

# デジタル表現論・第2回

劉 雪峰 (リュウ シュウフォン)

2016年4月18日

# 本日の目標

## Java プログラミングの基礎

1. 基本データ型（整数、浮動小数点、ブール、文字）
2. 変数
3. 配列
4. 制御文: for 文

# 変数 (Variable)

変数の例 :

---

```
/* 【例1】 ファイル名 : Code2_1.java */
public class Code2_1 {
    public static void main (String[] args) {
        int a=1, b=2;
        System.out.println("(Without variable) a+b is " + (1+2) );
        System.out.println("(With variable)      a+b is " + (a+b) );
        a=10;
        b=20;
        System.out.println("(With variable)      a+b is " + (a+b) );
    }
}
```

---

変数  $a, b$  の役割を考えよ。

- ▶ 2つ数  $a$  と  $b$  の和の計算について、 $a$  と  $b$  の値を修正する場合、コードの最初に  $a, b$  の値のみ修正すれば、他の部分を修正なし和の計算ができる。

# 変数と基本データ型

## 変数

- ▶ 数値や文字などのデータを格納する
- ▶ 変数を使用する前に、「宣言」が必要。例：「`int a;`」

## 基本データ型

Java 言語は以下の基本データ型を持っている。

- ▶ `boolean`: `true` または `false`
- ▶ `char`: 16 ビット Unicode UTF-16 の文字
- ▶ `byte`, `short`, `int`, `long`: それぞれは 8 ビット、16 ビット、32 ビット、64 ビットの整数である (符号 +/- 付き)
- ▶ `float`, `double`: それぞれは 32 ビット、64 ビットの浮動小数点である。(IEEE754 標準)

# 変数の名前

## 変数名のルール

- ・ アルファベットおよび “\_” (アンダースコア/アンダーライン) が使える。(例: Abc\_def)
- ・ 2文字目以降は 0-9 の数字が使える。(例: A1)
- ・ 予約語は使えない。
- ・ 大文字と小文字は区別される。(例: Abc と abc は異なる変数である。)

## Java 言語の予約語

abstract	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	enum	extends	final	finally
float	for	goto	if	implements	import
instanceof	int	interface	long	native	new
package	private	protected	public	return	short
static	strictfp	super	switch	synchronized	this
throw	throws	transient	try	void	volatile
while					

# 変数の宣言と初期値化

## 変数の宣言

変数を使用する前に、変数の宣言が必要である。宣言後、値を変数に代入することができる。

```
int a;  
int b,c;  
double x,y;  
a=10;  
x=1.1;  
b=c=1;
```

## 宣言する同時に値の代入ができる = 「初期化」

```
int a=11;  
int b, c=12;  
double x=20,y;
```

# 算術演算

- 1) 整数同士の演算結果は整数値となる。
- 2) 整数同士の割り算の場合は小数点以下が切り捨てられ整数値として計算される。

例:

---

```
int a=1, b=2, c;  
c = a/b; // cの値が0となる。
```

---

## 算術演算子

+ 加算	- 減算	* 乗算	/ 除算	% 剰余
------	------	------	------	------

例:

---

```
int a=12, b=10, c;  
c = a % b; // cの値が2となる。
```

---

# 演習課題 [1]

演習：以下の  $a, b$  に関する計算コードを書いてください。

```
a = 121; b = 11;  
c1 = (a + b) * (a - b)  
c2 = a2 - b2  
c3 = (a3 - b3) / (a - b)
```

以下のように出力しなさい。

```
c1 = (a+b)*(a-b) = ???;  
c2 = (a2-b2) = ???;  
c3 = (a3 - b3) / (a-b) = ???;  
// 「???' は実際の計算結果である。
```



# 配列

配列は同じ型の変数の集まりである。配列である変数の宣言は以下の形である。

データ型 [] 変数名

配列変数の初期化には「new」の使用が必要である。

整数の配列の例：

初期化方法 1

```
int [] a;  
a = new int [3];  
a [0]=5, a [1]=4, a [2]=3;
```

初期化方法 2

```
int [] a = new int [3];  
a [0]=5, a [1]=4;  
a [2]=3;
```

注意：

- ▶ "new int[3]" は長さが3である配列を作成する。即ち、配列の要素の数 = 3。
- ▶ 配列の要素をアクセスする時、インデックスは0から始まる。
- ▶ 配列 x の要素の数は x.length となる。
- ▶ 上記の例の配列について、a[3] は存在しない。

# 例

---

```
/* 【例2】 ファイル名 : Code2_2.java */
```

```
public class Code2_2 {  
    public static void main (String[] args) {  
        int[] x = new int[3];  
        x[0]=10;  
        x[1]=20;  
        System.out.println("The first element is x[0]=" + x[0]);  
        System.out.println("The last element is x[2]=" + x[2]);  
        System.out.println("Number of elements: " + x.length);  
    }  
}
```

---

注意 : x[2] は初期化されていないので、デフォルトの値は 0 である。

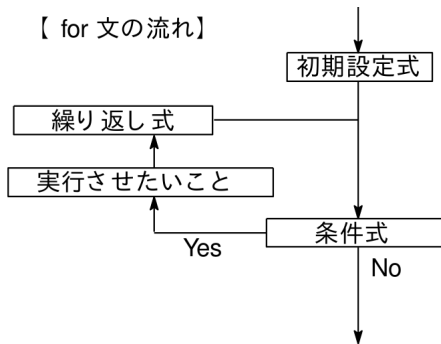
# 制御文: for 文

配列の要素を一つずつアクセス時、「for」文の使用は重要である。

## for 文

```
for (初期設定式 ; 条件式 ; 繰り返し式) {  
    繰り返し実行させたいこと  
}
```

【 for 文の流れ】



### 注意：

- ▶ for 文は ( ) の中に、2つの「;」で区切り3つの式を記述する。
- ▶ 初期設定式は最初に1回だけ実行される。通常、最初の値を代入しておく。

# for 文使用プログラム例

## 1 から 100 までの合計を求める

```
/* 【例2】 ファイル名 : Code2_3.java */

public class Code2_3 {
public static void main (String[] args) {

    int Sum =0, i;    /* 変数の宣言・初期化 */
    for (i=1; i<=100; i=i+1){/* for 文による繰り返し*/
        Sum = Sum+i; /* 変数Sumに値を1つ1つ加算する*/
    }
    System.out.println("Sum is " + Sum );/* 結果の表示 */

}
}
```

- ▶ 変数の宣言は最初にまとめて行う。
- ▶ for 後の { } は、 $i = 1, 2, \dots, 100$  となって繰り返し実行される。
- ▶ 「 $i=i+1$ 」は「 $i++$ 」と書いても同じ意味
- ▶ 「 $Sum=Sum+i$ 」は「 $Sum+=i$ 」としてもよい。

# レポート課題 1

以下のように数字を出力する。

```
1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4
5 5 5 5 5
6 6 6 6
7 7 7
8 8
9
```

## ヒント

下記のコードを参考・修正してから該当レポートのコードを作りなさい。

```
int i,j;
for(i=1;i<=9;i++){
    for(j=1;j<=9;j++){
        System.out.print(i + " ");
    }
    System.out.print("\n");//改行する。
}
```

## レポート課題2 (オプション)

四則演算の演算子のみを利用して剰余算の計算方法を考えなさい。

---

```
int a=17; int b=5; int c;
```

```
c = ?; // a=17,b=5時、c=a % b = 2
```

---