OPEN STREAM

210-01.Java概要

Make IT your Success

目次

この講義について

目的：

- プログラム言語であるJavaの特徴・利点を学ぶ。
- Javaで開発されたアプリケーションがどのように動くのかについて学ぶ。
- Javaプログラム(プロジェクト)作成の手順を学ぶ。

ゴール：

Javaで開発されたアプリケーションがどのようにして動作するのかを理解し、Javaアプリケーション開発の基本的な手順を習得する。

Javaとは

サンマイクロシステムズ社が開発したプログラミング言語。
（2010年1月にOracle社が買収）
※ プログラミング言語とはコンピューターに対する命令を記述するもの。

Javaの特徴

① オブジェクト指向プログラミング言語

プログラムの再利用やメンテナンスが容易で、規模の大きな開発にも向いている。

② コンパイルが必要

プログラミング言語を中間言語(バイトコード)に変換する作業が必要。

→ 実行時のチェックが厳密

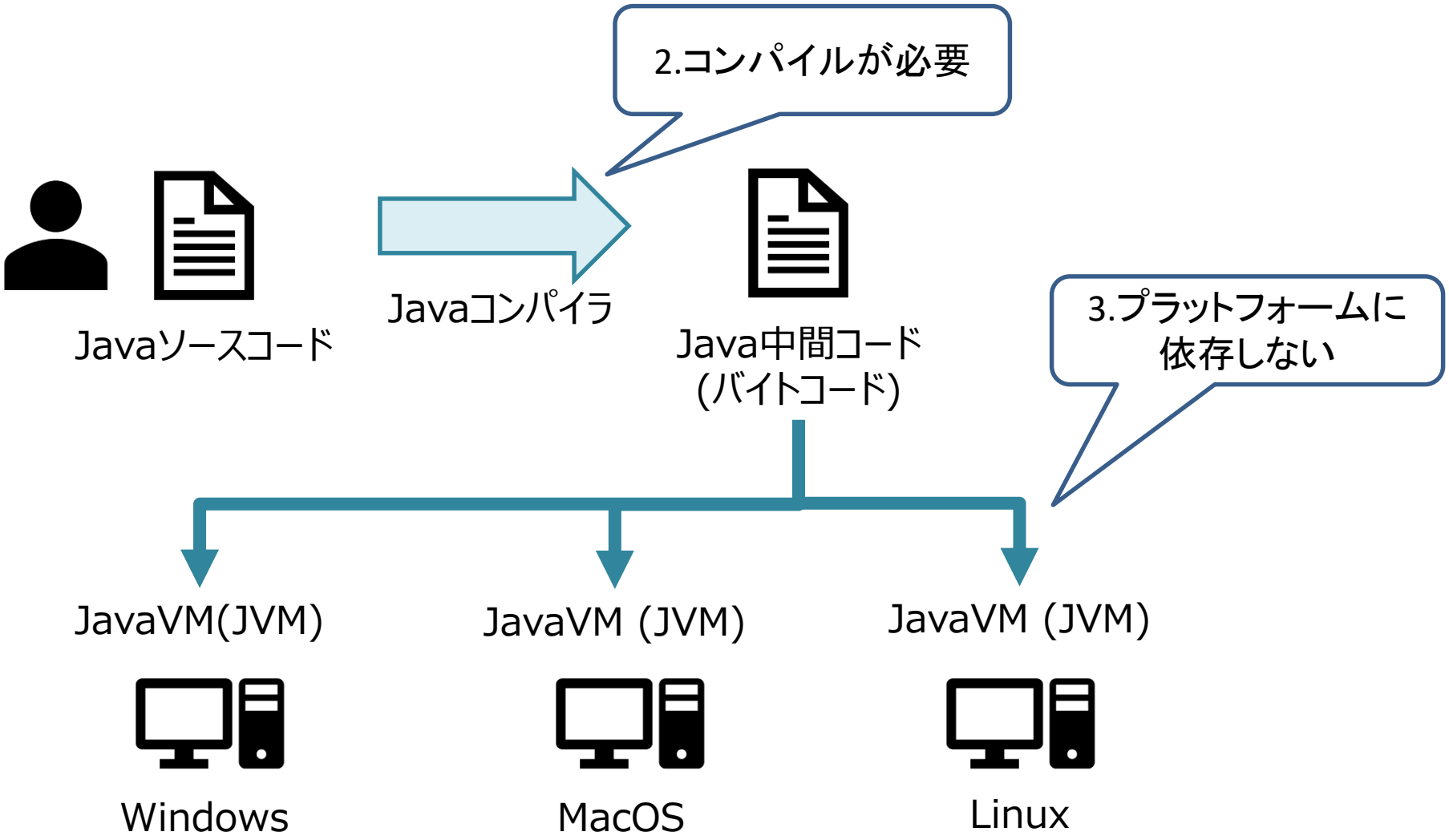
③ プラットフォーム(OS)に依存しない

あるプラットフォーム(OS)のJavaVM上で動作するプログラムは、別プラットフォーム(OS)のJavaVM上でも動作する。

- **その他**

  文法がシンプル、記憶領域を自動回収(ガベージコレクション)、マルチスレッド対応など・・・・・・。

Java概要



プログラムが動く仕組み(1/3)

プログラムとは

コンピュータへの命令。
コンピュータはプログラムに書かれている命令を実行(処理)する事で動く。

プログラムの作り方

コンピュータは電気で動くため、データも命令も2進数(bit)でしか扱えない。
処理装置(CPU)が実行できるプログラムは、0,1の2進数の集まりになる。
CPUが直接実行できるプログラムをネイティブコード(機械語)と言う。
ネイティブコードは人間が書くのは困難（ミスも多くなる） なので、
現在は、人間が読み書きしやすいプログラミング言語でプログラムを書き、
