

# Java開発環境

---

# Java言語の特徴(1)

- Sun Microsystems社が開発(1996年)
- インタプリタ型
  - ◆ コンパイルにより中間言語形式のバイトコードを生成
  - ◆ CやC++と同様のパフォーマンス
  - ◆ 迅速なプロトタイピングにより開発期間を短縮可能
- オブジェクト指向
  - ◆ C++やObjective-Cの機能を統合
  - ◆ プラグ&プレイのプログラムを容易に作成可能

# Java言語の特徴(2)

- シンプル
  - ◆ CやC++のプログラマなら容易にマスター可能
  - ◆ CやC++の基本データ型を踏襲しており複雑さは逆に排除
  - ◆ 自動ガベジコレクションにより記憶領域の管理不要
- 分散型
  - ◆ オブジェクトは、ローカル上でもネットワークを経由して遠隔でも実行可能
  - ◆ TCP/IPプロトコル用の拡張可能な内蔵ライブラリ

# Java言語の特徴(3)

- 安定性
  - ◆ 厳格な型指定言語
  - ◆ コンパイル時および実行時の2重チェック
  - ◆ ポインタ演算を使用せず、メモリ領域の衝突を排除
  - ◆ データ型の恣意的なキャストはできない
- セキュリティ
  - ◆ コードチェック機能を内蔵
  - ◆ ファイルアクセスの保護
  - ◆ 公開鍵暗号化および認証

# Java言語の特徴(4)

- プラットフォーム非依存
  - ◆ アーキテクチャに依存しないコード転送
  - ◆ アプリケーションは再コンパイルせずに複数プラットフォームで動作
- 移植性
  - ◆ CやC++のような処理系依存を解消
  - ◆ 抽象GUIクラス
  - ◆ コンパイラがJava言語で記述されておりJavaシステム自体の移植性を保証

# Java言語の特徴(5)

- コンパクト
  - ◆ パーソナルデバイスでも使用できる設計
  - ◆ コンパクトな実行環境
- マルチスレッド
  - ◆ リアルタイム動作が要求されるマルチメディアのアプリケーションに最適
  - ◆ 豊富な高機能同期プリミティブ群

# Java言語の特徴(6)

- ダイナミック
  - ◆ 環境の進化にも適応できる設計
  - ◆ 新たなコードをon the fly方式でリンク
  - ◆ ダイナミックプラグインが可能なコードモジュール
- ソフトウェアプラットフォーム
  - ◆ UNIX, Windows, MacOS で利用可能

# Javaで開発されているシステム例

- Androidアプリ
- Twitter
- Google
- Minecraft
- コンビニのレジ
- ATM



# Java SE とは

- Java Standard Edition
- 多くのJavaプラットフォームプログラムで利用される Java API \* の集合体で標準的機能を提供
- JVM(Java仮想マシン)やAPIなどで構成

\*API: Application Programming Interface

# Java SE Development Kit: JDK

- Java言語でプログラム開発を行う際に必要な開発用ソフトウェアをまとめたパッケージ
- コードエディタやIDE(統合開発環境)などは付属しない
- JDKには以下のものが含まれる
  - ◆ コンパイラ
  - ◆ デバッガ
  - ◆ アーカイバ
  - ◆ Javaプログラム実行環境(JRE) など

# Oracle JDK と OpenJDK

- Oracle JDK
  - ◆ Oracle社が提供するJDK
  - ◆ 半年ごとに次のバージョンがリリース
  - ◆ 有償のサポート有り
  - ◆ 無償利用は「開発・個人のみ」
- OpenJDK
  - ◆ Oracle JDKのオープンソース実装
  - ◆ Oracle JDKに比べ無い機能がいくつかある
    - Java Web Start、JavaFXなど
  - ◆ サポートは半年（次のバージョンが出るまで）

# Java開発環境の準備

- 各自のPC上にJava開発環境を構築してみよう
- 参考サイト
  - ◆ Javaの開発環境構築方法！おすすめの開発環境5選  
<https://kuguru.jp/2930>
  - ◆ 現役システムエンジニアが教えるJavaプログラミング  
<https://note.com/mikanactive/n/nce3d5337c4b1>
  - ◆ 情報科学演習1ではAmazon Corretto 8の利用を推奨  
<https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html>

# Javaアプリケーションプログラムの作成

- 情報科学演習1の  
簡単なプログラムを作成してみよう
  - ◆ <http://www.is.noda.tus.ac.jp/2is/enshu1/index.php>?簡単なプログラム
  - ◆ このサイトはVPNで大学に接続してからアクセスします
- “Hello World”を表示するプログラム
  - ◆ 以降のスライドはこのサイトのスナップショットです

