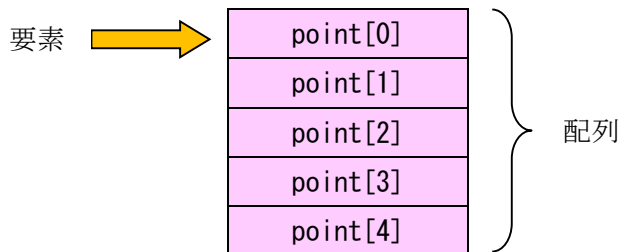


2023 年 11 月 9 日 (木) 実施

## 配列

同種のデータ型を有する複数のデータ (要素) を番号付けして、ひとまとまりの対象として扱うものを配列と呼ぶ。



配列の取り扱いに関して、次のような特徴がある。

1. プログラム中で用いる配列変数 (配列の本体を参照する参照型の変数) は必ず宣言しておく。

例) `int[] point; /* int 型の配列変数 point を宣言 */`

2. 配列の本体を生成し、配列変数に代入して参照させる。

例) `point = new int[5]; /* int 型の 5 個の要素を持つ配列を生成 要素数は配列が生成された時点で定まる */`

3. 配列変数を宣言する際の初期化で、配列の本体を生成することが出来る。

例) `int[] point = new int[5];`

4. 配列の生成時に、各要素には初期値 (int 型の場合は 0, char 型の場合は `\u0000`, boolean 型の場合は `false`, 参照型の場合は `null`) が設定される。

5. 配列変数を宣言する際に、中括弧 { } の間に初期値を与えて初期化することが出来る。

例) `int[] point = { 98, 76, 85, 67, 59 };` この場合には、`new` で配列を生成する必要はない。

6. 要素数を `n` とすると、配列の要素は 0 番から `n-1` 番までの添え字を用いて表され、各要素は変数と同等に扱うことが出来る。

### char 型の配列

Java 言語では、char 型の配列は文字列の扱いとはならない。一方、String クラスの変数を宣言して初期化するには次の様な書き方をすれば良かった。

例) `String str = "CUC";`

char 型の配列を用いて上の例と同等な宣言を行うには

```
char[] charray = {'C', 'U', 'C'};
String str = new String(charray);
```

とする。

### 例題 1 (int 型のデータによる配列)

次のプログラムは、標準入力装置から入力された点数によってランク分けを行い、ランクに応じた成績を表示させるものである。これを入力して、実行せよ。なお、データは 5 個分入力する必要がある。ここで、クラス名は Sample5\_1, ソースファイル名は Sample5\_1.java とする。

```
import java.util.Scanner;

public class Sample5_1 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM=5;
        int[] point = new int[NUM];
        int[] rank = new int[NUM];
        Scanner sc = new Scanner(System.in);

        for (int i=0; i<NUM; i++) {
            System.out.println((i+1) + "番目の点数(0~100)を入力して下さい。");
            point[i] = sc.nextInt();

            if (point[i] < 0 || point[i] > 100) {
                System.out.println("入力された点数は範囲外です。");
                System.exit(1);
            }
        }

        System.out.println("入力終了\n\n[結果表示]");

        for (int i=0; i<NUM; i++) {
            System.out.println((i+1) + "番目の点数:" + point[i]);

            rank[i] = point[i]/10;

            switch (rank[i]) {
                case 6: System.out.println("⇒ 成績は可です。"); break;
                case 7: System.out.println("⇒ 成績は良です。"); break;
                case 8: System.out.println("⇒ 成績は優です。"); break;
                case 9:
                case 10: System.out.println("⇒ 成績は秀です。"); break;
                default: System.out.println("⇒ 成績は不可です。");
            }
        }
    }
}
```

## 【解説】

1. 配列の要素の添え字は 0 番から始まるため、ループ変数  $i$  は 0 で初期化している。
2. 配列の要素の添え字は [要素数] - 1 番までなので、for 文の継続条件は  $i$  が NUM 未満となる。
3. System.out.println で『番目の点数』の箇所で  $(i+1)$  としているのは、 $i$  の初期値が 0 となっているが、画面表示では 1 番目から始めたいからであり、括弧で括っているのは、数値としての足し算を行った上で他の文字列と結合するためである。(括弧で括らないと、01 番目、11 番目の様に表示される。)

