

# 進め!! ~はじめの一步~ Java 新人



山田 猛矢・比嘉 築 著 持増 宏大 絵

## はじめに

みなさーん。こんにちわわわわわー。再び帰ってきましたー。「卒 Java 原人 君は Java 新人に進化できるか!?’の続編でーす。

「卒 Java 原人」読者から、次に何の本を勉強すればいいかと聞かれ、一般のやさしいと思われる入門書を紹介したところ、やっぱり挫折・・・ザセツ・ざせつ ということで、本書はもうちょっと Java を身近に感じてもらうために、なんとなくわかってもらうために、わかった気になってもらうために、Java 新人としての一步を踏み出してもらえるように書きましたー。

なので、「卒 Java 原人」と同様、本書をすべてマスターしたからといって、“これで Java はもう完璧”というわけではありませーん。Java 入門のための入門書という位置づけなのでお忘れなく。

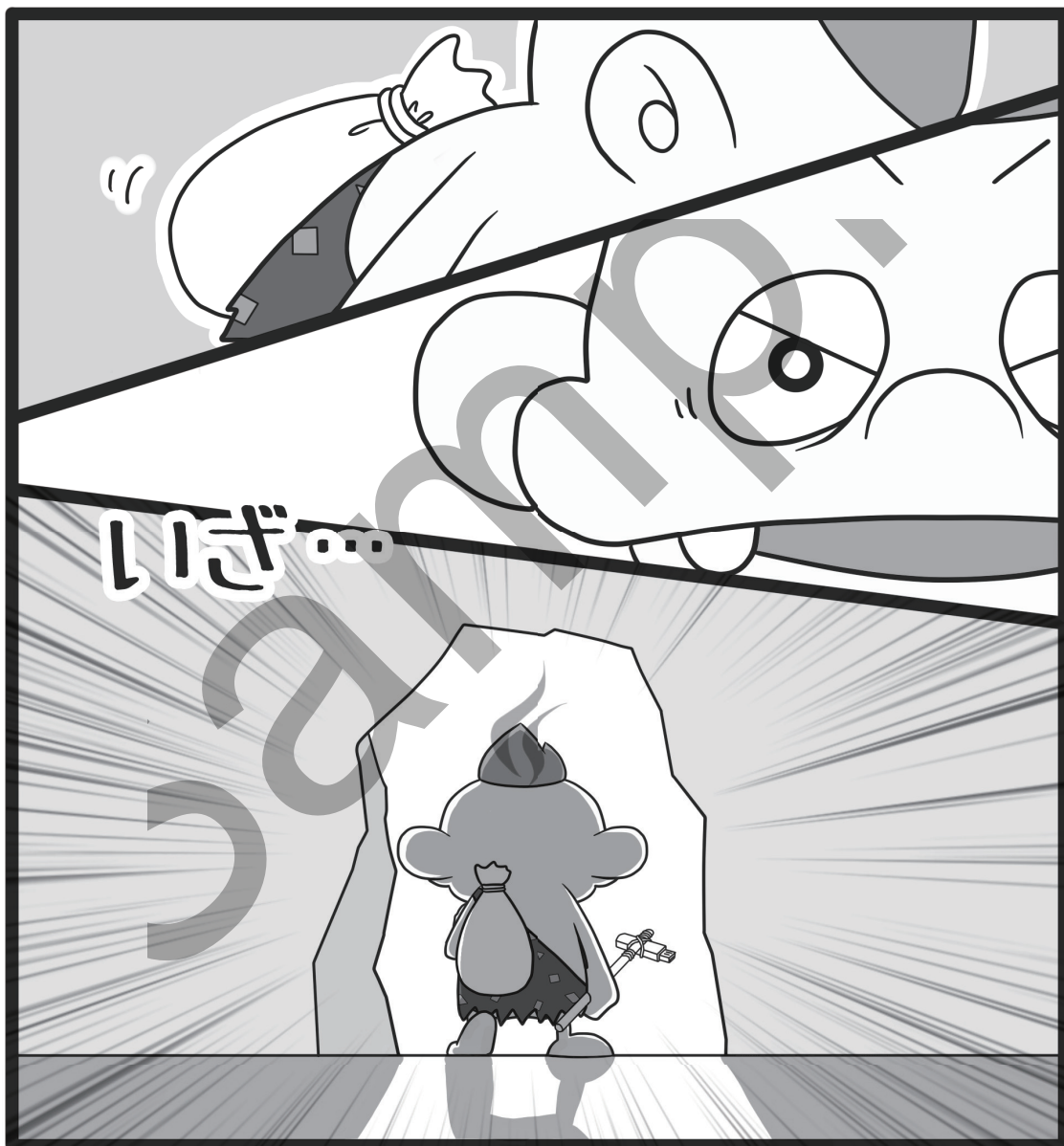
で、本書でやる内容ですが、基本的に「卒 Java 原人」の最後に作ったプログラム（再生ボタンを押すと車が動くアレ）と同じようなものを作っていくまーす。って、がっかりしないでね。もちろん違いはあるんで。

