

4章 プリミティブタイプと演算子

Chapter

04

4-1 Javaの“タイプ”の種類

前の第3章では、あなたがプログラムの中に書く名前、とくにメソッド名と変数名には、名前を宣言するときに「タイプ指定」が必要だ、と述べました。タイプ指定がないとコンピュータは、その変数や、メソッドの返し値のために、どれだけの大きさのメモリを確保したらいいか、あるいは、その変数のためにどんな処理をしたらいいのか、わからないからです。

Javaのタイプは「プリミティブタイプ」と「オブジェクトタイプ」の二種類に分かれます。オブジェクトタイプのタイプ指定としては、そのオブジェクトが属するクラスの名前を書きます。たとえば：

```
String myName;                                //Stringクラスのオブジェクト  
myName = "山田花子";  
Color black = new Color(0, 0, 0);           //Colorクラスのオブジェクト：真っ黒
```

オブジェクトタイプはクラスのオブジェクトですから、それらのタイプは必ず人間がプログラムを書いて作ったものです。Stringクラスなどは、コンパイラjavacやインタープリタjavaと一体的に提供されている大規模なクラス集——JavaのAPIと呼ぶ——に含まれていますが、それも、String.javaというソースファイルを誰かが書いて、そのコンパイル結果であるString.classをわれわれは使うのです。

本書でクラスを説明するときの例として取り上げている Color クラスも、もっと実用性のある本格的な Color クラスが、やはり API のクラス集に含まれています。それもやはり、誰かが Color.java というソースプログラムを書いたお蔭です。

そのほかのオブジェクトタイプも、すべて、企業、インターネットの上の見知らぬ人、友人、知人、あるいはあなた自身、などなどがクラスを書いて作るタイプです。

一方プリミティブタイプは、int、double などの数値タイプと、boolean という二値タイプ（ブーリアンタイプ）の二種類ですが、それらは“人間が書いて作ったものではなく、元々コンピュータが物理的に持っているタイプである”という点で共通しています。コンピュータの上に最初からある、という意味で、プリミティブ（primitive、原始的、本源的）と呼ぶのでしょう。

数値タイプは整数タイプと浮動小数点数タイプの二種類に分かれます。後者の浮動小数点数は、小数部のある数、いわゆる“実数”を、コンピュータの上で擬似的に表現する数値タイプです。整数は言うまでもなく小数部のない数、0、1、-1、2、…… 100、-8、12893、などなどです。

本物の実数は、私たちが日常的にささいな計算をしていても、小数部がひんぱんに、ものすごく長くなったり、無限桁になったりします。有限機械であるコンピュータは、ものすごく長い桁や無限に長い桁を表現できませんから、浮動小数点数（floating point number）という特殊な約束事を使って「擬似的な実数」を表現します。ですから、私たちの常識的な実数計算の結果と、コンピュータの上の実数計算（浮動小数点数を使った計算）の結果が食い違うことはしょっちゅうあります。浮動小数点数の詳しい仕組みの説明は、ここでは略します。

そしてまた整数も、コンピュータはやはり無限に大きな数は表現できませんから、32ビット幅なら32ビット幅という、一定のサイズを意識した計算を私たちコンピュータユーザはしなければなりません。また int のような有符号整数（正負のある整数）は、一定の幅のビットパターン（たとえば8ビットなら00000000から11111111まで）の半分を正の数用、残り半分を負の数用に使います。8ビットを全部正の数に使うと、0から255までを表現できますが、有符号整数として使うと、-128から+127までを表現します。

ちなみに、整数の正負は、こんな約束で表現します（8ビットの場合）：

00000000 から 01111111 まで…… 0 から 127 まで
 11111111 から 10000000 まで…… -1 から -128 まで

00

01

02

03

04

05

06

07

08

09

10

このように、最上位ビットが1なら負数、0なら正数、という約束事をコンピュータは伝統的に使います。

Javaのプログラミングの学習は、クラスとオブジェクトの作り方や使い方が主になりますから、本書のはじめのほうの部分で、ついつい注意が行き届かなくなりがちなプリミティブタイプとその演算について、ざっと学んでおこうと思います。

