

1 Java プログラムの実行/標準出力/文字列/配列

```

1: class MyClass {
2:     public static void main(String[] args) {
3:         int a, b;
4:         String str1, str2;
5:
6:         a = 1;
7:         b = 2;
8:
9:         str1 = "Hello, ";
10:        str2 = "World!";
11:
12:        System.out.printf("%d\n", a + b);
13:        System.out.printf("%s\n", str1 + str2);
14:
15:        System.out.println(str1 + str2);
16:        System.out.println(a + str1 + b + str2);
17:
18:        int[] test;
19:        test = new int[5];
20:
21:        test[0] = 80;
22:        test[1] = 60;
23:        test[2] = 22;
24:        test[3] = 50;
25:        test[4] = 75;
26:
27:        for(int i=0; i<test.length; i++) {
28:            System.out.println(i + ": " + test[i]);
29:        }
30:    }
31: }

```

【Point 1】 main() という関数 (Javaではメソッドと呼ぶ) から実行される (p.22)

【Point 2】 文字列を扱うための型 (正確にはクラス) Cの様に「文字の配列」ではない (p.303)

【Point 3】 文字列は、=で代入可能 (p.306)

【Point 4】 標準出力への出力には、Cと同じprintf()が使える

【Point 5】 「+」演算子は、文字列を連結することもある (p.80)

【Point 6】 出力には、println()を使うとより簡単に出力できる (p.19)

【Point 7】 int型とString型 (クラス) 同士でも「+」演算子は使える (この場合はintがStringに変換される)

【Point 8】 配列変数 (配列本体を参照する変数) を宣言する (p.178)

【Point 9】 配列本体 (要素となるハコが連続したもの) を作る (p.179 図7-2) 作る要素数には変数も利用可能 (p.186 Sample2.java)

【Point 10】 配列変数は、「length」を使って要素数を取得できる (p.199)

【練習 1-1】 テキストエディタを使って、上記の Java プログラムを入力してみましょう。作成したプログラムの名前は、**拡張子「java」で保存して下さい。**

【練習 1-2】 入力したプログラムを、次のように「javac」コマンドを使ってコンパイルしてみましょう。

```
$ javac 「保存した Java ファイル名」.java
```

【練習 1-3】 コンパイル後にクラスファイルが生成されます。次のようにファイルを確認してみましょう。

```
$ ls
MyClass.class  MyClass.java
```

実行するには、**main() が含まれたクラスファイル**を指定します。今回は次のように入力して実行します。

```
$ java MyClass
```

【課題 1-1】 先程の main() 内の処理を改良して、「配列 test の全要素の値の合計を計算し、出力する」Java プログラムを作成してください。

[実行結果]
合計は 287

【課題 1-2】 先程の main() 内の処理を改良して、「配列 test の全要素の値の中から最大値を調べて、その値を出力する」Java プログラムを作成してください。

[実行結果]
最大値は 80

Java の特徴

Java とは Sun Microsystems 社が開発したプログラミング言語です（現在は Oracle 社に買収され開発が続いています）。次のような特徴があります。

オブジェクト指向言語である オブジェクト指向プログラミングとは、物の持つ情報（変数）とその振舞い（関数）を 1 つの部品（オブジェクト）として作成し、これらの部品を組み合わせてプログラミングをするという考え方です。

プラットフォーム独立である Java は、コンパイル後のバイナリファイル（バイトコードと呼ばれる言語で書かれている）において、計算機の機種（CPU や OS の種類）に依存しないプラットフォーム独立な言語です。つまり、ある機種でコンパイルして得たバイナリファイルを、他の機種でコンパイルし直すことなくそのまま実行することができます。

Java の利用形態

Java プログラムは、次のように様々な形態で利用されます。この授業ではアプリケーションの形態で Java プログラミングを学びます。

アプレット これは WWW を経由してクライアント側の Web ブラウザ内でダウンロードされ実行されるプログラムです。ただし、Web ブラウザには Java アプレットを実行できる機能が必要です。

アプリケーション コンピュータ上で実行される通常のプログラムです。スタンドアロンアプリケーションとも呼ばれます。これはアプレットとは異なり実行する際に Web ブラウザを必要としません。

サーブレット, JSP (Java Server Pages) Java プログラムがサーバ側で実行されて、クライアント側には実行結果（例えば HTML データ）のみが返ってくるプログラムです。

Java の開発環境の種類

Java プログラムを開発するには、SDK (Software Development Kit) と呼ばれる開発環境を使います。SDK には用途に応じて 3 種類用意されています。この授業では Java SE を使います。

Java SE (Java Platform, Standard Edition) スタンドアロンアプリケーション向け

Java EE (Java Platform, Enterprise Edition) サーブレット向け

Java ME (Java Platform, Micro Edition) 携帯電話, PDA 向け

Java の開発環境を自分の PC にインストールするなら…

Java SDK Windows, Linux 向けに、Java をコンパイル・実行するためのコマンドを Java の開発元である Oracle 社から無償でダウンロードできます。Mac OS X 向けには Apple が配布しています。標準でこれらのコマンドが組み込まれています。

<http://www.oracle.com/technetwork/java/javase/overview/index.html>

Eclipse Eclipse とは、Java 言語向けのプログラミング、デバッグなどを実現する開発環境です。Java SDK とは異なり、エディタやデバッガなど Java で開発するために必要な機能が一通りそろったツールです。以下から無償でダウンロードできます。

<http://www.eclipse.org/> (公式 Web サイト)

<http://mergedoc.sourceforge.jp/> (Windows 用日本語版の配布サイト)