Java プログラミング入門

Agenda

1. **Java入門**
   Java の概要と Java プログラムを作成するための基本事項
2. **Java言語入門**
   Java 言語の基本的な文法
3. **Class入門**
   Java のクラスと、クラスから生成されるオブジェクトについて
4. **継承とClassの利用**
   Java言語でのクラスの継承、パッケージ、インタフェースなどについて
5. **例外処理**
   エラーや例外に対する処理方法について
6. **Javaの基本クラス**
   Java 2 SDK で用意されている基本的なクラスについて

1. Java入門

Java の概要と Java プログラムを作成するめの基本事項を説明します。

- ☐ Java とは
- ☐ コンパイルと実行
- ☐ Java プログラムの構成
- ☐ 簡単な標準出力

Javaとは

Java とは、以下の側面を持つテクノロジーです。

- ☐ プログラミング言語
- ☐ プログラム実行環境
- ☐ 開発環境

- **プログラミング言語**

  プログラミング言語としての Java は、Java 言語と呼ばれます。

  Java 言語の主な特徴には、次のものがあります。

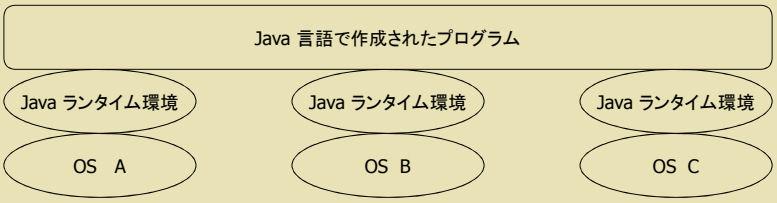
  - ☐ プラットフォーム非依存
  - ☐ インタープリタ
  - ☐ シンプル
  - ☐ オブジェクト指向
  - ☐ 分散型
  - ☐ セキュリティ
  - ☐ マルチ・スレッド

アプリケーションとアプレット

Java 言語で作成された、プログラムは大きく分けると次のように呼ばれます。

- □ アプリケーション　　スタンドアローンで動作するプログラム
- □ アプレット　　Java 対応のブラウザ上で動作するプログラム

プログラム実行環境

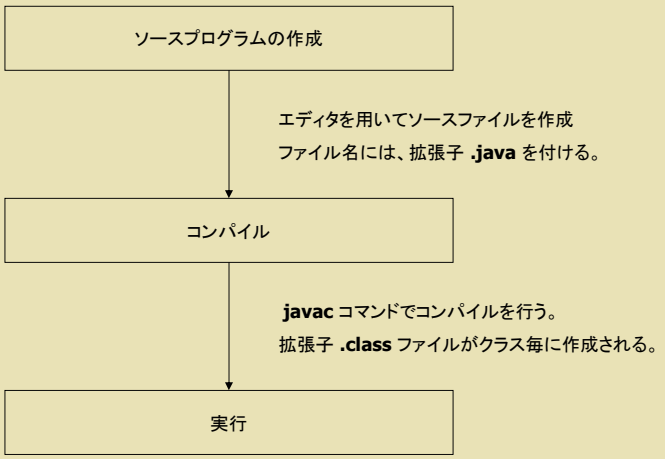


開発環境

Java プログラムの開発には、Java 2 Software Development Kit (Java 2 SDK) と呼ばれる開発環境が必要です。Java 2 SDK にはコンパイラ、インタープリタ、ライブラリなどが含まれています。
また、Jbuilder などの統合開発環境（IDE:Integrated Development Environment）を使用することもできます。

コンパイルと実行手順

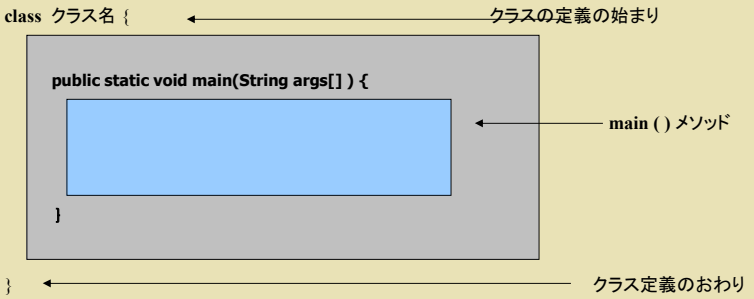
Java プログラムのソースプログラムの作成から、実行手順は以下のようになります。



Java プログラムの構成

Java プログラムは、クラスの集合で構成されています。

- Java プログラムは、クラスの定義です。
- Java アプリケーションは、最初に main ( ) メソットが実行されます。



注意 – Java では、クラス内の関数をメソットと呼びます。

参考 – main ( ) メソッドの引数 String args [] は、プログラム実行時のコマンドラインからの引数を受け取るために使用します。

文字列をするプログラム:


```
class  Hello{
    public static void main(String args[]) {
        System.out.println("Hello");   //文字を表示する
    }
}
```

基本データ型の表示プログラム

```
/*    各基本データを使用して値を代入、表示するプログラムです。*/
class Hello{
    public static void main( String args[ ]) {

        System.out.println("Hello World!");
    }
}
```