それをネイティブコードに変換して、実行するのが通常になっている。

プログラムが動く仕組み(2/3)

プログラミング言語

人間がコンピュータに命令を伝える為の言語。
その為に人間が理解できる単語や文法になっているが、ネイティブコードに変換する為に、
コンピュータが解析、変換できるように曖昧が記述が無く、構文や規則などが厳密に決められている。
様々な開発ニーズを満たすために、多種多様なプログラム言語がある。
より人間が分かり易く、CPUに依存しないプログラミング言語を高水準言語
ネイティブコードに近く、CPUやOS特有の処理を書けるプログラミング言語を低水準言語という。

プログラムの実行

人間がプログラム言語で書いたプログラム（ソースコード）を、ネイティブコードに変換して、CPUが実行するが、この変換方法には幾つか種類ある。
主な変換方法として、コンパイラ、インタプリタがある。

プログラムが動く仕組み(3/3)

コンパイラ言語

コンパイラによって変換を行う。通常は、プログラムのソースコードを一括で変換する。

Javaコンパイラは直接ネイティブコードに変換するのではなく、中間コード(バイトコード)に変換する。

一括で変換後に実行するので、実行が早い。コンパイラがソースコードを一括で変換するので、実行前に文法的なミスを発見できる。などの利点がある。

ex). C, C++, Lispなど

インタプリタ言語

ソースコードを1行ずつ変換しながら実行する。

繰り返し処理などは、繰り返す度に変換する場合も多く、実行速度が遅い。また、実行して初めて文法的なミスが発見されるので実行時のエラーが発生しやすいが、プログラムが作成途中でも書いたところまでは実行できる、素早く修正、実行、確認を行う事ができるという利点もある。

ex). PHP, Python, Rubyなど(Jrubyなどコンパイル処理系もある)

アセンブリ言語

2進数のネイティブコードをニーモックという人間が理解しやすい単語に置き換えたもの。

よりネイティブコードに近いプログラミングを行う事ができる（変換はアセンブリが行う）。

JREとJDK

JRE(Java Runtime Environment)

