

第1章

Java言語のプログラムの流れ

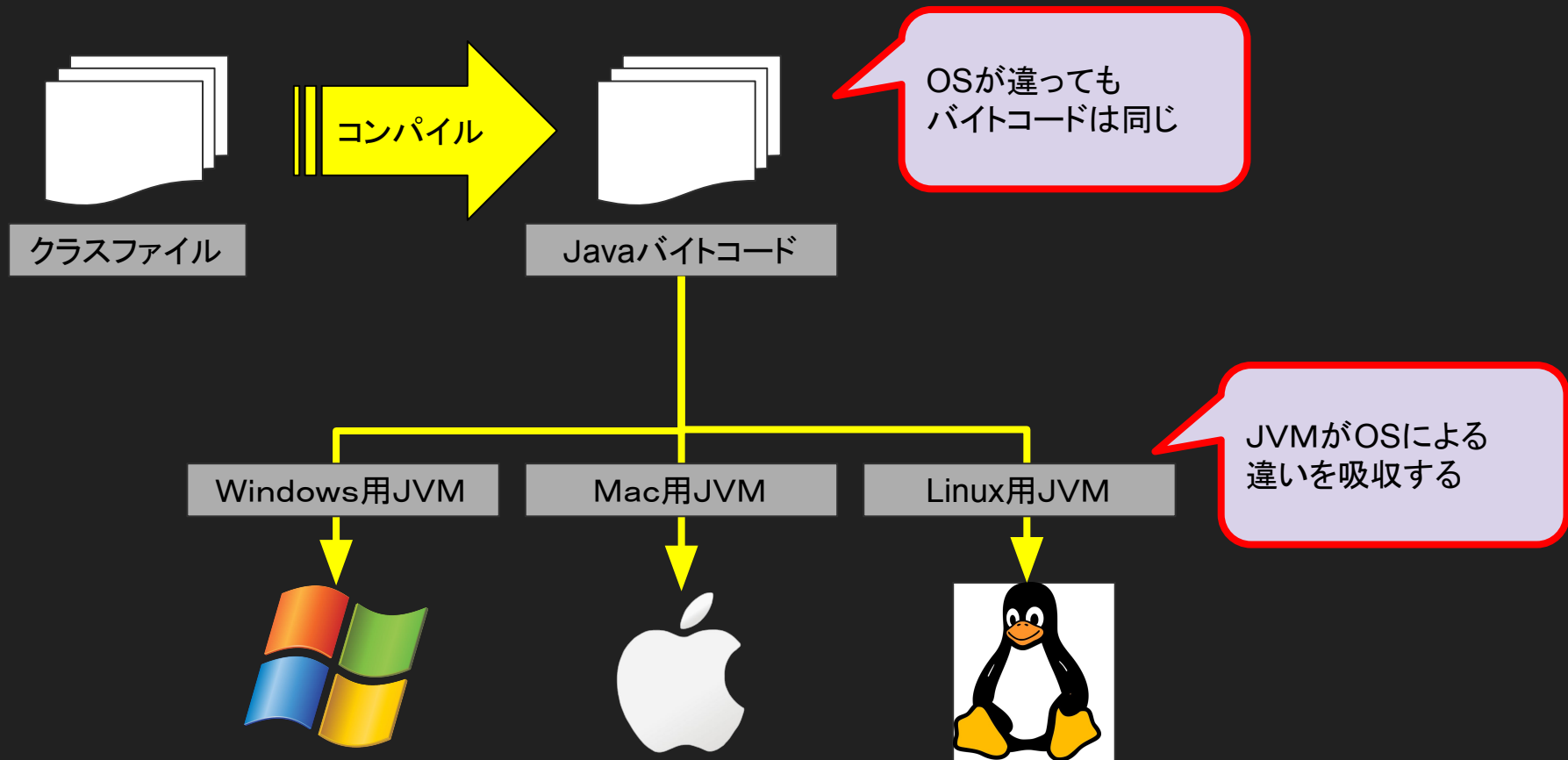
■Javaテクノロジーの特徴

1. Write Once, Run Anywhere(一度書けばどこでも動く)
2. JVMによる処理
3. 開発環境と実行環境の無償公開
4. オブジェクト指向
5. 充実したクラスライブラリ

■Javaテクノロジーの特徴

1. Write Once, Run Anywhere(一度書けばどこでも動く)
Javaは同じコードがWindows/Mac/Linux等、どのOSでも動作させることができる。
※C言語などは、OSによって書き方が異なる部分がある。
2. JVMによる処理
JavaはJVMという「仮想マシン」の上で動作する。

■Javaテクノロジーの特徴



■Javaテクノロジーの特徴

3. 開発環境と実行環境の無償公開

Javaは基本技術がすべて無償公開されている。
無償ツールも充実している。

→最近は珍しいことでもないけど...