環境変数 CLASSPATH

Javaを実行するとき、 JVM は実行に必要なクラスを環境変数 CLASSPATH に指定されたパス内を検索します。

CLASSPATH は、通常のコマンド検索パスと同様に複数指定することができます。

また、javac コマンドや java コマンドの使用時に、オプションを指定して、クラスファイルが保存されているパスを指定することもできます。

☐ javac コマンド

Javac – classpath ユーザクラスファイルを探す場所、ソースファイル

☐ java コマンド

java = cp ユーザクラスファイルを探す場所　クラス名

2. Java 言語入門

Java 言語の基本的な文法を紹介します。

- ☐ コメント
- ☐ 識別子
- ☐ キーワード
- ☐ Java 言語の基本データ型
- ☐ 文字列
- ☐ リテラル
- ☐ 配列
- ☐ Java アプリケーションへの引数
- ☐ 演算子と演算子の優先順位
- ☐ 分技文
- ☐ ループ文
- ☐ ループ制御

コメント

Java 言語におけるコメントの指定は、以下のとおりです。

//	この行の最後までをコメントとして扱う。	
/*	複数行に渡ってコメント指定ができる。	*/
/**	ドキュメント・コメント。複数行指定可能。	*/

参考 - ドキュメント・コメントは、Java 2 SDK に含まれるドキュメント作成コマンド javadoc で利用されます

識別子

識別子は、変数やラベル、クラス名、メソット名、インターフェース名に使用する名前です。
長さの制限はありません。識別子におけるルールは以下のとおりです。

- 1文字目は、英字、‘ _ ’ (アンダースコア) 、‘ $ ’ (ドル記号) のいずれか。
- 2文字目以降は、上記に加えて数字を用いることもできる。

userName　　user_name　USERNAME　　_WIDTH

キーワード

Java 言語で予約されている、キーワードは以下の通りです。

予約語

abstract	boolean	break	byte	case	catch	char
class	const	continue	default	do	double	else
extends	final	finally	float	for	goto	if
implements	import	instanceof	int	interface	long	native
new	package	private	protected	public	return	short
static	strictfp	super	switch	synchronized		this
throw	throws	transient	try	void	volatile	while

参考 – goto および const は、キーワードですが、現在は使用されていません。
注 – true、false および null は、値そのものであるためキーワードではなく、リテラルとして予約されている。

予め演習ファイルをc:¥srcにコピー(展開)しておくこと

演習１：計算をするプログラム：

開始 → 表示 → 終了

```
class  Plus{        /* 計算結果表示する */
   public static void main(String args[]) {
      System.out.println(1+2+3);   //足し算表示
      System.out.println(1+2/3);   //割り算と足し算表示
   }
}
```

```
C:¥Users¥user> cd  c:¥src
c:¥src>notepad  Plus.java
c:¥src>javac   Plus.java
c:¥src>java     Plus
6
1
```

作業場所を c:¥src に移動させる命令

プログラムをメモ帳で開く命令

プログラムをコンパイルする命令

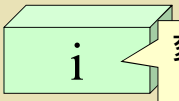
プログラムを実行する命令

プログラムの計算結果

変数の利用方法

- 変数
  - 変数の宣言

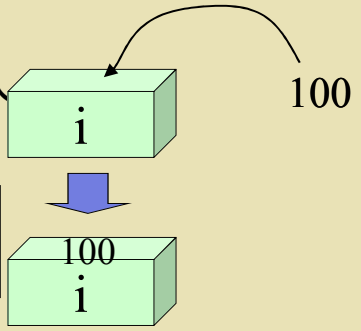
```
int  i;
```



変数とは、整数を入れる箱のようなもの

  - 変数への代入

```
i = 100;
```



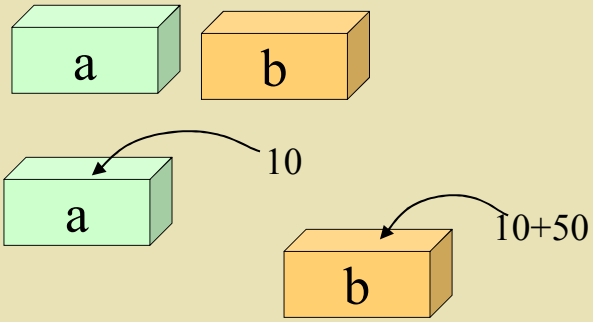
これは数学の＝では無く
i←100　の代入の意味

変数の利用例

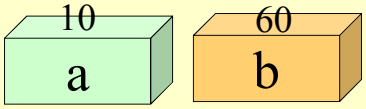
```
int a;
int b;

a = 10;
b = a+50;
```

プログラム部分



実行後は下記の値が保存されている



予め演習ファイルをc:¥srcにコピー(展開)しておくこと

演習２：変数を使ったプログラム：

開始

表示

終了

```
class Variable {        /* カッコの中がメイン文 */
  public static void main(String args[]) {
        int a;
        int b;
        a = 10 ;
        b = a + 50 ;
        System.out.println(a);
        System.out.println(b);
  }
}
```

