドに分ける方がよい。

そうは言っても、私はこの豆旅ではそこまで真剣に考えてない。もともと作るプログラムが小さいことと、これをもとに何かを作るわけではないからだ。あくまでも気まぐれな豆旅ということである。

TRY! Java で 2^x が計算できれば `powf()` メソッドは不要で、11:行目の `.println()` 内の書式は `"2^" + n + " = " + 2^x` で済む。が、これではダメなことを確かめてみよ。また、Java で 2^x を求める命令が何か調べよ。