

1回目 はじめての実行

■ 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 Visual Studio Code 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

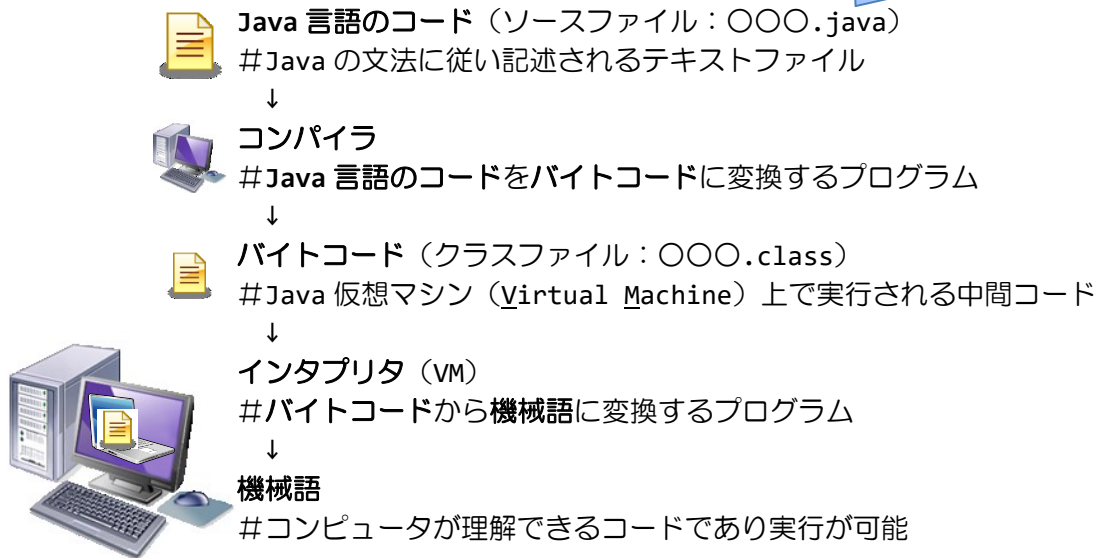


\*1 2010年1月27日（米国時間）Oracle 社が Sun Microsystems 社を買収

## 拡張子

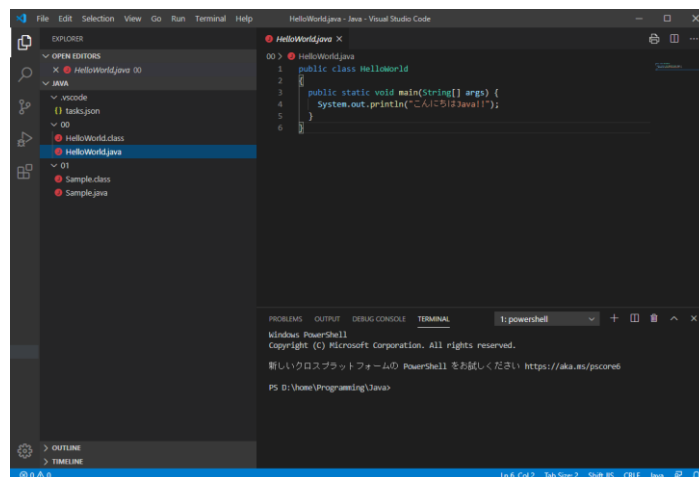
ファイル名に続く.(ドット)の後の記号です。ファイルの種類を表します。

## 2. 実行までの流れ



## 3. Java プログラミング開発環境

コンパイラやインタプリタは Oracle 社から Oracle Open JDK (Java Development Kit) として無償で提供されています。本講義では Java コードの入力、コンパイルと実行ができるフリーソフトである Microsoft 社の Visual Studio Code を使用します。



Java の実行環境の作り方を下記ページで紹介しております。

○Java の講義ページ の「開発環境などの事前準備」

<https://www.fit.ac.jp/~m-ishihara/Lectures/JavaProgramming1/howToSetupOnWin10new/howToSetup.html>



#### 4. 次のコードを打ち込んでみよう

次の Java 言語のコードを入力します。

ソースファイル名：Sample.java

The diagram shows a code editor window with the following code:

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

Callouts and annotations:

- [Tab]タブ: Points to the indentation of the main method.
- ファイル名 (拡張子より前) と一致させます: Points to the class name 'Sample'.
- 半角英数字で入力します: Points to the characters in the code.
- 大文字と小文字は区別されます: Points to the lowercase 's' in 'str'.

※入力する上での注意点

- 基本的に半角英数字で入力します
- 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 (a と A は異なる文字です)

ソースファイルの作成と管理の仕方について下記ページで紹介しております。

○Java の講義ページ の「開発環境などの事前準備」

<https://www.fit.ac.jp/~m-ishihara/Lectures/JavaProgramming1/howToSetupOnWin10new/howToSetup.html>



#### 5. 次にコードをコンパイルして実行してみましょう

実行したい Java 言語のコードを選択表示して、メニュー [Terminal] → [Run Build Task] でコンパイルと実行をしてみましょう。コンパイルエラーがある場合は、エラーメッセージが表示されますので、もう一度 Java 言語のコードを再確認しましょう。

The screenshot shows the Visual Studio Code interface with the following elements:

- Code Editor:** Displays the Java code from the previous section.
- Terminal:** Shows the execution output:

```
> Executing task: javac d:\home\Programming\Java\01\Sample.java <
Terminal will be reused by tasks, press any key to close it.
> Executing task: java -cp ".;d:\home\Programming\Java\01" Sample <
ようこそJavaへ!!
Terminal will be reused by tasks, press any key to close it.
```
- Callouts:**
  - コンパイル&実行  
ショートカット Ctrl + Shift + B  
または  
メニュー [Terminal] → [Run Build Task]
  - コンパイル結果  
実行結果

## ■ 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。

- 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。

