

【実習 7】 配列

配列処理をします。

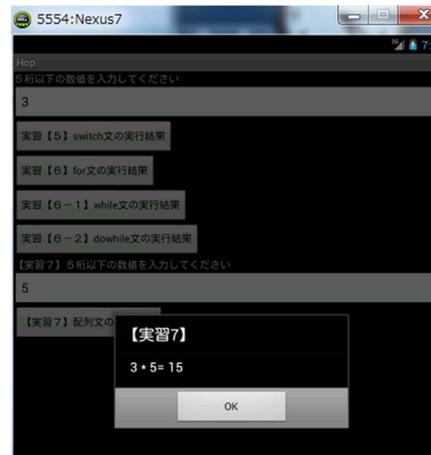
1 ソースコード

テンプレート Hop020 の「// 【実習 7】 配列」の箇所へ次のソースを追加入力します。

```
int q[] = new int[3]; //要素数が3つの int 型の配列 q を用意する
q[0] = num1; //q の 0 番に num1 を代入する
q[1] = num2; //q の 1 番に num2 を代入する
q[2] = q[0] * q[1]; //q の 2 番に 0 番と 1 番の積を代入する
//計算結果を表示する
showDialog(this, "【実習 7】", Integer.toString(q[0]) + " * " + q[1] + "
                = " + q[2]);
```

2 実行結果

「[num1 の数値] * [num2 の数値] = 」が表示されます。画面は、num1 の数値が 1、num2 の数値が 5 の場合です。



3 解説

配列 (array) は同じ型の値をまとめて記憶する機能を持ちます。

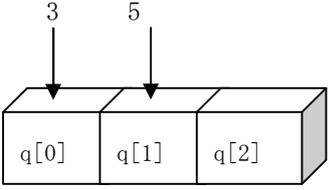
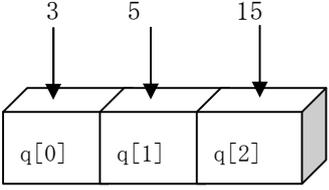
配列の宣言：
 型 配列名[] = new 型[要素の数]

※[] : ブラケットといいます。

配列につける名前を配列名といい、個々の配列の要素は添字 (インデックス) という番号で区別されます。添え字は 0 番から順番に配列の要素にふられていきます。

添字には int 型変数が使用できます。例えば、int n=1; int p=2; が設定した時の q[p]=q[n]; は q[2]=q[1] と同じ意味です。

<pre>int q[] = new int[3];</pre> <p>要素数が 3 の配列 q が用意されます。添え字は 0 番からふられるため、各要素には 0, 1, 2 と添え字が付きます。</p>	
<pre>q[0] = num1;</pre> <p>q の 0 番目に num1 の数値 (例 3) が代入されます。</p>	

<p><code>q[1] = 5;</code></p> <p>q の 1 番目に 5 が代入されます。</p>	
<p><code>q[2] = q[0] * q[1];</code></p> <p>0 番目と 1 番目の値の積 15 が 2 番目に代入されます。</p>	

q を「配列変数」といいます。int 型などの変数は、値を格納する記憶領域「そのもの」を表していますが、配列変数は値を格納する記憶領域が「メモリ上のどこに存在しているか」という配列の「場所」を表しているものとなります。このような種類の変数のことを「参照型の変数」と呼びます。

4 配列の利用

- ・各要素を指定するときの添字に変数を使います。
- ・配列の大きさを超えた要素は利用することができません。

上記の例では `q[3]` は利用できません。

5 配列の初期化

配列を初期化すると、配列の宣言、要素の確保、値の代入が同時に行えます。

```
型名 配列変数名[] = {値1, 値2, 値3, ...};
```

例 `int q[] = {3, 5, 15};` ...new 演算子を省略できる

6 2次元配列

配列の宣言：

```
型 配列名[][] = new 型[要素の数][要素の数]
```

例 `int q[][] = new int[3][2];` ...2次元配列の作成

7 2次元配列の初期化

配列を初期化すると、配列の宣言、要素の確保、値の代入が同時に行えます。

```
型名 配列変数名[][] = {{値1, 値2, 値3, ...}};
```

例 `int q[][] = {{3, 5, 15}, {6, 10, 30}};` ...new 演算子を省略できる

【演習】**■ 次のプロジェクトとプログラムを作成してください。**

プロジェクト名 : HopPro???? (???? : 年組席) (既存)

アプリケーション名 : HopEx070View

①Hop010View のソースをコピーして、Hop010View の2箇所の文字を、HopEx070View に変更してください。

②テンプレート(Hop010.java) の①の箇所に HopEx070View を入力してください。

③Hop010.java を実行してください。

【演習71】

配列を利用して次のようなメッセージを出力するコードを作成してください。

1 番目の人の点数は 80 です。
2 番目の人の点数は 60 です。
3 番目の人の点数は 22 です。
4 番目の人の点数は 50 です。

【演習72】

中間テストの各自 (A さん、B さん) の科目別点数が次の表で与えられるときに、配列を利用して、各科目ごとの平均点を求めるコードを作成してください。

	国語	社会	数学	理科	英語
A さんの点数	67	78	72	91	71
B さんの点数	86	69	66	87	81
科目平均点					

ヒント

■ 列順に表示していく場合

```
double ave; //平均点の変数
int i;
int x[] = {67, 78, 72, 91, 71};
int y[] = {86, 69, 66, 87, 81};
String k[] = {"国語", "社会", "数学", "理科", "英語"};
for(i=0; ~「省略」~
    ave=(x[i]+y[i])/2.0 //小数まで算出する場合、2.0にする
    以下 省略
```