4. オブジェクト指向

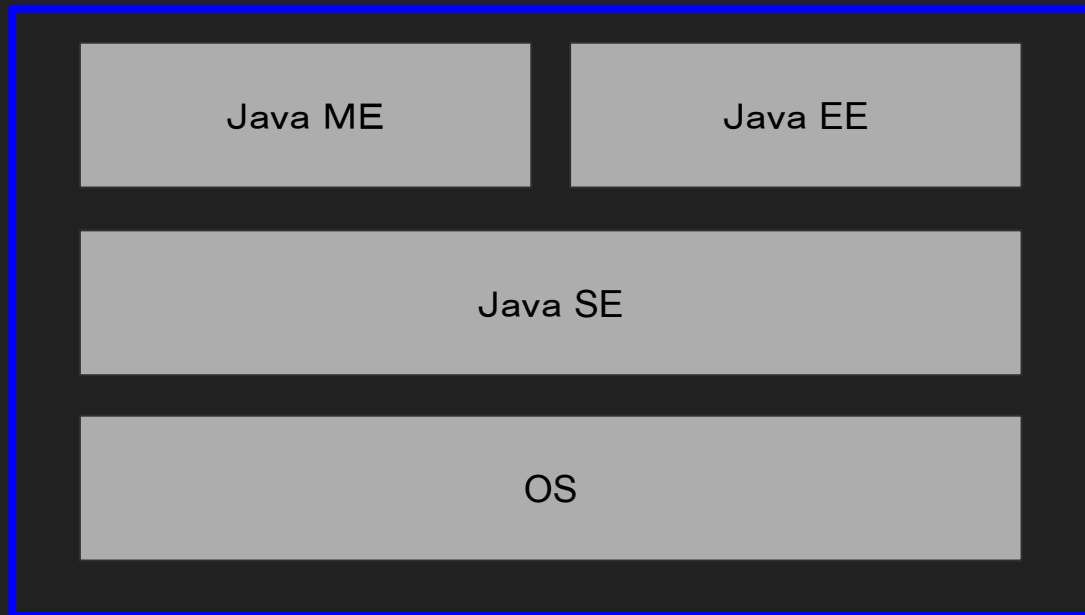
詳しい話は第5章で！！！！

5. 充実したクラスライブラリ

クラスライブラリ:よく使う機能の部品群。APIとも呼ぶ。
プログラムを効率よく作成できるようになる。

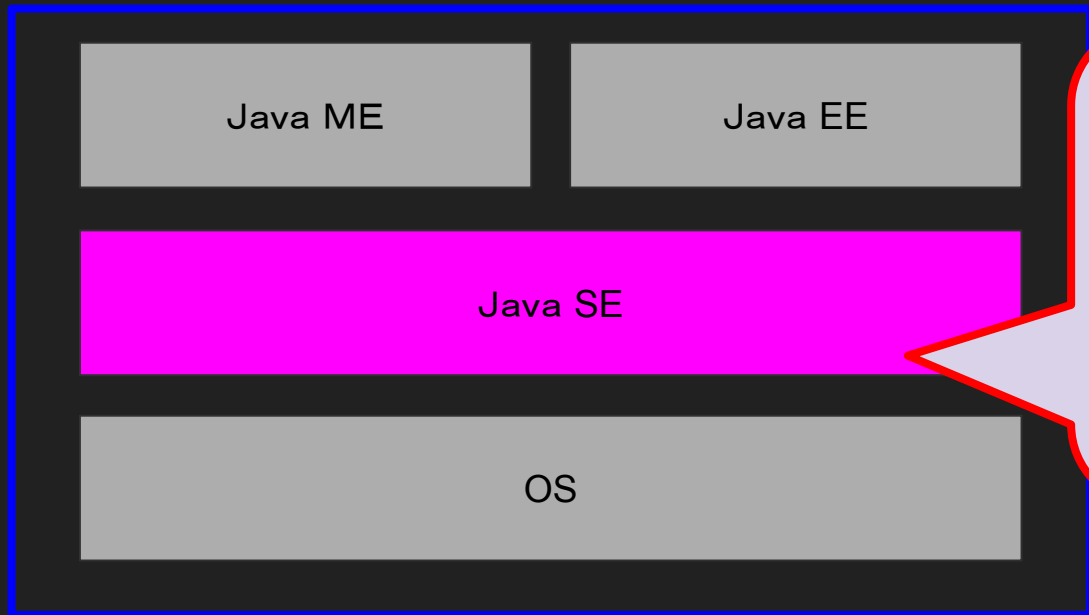
■Javaプラットフォーム各エディションの特徴

エディションの種類



■Javaプラットフォーム各エディションの特徴

エディションの種類



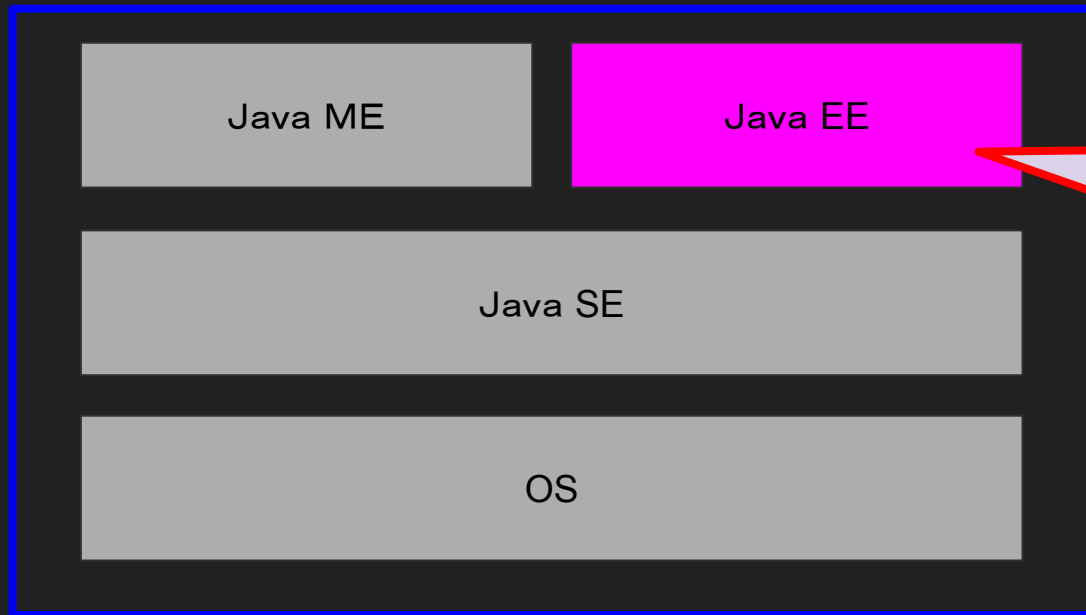
Javaの基本的な
ソフトウェア開発に必要な開発環境。

Java MEもJava EEもJava SEを
ベースに成り立っている。

Bronzeの学習範囲はこの範囲。

■Javaプラットフォーム各エディションの特徴

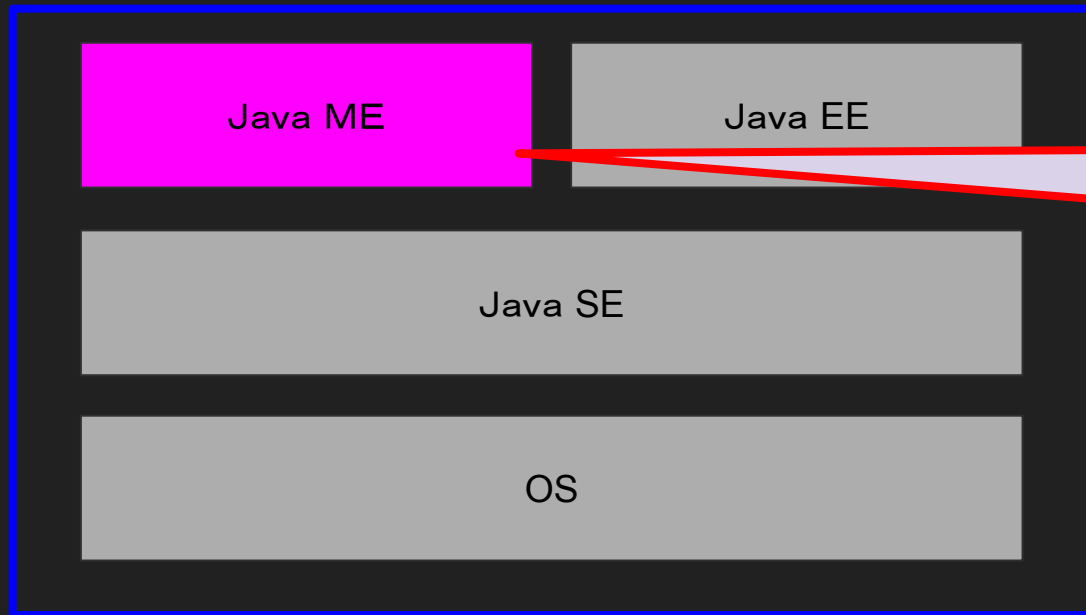
エディションの種類



Webアプリケーション用。
HTTP通信やサーバ間連携用の
処理が用意されている。

■Javaプラットフォーム各エディションの特徴

エディションの種類



組み込み機器用。

■Javaプラットフォーム各エディションの特徴

1. Java SE

全てのJavaプログラムの基礎となる機能を提供。

Java ME、Java EEの機能が必要でさえなければ、Java SEのみでプログラムを書くことができる。

Java SEは、①Java SE API ②JRE ③JDKで構成される。

■Javaプラットフォーム各エディションの特徴

1. Java SE

①Java SE API

Javaが提供するクラス群。よく使用されるクラスが含まれる。

例) Stringクラス、ArrayListクラス など

■Javaプラットフォーム各エディションの特徴

1. Java SE

②JRE

Java Runtime Environment、つまり実行環境のこと。
Javaで書かれたプログラムを実行するだけであれば、JRE
さえあれば良い。

■Javaプラットフォーム各エディションの特徴