## 例題 2 (double 型のデータによる配列)

次のプログラムは、標準入力装置から入力された実数データの平均値を求め、表示させるものである。これを入力して、実行せよ。なお、データは 5 個分入力する必要がある。ここで、クラス名は Sample5\_2, ソースファイル名は Sample5\_2.java とする。

```
import java.util.Scanner;

public class Sample5_2 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM=5;
        double sum=0.0, ave;
        double[] data = new double[NUM];
        Scanner sc = new Scanner(System.in);

        for (int i=0; i<NUM; i++) {
            System.out.print((i+1) + "番目の実数データを入力してください: ");
            data[i] = sc.nextDouble();

            sum = sum + data[i];
        }

        ave = sum / NUM;

        System.out.println("入力終了\n\n[結果表示]");

        for (int i=0; i<NUM; i++)
            System.out.println((i+1) + "番目のデータ: " + data[i]);

        System.out.println("データの合計: " + sum);
        System.out.println("データの平均値: " + ave);
    }
}
```

## 例題 3 (String 型のデータによる配列)

次のプログラムは、標準入力装置から入力された氏名を配列によって格納し、編集して表示させるものである。これを入力して、実行せよ。なお、氏名は 5 人分入力する必要がある。ここで、

クラス名は `Sample5_3`, ソースファイル名は `Sample5_3.java` とする。

```
import java.util.Scanner;

public class Sample5_3 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM=5;
        String[] str = new String[NUM];
        Scanner sc = new Scanner(System.in);

        for (int i=0; i<NUM; i++) {
            System.out.print((i+1) + "人目の氏名を入力してください: ");
            str[i] = sc.nextLine();
        }

        System.out.println("入力終了\n\n[結果表示]");

        for (int i=0; i<NUM; i++)
            System.out.println(" Name (No. " + (i+1) + ") : " + str[i]);
    }
}
```

**【解説】** `nextLine()` は 1 行分の文字列を読み込む。これまで用いていた `next()` の場合は、文字列に空白やタブが含まれるとそれらを区切り記号とみなし、その前までの文字列を読み込む。

### 演習

次のプログラムは、標準入力装置から A, I, U, E, O の文字を組み合わせる **5 文字以上** の文字列を入力すると、入力された文字列の **先頭の 5 文字分及び変換された平仮名** を表示させるものである。但し、変換ルールが無い場合には、**平仮名の代わりに□ (四角)** が用いられる。このプログラムリストの空欄に **適切な語句を埋めて** 完成させたプログラムを入力し、実行せよ。ここで、クラス名は `Ex5`, ソースプログラム名は `Ex5.java` とする。

```
import java.util.Scanner;

public class Ex5 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM=5;
        String str1;
        1) _____ romaji = new char[NUM];
        2) _____ hiragana = new char[NUM];
        Scanner sc = new Scanner(System.in);
    }
}
```

```
System.out.println("A, I, U, E, O の文字を組み合わせて 5 文字以上の文字列を入力して下さい。");
str1 = sc.nextLine();

for (int i=0; i<NUM; i++) {
    romaji[i] = str1.charAt(i);

    switch (3) _____ ) {
        case 'A':
        case 'a': hiragana[i] = 'あ'; break;
        case 'I':
        case 'i': hiragana[i] = 'い'; break;
        case 'U':
        case 'u': hiragana[i] = 'う'; break;
        case 'E':
        case 'e': hiragana[i] = 'え'; break;
        case 'O':
        case 'o': hiragana[i] = 'お'; break;
        default : hiragana[i] = '□';
    }
}

System.out.println("¥n[結果表示]");

String str2 = new String(romaji);
String str3 = new String(hiragana);
System.out.print("入力された文字列の先頭 5 文字分は:" + str2);
System.out.print("¥n 変換された平仮名は:" + str3);
}
}
```

**提出物 :**

- 1) **例題 1**, **例題 2**, **例題 3** 及び**演習**のプログラムの**コンソールへの出力結果**をコピーして貼り付けた**テキストファイル res5.txt** をメールに添付する。
- 2) **演習**のソースプログラムのファイル **Ex5.java** をメールに添付する。
- 3) 第 5 回の理解度確認用の**質問ファイル Prog1\_Questions\_5th.txt** に解答を記入して、メールに添付する。

\* メールのは名は『**プログラミング 1 第 5 回課題**』(鍵括弧は要らない) とする。