

デジタル表現論・第一回

劉 雪峰 (リュウ シュウフォン)

2016 年 4 月 11 日

本日の目標

Java プログラミング言語の紹介
Java プログラミングの基本概念
プログラミングの環境の整備

デジタル表現論

世の中の様々な情報 (画像、音声、動画など) はデジタルデータで再現できる。情報をデジタルで表現するために、計算機との会話が必要となる。この授業は計算機と会話するための言葉、即ち、「プログラミング言語」を学ぶ。

Java プログラミング言語を通じて、様々な情報の基礎的なデジタル表現方法を勉強する。

Java 言語はあくまでもデジタル表現の 1 つの道具である。この授業は「Java 言語の入門とデジタル表現への応用」と考えても良い。プログラミングの以外に、たくさんのソフトウェアを利用することで、情報のデジタル処理も可能。

提示 この授業では Java 言語の基礎の知識のみを説明するので、全面的な勉強と理解は随時に教科書、Google などの検索エンジン、Wikipedia などを利用してやりましょう。

なぜ Java ?

Java の特徴

Java は世界中一番使用されるプログラミング言語と言われる。
(ランキングの標準によって、Java または C が 1 位となることがある。)

コードの実行は OS 環境に依存しないので、一度コードを書けばどこでも実行できる。(Windows+Linux+Mac)

オブジェクト指向言語であって、柔軟で拡張性の高いプログラムを作成できる。

たくさんのライブラリが用意されて、複雑な機能又は計算は数行のコードで実現できる。

豆知識： Java は 1990 年代前半にサン・マイクロシステムズ社によって開発された言語です。

計算機の考え方

計算機と会話するために、計算機の考え方を理解するのは大切である。
以下の例を考えよ。

例：変数 x, y の値を入れ替えること。

最初の値は $x = 1, y = 2$ であれば、入れ替え後、 $x = 2, y = 1$ となる。

計算機の考え方

計算機と会話するために、計算機の考え方を理解するのは大切である。
以下の例を考えよ。

例：変数 x, y の値を入れ替えること。

最初の値は $x = 1, y = 2$ であれば、入れ替え後、 $x = 2, y = 1$ となる。
計算機の考え方（操作の手順）：

- 1) 変数 z を用意する。 x の値を z に代入する。 ($z = x$)
- 2) y の値を x に代入する。 ($x = y$)
- 3) z の値を y に代入する。 ($y = z$)

コメント：

1. 人間に対して「入れ替える」の作業は簡単ですが、計算機では、必ず幾つかの基本操作に分解してから作業を行うことができる。
2. z という変数は不可欠である。

演習

「変数 x_1, x_2, x_3 の中の最大値を求める」という作業を以下の基本操作までに分解しなさい。

基本操作：2つの数の比較、値の代入。

演習

「変数 x_1, x_2, x_3 の中の最大値を求める」という作業を以下の基本操作までに分解しなさい。

基本操作：2つの数の比較、値の代入。

計算の例：

- 1) max という変数を宣言して、 $max = x_1$ 。
- 2) max と x_2 を比較して、 x_2 が max により大きいであれば、 $max = x_2$ 。
- 3) max と x_3 を比較して、 x_3 が max により大きいであれば、 $max = x_3$ 。

計算後、 max は x_1, x_2, x_3 の最大値となる。

コード

実際の計算機への指示（プログラムコード）

```
int x1=1, x2=5, x3=2; //準備
```

```
int max;
```

```
max = x1;
```

```
if( x2 > max ){ max=x2; }
```

```
if( x3 > max ){ max=x3; }
```

Java の入門

三つの例から Java を始めましょう。

```
/*  
【例1】ファイル名： HelloWorld.java  
*/
```

```
public class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Java コードの実行方法

ファイルの準備

- 1) [ドキュメント ¥ Java2016 ¥ Course1 ¥] というフォルダを作る。
- 2) エディタ TeraPad を使って、前ページのソースコードを入力し、「HelloWorld.java」に保存する。

コンパイル

- 1) Windows の中で、「スタートメニュー」 → 「アクセサリ」の中で、「コマンド プロンプト」をクリックする。
- 2) `cd`というコマンドを使って、作業の場所を「ドキュメント ¥ Java2016 ¥ Course1 ¥」に移動する。

`cd`の使用 : `> cd C:¥... ¥ Java2016 ¥ Course1`

- 3) `javac` を使って、Java のコードをコンパイルする。

```
> javac HelloWorld.java
```

エラーがなく、コンパイルが成功すれば、HelloWorld.class が生成される。

class ファイルの実行

コマンドプロンプトの画面で `java` というコマンドを使って、コンパイルされたファイルを実行できる。

```
> java HelloWorld
```

豆知識 : 「コマンドプロンプト」を起動するための2つの便利方法

1. Windows キー + 「r」を押して、[cmd] を入力して実行する。
2. Windows のスタートメニューで「cmd」を使って検索して、「cmd」というプログラムをクリックする。)

コードの説明

- 1) HelloWorld という class が定義されている。class の名前はファイルの名前に一致にすることが必要。
- 2) class の中の main というメソッドは最初に実行される。
- 3) System.out.println は結果の出力用のメソッドである。

「オプション」

ここまでの例を理解でき、余裕を持っていれば、次ぎの Java の例も考えてください。

注：クラスの人々の個人差を配慮する上、よりレベルの高い内容も用意している。

変数の宣言、メソッド

```
/* 【例2】 ファイル名 : Code2.java */
public class Code2 {
    public static int add_two_number(int a, int b){
        return a*10+b;
    }
    public static void main (String[] args) {
        int a,b;
        double x;
        a = 1; b = 2;
        x = add_two_number(a,b);
        System.out.println("Calculation of a+b:");
        System.out.println("Result is " +
            Double.toString(x) );
        System.out.print("Calculation of a+b:");
        System.out.print("result is ");
        System.out.println(x);
    }
}
```

コードの説明

- 1) 整数型 (int) の変数と倍精度の浮動小数点数型 (double) の変数が宣言されている。
- 2) `System.out.println` は出力のデータの最後に改行を入れる。
`System.out.print` の場合、改行が加えない。
- 3) `Double.toString()` は `Double` 型の変数を文字列 (String) に変換する。
- 4) `class` の中で、メソッド `add_two_number` が定義されている。メソッドの戻り型が整数である。
- 5) `x = add_two_number(a,b);`
上記のコードでは、`x` が `Double` 型なので、メソッドの戻り値は自動的に整数型から `Double` 型に変換してくれる。また、`println(x)` の中で、型の変換も行われている。

制御文

```
/* 【例3】 ファイル名 : Code3.java */
public class Code3 {
    public static void main (String[] args) {
        int a;
        for(a=1; a<=100;a++){
            if( a % 4 ==0 ){
                System.out.println(a);
            }else{
                System.out.println("-----" +
Integer.toString(a) );
            }
        }
    }
}
```

コードの説明

- 1) 制御文 if と for が使用されている。
- 2) 剰余算は% を使う。 例、 $2\%4$ が 2 となる。