オブジェクト指向プログラミング　第 1 回
簡単な Java プログラムの作成と実行

担当：上浦

1　はじめに

- 参考書として
  - 内田智史「Java プログラミング徹底入門　基礎編」電波新聞社，改訂新版 2009 (旧版 2002).

  を挙げておきます．一部で同書の姉妹編である「応用編」の内容についても触れることがあります．やや古い本のため，使用されるメソッドなどに注意が必要ですが，Java の「考え方」を学ぶ上で有益な本だと思います．
- Java プログラムは，Java Virtual Machine（Java 仮想マシン，JVM）を介して実行するよう設計されており，OS に依存せずにプログラムを開発・実行することができます．講義では Windows 環境を用いますが，各自の PC に Java 環境を整えて受講しても構いません．

2　プログラム実行環境の確認（Windows）

- Java Standard Edition Development Kit (JDK)
  - JDK は，Java アプリケーションや，Java アプレット（Web ブラウザに読み込まれ実行される Java のアプリケーション），Java コンポーネント（他のプログラムで呼び出されて使用されるプログラム部品）を，Java 言語を用いて開発するための開発環境です．JDK をコンピュータにインストールすることにより，Java 言語で書かれた Java プログラムファイル (xxx.java) をコンパイルし，実行することができます．JDK は Oracle 社の Web ページ (`http://www.oracle.com/technetwork/java/javase/downloads/index.html`) からダウンロードすることができます．
  - JDK と JRE(Java SE Runtime Environment) を間違えないよう注意してください．JDK は Java プログラムのコンパイルおよび実行が可能な環境ですが，JRE は実行のみが可能な環境であり，JRE のみではコンパイルができません．
  - JDK をインストールしたら，ディレクトリ
    C:¥Program Files¥Java ¥jdk1.x.x_xx¥bin　　…（※）
    (1.x.x_xx の部分は Java のバージョンを表します．) に実行ファイル `javac.exe` があることを確認しましょう．
  - 【重要】Path の確認
    JDK インストール後，
    [スタート] > [コントロールパネル] > [システム] > [システムの詳細設定]
    > [環境変数] > [システム環境変数] > [Path]
    を開き，「C:¥WINDOWS¥SYSTEM32; C:¥WINDOWS; C:¥Program Files; **C:¥Program Files¥Java¥jdk1.x.x_xx¥bin**」などと，前述のディレクトリ（※）を [Path] の項に追記しましょう．各ディレクトリの記述は，ゼミコロン (;) で区切ります．**システムによっては，[システム環境変数] に [Path] を追記しても動作せず，[○○のユーザ環境変数] に [Path] を追記する必要がある場合もありますので注意してください．**

- コマンドプロンプト
  [スタート] > [すべてのプログラム] > [アクセサリ] > [コマンドプロンプト]
  コマンドプロンプトで「java -version」とタイプすると，Java のバージョン情報が確認できます.
- TeraPad（テキストエディタ）
  無料で入手できる，各種のプログラミング言語のコーディングに便利なテキストエディタです.
- Eclipse（統合開発環境）
  プログラムの規模が大きくなってきたときには，Eclipse などの統合開発環境を用いると，プログラミングをより効率的に行うことができます．特に，`http://mergedoc.osdn.jp/` からは，日本語化された Eclipse(Pleiades All in One) を簡単にインストールすることができます.

プログラムの開発と実行は,「TeraPad ＋コマンドプロンプト」および「Eclipse」の両方で行えるようにしておきましょう.

3　はじめての Java プログラム

Java プログラムの書き方を例示します.

クラス名.java

```
class クラス名{

    public static void main(String [] args){

        ここにプログラムコードを書いていきます.

    }
}
```

「クラスって何だ？」と思ってしまいますが，しばらくは「特定の種類の関数や変数を集めたもの」というぐらいに理解しておいてください.

コーディングの際の注意点

- Java プログラムのファイル名は「クラス名.java」とします.
- 英字の大文字・小文字は区別してください.
- ファイル名・クラス名は，単語の頭文字を大文字にします.
  （例）class Welcome, Welcome.java など
- ファイル名・クラス名で，複数の単語が並ぶ場合はスペースやアンダーバー，ハイフンなどを入れずに頭文字を大文字にして連接します.
  （例）class HelloWorld, BusinessCard.java など

以下のようなプログラムを書き，コンパイルして実行してみましょう．プログラムを書くときには，タブ（Tab）を適切に使って読みやすいコードを書くようにしましょう.

また，1 行分のコメントアウトは文頭に「 // 」を置き，複数行のコメントアウトは文の前後に「 /* 」と「 */ 」を置くことで可能になります.

