

10回目 配列

■ 今日の講義で学ぶ内容 ■

- ・配列とその使い方
- ・基本型変数と参照型変数
- ・エンハンスド for 文

配列

配列

同じ型である複数の変数を一括して管理する機能です
直観的には、



配列の準備

配列の準備手順

1. 配列変数の宣言
2. 配列要素の確保
3. 配列要素の初期化

1. 配列変数の宣言

配列変数とは配列を扱う変数です

配列変数は通常の変数と同様に型と識別子をもちます

型と識別子を指定して次のように行います

型 識別子[];

または

型[] 識別子; // Java での標準のスタイル

2. 配列要素の確保

配列要素とは値を格納するための一連の領域です
各配列要素は一つの変数としての機能をもちます

型と配列要素の個数を指定して次のように行います

```
識別子 = new 型 [ 配列要素の個数 ]:
```

⚠️ new 演算子は、指定された個数の配列要素をコンピュータのメモリ上に確保します

3. 配列要素の初期化

配列要素を確保した後に適当な値を入れておきます

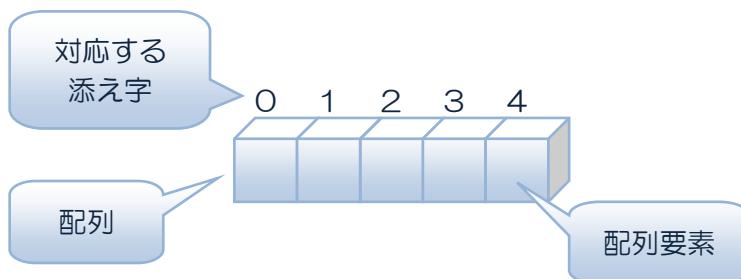
各配列要素の参照は、配列変数の識別子と添え字を用いて次のようにします

```
識別子 [ 添え字 ]
```

添え字には個々の配列要素の位置を表す値を指定します

- 最初の配列要素を参照するには、添え字に 0 を指定します
- 最後の配列要素を参照するには、添え字に 配列要素の個数 - 1 を指定します

例えば、5個の配列要素を持つ場合は次のように0から4までの添え字を指定します



配列要素の初期化は、各配列要素を参照して次のように行います

```
識別子 [ 添え字 ] = 値:
```

⚠️ 添え字に指定する値は int 型である必要があります

⚠️ 配列要素は初期化を行う以前に予め以下の値が代入されています

(型)	(デフォルト値)
boolean	false
char	0 (¥u0000)
byte、short、int、long	0
float、double、	0.0

ソースコード例

ソースファイル名 : Sample10_1.java

```
// 配列を用いて5人の学生の点数を管理する
class Sample10_1
{
    public static void main(String[] args)
    {
        int i;

        // 配列変数の宣言
        int test[];
        // int[] test; とも記述可能

        // 配列要素の確保
        test = new int[5];

        // 配列変数の宣言と配列要素の確保は同時に記述可能
        // int test[] = new int[5];
        // int[] test = new int[5];

        // 各配列要素の初期化（兼、値の代入）
        test[0]=80;
        test[1]=60;
        test[2]=22;
        test[3]=50;
        test[4]=75;

        // 各配列要素（添え字は 0 から 4 まで）を順番に出力
        for(i=0; i<5; i++)
            System.out.println((i+1)+"番目の学生の点数は"+test[i]+"です。");
    }
}
```

実行画面

```
>java Sample10_1
1 番目の学生の点数は 80 です。
2 番目の学生の点数は 60 です。
3 番目の学生の点数は 22 です。
4 番目の学生の点数は 50 です。
5 番目の学生の点数は 75 です。
-- Press any key to exit (Input "c" to continue) --
```

ソースコード例

ソースファイル名 : Sample10_2.java

```
// 配列要素の動的な確保
import java.io.*;
class Sample10_2
{
    public static void main(String[] args) throws IOException
    {
        int num;      // 学生数
        int[] test;   // 学生の点数を保存する配列変数

        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("学生の人数を入力してください。");
        num=Integer.parseInt(br.readLine());

        // 学生数分の配列要素を確保
        test = new int[num];

        // キーボードから点数を配列要素へ順番に入力する
        for(int i=0;i<num;i++)
            test[i]= Integer.parseInt(br.readLine());

        // 配列要素に入力されている点数を順番に出力する
        for(int i=0;i<num;i++)
            System.out.println((i+1)+"番目の学生の点数は"+test[i]+"です。");
    }
}
```

実行画面

```
>java Sample10_2
学生の人数を入力してください。
3
89
75
95
1 番目の学生の点数は 89 です。
2 番目の学生の点数は 75 です。
3 番目の学生の点数は 95 です。
-- Press any key to exit (Input "c" to continue) --
```



配列要素の確保されていない領域へアクセスしたら？

ソースファイル名：Ext1O_1.java

```
// 添え字の範囲のミス
class Ext1O_1
{
    public static void main(String[] args)
    {
        int i;

        // 配列変数の宣言と 5 個の配列要素の確保
        int[] test = new int[5];

        // 6 番目の配列要素へアクセス
        for(i=0; i<6; i++)
            System.out.println((i+1)+"番目の配列要素は"+test[i]+"です。");
    }
}
```

実行画面

```
>java Ext1O_1
1 番目の配列要素は 0 です。
2 番目の配列要素は 0 です。
3 番目の配列要素は 0 です。
4 番目の配列要素は 0 です。
5 番目の配列要素は 0 です。
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at Ext1O_1.main(Ext1O_1.java:13)
-- Press any key to exit (Input "c" to continue) --
```

