

目次

- オブジェクトの生成とメソッド呼び出し (承前)
- クラス変数とクラスメソッド
- 宿題?

★2 オブジェクトの生成とメソッド呼び出し (承前)

「承前」って? ⇒ 辞書引きましょう

★2.7 オブジェクトの参照とヒープ

Java の変数の型は、**参照型** (クラス型を含む) と **プリミティブ型** に大別できる。この2つの型では、「変数への代入」の意味が異なる。右のプログラムの動作を考えてみよう。

このプログラムで観察できるような参照型とプリミティブ型の違いは、それらの値の記憶のされ方から理解できる。プリミティブ値はその変数用の記憶場所にそのまま格納されるのに対して、オブジェクトはすべて「**ヒープ**」という領域に置かれ、変数には「そのオブジェクトがヒープ上のどこにあるか」を表す情報 (これを「**参照**」という) が格納される。そのため、右記で `m1 = m` という代入文を実行すると、変数 `m1` は `m` と同じオブジェクトを「参照している (指している)」ことになる。

プリミティブ型と参照型

```

1 // プリミティブ型の場合
2 int x, y;
3 x = 50;
4 y = x;
5 x += 50;
6 //参照型 (オブジェクト) の場合
7 Turtle m, m1;
8 m = new Turtle(); frame.add(m);
9 m1 = m;
10 m.fd(100);
11 m1.bk(100);

```

★2.8 インスタンス変数

Turtle クラスのインスタンス (個々のかめたち) は、自分の位置、向き、色や大きさなどの情報を持っている。インスタンス毎に保持されるこのような情報は、**インスタンス変数** と呼ばれる変数に格納されている (☆1)。T23 の例 (p.18,20) で考えてみよう。

☆1) 詳しくは先の授業で説明するが、存在していてもユーザには公開されない (API に示されない) インスタンス変数もある。例えば Turtle クラスの API (p.14) には、インスタンス変数として色 (`tcolor`) と大きさ (`tscale`) のみが示されているが、実際にはかめの位置や向きなどを表すインスタンス変数も存在している。

★3 クラス変数とクラスメソッド (第3章)

★3.1 パッケージ, import 宣言, Java API (p.23)

前回の例で, `java.awt.Color` というクラスのインスタンスを以下のように生成して利用した (この例では黄色のオブジェクトが生成される).

```
————— java.awt.Color クラスのインスタンスの生成式 (1) —————  
java.awt.Color c = new java.awt.Color(255, 255, 0);
```

このクラスは, Java に標準で用意された「標準クラスライブラリ」に属するクラスである. 標準クラスライブラリには膨大な数のクラスがあるので, それらは「パッケージ」にまとめられている. 例えば, `java.awt.Color` は, `java` パッケージの中の `awt` パッケージの中の `Color` クラスである.

標準クラスライブラリに含まれるクラスを利用するには, `java.awt.Color` のようにその名前を指定すればよいが, ソースファイルの先頭に「import 宣言」を書いておくと, クラス名を省略形で書けるようになる.

```
————— import 宣言の例 1 —————   ————— import 宣言の例 2 —————  
import java.awt.Color;                   import java.awt.*;
```

いずれの場合も, `java.awt.Color` を `Color` と書けるようになる. 例 2 の場合, `java.awt` パッケージ内の全てのクラスについて, 省略形を使えるようになる.

```
————— java.awt.Color クラスのインスタンスの生成式 (2) —————  
Color c = new Color(255, 255, 0); // 黄色オブジェクトの生成
```

標準クラスライブラリにどんなクラスが含まれているのか, それを使うにはどうしたらよいのかを知るためには, それらの API 仕様を参照する必要がある. web で公開されている.

- Java SE7 API 仕様 <http://docs.oracle.com/javase/jp/7/api/>
- この科目のページからもリンクしている

★3.2 クラス変数とクラスメソッド (p.25)

前の節で説明したように, インスタンス変数は, 個々のインスタンス (オブジェクト) が保持する値である. また, インスタンスメソッドを呼び出すと, 特定のインスタンスに処理を実行させたり状態を変化させたりすることができる. これに対して, 個々のインスタンスに属するのではなく, あるクラスの中で共通の変数やメソッドがあると便利なことがある. それが, 「**クラス変数**」, 「**クラスメソッド**」である.