Welcome.java

```
class Welcome{

    public static void main(String [] args){

        System.out.println("Hello, World!");

    }

}
```

このようなプログラムが書けたら,「H:¥oopJava¥Lec01」のようにフォルダ（ディレクトリ）を作り，ファイルを保存します.

次に，コマンドプロンプトを立ち上げ，Welcome.java が保存されているカレントディレクトリ（今は「H:¥oopJava¥Lec01」）へ cd コマンドで移動します.

移動できたら「javac Welcome.java」とタイプし enter すると，Java プログラム (Welcome.java) が **Java コンパイラ**によってコンパイルされます．これによって，Java プログラムは**バイトコード**と呼ばれる形式の中間的な言語に変換されます（「Welcome.class」というのが生成されたバイトコードのファイルです）.

```
H:¥oopJava¥Lec01 > javac Welcome.java
```

エラーが出なければ「java Welcome」タイプし enter すると Java バーチャルマシン (Java Virtual Machine;JVM) と呼ばれるインタプリタ (interpreter) によって，バイトコードが実行されます.

```
H:¥oopJava¥Lec01 > java Welcome
Hello, World !
```

以下の使い分けにも注意してください.

Check

```
System.out.print( ); // 改行なしで標準出力
System.out.println( ); // 改行ありで標準出力
```

練習 1.1

標準出力に

```
情報システムデザイン学系
(空白行)
氏名
学籍番号
```

を表示する「名刺プログラム」BusinessCard.java を作成し，実行せよ.

4　プリミティブ型変数と簡単な計算

4.1　整数の計算

2 数の和を表示するプログラムを書き，実行してみましょう.

AddSample.java

```
class AddSample{
    public static void main(String [] args){
        int a,b,c;

        a = 100;
        b = 200;
        c = a + b;

        System.out.println("c = " + c);

    }
}
```

練習 1.2

a=5，b=2 として，差 (a-b)，積 (a*b)，商 (a/b)，余り (a%b) を計算して表示するプログラム Calculation01.java を書き，実行しなさい.

4.2　実数の計算

int a=5, b=3; double ra=5, rb=3; とし，それぞれの型で $5 \div 3$ を計算してみましょう.

DivSample.java

```
class DivSample{
    public static void main(String [] args){

        int a = 5, b = 3;

        double ra = 5.0, rb = 3.0;

        System.out.println(ra +" / "+ rb +" = "+ ra/rb);
        System.out.println(ra +" / "+  b +" = "+ ra/b);
        System.out.println( a +" / "+ rb +" = "+ a/rb);
        System.out.println( a +" / "+  b +" = "+ a/b);

    }
}
```

Tabel 1.1 に示すように，**int** や **double** 以外にも様々な変数の型があります．**int** や **double** などの「ふつうの」変数を定義する型を**プリミティブ型 (primitive type)**（または**基本型**）といいます[1].

[1] プリミティブ型でない型として**参照型**というのが後の回で出てきます.

Table 1.1　Java で用いるプリミティブ型

区分	型	値の範囲	バイト数	有効桁数
整数型	byte	-128 〜 127	1	
	short	-32768 〜 32767	2	
	int	-2147483648 〜 2147483647	4	
	long	-9223372036854775808 〜 9223372036854775807	8	
	char	0 〜 65535 の Unicode	1	
浮動小数点型	float	$-\infty$, -3.4×10^{38} 〜 -1.4×10^{-45}, 0.0, 1.4×10^{-45} 〜 3.4×10^{-38}, ∞	4	6 桁程度
	double	$-\infty$, -1.798×10^{308} 〜 -4.9×10^{-324}, 0.0, 4.9×10^{-324} 〜 1.798×10^{308}, ∞	8	16 桁程度
論理型	boolean	false, true	※	

※特定の仕様が決められていません.

4.3　String 型と文字列

Java では，文字列を変数の値のように扱うことができます.

StringSample.java

```
class StringSample{
    public static void main(String [] args){

        String st = "Hello, World!";
        st = "Hello, Japan!";

        System.out.println(st);
    }
}
```

上のプログラムの st は，プリミティブ型変数ではなく，String クラスの**参照型変数 (reference variable)** なのですが，ここでは詳しく述べません．とりあえず，プリミティブ型でない変数なのだということだけ覚えておいてください[2].

4.4　Math クラスにある数学関数の使用

球の半径が与えられたとき，表面積と体積を計算しましょう.

