

プログラミング I

Java言語 (JDK)

開発環境Eclipseの使い方

文教大学 情報学部 経営情報学科
堀田 敬介

Java言語 —概要—

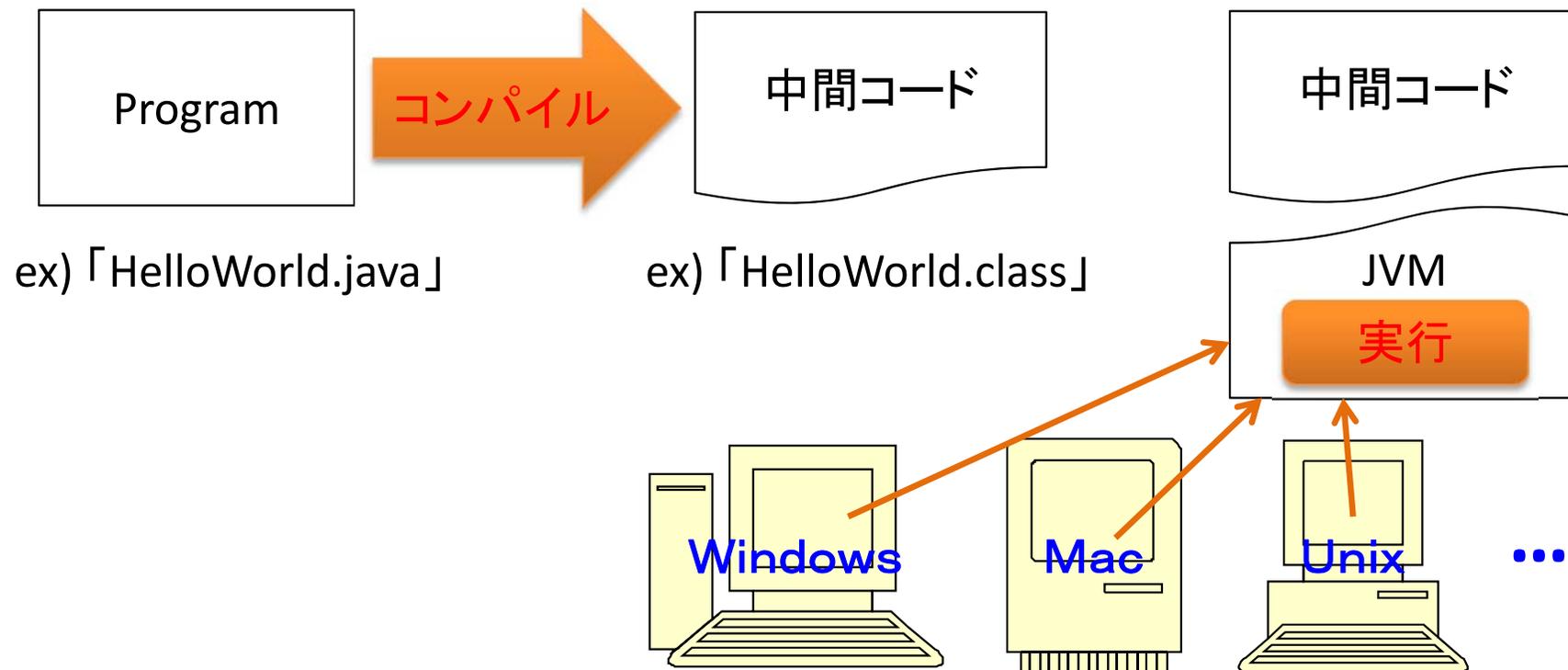
プログラミング言語

➤ 代表的なプログラミング言語

- Basic/Visual Basic, C/C++, C#, Java, ...
- JavaScript, PHP, Perl, Python, ...

➤ プログラミング言語の種類

- **インタプリタ型**: 「プログラムファイル」を1行ずつ解釈して実行
- **コンパイラ型**: 「プログラムファイル」をコンパイルして「実行ファイル」を生成し、それを実行. Javaは「実行ファイル」の代わりに「中間コード」を生成し、「JVM (Java Virtual Machine, Java仮想機械)」上で実行する



Java言語 —programming—

Javaによるプログラミング

➤ Java言語:

➤ 現在公開されているバージョン

➤ JDK7, JDK6 (Java Development Kit)

➤ ダウンロードサイト: <http://java.sun.com/javase/ja/6/download.html>

➤ Eclipse : 代表的なJava開発環境の一つ

➤ 現在公開されているバージョン

➤ Eclipse3.7 (Indigo), 3.6 (Helios), 3.5 (Galileo), 3.4 (Ganymede), 3.3 (Europa)

➤ Pleiades【日本語化プラグイン】: <http://mergedoc.sourceforge.jp/>

EclipseでJavaプログラミング —簡易版—

プロジェクトの作成

- Step1. 新規プロジェクト作成
- Step2. プロジェクト名の設定

- 完了

[ファイル]—[新規]—[Javaプロジェクト]
[プロジェクト名]に好きな名前を入力 ex)sample01
[ロケーション]で作業場所(フォルダ)を設定

ソースコードファイルの作成

- Step0. プロジェクトの選択
- Step1. 新クラス作成
- Step2. クラス設定

- 完了

[パッケージ・エクスプローラ]で上記プロジェクト選択
[ファイル]—[新規]—[クラス]
[パッケージ]設定 ex) jp.taro
[名前]にクラス名入力(半角英数・1文字目英大文字)
mainクラスなら, 《public static void ...》にチェック

