2014年9月22日

プログラミング

Javaによるプログラミング

堀田　敬介

プログラミングについて

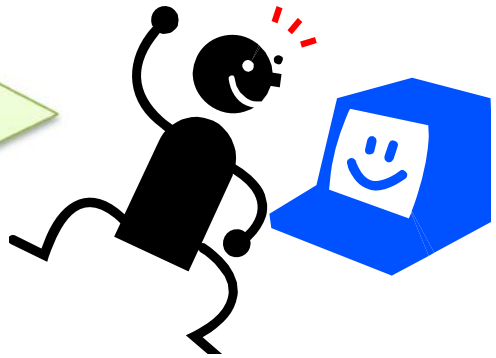
- ありとあらゆる場所でコンピュータが使われている
  - 通信機器（スマホ，携帯，…）
  - 家電（冷蔵庫，エアコン，TV，電子レンジ，…）
  - 社会インフラ（金融業ATM，コンビニ等POS，…）
  - 輸送・流通（鉄道，航空，車，宅配，…）
  - 通貨（電子マネー，クレジット，デビットカード，…）
  - 製造（工場，SCM，建築，…）
  - ITC（SE，コンテンツ，クリエイター，Web，…）
  - 問題解決ツール（コンサルタント，研究開発，…）
  - 政策立案（都市計画，年金，介護，病院，…）
  - etc….
- そして，
  コンピュータを動かしているのはプログラムである

プログラミングについて

- **人間** と **コンピュータ** の **役割分担**

それぞれの得意分野で能力を発揮しよう！

人間
創造性，演繹，推論，
直感，臨機応変…



コンピュータ
大量，高速，反復，
持久力，正確，愚直
…

- **プログラム** …コンピュータに指示を出す命令セット
  - 身の回りの多くのものがプログラムで動いている
  - プログラムを組む（プログラミング）ために様々な言語がある

- **プログラミング** …命令の出し方によって，指示されたコンピュータの処理性能が大幅に異なる

上手い命令の出し方 ＝ 創造的な仕事をする人間の役割 ＝ アルゴリズム データ構造

プログラミング言語について

- どの言語を勉強したら良いのかな…　（マイナビニュース 2014/5/16）

5月PYPLプログラミング言語人気 - 引き続きPythonが上昇

後藤大地　[2014/05/16]

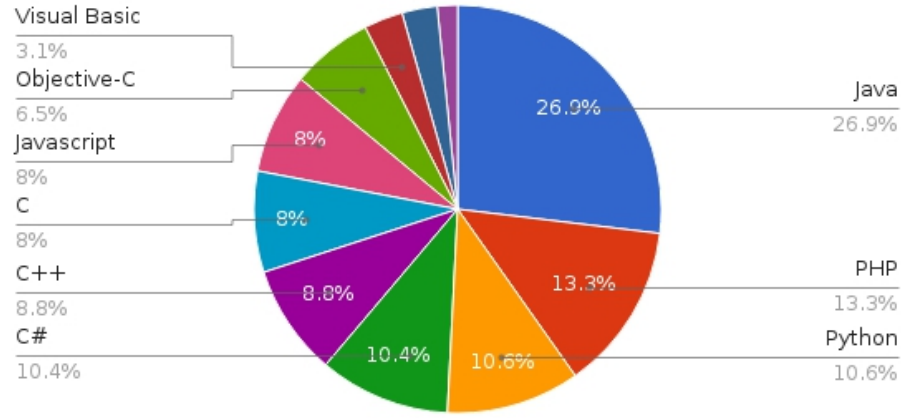
2014年5月の「PYPL PopularitY of Programming Language Index」が公開された。

