

Javaプログラミング実習

31. ライブライ

株式会社ジーードライブ

今回学ぶこと

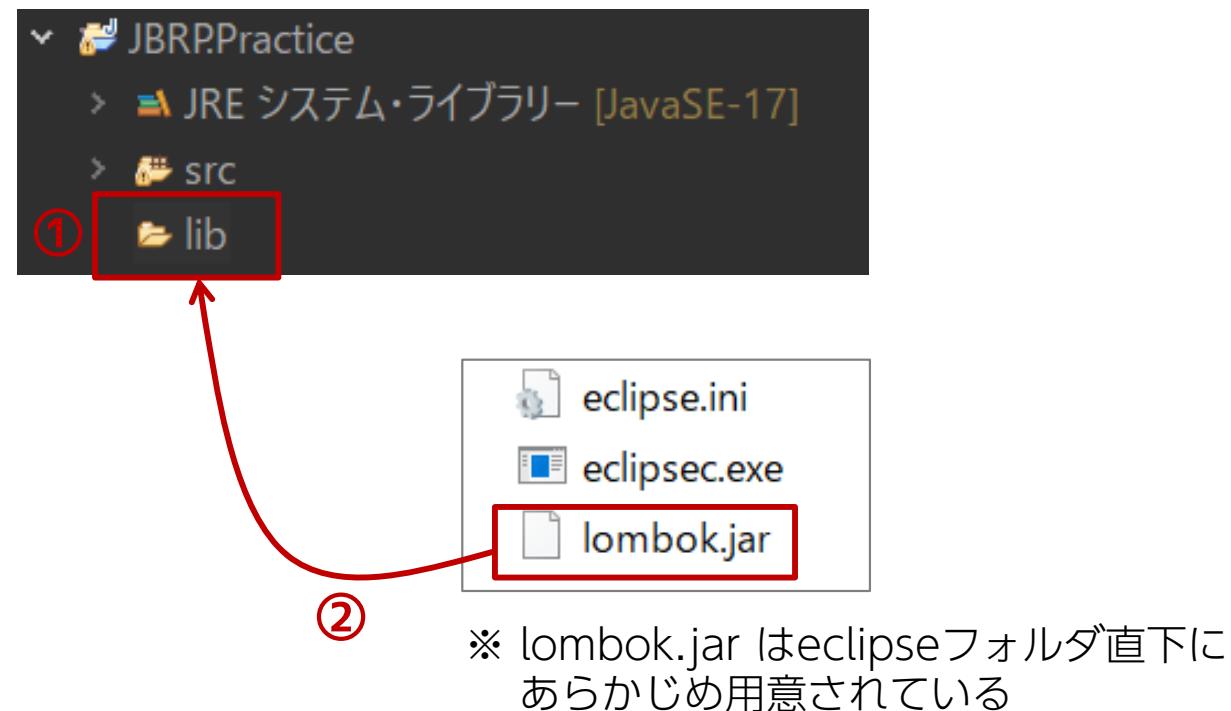
- ライブラリとは
- ライブラリの利用方法
- ライブラリの例: Lombok
- 補足: Maven

ライブラリとは

- 再利用を目的としたプログラムの部品(プログラムのファイル)
 - 独自のライブラリを作成することも可能だが、第三者が作成したライブラリをダウンロードして利用することも可能
 - ライブラリを導入することで、使用できる機能が増える
- Javaの場合は、クラスの集合であるクラスライブラリとして提供されている
 - Javaには標準で使えるライブラリが数多く存在する
 - 別途ライブラリを導入することで、使用できるクラスが増える
 - ライブラリは、JARという形式で提供されており、JARファイル(拡張子は.jar)を、自身のプロジェクトに読み込んで利用する