1. Java SE

③JDK

人間が書いたJavaのクラスファイルをバイトコードに変換するためのjavacツールなど、開発に必要な機能が含まれる。

■Javaプラットフォーム各エディションの特徴

2. Java ME

組み込み機器に向けたJava開発を行うためのプラットフォーム。

※

Android端末のコードもJavaで書かれているけど、Java MEは使っていない。

Java SEによって作られた『Android』という独自のプラットフォームによって動いている。

■Javaプラットフォーム各エディションの特徴

3. Java EE

Webアプリケーション用の開発を行うためのプラットフォーム。
JSP(HTMLを動的に生成するための仕組み)や
EJB(サーバ間連携の仕組み)を持っている。

※

S2以外では、Javaの仕事 = Java EEの出番となります。

■問題の回答例

問題 1 - 11 正解:B

The screenshot shows an IDE interface with a code editor and a console. The code editor contains the following Java code:

```
1 public class Main {  
2     public static void main() {  
3         System.out.println("Hello");  
4     }  
5 }  
6  
7
```

Annotations on the screenshot include:

- A red box around the word `Main` in the code, with a callout bubble asking: クラス名はMainでないといけない？ (The class name must not be Main, is it?)
- A red box around the "実行 (Ctrl-Enter)" button, with a callout bubble asking: 実行時エラー (Runtime error)
- A red box around the "実行時エラー" (Runtime error) label in the console, with a callout bubble asking: 実行時エラー (Runtime error)

The console output shows the following error message:

```
Error: Main method not found in class Main, please define the main method as:  
    public static void main(String[] args)  
or a JavaFX application class must extend javafx.application.Application
```


■問題の回答例

問題 1 - 11 正解:B

クラス名はMainでないといけない？
paiza.ioを使う上でのルール？

```
1 public class Main {
2     public static void main() {
3         System.out.println("Hello");
4     }
5 }
6
7
```

実行 (Ctrl-Enter)

実行時エラー

Error: Main method not found in class Main, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application

ちなみに...

```
1 public class Main2 {
2     public static void main() {
3         System.out.println("Hello");
4     }
5 }
6
7
```

コンパイルエラーだとこうなる

コンパイルエラー

Main.java:1: error: class Main2 is public, should be declared in a file named Main2.java
public class Main2 {
 ^
1 error