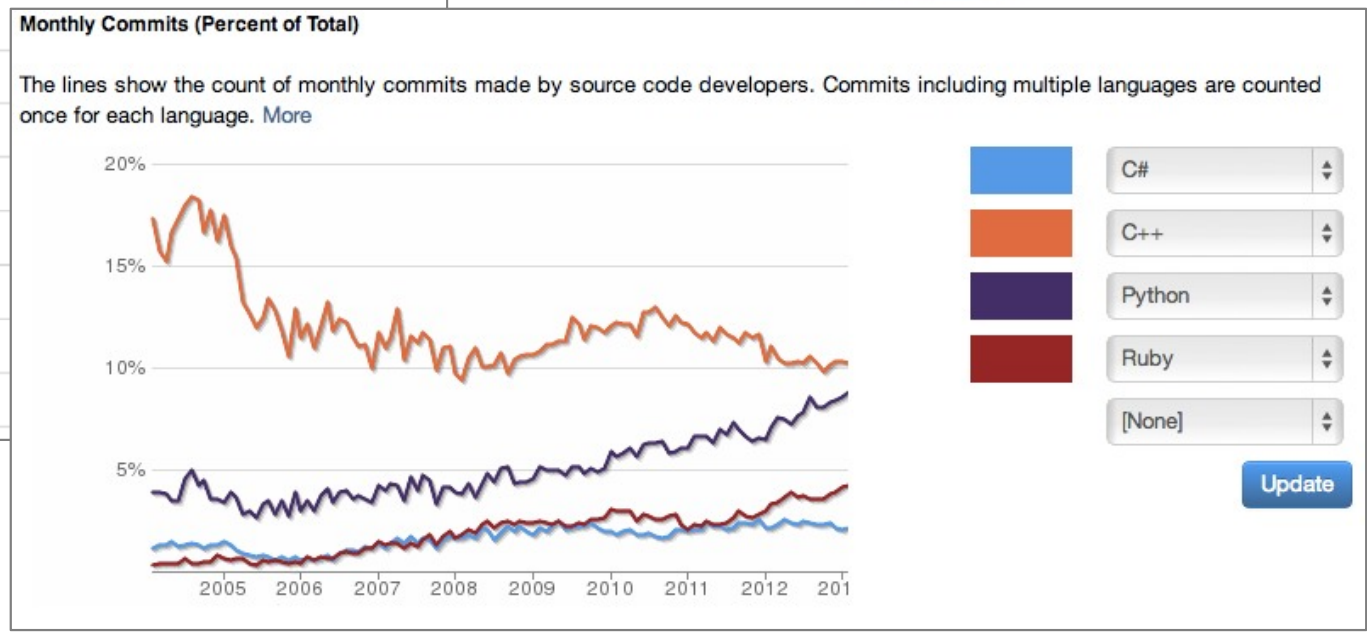
PYPLはGoogle検索エンジンにおいてプログラミング言語のチュート
なるプログラミング言語がどれだけ話題になっているかをインデック
回数を人気度と位置づけてランキングしている。Google Trendのデー
おり、PYPLページのリンクを辿っていけば自国におけるそれぞれの害

2014年5月におけるインデックスは次のとおり。

順位	プログラミング言語
1	Java
2	PHP
3	Python
4	C#
5	C++
6	C
7	Javascript
8	Objective-C
9	Visual Basic
10	Ruby

(Popularity of Programming Lang. Index 2014/5)





プログラミング言語について

- どの言語を勉強したら良いのかな…

RedMonkによる四半期毎プログラミング言語の人気ランキング（GitHub/StackOverflowで）

ランク算出方法　Drew Conway, John Myles White（2010）
RedMonk　産業調査会社
GitHub　オープンソースのプロジェクトホスティングサイト
StackOverflow　プログラミングのQ&Aサイト

	言語	2012年 9月	2013年 1月	2013年 6月	2014年 1月
1	JavaScript	1位	1位	2位	1位
2	Java	2位	2位	1位	2位
3	PHP	3位	3位	3位	3位
4	C#	6位	6位	6位	4位
5	Python	4位	4位	4位	5位
6	C++	7位	7位	7位	6位
7	Ruby	5位	5位	5位	7位
8	C	8位	8位	8位	8位
9	Objective-C	9位	9位	9位	9位
10	CSS	圏外	圏外	圏外	10位
11	Perl	11位	10位	11位	11位
12	Shell	10位	11位	10位	12位
13	Scala	12位	12位	12位	13位
14	Haskell	13位	14位	14位	14位
15	R	17位	17位	16位	15位
16	Matlab	圏外	20位	19位	16位
17	Clojure	圏外	圏外	圏外	17位
18	CoffeeScript	19位	18位	17位	18位
19	Visual Basic	18位	19位	20位	19位
20	Groovy	20位	圏外	18位	20位