BallVolume.java

```
class BallVolume{
    public static void main(String [] args){

        double r, S, V;

        r = 10; //球の半径

        S = 4 * Math.PI * Math.pow(r,2.0);  //球の表面積

        V = (4.0/3.0) * Math.PI * Math.pow(r,3.0);  //球の体積
```

[2]詳しくは http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/java/lang/String.html を参照してください.

```
        System.out.println("r = "+ r);
        System.out.println("S = "+ S);
        System.out.println("V = "+ V);
        System.out.println("π = "+ Math.PI);     //円周率

    }
}
```

Math クラスには様々な数学関数が用意されています．以下にその一例をあげます[3]．

Check

```
Math.sin(x); //sin x の値を double 型で返す.
Math.cos(x); //cos x の値を double 型で返す.
Math.tan(x); //tan x の値を double 型で返す.
Math.exp(x); //e^x の値を double 型で返す．e は自然対数の底.
Math.pow(x,y); //x^y の値を double 型で返す.
Math.sqrt(x); //√x を double 値で返す.
Math.abs(q); //q の絶対値を引数と同じ型で返す.
Math.random(); //0.0 以上 1.0 未満の一様乱数を double 値で返す.
```

※ 引数 x，y は double 型．引数 q は double 型，float 型，long 型，int 型のどれか

4.5　定数（final 変数），浮動小数点表現

次のプログラムを書いて，実行してみましょう．また，最後の 2 行をコメントアウトせずに実行した場合に何が起こるかを確認しましょう．

FinalSample.java

```
class FinalSample{
    public static void main(String [] args){

        final double a = 1234.5;

        final double b = 1.2345e3;

        double c = a/b;

        System.out.println(a +" / " + b + " = " + c);


/*
            double a = 3.0;
            System.out.println("a  = " + a);
*/
    }
}
```

[3]詳しくは http://java.sun.com/j2se/1.4/ja/docs/ja/api/java/lang/Math.html を参照してください．

Check

変数の定義の前に **final** をつけると，その変数の値は変更できなくなります．このような変数を final 変数と呼び，プログラム中で定数の役割を果たします．

Check

Java では，1.2345×10^3 のように，10 のべき乗を使った浮動小数点データを `1.2345e3`（あるいは `1.2345E3`）と表現します．

4.6　キャスト

変数の型を変換することを**キャスト (cast)** といいます．以下の CastSample.java はキャストを例示したものです．これを実行して結果を確認してみましょう．

CastSample.java

```
class CastSample{
    public static void main(String [] args){

        int k = (int)3.6; //double 型 → int 型
        double x = (double)12345; //int 型 → double 型

        int a = (int)'A'; //char 型 → int 型
        char c = (char)65; //int 型 → char 型

        int m = k * (int)3.1415926535 + 100; //double 型 → int 型
        int n = k * (int)(1.5 * k ) * (k - 13); //double 型→ int 型

        System.out.println("k = " + k);
        System.out.println("x = " + x);
        System.out.println("a = " + a);
        System.out.println("c = " + c);
        System.out.println("m = " + m);
        System.out.println("n = " + n);
    }
}
```

5　ループ

次のプログラム **LoopSample.java** は for 文，while 文，do-while 文を用いたループの例です．2 つある for 文では，変数 i と j の定義位置の違いに注意してください．2 つ目の for 文は () 内で変数を定義していますが，これによって変数 j の有効範囲が for 文の内部に限定されます．また，while 文と do-while 文では，変数の初期値を k=5 としたときに違いが出ます．条件式 k<5 を判定する位置が異なるからです．

LoopSample.java

```
class LoopSample{
    public static void main(String [] args){

        System.out.println("for 文 その 1");
        int i;
        for(i=0; i<5; i++){
            System.out.println("i = "+i);
        }
        System.out.print("for 文終了後に i の値はどうなっているか ?: ");
        System.out.println("i = "+i);

        System.out.println("\n"+"for 文 その 2");
        for(int j=0; j<5; j++){
            System.out.println("j = "+j);
        }
        //System.out.println("j = "+j);
        //上の for 文の外で j は定義されていない

        //while 文と do-while 文は k=5 としたとき違いが出ます.
        int k = 0;
        System.out.println("\n"+ "while 文");
        while(k<5){
            System.out.println("k = " + k);
            k++;
        }

        k = 0;
        System.out.println("\n"+ "do-while 文");
        do{
            System.out.println("k = " + k);
            k++;
        }while(k<5);
    }
}
```

課題

for 文を用いて 1 から 10 までの和を計算して標準出力する Java プログラムを作成し，提出しなさい（$1+2+\cdots+10=55$）．ファイル名は **Lec01Kadai_00rd000.java** とし，プログラムコードの冒頭に学籍番号と氏名をコメントとして記入すること．また，ファイル名の **00rd000** は各自の学生番号とし，**rd** は小文字とする．