

Java言語の概要

Javaの歴史

- Sun Microsystems社で開発され、1995年に発表された新しいプログラム言語
- Javaの原形はOak(オーク)という言語
- Oak: 家電製品をコントロールする目的の言語から、ネットワーク対応言語へ
- Java Coffee(ジャワ・コーヒー)から、Java(ジャバ)言語へ
- Javaに対応したSun製Webブラウザ「HotJava」…Netscape NavigatorもJavaサポート
- 1996年、Java開発キット「JDK 1.0(Java Development Kit)」を無償公開
- Netscape Navigator 2.0から動作する「JavaScript」とは無関係、別物!

Javaの仕組み

- 「Write Once, Run Anywhere: 一度作成すれば、どこでも実行できる」
- Javaプログラムを実行させるにはJVM(Java Virtual Machine: Java仮想マシン)が必要
- Java以外のプログラム(C, C++等)では
 - a. 「ソースプログラム」の作成
 - b. 【各言語コンパイラ】
 - c. 「CPUに依存したマシンコード」
 - d. (実行)
- Javaプログラムでは(アプリケーションの場合)
 - a. 「ソースプログラム」の作成
 - b. 【Javaコンパイラ: javac】
 - c. 「CPUに依存しないJavaバイトコード」
 - d. 【Javaインタプリタ: java】
 - e. (実行)
- Javaアプリケーション: 一般のコンパイラと同様に、それ自身で動作する形式のプログラム
 - a. ~.java: 「アプリケーション・ソースプログラム」の作成
 - b. 【Javaコンパイラ: javac】
 - c. ~.class: 「CPUに依存しないJavaバイトコード」
 - d. 【Javaインタプリタ: java】
 - e. (実行)
- Javaアプレット: ユーザ側のWebブラウザ上で動作する形式のプログラム
 - a. ~.java: 「アプレット・ソースプログラム」の作成
 - b. 【Javaコンパイラ: javac】
 - c. ~.class: 「CPUに依存しないJavaバイトコード」
 - d. ~.html: 「~.classを定義したHTMLファイル」の作成
 - e. ~.html: (Webブラウザ上で実行) or (appletviewerで実行)
- Javaサーバアプレット: サーバ上で動作するプログラム

データを出力する (アプリケーション)

簡単なアプリケーションプログラムの例

```
public class Hello { // クラス名を Hello にする
    public static void main(String args[]) { // 決まり文句
        System.out.println("Hello, Java world!"); // 文字列出力
    }
}
```

まずこれを適当なエディタで入力してください。ただし行末の // から右側はコメント(注釈)ですから記述しなくてもかまいません。ファイル名は【クラス名+.java】の **Hello.java** にします。Java言語のプログラムは、ファイル名の最後に必ず **.java** を付けます。

次に、この **Hello.java** を保存したディレクトリで

```
javac Hello.java
```

と打ち込んでください。

すると、Javaコンパイラ(javac)が、この **Hello.java** というファイルを、Javaバイトコードの **Hello.class** というファイルに変換してくれます。

こうしてできた **Hello.class** を実行するには、

```
java Hello
```

のように打ち込みます。すると、Javaインタプリタ(java)が **Hello.class** を読み込み実行します。

●ポイント

1. Javaソースプログラムはテキストファイル。エディタで作成する。
2. プログラムファイルには **~.java** という名前をつける。
3. Javaプログラムではメインクラス名とファイル名(**~.java** の **~**)を一致させる。
4. Javaプログラムのコンパイルは **javac** コマンドで行う。
5. Javaプログラム(アプリケーション)の実行は **java** インタプリタコマンドで行う。
6. ファイル名、プログラム中の識別子は、大文字と小文字が厳密に区別される。
7. Javaアプリケーションには **main()** という名前の **メソッド** が必要である。
8. 文の末尾には **;** (セミコロン) が必要である。

メッセージを出力 & 円を描く (アプレット)

[1] 簡単なアプレットプログラムの例1

・ Helloa.java

```
import java.applet.Applet;           // 短名 Applet
import java.awt.Graphics;            // 短名 Graphics

public class Helloa extends Applet {  // サブクラス Helloa
    public void paint(Graphics g) {    // paint() × ソッド
        g.drawString("Hello, Java world!", 30, 40); // 文字列描画 × ソッド
    }
}
```

・ Helloa.html

```
<HTML>
<BODY>
<APPLET CODE="helloa.class" WIDTH=250 HEIGHT=100>
</APPLET>
</BODY>
</HTML>
```

Helloa.javaのコンパイル

```
javac Helloa.java
```

Helloa.class が出来ていることを確認してappletviewerで実行

```
appletviewer Helloa.html
```

または、Helloa.html を直接呼び出し実行

[2] 簡単なアプレットプログラムの例2

・ Circles.java

```
import java.awt.*;           // ウィンドウ関連ツール (Abstract Window Toolkit) を使う
import java.util.Random;    // 乱数を使う
import java.applet.Applet;

public class Circles extends Applet { // アプレットの名前は Circles にする
    public void paint(Graphics g) {   // 描画手順は以下の通り
        Random a = new Random();      // 新しい乱数列 a を用意する
        for (int i = 0; i < 1000; i++) { // 0 から 999 までループ
            g.setColor(new Color(a.nextInt())); // 整数の乱数を発生し、それで色を決める
            int x = (int) (300 * a.nextDouble()); // 整数変数 x に 300 × (0以上1未満の実数の乱数) を (切り捨てて整数
            int y = (int) (100 * a.nextDouble()); // 上と同様
            int r = (int) (40 * a.nextDouble()) + 10; // 上と同様
            g.fillOval(x - r, y - r, 2 * r, 2 * r); // 中心 (x,y), 半径 r の円板を描く
        }
    }
}
```

・ Circles.html

```
<html>
<body>
<applet code="Circles.class" width="300" height="100">
</applet>
</body>
</html>
```

Circles.javaのコンパイル

```
javac Circles.java
```

Circles.class が出来ていることを確認してappletviewerで実行

```
appletviewer Circles.html
```

または、Circles.html を直接呼び出し実行

●ポイント

1. Javaアプレットのコンパイルは `javac` コマンドで行う。
2. Javaアプレットの実行は `appletviewer` コマンドか、HTMLファイル (~.html) で行う。
3. WebブラウザのURLとして HTMLファイル を指定すると Javaアプレット が起動される。

4. JavaアプレットはJavaアプリケーションとは異なる記述方法をとる。

Next Step

[TANIGUCHI, Yuji](#)
Last modified: 2003/10/20 (Mon) 08:44