"get.stack.R"

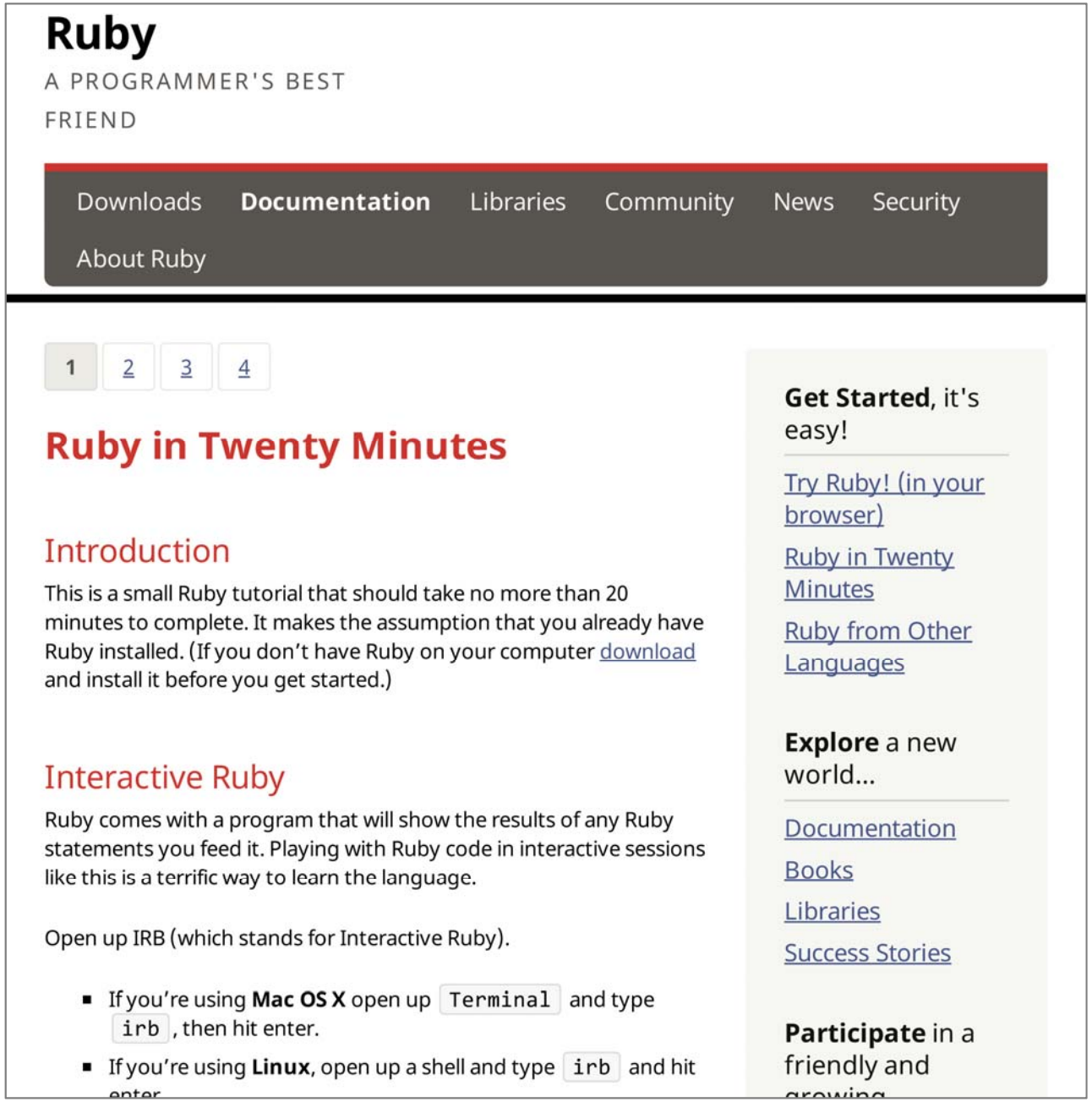
```
# Get StackOverflow data
get.stack<-function(tok) {
    # Must check for XML install, thanks onertipaday!
    if (!require(XML)) install.packages('XML')
    library(XML)
    # Enter a SO tag as character string, and number of tags are returned
    tok<-gsub("(/| )","-",tok)
    tok<-gsub("#","%23",tok,fixed=TRUE)
    base.stack<-"http://stackoverflow.com/questions/tagged/"
    stack.tree<-htmlTreeParse(paste(base.stack,tok,sep=""),useInternalNodes=TRUE)
    tag.count<-getNodeSet(stack.tree,"//div[@class='module']/div[@class='summarycount al']")
    tag.num<-suppressWarnings(as.numeric(gsub(",","",xmlValue(tag.count[[1]]),fixed=TRUE)))
    if(is.na(tag.num)) {
        warning(paste("Something went wrong trying to parse '",tok,"'.\nNA returned",sep=""))
    }
    return(tag.num)
}
```