# Hello Worldアプリケーション(1)

## HelloWorldアプリケーション (HelloWorld.java)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- プログラム全体はHelloWorldと名付けられたpublic（公開，外部から呼び出せる）のクラス定義である。
- プログラムの本体は，main()というメソッドから成る。（C言語では「関数」と呼んだが，Javaでは「メソッド」という。）
- Javaアプリケーションでは，プログラムが実行されるとmainという名前のメソッドが最初に呼ばれる。
- Javaのソースファイル名は，その中で定義しているpublicなクラス名と同じ名前にして，それに拡張子.javaをつける。
- 大文字・小文字の区別があるので注意が必要。
- System.out.printlnは，画面に文字列を表示するメソッド。
- main()メソッドの引数として複数の文字列を取ることができる (String[] args, 文字列クラスの配列)。この例では引数は利用していない。

# Hello Worldアプリケーション(2)

## プログラムのコンパイル <sup>+</sup>

プログラムのコンパイルはjavacコマンドを用いる。  
引数には、コンパイルするソースファイル名を指定する。

プログラムのコンパイル

```
prompt% javac HelloWorld.java
```

プログラムをコンパイルすると、ソースファイルと同じディレクトリ内に  
HelloWorld.class というファイルが生成される。  
これはJavaバイトコードのファイルであり、クラスファイルと呼ばれる。

# Hello Worldアプリケーション(3)

## プログラムの実行 <sup>↑</sup>

バイトコードはインタプリタ（コマンド名はjava）で実行できる。  
引数には、実行するクラスファイルの拡張子.classを取り除いたファイル名を指定する。

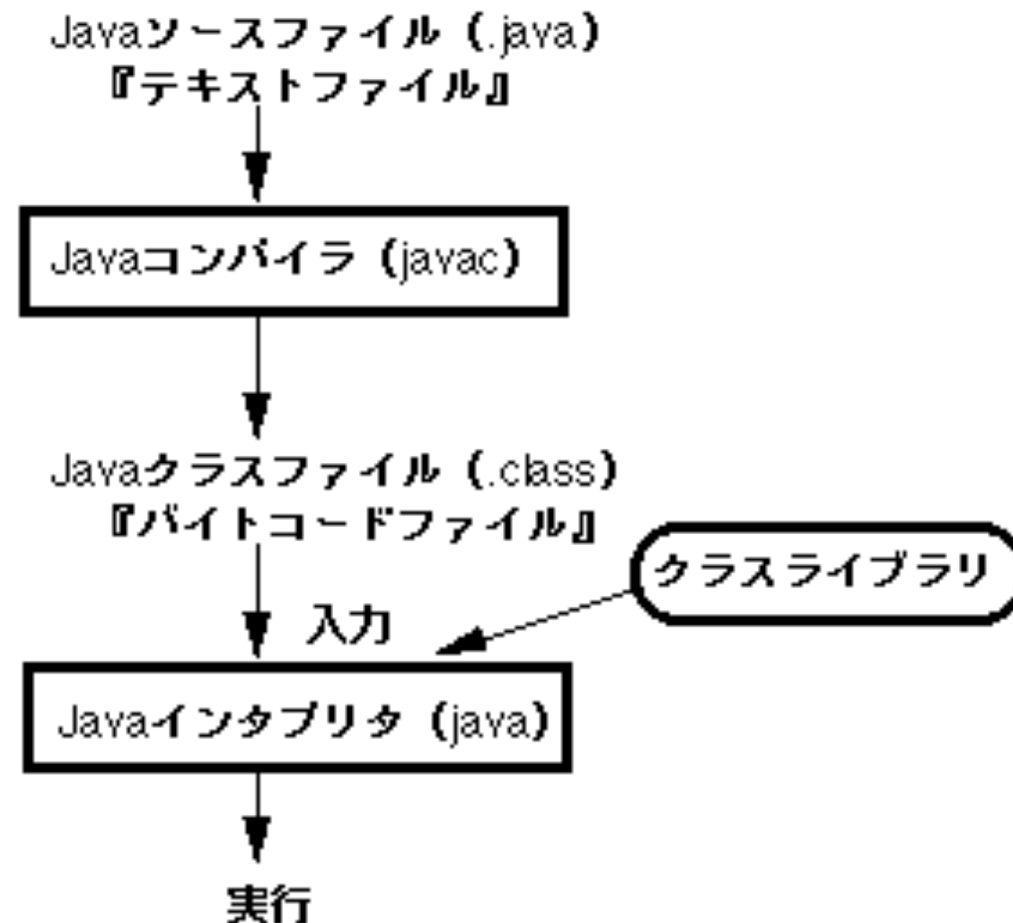
バイトコードの実行

```
prompt% java HelloWorld  
Hello World!
```



# Hello Worldアプリケーション(4)

Javaアプリケーション実行の流れ



# レポート課題

- 前述した情報科学演習1のHello Worldアプリケーションプログラムを修正して“各自の学籍番号 氏名”を表示するプログラムを作成してみよう  
出力例:  
6319121 理大太郎
- レポートにはソースプログラム(HelloWorld.java)の記述と実行結果の画面(スクリーンショット)を挿入すること  
\*使用しているPC上のスクリーンショットを取ることも学習のひとつです
- 詳細はLETUSを参照