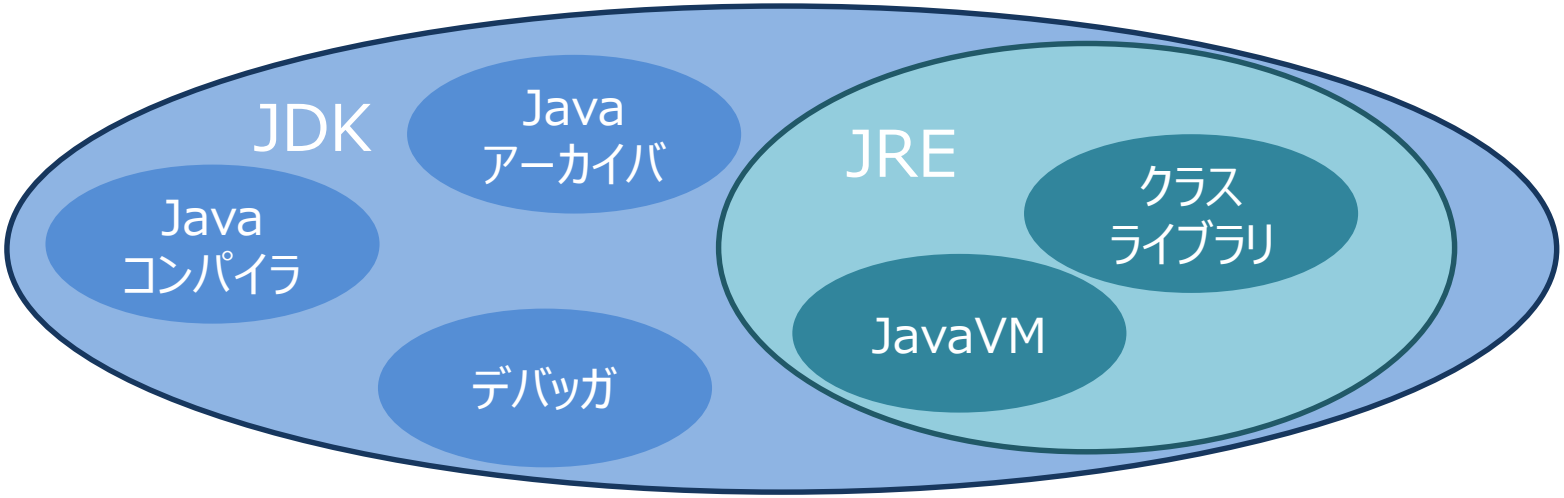
Javaアプリケーションを動かすための実行環境。

JavaVM(Java仮想マシン:JVM)とAPI(Javaクラスライブラリ)。

JDK(Java Development Kit)

Javaアプリケーションを開発するための開発キット。

主にJavaコンパイラ、デバッガ、アーカイバなど。JREも同梱している。



Javaプラットフォームの種類

Java SE(Standard Edition)

Javaの基本的なAPIセットで標準的な機能が備わっている。

Java EE(Enterprise Edition)

Java SEの機能拡張でサーバ向け機能のAPIセットが拡張されている。

※2017年Oracle社はJavaEEの開発と管理をEclipse Foundationに移管し
名称をJakarta EEに改称した。

Java FX

GUIアプリケーション開発用のライブラリ。Java7update2からJava SEに同梱されていたが、Java11から同梱されなくなった。

現在はオープンソース実装のOpenFXを利用する。

など……

JavaSEのバージョンとOracleJDKの有償化

Javaは登場から25年程経過している。改良・拡張が施されており最新はバージョン20(2023年3月リリース)
Oracleが配布するJDKは、OracleJDKとオープンソースのOpenJDKがある。
OracleによるOracleJDKの無償配布はバージョン10までになる。(2018年3月リリース)
バージョン11(2018年9月リリース)以降はオープンソースのOpenJDKが無償配布されている。
しかし、バージョン17からは再び無償利用が可能になった。(サポートは有償)

バージョン8までは、基本的には互換性は保たれている。
しかし、バージョン9以降は内部実装の変更や削除される機能があるので互換性に注意する。
今回の研修はLTSであるバージョン17で行う。

※バージョン表記に関して、
JavaSE9まではJDK1.9.xとなるが、
以降はJavaSE10がJDK18.3、
JavaSE11がJDK18.9と
リリース年月になるので注意

OracleJDKのバージョンと無償サポート

	JDK	リリース	無償サポート
8	1.8	2014年3月	商用:2019年1月 個人:2020年12月
9 (non-LTS)	1.9	2017年9月	2018年3月
10 (non-LTS)	18.3	2018年3月	2018年9月
11 (LTS)	18.9	2018年9月	無し
12 (non-LTS)	19.3	2019年3月	無し
13 (non-LTS)	19.9	2019年9月	無し
14 (non-LTS)	20.3	2020年3月	無し
15 (non-LTS)	20.9	2020年9月	無し
16 (non-LTS)	21.3	2021年3月	無し
17(LTS)	21.9	2021年9月	無し