プログラミング言語について

- どの言語を勉強したら…？

- 各言語には特徴があり，向き不向きがある
- 初心者は好きなのを（偶々出会った言語を）勉強すれば良い
- 1つ勉強すれば2つ目以降の修得は容易
- 今はWeb上でいくらでも勉強できる！

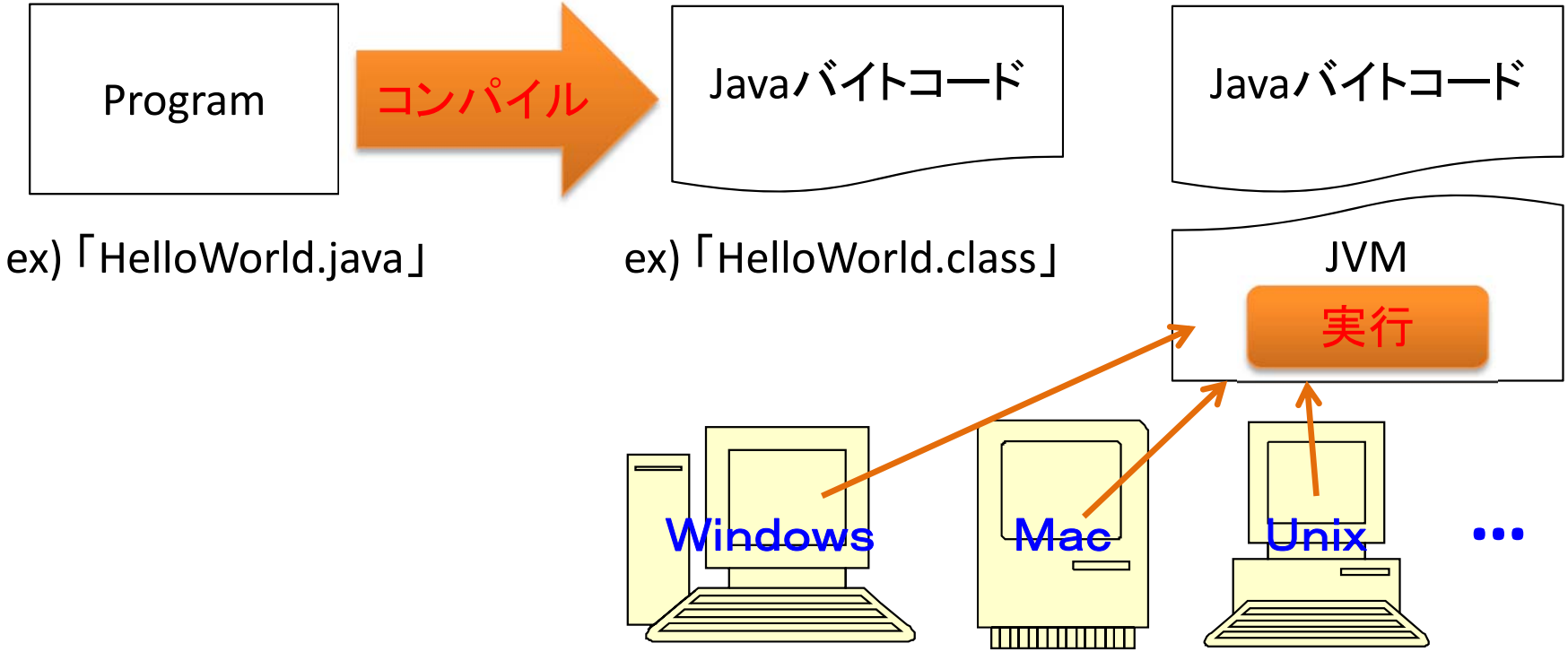
https://www.ruby-lang.org/en/documentation/quickstart/