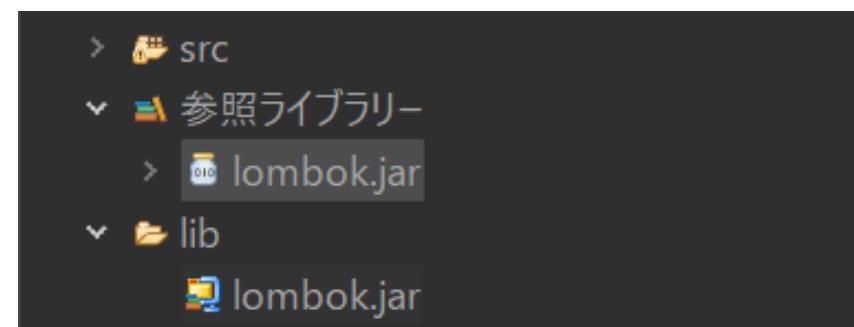
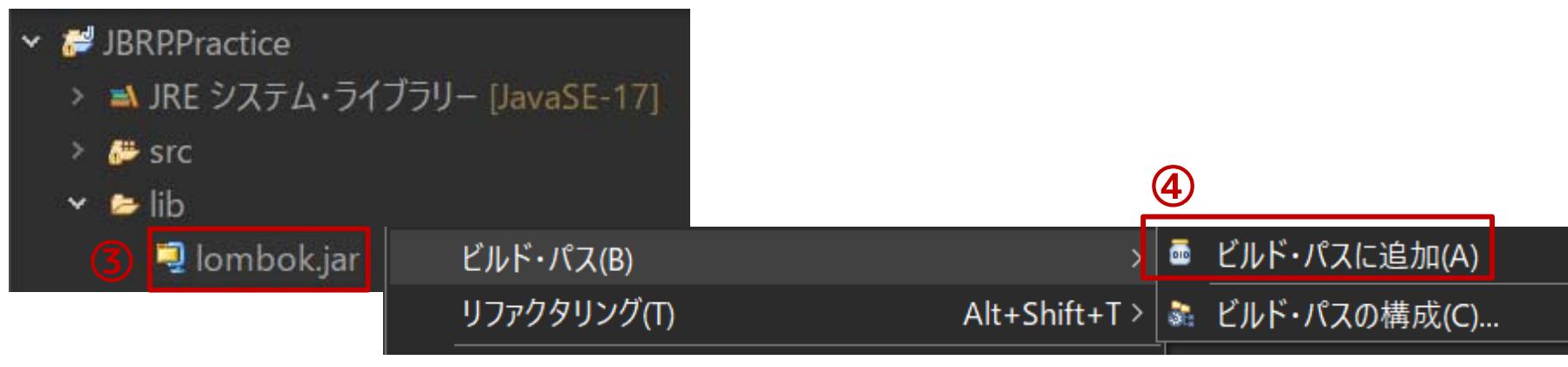
ライブラリの利用方法

- ① プロジェクト内に任意のフォルダを用意する
- ② JARファイルをドラッグ&ドロップ
 - ファイルをコピーにチェックを入れて、OKを押す



ライブラリの利用方法

- ③ JARファイル上で右クリック
- ④ ビルド・パス → ビルド・パスに追加
 - 参照ライブラリーとして追加される



ライブラリの例: Lombok

- Lombokは、アクセッサやコンストラクタなど、何度も書かなければいけない定型のコード(ボイラープレートコード)を自動生成するライブラリ
- クラスやフィールドにアノテーションを付与することで生成するメソッドを制御することができる

Lombokの利用例：引数付きのコンストラクタ、アクセッサの自動生成

@AllArgsConstructor

@Data

```
public class Item {  
    private String name;  
    private int price;  
}
```

ライブラリの例: Lombok

- 基本的なアノテーション

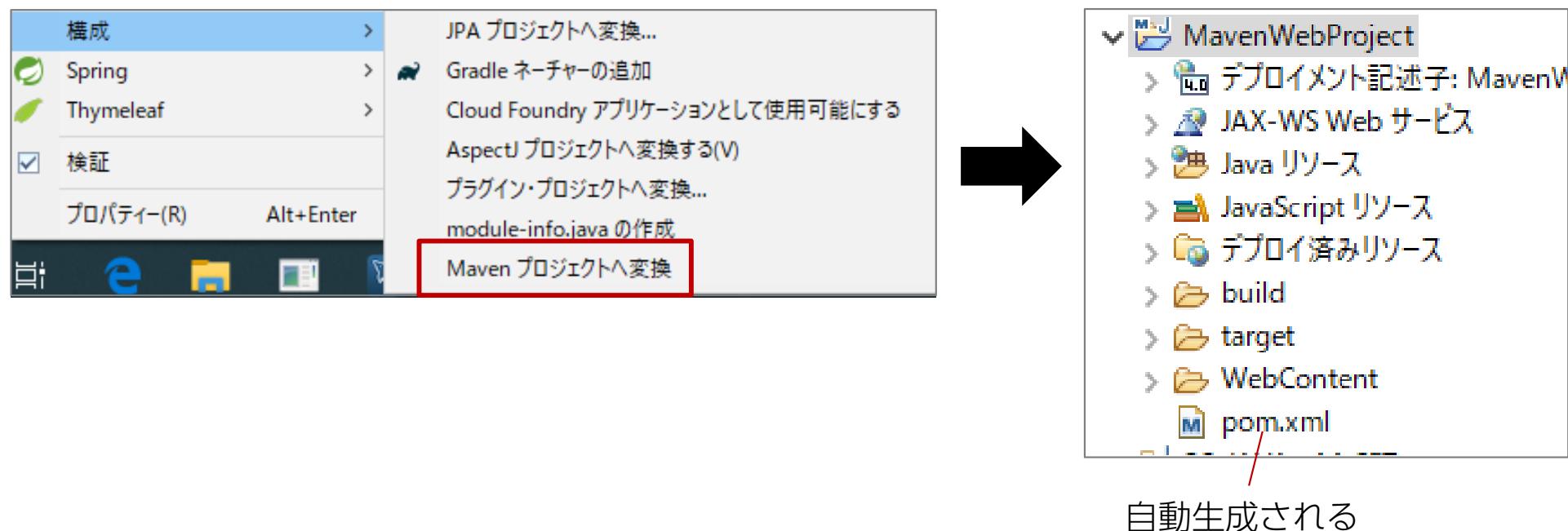
アノテーション	説明
@NoArgsConstructor	引数なしのコンストラクタを生成
@AllArgsConstructor	全フィールドを引数にとるコンストラクタを生成
@Getter	ゲッターメソッドを生成
@Setter	セッターメソッドを生成
@ToString	toString()メソッドの生成
@Data	アクセッサやtoString()などを生成
@Value	ゲッターメソッドやtoString()などを生成

