



# 独習 Java (第3版)

---

- 1.15 1次元配列
- 1.16 多次元配列
- 1.17 Javaキーワード

発表者 田島 勇樹



## 1.15 1次元配列(1)

---

- **1次元配列**とは…  
共通の名前を利用してアクセスすることが可能な、  
同じ型を持つ一連の変数のこと

配列内の個々の変数は**配列要素**と呼ばれる

配列は関連するデータのグループを扱うのに便利



## 1.15 1次元配列(2)

---

- 1次元配列の宣言

型 配列名[];

例 int ia[];

配列用の領域が作成されていない

- 1次元配列の領域割り当て

配列名 = new 型[サイズ];

例 ia = new int[10];

new演算子はオブジェクトの生成に用いる

# 1.15 1次元配列(3)

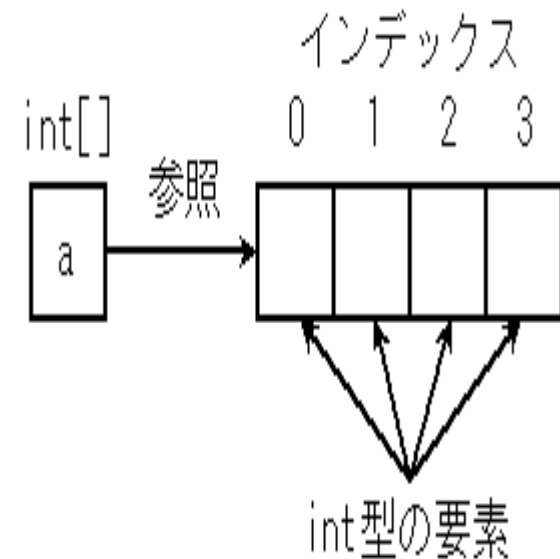
- 1次元配列の宣言と領域割り当ての組み合わせ

型 変数名[] = new 型[サイズ];

例 int a[] = new int[4];

int[] aのような記述も可

```
int[] a = new int[4];
```





## 1.15 1次元配列(4)

---

- 個々の要素にアクセス 配列にインデックス指定

例 `ia[2]=10;`

意味: ia配列の3つ目の要素に値10を代入

注意: 配列のインデックスは0からはじまる。



# 1.15 1次元配列(5)

- 1次元配列の要素数取得  
配列名.length

実行時に配列インデックスの境界  
チェックが実行  
配列の最後を越えて参照でき  
ない。

参照すると**例外**が生成

```
class test {  
    public static void main ( String  
        args[] ) {  
        int i;  
        //配列を宣言  
        int array[] = new int[ 5 ];  
  
        System.out.print( "配列arrayの  
            要素数は");  
        System.out.println( array.length  
            + "です。" );  
    }  
}
```

出力結果

配列arrayの要素数は5です。



## 1.15 1次元配列(6)

---

- 1次元配列の初期化

型 配列名[]={e0 ... en};

一つの配列変数を別の配列変数に代入することが出来る

配列を初期化する式 **配列初期化子**

例 `int j[]={0,1,2,3,4,5};`

`int k[];`

`k=j;`

ただし、配列そのものが物理的にコピーされるわけではない。



## 1.16 多次元配列(1)

---

- **多次元配列**とは・・・

2次元またはそれ以上の配列のこと  
配列の配列として実装される

配列を宣言し、配列の要素に領域を割り当てること  
は1次元配列と同じ手順

以降は2次元配列を例にして説明する



## 1.16 多次元配列(2)

---

- 2次元配列の宣言

型 配列名[][];

例 float fa[][];

- 2次元配列の領域割り当て

配列名 = new 型[サイズ1][サイズ2];

例 fa = new float[2][3];



## 1.16 多次元配列(3)

- 2次元配列の宣言と領域割り当て

型 配列名[][] = new 型[サイズ1][サイズ2]

例 float fa[][] = new float[2][3];

float [][]fa = new float[2][3];も可

float fa[][] = new float[2][3]

|          |          |          |
|----------|----------|----------|
| fa[0][0] | fa[0][1] | fa[0][2] |
| fa[1][0] | fa[1][1] | fa[1][2] |

図 2次元配列



## 1.16 多次元配列(4)

- **多次元配列の要素数取得**  
配列名.length  
第1次元の要素数が返される。
- **個々の配列の要素数取得**  
配列名[インデックス].length

```
class test {  
    public static void main ( String  
        args[] ) {  
        int i;  
        //配列を宣言  
        int array[][]= new int[5][10];  
        System.out.print( "配列arrayの要素  
        数は");  
        System.out.println( array.length +  
        "です。" );  
        System.out.println( array[0].length  
        + "です。" );  
    }  
}
```

出力結果  
配列arrayの要素数は5です。  
10です。



## 1.16 多次元配列 (5)

- 多次元配列の初期化

3つの要素を持つ配列の場合は・・・

型 配列名[][] = {{e00...e0x}, {e10...e1y}, {e20...e2z}};

配列名の最初の要素はe00からe0xまでの要素を持つ配列

2つ目の要素はe10からe1yまでの要素を持つ配列

3つ目の要素はe20からe2zまでの要素を持つ配列



# 1.17 Javaキーワード

## Javaキーワードの特徴

- ・変数名またはメソッド名として使えない
- ・キーワードは英字の小文字で表記されている

これらのキーワードに加えてtrue,false,null

|           |            |        |            |              |
|-----------|------------|--------|------------|--------------|
| abstract  | boolean    | break  | byte       | case         |
| catch     | char       | class  | const      | continue     |
| default   | do         | double | else       | 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  |            |              |

Javaのキーワード



# 練習問題

- 5人の学生がいる。それぞれの身長は167cm, 175cm, 159cm, 183cm, 160cmである。これらの平均を出し、人数と平均を出力せよ。ただし1次元配列で表し、人数は要素数と同じである。
- 2次元配列を使って1つ目の要素に3つ、2つ目の要素に4つ、3つ目の要素に2つずつ文字列をいれて配列と配列の要素の要素数を表示し、また要素をそれぞれ表示せよ。文字列は何でも良い。
- 次の宣言があるとき、下の各文1から5をコンパイル、実行したとき、正常に実行されるものをA～Eからすべて選べ。

1: int a[] = {1, 2, 3}; 2: int b[] = {4, 5, 6}; 3: short x = 2; 4: long y = 2;  
5: double z = 2;

解答[A. a[x]=10; B. a[y]=20; C. a[z]=30; D. a[3]=50; E. a=b;]