配列の初期化

配列の初期化

指定された値の列を用いて配列要素を確保し同時に初期化します

配列の初期化は、配列変数の宣言時に次のように行います
指定する値の列が代入された配列要素をもつ配列変数が宣言されます

```
型 [識別子] = { 値1, 値2, 値3, ..., 値n };
```

または

```
型[] [識別子] = { 値1, 値2, 値3, ..., 値n }; // Java での標準のスタイル
```

ソースコード例

ソースファイル名 : Sample10_3.java

```
// 配列の初期化
class Sample10_3
{
    public static void main(String[] args)
    {
        // 配列の初期化
        int test[]={80,60,22,50,75};
        // int[] test={80,60,22,50,75}; とも記述可能

        // 配列要素を出力
        for(int i=0;i<5;i++)
            System.out.println((i+1)+"番目の学生の点数は"+test[i]+"です。");
    }
}
```

実行画面

```
>java Sample10_3
1 番目の学生の点数は 80 です。
2 番目の学生の点数は 60 です。
3 番目の学生の点数は 22 です。
4 番目の学生の点数は 50 です。
5 番目の学生の点数は 75 です。
-- Press any key to exit (Input "c" to continue) -
```

配列要素の初期化 指定された値の列を用いて配列要素を初期化します

配列要素の初期化は、配列要素の確保時に次のように行います
指定する値の列が代入された配列要素が確保されます

```
型[] 識別子 = new 型[] { 値1, 値2, 値3, …, 値n };
```

⚠ 配列の初期化では配列要素の数を指定しません
配列要素の数は値の列より自動的に計算されます

配列変数と参照型変数

変数には

基本型変数

値として“値”そのものをもつ
例えば、int型変数、double型変数など



参照型変数

値として“値のある場所”をもつ
例えば、配列変数、クラス型変数など



参考：例題 Sample10_1 での配列変数の宣言と配列要素の確保は次のように理解できる

①配列変数の宣言
int[] test;

②配列要素の確保
new int[5];

③配列要素を配列変数へ代入
test = new int[5];

ソースコード例

ソースファイル名 : Sample10_4.java

```
// 配列変数へ代入するということは？
class Sample10_4
{
    public static void main(String[] args)
    {
        int i;
        int n1=1, n2;
        int[] ary1={1,2,3}, ary2;

        // int 型変数へ代入
        n2=n1;
        // 配列変数へ代入
        ary2=ary1;

        // 変数と配列要素を出力
        System.out.println("n1=" + n1 + ", n2=" + n2);
        System.out.print("ary1={");
        for(i=0;i<3;i++)
            System.out.print(ary1[i] + " ");
        System.out.println("}");

        System.out.print("ary2={");
        for(i=0;i<3;i++)
            System.out.print(ary2[i] + " ");
        System.out.println("}");

        // 一方の int 型変数の値を変更
        System.out.println("n2 := 2;");
        n2=2;
        // 一方の配列要素の値を変更
        System.out.println("ary2[1] := 4;");
        ary2[1]=4;

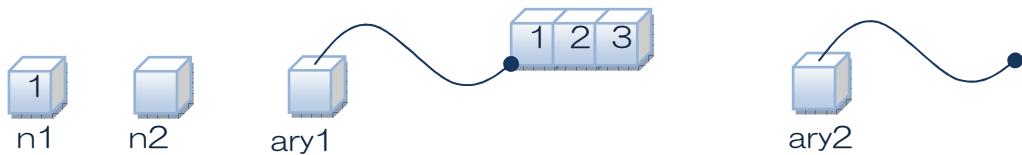
        // 変数と配列要素を出力
        System.out.println("n1=" + n1 + ", n2=" + n2);
        System.out.print("ary1={");
        for(i=0;i<3;i++)
            System.out.print(ary1[i] + " ");
        System.out.println("}");

        System.out.print("ary2={");
        for(i=0;i<3;i++)
            System.out.print(ary2[i] + " ");
        System.out.println("}");
    }
}
```

実行画面

```
>java Sample10_4
n1=1, n2=1
ary1={1 2 3 }
ary2={1 2 3 }
n2 := 2;
ary2[1] := 4;
n1=1, n2=2
ary1={1 4 3 }
ary2={1 4 3 }
-- Press any key to exit (Input "c" to continue) --
```

参考：例題 Sample10_4 の変数の振る舞いは次のように図的に理解できる
(変数の宣言時)



(変数へ代入後)



(値の変更後)



エンハンスド for 文

Java では配列を処理するための特別な for 文が準備されています

エンハンスド for 文 配列に格納されている値が順番に取り出されます
取り出された値が変数に代入され文が処理されて、繰り返します

```
for( 型 変数名 : 配列変数名 ) 文
```



配列の終端に到達するまで for 文は繰り返されます

1回目の繰り返し：配列の第1要素の値が変数に代入されます

2回目の繰り返し：配列の第2要素の値が変数に代入されます

⋮

n回目の繰り返し：配列の第n要素の値が変数に代入されます

ソースコード例

ソースファイル名：Ext10_2.java

```
// エンハンスド for 文
class Ext10_2
{
    public static void main(String[] args)
    {
        // 配列の初期化
        int[] test={80,60,22,50,75};

        // 配列要素を出力
        for(int num:test)
            System.out.println(num);
    }
}
```

実行画面

```
>java Ext10_2
80
60
22
50
75
-- Press any key to exit (Input "c" to continue) --
```