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「Javaバイトコード」を生成し，「JVM（Java Virtual Machine, Java仮想機械）」上で実行する



Javaによるプログラミング

- **Java言語**:
  - **Java SE**（Java Platform, Standard Edition）
  - 現在公開されているバージョン 8u20 [version 8, update20] (2014/9/21)
  - ダウンロードサイト:
    http://www.oracle.com/technetwork/java/javase/overview/index.html

- **開発環境**: **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
    - Eclipse4.4（Luna）, 4.3（Kepler）, 4.2（Juno）
    - 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】
  - ダウンロードサイト：http://mergedoc.sourceforge.jp/

- **簡易開発環境**: **Cpad for Java2SDK**
  - Cpad Suite：他にCpad for Borland C++Compiler, Cpad for C#.NET など
  - 現在公開されているバージョン：2.31
  - ダウンロードサイト：http://hp.vector.co.jp/authors/VA017148/

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 返値 メソッド名（引数の指定）{
        実行処理
            …
    }
    修飾子 返値 メソッド名（引数の指定）{
        実行処理
            …
    }
}
```

```
public class Sample {

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

}
```

メソッドの戻り値・返り値

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

アクセス修飾子

ex) public, protected, private

その他の修飾子

ex) static, abstract, final, …

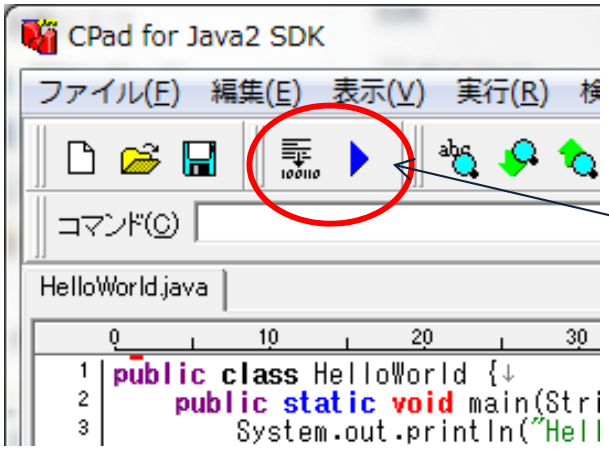
例）HelloWorld.java

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println(“Hello World!”);
    }

}
```

ファイル名：　HelloWorld.java

- 半角・英数字
  （使える記号は限られる）
- 「クラス名」と「ファイル名」は，大文字・小文字など完全に一致していないとエラー
- ファイル拡張子は[.java]
- クラス名は大文字で始めるのが慣例

「コンパイル」すると，Javaバイトコードを含むファイル「HelloWorld.class」が作成される



プログラムを書いて保存したら，「コンパイル」して「実行」しよう

- 左＝コンパイル・ボタン　…　コンパイルのみ行う
- 右＝コンパイル&実行ボタン … コンパイルし，実行する

エラーの時は，「エラーメッセージ」をよく読んで対処
上手くいけば，「コマンドプロンプト」画面（黒い画面）が出て，実行結果が表示される

参考：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　[ファイル]ー[新規]ー[Javaプロジェクト]
- Step2. プロジェクト名の設定　[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
- 完了