■ 行順に表示していく場合

```
double ave[]=new double[5];
int i;
int x[] = {67, 78, 72, 91, 71};
```

```
int y[] = {86, 69, 66, 87, 81};
String k[] = {"国語", "社会", "数学", "理科", "英語"};
for(i=0; ~「省略」~
    ave[i]=(x[i]+y[i])/2.0 //小数まで算出する場合、2.0にする
    以下 省略
```

■次のプロジェクトとプログラムを作成してください。

プロジェクト名 : HopPro???? (???? : 年組席) (既存)

アプリケーション名 : HopEx071View

①Hop070Viewのソースをコピーして、Hop070Viewの2箇所の文字を、HopEx071Viewに変更してください。

②テンプレート(Hop010.java)の①の箇所に HopEx071View を入力してください。

③Hop010.javaを実行してください。

【演習73】

ラーメン屋の1週間の売上が、次のような表になるとき、1週間の合計数量と売上金額を求めて表示するコードを作成してください。

種類	単価	月	火	水	木
しょうゆ	¥550	72	75	88	56
味噌	¥650	60	62	45	35
塩	¥600	80	75	86	83
ねぎ	¥750	50	45	46	23
ワカメ	¥700	35	37	23	24

種類	金	土	日	合計数量	売上金額
しょうゆ	95	98	99		
味噌	54	77	68		
塩	88	87	84		
ねぎ	51	39	45		
ワカメ	43	41	51		

ヒント：¥記号の表示・・・"¥u00A5"

【演習74 進んだ学習】

配列を使って九九の表を表示するコードを作成してください。

条件：二次元配列を使用してください。

*****九九の表*****

```

1  2  3  4  5  6  7  8  9
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63

```

画面サイズ：800×1183

種類	単価	月	火	水	木
しょうゆ	550	72	75	88	56
味噌	650	60	62	45	35
塩	600	80	75	86	83
ねぎ	750	50	45	46	23
ワカメ	700	35	37	23	24

種類	全	土	日	合計数量	売上金額
しょうゆ	95	98	99	583	320650
味噌	54	77	68	401	260650
塩	88	87	84	583	349800
ねぎ	51	39	45	299	224250
ワカメ	43	41	51	254	177800

***** 配列九九 *****

```

1  2  3  4  5  6  7  8  9
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81

```

■次のプロジェクトとプログラムを作成してください。

プロジェクト名 : HopPro???? (???? : 年組席) (既存)

アプリケーション名 : HopEx072View

①Hop071Viewのソースをコピーして、Hop071Viewの2箇所の文字を、HopEx072Viewに変更してください。

②テンプレート(Hop010.java)の①の箇所に HopEx072View を入力してください。

③Hop010.javaを実行してください。

【演習75 進んだ学習】

演習72の問題を二次元配列を使用してプログラミングしてください。

中間テストの各自(Aさん、Bさん)の科目別点数が次の表で与えられるとき、各科目ごとの平均点を求めるコードを作成してください。

【演習76 進んだ学習】

演習73の問題を二次元配列を使用してプログラミングしてください。

ラーメン屋の1週間の売上が、次のような表になるとき、1週間の合計数量と売上金額を求めて表示するコードを作成してください。

【解説】

■Javaの文法

概念	意味	例	キーワード
クラス	モノに関するデータ・機能をまとめるしくみ	車	<pre>class Car { int num; double gas; }</pre>
フィールド	モノに関するデータをあらわすしくみ	ナンバー ガソリン量	
メソッド	モノに関する機能・操作をまとめるしくみ	ナンバーを決める ガソリンを入れる	<pre>public static void main(string args[]) { Car car1; car1=new Car(); car1.num=1234;</pre>
メンバ	フィールドとメソッドのこと		
オブジェクト (インスタンス)	クラスをもとに作成される実際のモノのこと	コード上で作成される車1台1台のこと	car1=new Car();
コンストラクタ	そのクラスのオブジェクトが作成されたときに定義しておいたコンストラクタ内の処理が自動的に行われる	オブジェクトを初期化するためのしくみ	<pre>Public Car() { num=0; gas=0.0; }</pre>
拡張・継承	あるクラス(スーパークラス)のメンバを受け継ぐクラス(サブクラス)を設計すること	「車」クラス・・・スーパークラス 「レーシングカー」クラス・・・サブクラス	<pre>class RacingCar extends Car { }</pre>
スーパークラス	拡張されるクラスのこと		
サブクラス	スーパークラスを拡張したクラスのこと		
インターフェイス・抽象クラス	メソッドの名前や定数をまとめるしくみ	オブジェクトを作成することはできない	<pre>interface iVehicle { void show(); }</pre>
実装	あるインターフェイスで宣言されているメソッドの内容をクラスで実際に定義すること		implements
オーバーロード	引数・型の異なる同名のメソッドを定義すること	似たような複数の処理をオーバーロードしておけば1つのメソッド名を覚えて使うだけで自動的にその型・個数に応じた処理が行われる	
オーバーライド	スーパークラスのメソッドにかわってサブクラスのメソッドが機能すること		

多態性	1つのメソッド名がその状況に応じていろいろな働きをすること		
パッケージ	クラスの名前を分類するしくみ		package
インポート	パッケージ内のクラス名を取り込むしくみ		import
修飾子	クラスやメンバへのアクセス範囲を指定するキーワードのこと		private public