変数 aの値(内容)を表示させる命令

変数 bの値(内容)を表示させる命令

作業場所を c:¥src に移動させる命令

```
C:¥Users¥user> cd c:¥src
c:¥src>notepad Variable.java
c:¥src>javac Variable.java
c:¥src>java Variable
```

プログラムをメモ帳で開く命令

プログラムをコンパイルする命令

プログラムを実行する命令

Java 言語の基本データ型

Java 言語には、8つの基本データ型が定義されています。Java で使用できる基本データ型は、以下のとおりです。

	データ型	サイズ	表現できる値
整数	byte	8ビット	−128 ～ 127
	short	16ビット	−32,768 ～ 32,767
	int	32ビット	−2147483648 ～ 2147483647
	long	64ビット	−9223372036854775808 ～ 9223372036854775807
実数	float	32ビット	$-3.40282347 \times 10^{38}$ ～ $3.40282347 \times 10^{38}$
	double	64ビット	$-1.79769313486231570 \times 10^{38}$ ～ $1.79769313486231570 \times 10^{38}$
文字	char	16ビット	Unicode で表現できる1文字
真偽値	boolean	−	true (真) または false (偽)

参考 – 基本データ型以外のデータ型を参照型と呼びます。
参考 – Java 言語では、文字コードとして UNICODE を採用しています。
注意 – Java 言語では、unsigned (符号なし)タイプはありません。

オブジェクトを参照するクラス型変数は、基本データ型の変数に対して、参照型変数と呼ばれる。配列の変数も参照型変数であり、これらの変数に共通することは、型宣言と初期化のほかに、生成という手順が必要となる。

文字列

文字列を表現するための基本データ型は定義されていません。代わりに String クラスのクラス型を使用しています。文字列はオブジェクトとして扱います。

☐ 文字列は、String クラス型で扱う。
☐ ダブル・クォートで囲まれた文字列は String クラスのオブジェクトを生成する。
☐ 文字列は + 演算子を使用することみより、文字列を結合することができる。

```
String    myobject   = "Hello World! ";
```

参照 – クラスおよびオブジェクトについての詳細は、3. Class入門で説明します。

リテラル

整数リテラル

整数リテラルのデフォルトは、 int 型です。long 型のリテラルとして利用したい場合は、数値の後に明示的に L または l を指定することが必要になります。
また、10進数以外の基数表現をすることもできます。

2	**int** 型としての**10進数値**
123L	**long** 型としての**10進数値**
077	**int** 型としての**8進数値**
077L	**long** 型としての**8進数値**
0xFF	**int** 型としての**16進数値**
0xFFL	**long** 型としての**16進数値**

参考 – 0 (ゼロ)で始まる数値は8進数値、0x で始まる数値は16進数値とみなされます。

浮動小数リテラル

浮動少数リテラルのデフォルトは、double 型です。Float 型のリテラルとして利用したい場合は、数値の後に明示的に F または f を指定することが必要となります。
double 型のリテラルとして明示的に表す場合は、D または d を指定します。
また、指数表現 (E または e) をすることもできます。

3.14	**double** 型としての浮動小数値 (**3.14D** としても同じです)
2.718F	**float** 型としの浮動小数値
6.02E23	**double** 型としての浮動小数値、指数
2e12	**double** 型としての浮動小数値、指数

文字リテラル

1文字を表現するときは、char 型です。シングル・クォート (') で囲みます。

'a'　　a

'¥n'　　改行

文字列リテラル

文字列は、文字列全体をダブル・クォート (") で囲みます。

" Hello World "

" This is a string data! "

Boolean リテラル

真偽を表現するときは、true または false のいずれかのリテラル値を使用します。

予め演習ファイルをc:¥srcにコピー(展開)しておくこと

演習2：平均を計算するプログラム：

開始

表示

終了

```
class Average {  /* 平均値を表示する  ここに学番氏名 */
  public static void main(String args[]) {
    int  japanese    = 70;  //国語の点数
    int  english     = 80;  //英語の点数
    int  mathematics = 90; //数学の点数
    int  average = (japanese+ english)/2; //平均点計算
    System.out.println(average);          //平均点表示
  }
}
```

変数はxでもaでも良いが、それでは何を意味する変数か
が分からないので、誰でも分かる名前にする

```
C:¥Users¥user> cd c:¥src
c:¥src>notepad Average.java
c:¥src>javac  Average.java
c:¥src>java Average
```

作業場所を c:¥src に移動させる命令

プログラムをメモ帳で開く命令

プログラムをコンパイルする命令

プログラムを実行する命令

課題2：5科目平均を計算するプログラム：

前のページのプログラムを基にして、
5科目の平均を求めるプログラムに改良しなさい。

ただし、変数は下記とすること
社会科の変数　social　として　70点 で計算
理科の変数　science　として　80点で計算

```
c:¥src>notepad Average.java
c:¥src>javac  Average.java
c:¥src>java Average
```

プログラムをメモ帳で開く命令（メモ帳で編集後は、上書きで更新）

プログラムをコンパイルする命令

プログラムを実行する命令