[補足]

Mavenによるライブラリ管理

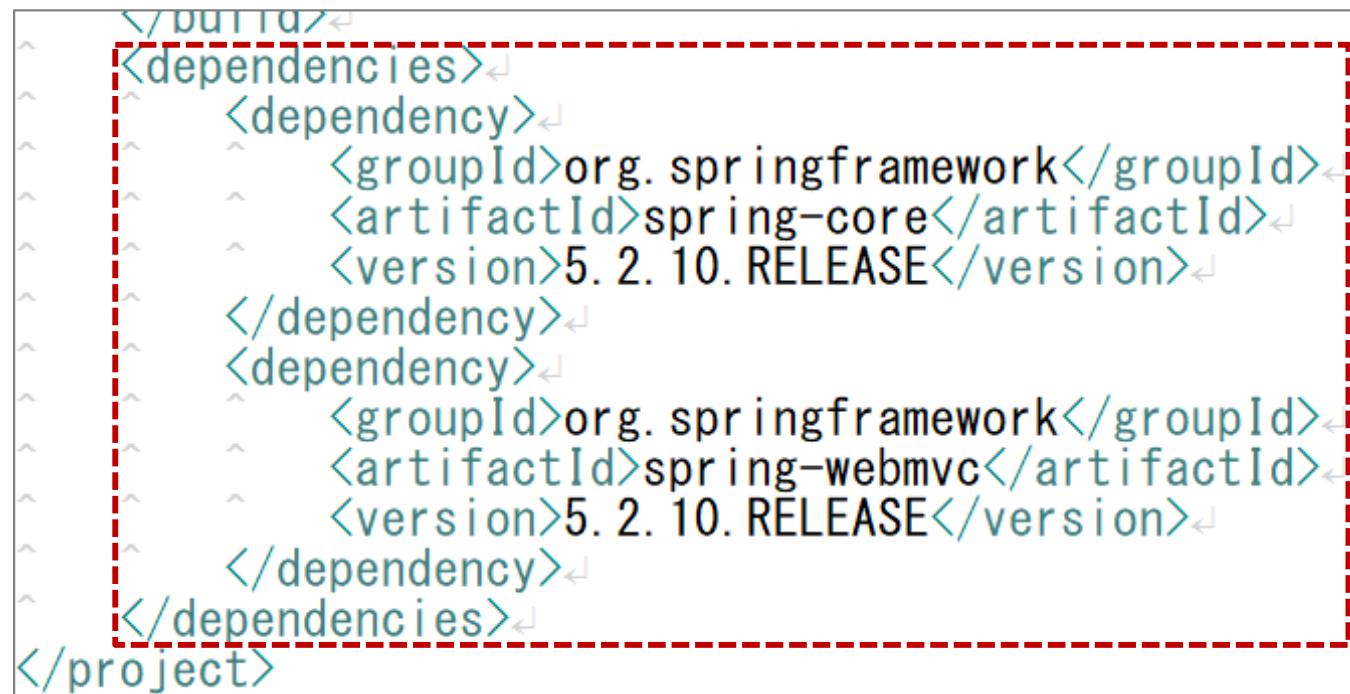
Mavenの利用方法

- MavenはJava用のプロジェクト管理ツール
- Eclipseでプロジェクトフォルダを右クリックし、「構成 ⇒ Mavenプロジェクトへ変換」する
 - pom.xmlというファイルが生成され、Maven管理のプロジェクトになる



pom.xmlの記述

- pom.xmlには、プロジェクトについての基本情報や使用するライブラリについての記述を行う
 - 使用するライブラリの情報は、dependencies要素内に記述