LTS……Long Term Support

JDKのサポートと実装

Javaはプログラミング言語であり、その実行環境でもある。

つまり、構文や命令といった文法であるが、JDK、JREアプリケーションの仕様でもある。

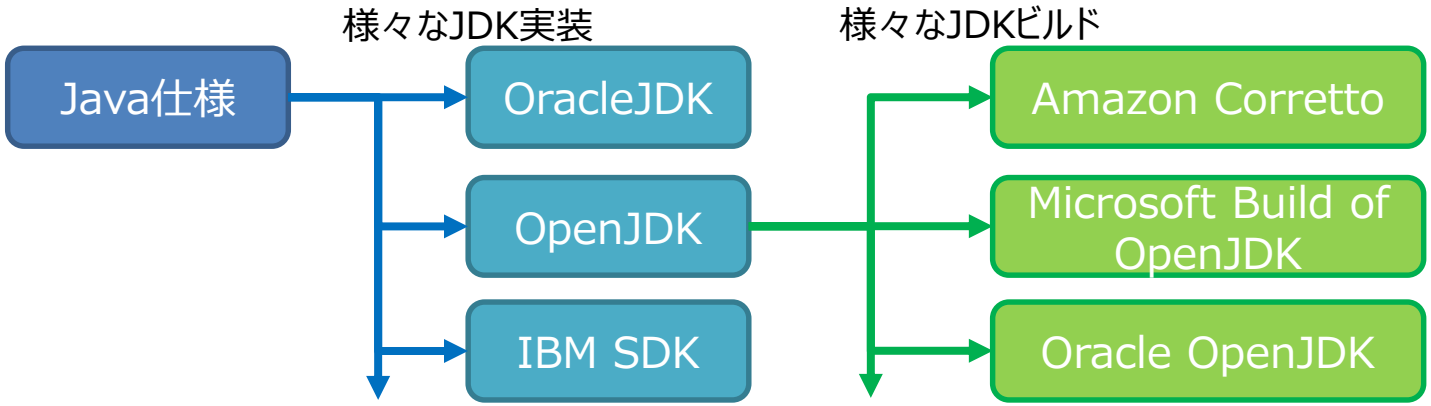
Javaの仕様はJCP(Java Community Process)によって管理され標準化されている。

OracleJDKはOracle社が提供するJava(JDK)実装の一つ。

(IBM SDK、OpenJDKなど様々なJDK実装がある)

OacleJDKはバージョン11以降は有償サポートになっている為、無償での利用はオープンソースのOpenJDKを利用する必要がある。

OpenJDKはオープンソース(リンク例外付きGNU GPL)で、いくつかのOpenJDKプロバイダがOpenJDKのビルドを提供している。OracleはOracleJDKの提供の他に、OpenJDKのビルドも提供している。



参考：
https://docs.google.com/document/d/1nFGazvrCvHMZJgFstlbzoHjpAVwv5DEdnaBr_5pKuHo/edit#heading=h.p3qt2oh5eczi

Javaの設定・環境変数

Javaコンパイラ・JavaVMを動作させるためには環境変数を設定しなければならない。

環境変数

OSが持つ変数。環境毎に異なる値を設定することが出来る。

環境変数はOS上で動作するプログラムから参照出来る。

最低限設定が必要な環境変数

① JAVA_HOME

JDK(JavaコンパイラやJavaVMなど含んだ開発キット)をインストールした場所を設定

② PATH

JavaコンパイラやJavaVMがある場所を追加設定（JAVA_HOME配下のbinフォルダ）

開発の基本的な流れ

手順

① Javaプログラムのソースファイルを作成する
② 作成したソースファイルをコンパイルする
③ コンパイルが正常終了するとクラスファイルが作成される
④ クラスファイルをJavaVM上で実行する
⑤ 実行結果を得る

※ eclipseでは②～⑤が一括で行える

開発の基本的な流れ

ソースファイル

人が読むことが出来る状態のもの。開発者が作成する。

コンパイル

ソースファイルをコンピュータが理解できる状態に変換。

Javaではjavacコマンド（コンパイラ）でコンパイルする。(※1)

クラスファイル

コンピュータ(JavaではJavaVM)が理解できる状態。中間コード(バイトコード)

JavaVM

クラスファイルを読み込んで解釈・実行を行う。

javaコマンドでJavaVMを起動する。(※2)

※ 1.今回の研修ではeclipseを使う。

※ 2.JavaVMはJIT(Just In Time)コンパイラで実行直前にネイティブコードにコンパイルする。

※ プログラミング言語の中には、コンパイルを必要とせずソースファイルを逐次解釈しながら実行出来るものがある。（インタプリタ型）