ソースコードファイルの作成

- Step0. プロジェクトの選択　[パッケージ・エクスプローラ]で上記プロジェクト選択
- Step1. 新クラス作成　[ファイル]ー[新規]ー[クラス]
- Step2. クラス設定　[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
- 完了

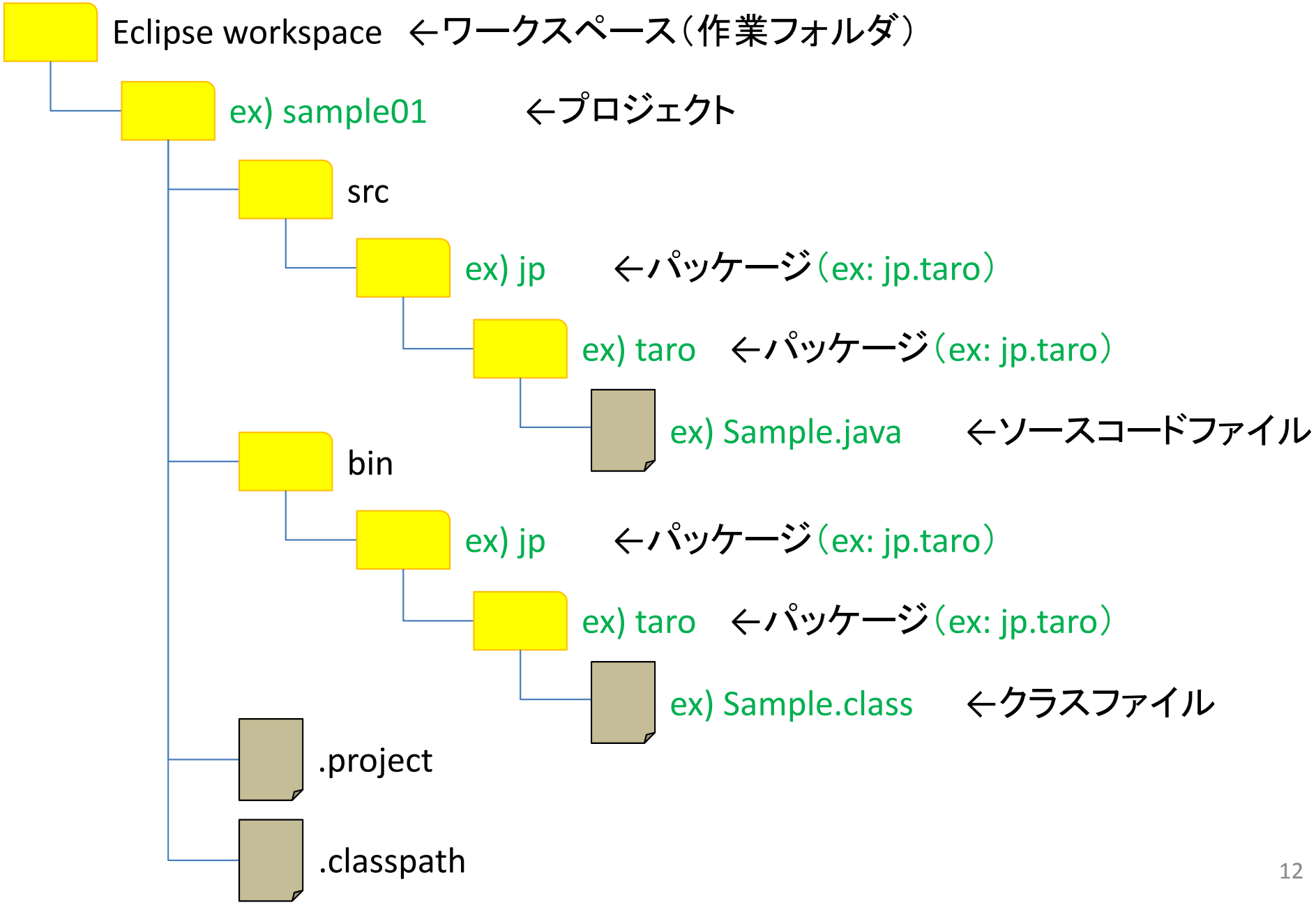
ソースコードの記述

- 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　[プロジェクト]ー[自動的にビルド]がチェック済ならOK
- Step1. 実行　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