本書では「卒 Java 原人」で使った AnimationFrame クラスは使わずに、Java の標準クラスライブラリだけを使って作っていくまーす。第 1 章から第 6 章までは、そのために必要なこと、知っといってもらいたい基本的なことを書いてまーす。基本的なクラスの話から始まって、継承の話まで。で、第 7 章以降で標準クラスライブラリを使ってプログラムを作っていくまーす。もちろん、第 7 章以降でも、第 6 章までで書ききれなかったインタフェースや例外の話を書いてまーす。ぜひ最後まで読んで Java 新人としての一步を踏み出してもらえたらと思いまーす。

なお、「卒 Java 原人」をまだ読んでないって人は、ぜひ「卒 Java 原人」から読んでね♡ だってー、本書は「卒 Java 原人」を読んだっていう前提で書いてるんで。

では、早速。

いざ、第2ステージへ



# 目次

## CHAPTER 1

フィールドのみのクラスを作ってみよう!!使ってみよう!!

- 1.1 フィールド1つのクラスを作ってみよう!!使ってみよう!! 6
- 1.2 フィールド2つのクラスを作ってみよう!!使ってみよう!! 12

## CHAPTER 2

クラスにメソッドを追加しよう!!

- 2.1 引数と戻り値 14
- 2.2 「引数なし」「戻り値あり」のメソッド 16

## CHAPTER 3

引数を持つメソッドを追加しよう!!

- 3.1 「引数あり」のメソッド 25
- 3.2 「引数2つ」のメソッド 29

## CHAPTER 4

アクセス修飾子でカプセル化

- 4.1 フィールドに値を代入するより良い方法は? 35
- 4.2 アクセス修飾子 40

コラムA パッケージ 45

コラムB staticなフィールドやメソッドを作ろう 47

## CHAPTER 5

コンストラクタってなに?

- 5.1 コンストラクタってなに? 50
- 5.2 引数を持つコンストラクタ 53
- 5.3 コンストラクタとインスタンス 56

## CHAPTER 6

継承ってなに?

- 6.1 継承ってなに? 59
- 6.2 サブクラスへの機能追加 62
- 6.3 メソッドのオーバーライド 65

コラムC Androidアプリにチャレンジ!? 69



## CHAPTER 7

JFrameクラスを使ってみよう!!

7.1 JFrameクラスを使ってみよう!!	70
7.2 コンストラクタに引数を指定	74
7.3 フレームを2つ作ってみよう!!	76

## CHAPTER 8

フレームに絵を描こう!!

8.1 フレームに絵を描こう!!	80
------------------	----

## CHAPTER 9

ボタンを追加しよう!!

9.1 ボタンを設置しよう	85
9.2 ボタンにイベント処理を追加しよう	87
9.3 ボタン2個での処理	93

コラムD レイアウトマネージャー	97
------------------	----

コラムE イベント処理	100
-------------	-----

## CHAPTER 10

スレッドを使ってみよう!!