ソースコードの記述

- Step1. プログラミング

上記ソースコードファイルにプログラムを書く

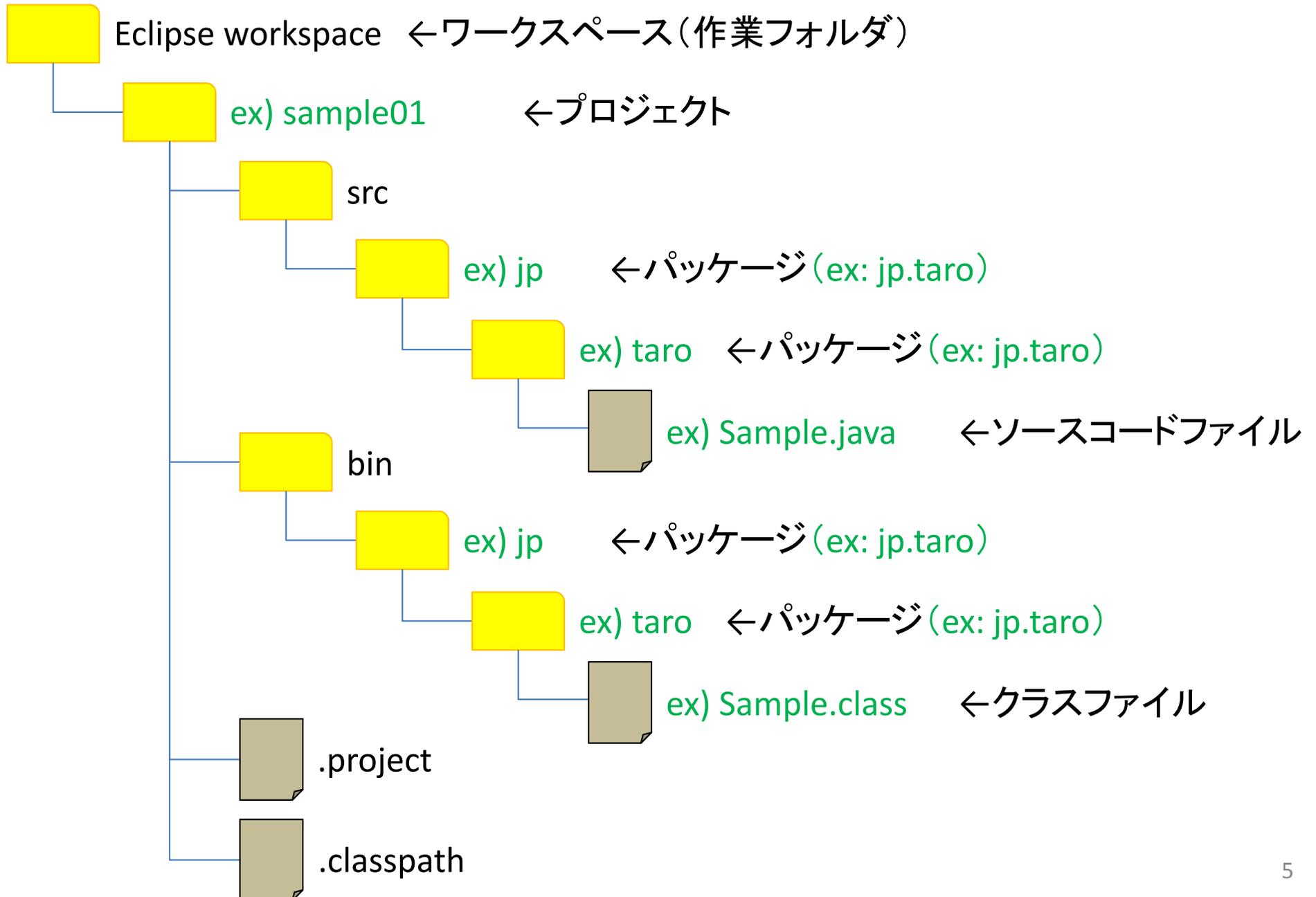
プロジェクトのビルドと実行

- Step0. ビルド
- Step1. 実行

[プロジェクト]—[自動的にビルド]がチェック済ならOK
三角アイコンを押して実行
→問題がなければ, コンソールウィンドウに結果表示

EclipseでJavaプログラミング —簡易版—

作成されるファイル



EclipseでJavaプログラミング —簡易版—

デバッグ

- ▶ デバッグモードで実行
 - ▶ ビュー: デバッグビュー, 変数ビュー, ブレークポイントビュー, 式ビュー, etc.
- ▶ ブレークポイント: ブレークポイント設定場所まで実行
- ▶ ステップイン: 1処理ずつ実行, メソッド呼び出し時はメソッド内移動で同様
- ▶ ステップオーバー: 1処理ずつ実行, メソッド呼び出し時はメソッド全実行
- ▶ ステップリターン: 実行中のメソッドを最後まで実行して呼び出し元に戻る
- ▶ 監視式
- ▶ 変数の値変更
- ▶ etc.

プロジェクトの終了

- ▶ プロジェクトの終了 [パッケージ・エクスプローラー]内のプロジェクト右クリック
→[プロジェクトを閉じる]

Javaプログラミングの基本

クラスの定義とメソッド

```
修飾子 class クラス名 {  
    フィールド
```

...

```
    修飾子 返回值 メソッド名(引数の指定) {  
        実行処理  
        ...  
    }
```

```
    修飾子 返回值 メソッド名(引数の指定) {  
        実行処理  
        ...  
    }
```

```
}
```

メソッドの戻り値・返り値

ex) void, byte, short, int, long,
char, float, double, boolean, ...

```
public class Sample {
```

```
    public static void main(String[] args) {  
        System.out.println("Hi!");  
        System.out.println("Ya!");  
    }
```

```
}
```

アクセス修飾子

ex) public, protected, private

その他の修飾子

ex) static, abstract, final, ...