作成されるファイル


Eclipse workspace　←ワークスペース（作業フォルダ）
ex) sample01　←プロジェクト
src
ex) jp　←パッケージ（ex: jp.taro）
ex) taro　←パッケージ（ex: jp.taro）
ex) Sample.java　←ソースコードファイル
bin
ex) jp　←パッケージ（ex: jp.taro）
ex) taro　←パッケージ（ex: jp.taro）
ex) Sample.class　←クラスファイル
.project
.classpath

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

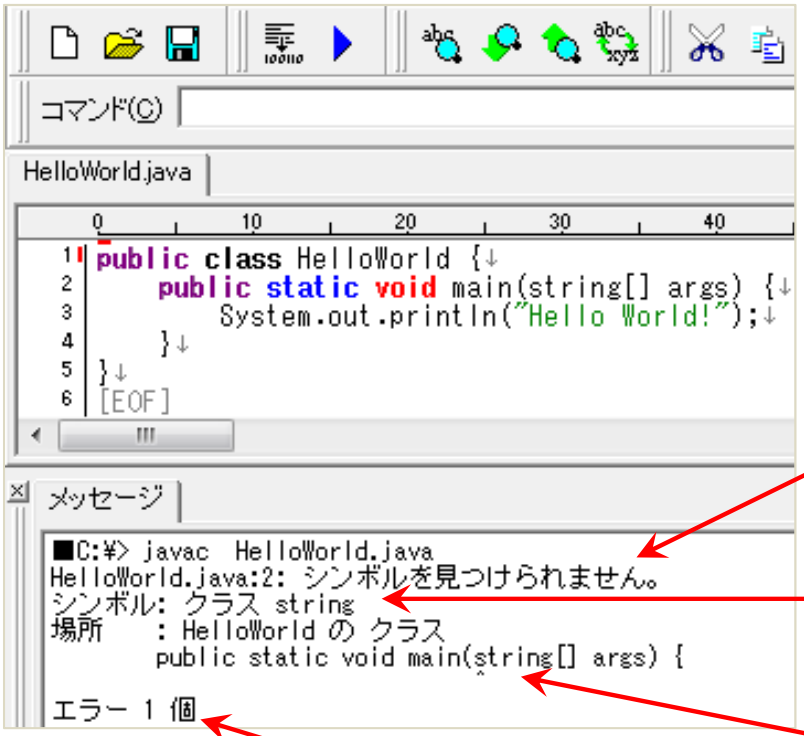
- プロジェクトの終了　[パッケージ・エクスプローラー]内のプロジェクト右クリック
  →[プロジェクトを閉じる]

コンパイルと実行


コンパイル
CPad for Java2 SDK
ファイル(F)
編集(E)
表示(V)
実行(R)
コマンド(C)
HelloWorld.java
0
10
20
30
1 public class HelloWorld {
2 public static void main(Stri
3 System.out.println("Hell
コンパイル & 実行

エラー処理（バグとり）

- プログラムの間違いを直す第1歩は，エラーメッセージを注意深く読むこと



エラーメッセージ（エラーの内容）
ex) 2行目で「シンボルを見つけられない」
→ 2行目にJavaの知らない言葉
→ 綴りを間違えた可能性が高い

そのシンボルとは【クラス string】らしい

場所はここ（ハット記号【^】で示してある）
string の頭文字を小文字で書いてるミス発見

今回のコンパイルで見つけたエラーは1個
（エラーを修正し，再コンパイルすると，別のエラーが見つかる場合がある）