10.1 スレッドって何?	102
---------------	-----

コラムF Threadクラスを継承	114
-------------------	-----

最後に	116
-----	-----

### 注意

- ・ 本書に記載されている社名、製品名などは、一般に各社の商標または登録商標です。本文中では©, ®, TM, などの表記は省略しております。
- ・ 本書に記載された内容は執筆時点の情報です。今後、本文中に記載されている URL や製品仕様等は予告なく変更される可能性があります。
- ・ 本書に記載された説明、プログラム等を運用した結果について著者および発行所は一切責任を負いません。個人の責任の範囲でご使用ください。

# CHAPTER 1

## フィールドのみのクラスを作ってみよう!!使ってみよう!!

### 1.1 フィールド1つのクラスを作ってみよう!!使ってみよう!!

ようこそ!! Java 新人のみなさん。「卒 Java 原人」での内容は、まだ覚えているかなー? Math クラスや AnimationFrame クラス (著者オリジナルクラス) のフィールド (変数) やメソッド (処理) の使い方, 覚えているかなー?



本書では、もう一歩、Java の世界に足を踏み出してもらいたいと思ってまーす。具体的には、Java の標準ライブラリ (JDK をインストールした時に標準で入ってるやつで、API ドキュメントに説明のあるクラス) のみ使って、「卒 Java 原人」でやったようなことをやっていきまーす。

ま、気楽にやっていこー。

まずは、クラスの理解をさらに深めるために、(AnimationFrame クラスのような) オリジナルのクラスを自分たちで作って、そのクラスを実際に使ってみましょー。

と、その前に、復習でーす。「卒 Java 原人」の 64 ページでやったんだけど、クラスはフィールド (変数) とメソッド (処理) の 2 つの部分からできてるって覚えているかなー? 忘れてたーって人は思い出して♥

クラス

フィールド

メソッド

最初は、フィールド（変数）のみを持つ YMath クラスというオリジナルのクラスを作って、それを使ってみましょー。

クラスの作り方は

```
class クラス名{・・・}
```

です。簡単でしょ。

今回はフィールド（変数）のみのクラスなんで、

```
public static void main(String[] args){・・・}
```

ってのはいりませーん。これメソッドだし。

じゃー、早速オリジナルのクラス、YMath クラスを作ってみましょー。次のプログラムを入力し、ファイル名 YMath.java で保存してコンパイルしてみて。(クラス名とファイル名が一致してないといけなかったこと覚えてた?)

ちなみに、実行はしなくていいよ。ってかできないよ。だって実行したときに、最初に呼ばれるメソッド main() (エントリーポイントってやつ) が無いもーん。

#### Source

```
1 class YMath{
2     double ye = 2.7;
3 }
```

どう？ 簡単でしょ。単純でしょ。

では解説をば。1行目はいいよね。Java はクラス単位で書かれるので1行目は「class クラス名」から始まりまーす。ここでは、クラス名は YMath。

2行目が YMath クラスの中身です。見てわかるように、このクラスはフィールド（変数）を1つだけもつクラスです。具体的には、double 型のフィールド ye を1つだけ持ってまーす。もちろん ye には 2.7 が代入されてまーす。

では、ちょっとこのオリジナルの YMath クラスを使ってみましょー。

使い方は「卒 Java 原人」で使った AnimationFrame クラスの使い方と一緒に。まずはインスタンス(クラスのコピーみたいなやつ)を生成(キーワードは「new」)して、フィールド(変数)にアクセスしてみます。

クラスのフィールド(変数)やメソッド(処理)を使うときは、インスタンス(クラスのコピーみたいなやつ)を生成してから使うんでござる。覚えとったかなー？



次のプログラムをコンパイル & 実行してみてください。もちろん YMath クラスを使ってるんで、YMath.class と同じフォルダに入れてね。

```
Source
1 class StepR1{
2     public static void main(String[] args){
3         YMath ym = new YMath();
4
5         System.out.println(ym.ye);
6     }
7 }
```