4-2 プリミティブタイプの種類

Javaのプリミティブタイプは、次の8種類です：

【Javaのプリミティブタイプ】

■ 数値タイプ ■

整数タイプ

char	16ビット無符号整数（正数のみ）…主に文字の文字コード値（Unicodeのコード値）を表す。 0から65535まで（16進数記法では0x0000から0xffffまで）
byte	8ビット有符号整数（正負がある整数） -128から+127まで
short	16ビット有符号整数（正負がある整数） -32768から+32767まで
int	32ビット有符号整数（正負がある整数） -2147483648から+2147483647まで
long	64ビット有符号整数（正負がある整数） -9223372036854775808から+92233720368547758071まで

浮動小数点タイプ

float	32ビット浮動小数点数（擬似的実数表現…小数点のある数） $1.401298464324817 \times 10^{-45}$ から $3.4028234663852886 \times 10^{38}$ まで
double	64ビット浮動小数点数（擬似的実数表現…小数点のある数） 4.9×10^{-324} から $1.7976931348623157 \times 10^{308}$ まで

■ ブーリアンタイプ ■

boolean true(真)・false(偽)の2値変数

最後のブーリアンタイプは、値が二値しかないものを表現するために使います。たとえば、ナニとナニが等しいか／等しくないか、という比較の結果は、true(真)／false(偽)の二値しかありません。ナニナニが正しいか正しくないか、という検証の結果も、やはり二値しかありません。今コレコレの処理をしてよい状態か

否か、という制御のための変数の値も、二つの値 (OK / NO) を表現できれば十分です。Java のキーワード、true と false は、ブーリアンタイプの値を表現するために使います。イギリスの Bool さんという数学者 (George Bool, 1815-1864) がこのような二値代数理論の開拓者なので、その名を取ってブーリアン (タイプ名は boolean) と呼ばれます。

昔のプログラミング言語は、1 や 0 という整数を使って、真偽や正否などの二値を疑似的に表現していました。そうすると、真を表す 1 がふつうの整数の変数に加算されたりして、難解なバグが発生することがときどきありました。Java は、true や false のコンピュータの上の物理的実体をユーザに対して隠していますし、われわれユーザ / プログラマが気にする必要もありません。たぶん物理的には 1 や 0 を使っているんでしょうけど、もしそうだとすると、Java では true や false を int タイプの変数に加算するなどの操作は許されません (やるとコンパイラがエラーメッセージを出す)。

整数の大きさや小数部の精度に限界のある、これらプリミティブタイプでは、たとえばお金の計算のような端数まで重要な計算はできません。そういう重要な計算のためには、プリミティブタイプではなく、Java が最初から提供しているクラス集 (Java API^{※1} と呼ぶ) の中にある特殊なクラス、すなわちプリミティブタイプではなくオブジェクトタイプを使います。

4-3 演算子

Java では、オブジェクトタイプ、すなわちクラスのオブジェクトに対しては、いろんな操作をメソッドというものを使って行います。しかしプリミティブタイプにはクラスがなく、したがってメソッドもありませんから、操作はもっぱら演算子を使って行います。たとえば、二つの整数タイプを加算して別の整数タイプに結果を取める操作は、演算子 = や + を使ってこんなふうに書きます：

```
int a, b, c;
a = 4;
b = 2009;
c = a + b; // c = 4 + 2009;でも同じ
```

※1 API は、Application Program Interface の頭字語で、われわれがアプリケーションプログラムを作る (書く) ときのための、Java 言語のサービス窓口、といった意味です。Java の API は、便利なクラスをたくさん揃えています。その中の System クラスや String クラスは、本書でもすでに使いました。