Java独習　第3版

1.8　はじめてのJavaアプリケーション
1.9　変数と代入
1.10　文字列とキャラクタ

2006年4月12日(水)　　南　慶典

1.8 はじめてのJavaアプリケーション

Javaプログラム作成の注意点

- ソースコードのファイル名はそのプログラムで宣言するクラス名と同一にする。
(コンパイルするとファイル名ではなくクラス名が付けられるため)
- 英字の大文字と小文字が区別される。
- ファイル保存時のフォーマットに注意する。
（拡張子を.txt等としないようにする。）

ソースコードファイルを作成してみる

適当なテキストエディタ（メモ帳など）を使用して、ソースコードを入力して、
ソースコードファイルを作成してみる。

```
Class Example1 {
  public static void main(String args[]){
    System.out.println("This is the output from Example1”);
  }
}
```

ファイル名はクラス名と同一にするため、"Example1.java"と付ける。

プログラムを実行する

ソースコードをコンパイル（バイトコードに変換）

javac　ファイル名 .java

⇒ 「クラス名.class」というファイルが作成される。

プログラムの実行

java　クラス名

1.9　変数と代入

変数の宣言と代入 ①

変数

名前のついたメモリ位置のことで、変数にはなんらかの値を格納することができる。

使用する変数はすべて使用前に宣言しなければならない。
⇒変数の型をコンパイラに知らせる。**Java**では**8**種類の基本データ型がある。

変数の宣言

型　変数名　;

例）　**int　counter ;**
　　float　x, y, z ;

表.　Javaの8つの基本データ型

型	説明
char	16ビットUnicodeキャラクタデータ
boolean	真偽値
byte	8ビット符号付き整数
short	16ビット符号付き整数
int	32ビット符号付き整数
long	64ビット符号付き整数
float	32ビット符号付き浮動小数点
double	64ビット符号付き浮動小数点

変数の宣言と代入 ②

ローカル変数

ローカル変数は1つのメソッド内（**main()**メソッドなど）で宣言する変数で、宣言したメソッド内でしか認識されず、またそのメソッド内からしかアクセスできない。

値の代入

変数名　＝　値　；

“＝”は代入を表し、定数または式を指定することができる。

変数の宣言時に値を代入することもできる。

例）　int num = 100 ;

変数の宣言と代入 ③

- キャラクタ（単一文字）定数を指定するには、キャラクタを単一引用符（‘）で囲む。
  例） char ch = ‘A’ ;
- Long型の値を表すには、接尾辞として「 L 」または「 l 」を明示。
  例） long m = 21478364L ;
- Float型の値を表すには、接尾語として「 F 」または「 f 」を明示
- 小数点または指数を含んだ数値リテラルは、double型
  例） .5 0.8 9e-2 -8.7e-5

変数名の規則

- 同じ名前の変数を2つ以上宣言することはできない
- 変数名に空白と記号（＋やーなど）は使うことができない
- 変数名は先頭に数字を使うことができない
- 変数名には予約語（キーワード）と同じものは使えない
  例）　int　public ; ←はダメ
- 変数名には「true」「false」はつかえない
- 変数名には長さの制限はない
- 変数名は大文字、小文字も区別される
- 変数名にはUnicode文字を使う
  例）　int　字, 整数 ; ←はOK

キーワード一覧

abstract boolean break byte
case char class default do
else float for if import
int interface long new package
private public return this
などなど…

print( ) と println( )

PrintIn()メソッドは、
　文字列引数の最後に改行文字を自動的に追加する。

Print()メソッドは、
　改行文字を追加しない。

```
class DisplayFloat {
  public static void main(string args[]) {
   float   price = 45.35f;
   System.out.print(“The price is”);
   System.out.println(price);
  }
}
```

1.10文字列とキャラクタ

文字列とその連結

- 文字列の型は String

  String s = “moziretsu” ;

- C言語のようなchar型の配列を指定する事は出来ない。

  char ch = ‘a’ ;

- 文字列の内部に二十引用符を含めるには、二重引用符の前に円記号(¥)を挿入する。

- 連結演算子「＋」

  二つの文字列を連結。引用符で囲んだ文字列を使用可能な場所ならどこでも使用できる。

プログラム例

```
class LincolnQuote {
  public static void main(String args[]) {
    String s = "Lincoln said: "  +
      "￥"Four score and seven yeas ago￥"";
    System.out.println(s);
  }
}
```

出力結果

```
  Lincoln said: "Four score and seven years ago"
```

```
class Concatenation {
  public static void main(String args[]) {
    System.out.println("My book " +
      "will teach you " +
        "about Java programming");
  }
}
```

出力結果

```
  My book will teach you about Java programming
```

プログラム例

```
class StringVariables {
  public static void main(String args[]) {
    String s1 = "My book teaches ";
    String s2 = "you how to ";
    String s3 = "use Java";
    System.out.println(s1 + s2 + s3);
  }
}
```

出力結果

```
My book teaches you how to use Java
```

練習問題

- 以下のソースコードの間違いを指摘しなさい。
  「**Name.java**」

```
public class MyName {
  public static void main(String args[]) {
    System.out.println(私の名前は北です)
  }
```

- ３つの整数変数を宣言、初期化し、print()とprintln()ステートメントを使って、以下のように表示させなさい。

```
12345
67890
```

- 文字列変数を1つ宣言し、1つの連結演算子を使って2つの文字列を連結させて、自分の名前を表示するプログラムを作れ。