

■ 今日の講義で学ぶ内容 ■

- Java 言語と C 言語の違い
- プログラム開発環境 C Pad の使用方法
- プログラムの実行の仕方

1. Java とは？

1995 年に Sun Microsystems 社（現在、Oracle 社）*1から発表されたオブジェクト指向言語（Object-oriented language）です。オブジェクト指向とは、ソフトウェアが扱う対象の振る舞いを物理的または抽象的な実体とそれらの相互作用として捉える考え方です。オブジェクト指向は、近年複雑・大規模化するソフトウェアシステムを見通し良く記述できる特徴をもち、様々なプログラミング言語に取り入れられています。

Java 言語 → オブジェクト（物体）に着目したプログラミングスタイル
たとえば、

PCは、CPUとメモリ、マウス、スクリーン・・・をもつ

と“何を”もつかを考えるスタイル



C 言語 → 手続きに着目したプログラミングスタイル
たとえば、

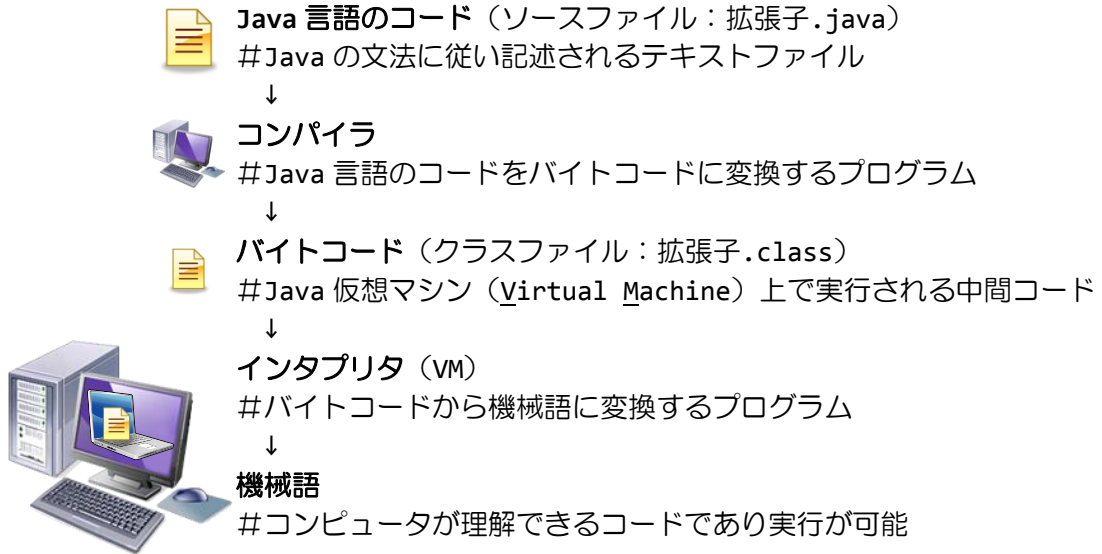
最初にooして、次にooして・・・、最後にooをする

と“どのように”するかを考えるスタイル



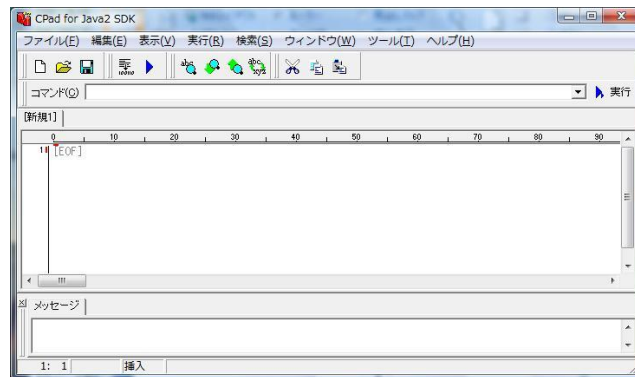
*1 2010年1月27日（米国時間）Oracle 社が Sun Microsystems 社を買収

2. 実行までの流れ



3. Java 言語開発環境

コンパイラやインタプリタは Oracle 社から JDK (Java 2 Standard Edition Development Kit) として無償で提供されています。本講義ではコンパイル・実行を GUI (Graphical User Interface) から簡単に起動できるフリーソフトである C Pad for Java2 SDK を使用します。



4. 次のコードを打ち込んでみよう

ソースファイル名：Sample1.java

The diagram shows a snippet of Java code with callouts explaining specific syntax rules:

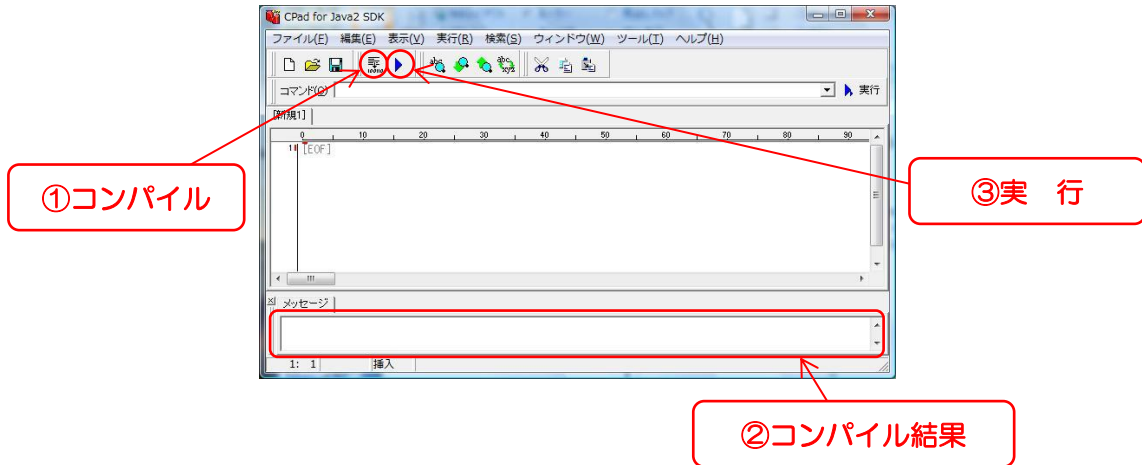
```
class Sample1
{
    public static void main(String[] str)
    {
        System.out.println("ようこそ Java へ!!");
    }
}
```

- Callout 1: [Tab]タブ (points to the indentation of the main method)
- Callout 2: ファイル名 (拡張子より前) と一致させます (points to 'Sample1' in the class declaration)
- Callout 3: 英数字は半角で入力します (points to the characters in the main method signature)
- Callout 4: 大文字と小文字を区別します (points to the 'S' in 'System.out.println')

※入力上の注意点

- 英数字は全角ではなく、半角で入力します
- 英字の大文字と小文字は異なる文字として処理されます

5. 次にコードをコンパイルし、実行してみよう



■ 今日の講義のまとめ ■

- Java 言語はオブジェクト指向言語であり、物（オブジェクト）に着目したプログラミングスタイルです。物が何をもっているのかを考えながらプログラムを行います。
- c 言語は手続きに着目したプログラミングスタイルです。処理の順番を考えながらプログラムを行います。
- プログラム開発環境 C Pad for Java2 SDK は GUI 環境で手軽にコンパイルや実行を行えるフリーソフトです。
- ソースファイルは、Java の文法にしたがい記述されるテキストファイルです。ソースファイルに記述される Java 言語のコードをコンパイラと呼ばれる変換プログラムにより、バイトコードに変換します。バイトコードはインタプリタと呼ばれる翻訳プログラムにより、機械語に翻訳され実行されます。