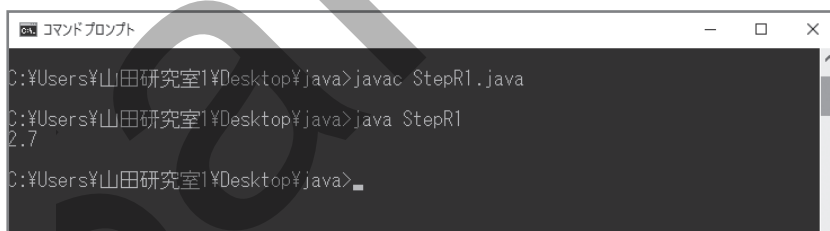


図 1.1 StepR1 の実行

できたー？ 図 1.1 みたいになっただけ？

簡単に説明をば。3 行目で、YMath 型の変数 ym を宣言して、それと同時に YMath クラスのインスタンス(クラスのコピーみたいなやつ)を生成して ym に代入してまーす。インスタンス(クラスのコピーみたいなやつ)の生成方法は

new クラス名()



だったよねー。覚えてたー? ちなみにインスタンス (クラスのコピーみたいなやつ) を代入する変数の型名は, 「クラス名」と同じになりまーす。これまた覚えてたー? AnimationFrame クラスを使うときも

```
AnimationFrame af = new AnimationFrame();
```

って書いてたでしょ。あの時と同じ。今回は YMath クラスを使うんで

```
YMath ym = new YMath();
```

ってなりまーす。

で, 5 行目でフィールド ye を出力してまーす。ym インスタンスのフィールド ye にアクセスするんで

インスタンスを代入した変数名.フィールド名

って書いてフィールド ye を出力してまーす。

じゃー次は, 2 つのインスタンス ym1, ym2 を生成し, インスタンス ym2 の方だけに 5.0 を代入してみましょー。

次のプログラムをコンパイル & 実行してみてみて。

```

Source
1  class StepR2{
2      public static void main(String[] args){
3          YMath ym1 = new YMath();
4          YMath ym2 = new YMath();
5
6          ym2.ye = 5.0;
7
8          System.out.println(ym1.ye);
9          System.out.println(ym2.ye);
10     }
11 }
```

```

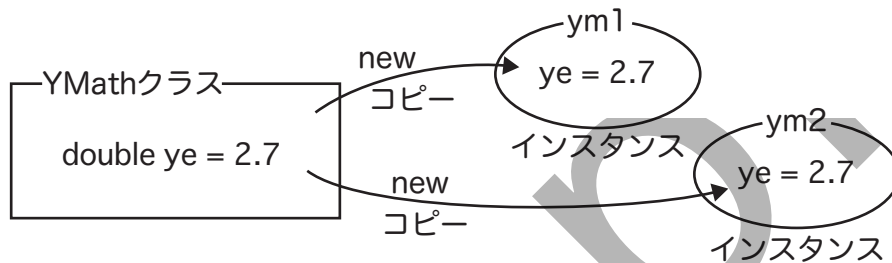
コマンドプロンプト
C:\Users\山田研究室1\Desktop\java> javac StepR2.java
C:\Users\山田研究室1\Desktop\java> java StepR2
2.7
5.0
C:\Users\山田研究室1\Desktop\java>

```

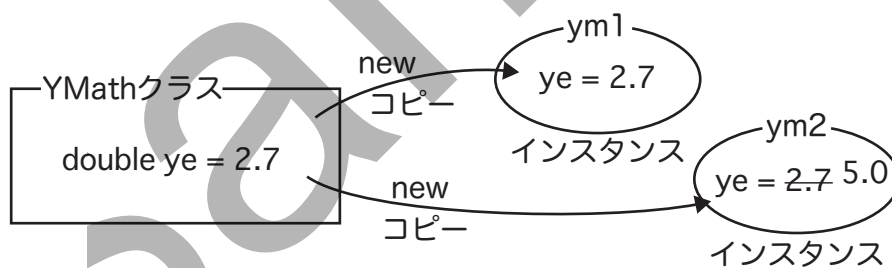
図 1.2 StepR2 の実行

図 1.2 みたいな結果が出ましたー？ 簡単に説明をば。

ym1 も ym2 も同じ YMath クラスから生成したインスタンス(クラスのコピーみたいなやつ) であす。



でも 6 行目で, ym2 の方のフィールド ye には 5.0 を代入してるんで, それぞれの ye を出力すると, 2.7 と 5.0 が出力されまーす。



こんな風に, 同じ YMath クラスからインスタンス (クラスのコピーみたいなやつ) を生成するけど, 一旦生成してしまえば, 後はそれぞれ処理されていくんで, それぞれのインスタンス (クラスのコピーみたいなやつ) で ye の値が変わっていきまーす。

