

第2章

データの宣言と使用

目次

- リテラル
- 変数や定数の宣言と初期化

リテラル

ソースコードに直接書き込んだ値や、書き込むための表記のことを、
リテラルと呼ぶ。

例えば...

「10+10」というコードの10は「10」という整数値を表すリテラル

`print("Hello")`というコードの"Hello"は「Hello」という文字を表すリテラル

リテラル

①整数リテラル

小数部をもたない数値を表現。

10進数、8進数、16進数、2進数が表現可能。

②浮動小数点リテラル

小数部をもつ数値を表現。10進数、指数を表現できる。

リテラル

③文字リテラル

一つの文字を表現する。文字は「**'**」(シングルクォーテーション)で囲む。
「田」などの一文字を表現する以外に、特殊文字を扱う、
エスケープシーケンスを表現することも可能。

④文字列リテラル

複数の文字の集合体である文字列を表現する。「**"**」(ダブルクォーテーション)で囲む。

リテラル

5

⑤論理値リテラル

真(true)か偽(false)の値を表現。

⑥nullリテラル

参照型のデータを利用する際に「何も参照していない」という意味を表す。

変数や定数の宣言と初期化

■ 変数とデータ型

名前をつけた入れ物を用意し、計算結果の値などを格納し、後から参照する入れ物を**変数**と呼ぶ。

Java言語では、扱う値の種類が**データ型**として決められている。

データ型には、**基本データ型**と**参照型**の二種類がある。

変数や定数の宣言と初期化

7

■ 変数宣言と代入

変数を宣言することを**変数宣言**と呼び、その変数に値を格納することを、**代入**と呼ぶ。

構文

```
データ型 変数名; //変数宣言  
変数名 = 値; //変数に値を代入
```

宣言した変数を使う前に最初に値(初期値)を代入しておくことを、変数の**初期化**と呼ぶ。

変数や定数の宣言と初期化

■ 定数

固定された値を扱うために使いたい場合、**定数**として宣言する。

変数を宣言するときは**final**修飾子を使用する。

構文

```
final データ型 定数名 = 初期値;
```

変数や定数の宣言と初期化

9

■ 文字列は参照型

複数の文字の集合体である文字列は、基本データ型に含まれず、文字列は参照型として扱われる。

文字列を扱う変数を宣言するには、参照型の一つのString型を使用。

構文

```
String str = "Hello";  
String name = "田中";
```

複数の文字のかたまりを一つの変数で扱う。この変数を**参照変数**と呼ぶ。

変数や定数の宣言と初期化

10

■ 識別子と予約語

変数名でなくクラスやメソッドなどの名前は、数字や文字を組み合わせで作成する。これらの名前を**識別子**と呼ぶ。
Java言語ですでに使用されている名前を**予約語**と呼ぶ。