クラス変数 API 仕様のフィールドの欄で先頭に `static` という修飾子がついているもの. `Turtle` クラスには `withTurtleAll` という名前のクラス変数がある (p.14 参照).

クラスメソッド API 仕様のメソッドの欄で先頭に `static` という修飾子がついているメソッド. `Turtle` クラスには `speedAll` というクラスメソッドがある (p.14 参照)

★ 3.3 使ってみよう

前回登場した T23.java を改造してクラス変数、クラスメソッドを使ってみよう (p.26 練習問題 3.3 & p.27 練習問題 3.4) .

```
----- T23.java -----
1 public class T23 {
2     public static void main(String[] args){
3         int d = 100, x, y, a;
4         TurtleFrame f = new TurtleFrame();
5         Turtle m = new Turtle(200,300,0);
6         f.add(m);
7         m.fd(d);
8         x = m.getX();           // m の X 座標のとり出し
9         y = m.getY();           // m の Y 座標のとり出し
10        a = m.getAngle() - 45;   // m の角度のとり出し
11        Turtle m1 = new Turtle(x, y, a); //m1 の生成
12        f.add(m1);
13        m1.fd(d);

14        Turtle m2 = m.clone();   //m2 の生成
15        f.add(m2);
16        m.rt(45);
17        m.fd(d);

18        m2.tcolor = new java.awt.Color(0,255,255); // m2 の亀の色を水色に変える
19        m2.tscale = m2.tscale * 4; // m2 の亀を現在の 4 倍の大きさにする
20        m2.fd(d);
21    }
22 }
```

【注意】 上記のプログラムの例でクラス変数を参照しているところやクラスメソッドを呼び出しているところは、クラス名を使って `Turtle.withTurtleAll` のように書くべきですが、実は `m.withTurtleAll` のように書いても正しく動作しちゃいます。 `m` が `Turtle` クラスのオブジェクトを指す変数なので、 `Turtle` クラスのクラス変数やクラスメソッドを扱いたいという気持ちを言語処理系がわかってくれちゃうからです。でも、動くからといってそういう書き方をしていると、インスタンス変数とクラス変数、インスタンスメソッドとクラスメソッドそれぞれの違いがわからなくなります。きちんと使い分けましょう。

★3.4 Java API に現れるクラス変数, クラスメソッド (p.27)

標準クラスライブラリに登場する様々なクラスのうち、クラス変数やクラスメソッドをよく使うものをいくつか紹介しておく。String クラスについては次回とりあげる。

Math クラス

java.lang というパッケージに、Math というクラスがある。数学関係の関数や定数がクラスメソッド/変数として用意されている (☆2)。このクラスの正式な名前は java.lang.Math ということになるけれど、java.lang パッケージのクラスは自分で import 宣言を書かなくても自動で import されるので、Math. ならぬように書くだけで使える。

☆2) 下記の PI のような定数は、final という修飾子をつけることで、代入不可のクラス変数として宣言されている。p.27 参照。

Math クラスの利用	左の実行結果
<pre>double x = Math.sqrt(2.0); // Math クラスのクラスメソッド sqrt() System.out.println(x); double theta = Math.PI; // Math クラスのクラス変数 PI System.out.println(theta); x = 30.0; theta = x*Math.PI/180.0; // theta = π/6 System.out.println(Math.sin(theta));</pre>	<pre>1.4142135623730951 3.141592653589793 0.49999999999999994</pre>

System クラス

System も、java.lang パッケージのクラスである。インスタンスは生成できない。メソッド System.out.print() や System.out.println() はこのクラスに関係している。詳しくは p.29。

宿題?

QUIZ

Q1. 下記のソース 1 のようなプログラムを実行した場合、変数 m1 (m ではない) が指すかめは何色ですか。

Q2. 下記のソース 2 の誤りを見つけなさい (いっぱいあります)。

ソース 1	ソース 2
<pre>import java.awt.*; : Turtle m = new Turtle(), m1 = m; : m.setColor(new Color(255, 255, 0)); :</pre>	<pre>: TurtleFrame f = new TurtleFrame(); Turtle m = new Turtle(); : f.add(m); TurtleFrame.addMesh(); m.fd(100); m.withTurtleAll = false; Turtle.tscale = 0.8; Turtle.moveto(0, 0); :</pre>

次回までに...

次回は教科書第 4 章の内容を説明します。あらかじめ読んでおくこと。例題のプログラムを作成し実行しておくこと。