ym1 と ym2 は, 同じ YMath クラスから生まれてくるんで兄弟みたいなもんだけど, 生まれた後は別々の道を歩いていきまーす。



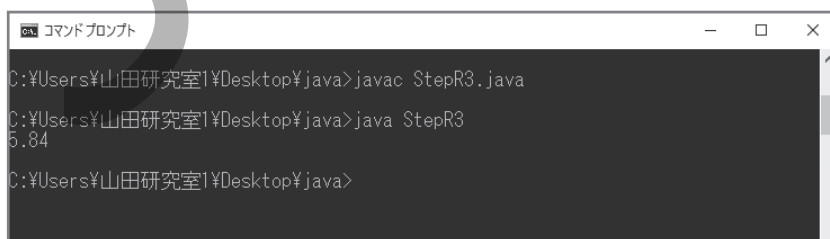
## 🔥 1.2 フィールド2つのクラスを作ってみよう!!使ってみよう!!

じゃー次に、YMathクラスにもう一つフィールド(変数)を追加し、フィールド(変数)を2つにしてみましょー。YMathクラスに網掛け部分を追加してね。

```
Source
1  class YMath{
2      double ye = 2.7;
3      double ypi = 3.14;
4  }
```

変更が終わったらコンパイルしといてね。YMathクラスのコンパイルが終わったら、次のプログラムをコンパイル & 実行してみて。

```
Source
1  class StepR3{
2      public static void main(String[] args){
3          YMath ym = new YMath();
4
5          double z = ym.ye + ym.ypi;
6
7          System.out.println(z);
8      }
9  }
```



```
コマンドプロンプト
C:\Users\山田研究室1\Desktop>javac StepR3.java
C:\Users\山田研究室1\Desktop>java StepR3
5.84
C:\Users\山田研究室1\Desktop>
```

図 1.3 StepR3 の実行





## 著者略歴

### 山田 猛矢 (やまだ たけし)

1975 年 誕生

2003 年 鹿児島大学大学院理工学研究科 博士前期課程 物理科学専攻 修了 修士 (理学)

2005 年 鹿児島第一高等学校 教諭

2006 年 第一工業大学 (現・第一工科大学) 工学部総合科学 講師

2010 年 第一工業大学 (現・第一工科大学) 工学部情報電子システム工学科 講師

2019 年 第一工業大学 (現・第一工科大学) 工学部情報電子システム工学科 准教授

2020 年 鹿児島大学大学院理工学研究科 博士後期課程 総合理工学専攻 修了 博士 (工学)

2022 年 第一工科大学工学部情報電子システム工学科 教授

現在に至る

### 比嘉 築 (ひが きづく)

1980 年 誕生

2007 年 鹿児島大学大学院理工学研究科 博士前期課程 物理科学専攻 修了 修士 (理学)

前後にプログラマ・システムエンジニア, 専門学校講師として勤務

2012 年 フリーとなり E Cube Factory 名で活動

2016 年 合同会社イー・シー・エフ 代表社員

現在に至る

## 進め!! Java 新人 ~はじめの一步~

---

2023 年 12 月 21 日 初版 1 刷発行

2023 年 12 月 31 日 初版 2 刷発行

著 者 山田 猛矢・比嘉 築

イラスト 持増 宏大

発行人 比嘉 築

発行所 E Cube Factory

〒904-0322 沖縄県中頭郡読谷村字波平1648

URL : <https://www.e3factory.com/book/>

---

©2023 Takeshi Yamada, Kizuku Higa, Koudai Mochimasu Printed in Japan

ISBN978-4-9907324-2-4

定価は裏表紙に表示してあります。

---

正誤表などの補足情報は以下の Web サイトをご確認ください。

<https://www.e3factory.com/book/susume-java-shinjin/>