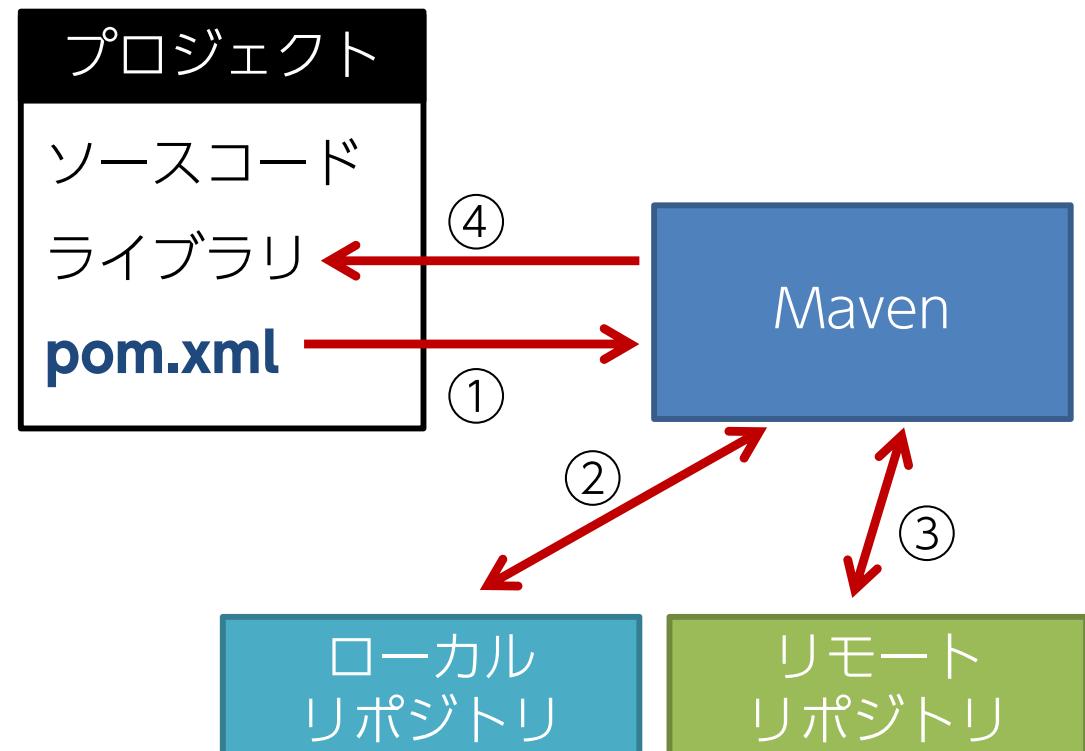


```
</build>
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>5.2.10.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>5.2.10.RELEASE</version>
    </dependency>
</dependencies>
</project>
```

Mavenによるライブラリの管理

- Mavenはpom.xmlに従い、必要なライブラリをダウンロードして、利用可能な状態にする

- pom.xmlに必要なライブラリを記載
- ローカルリポジトリをチェック
- ローカルにない場合、リモートリポジトリから取得し、ローカルに保存
- ライブラリとして使用可能になる



C:\Users\zdXXXX\.m2 内にダウンロードされる

Maven Repositoryの利用

- 使用するライブラリについては、Maven Repositoryで検索する (<https://mvnrepository.com/>)

The screenshot shows the search results for 'spring mvc' on the mvnrepository.com website. A red box highlights the search input field containing 'spring mvc'. To the right of the input field is a red circle with the number '①' and the text '検索' (Search). Below the search bar, the text 'Found 18042 results' is displayed. Underneath, sorting options are shown: 'Sort: relevance | popular | newest'. The first result is highlighted with a red box and labeled '1. Spring Web MVC'. It includes the package name 'org.springframework > spring-webmvc', the name 'Spring Web MVC', and the last release date 'Last Release on Oct 27, 2020'. A second result is partially visible below it, labeled '2. Spring Context'.

spring mvc ① 検索

Found 18042 results

Sort: [relevance](#) | [popular](#) | [newest](#)

1. Spring Web MVC
org.springframework > spring-webmvc
Spring Web MVC
Last Release on Oct 27, 2020

2. Spring Context
org.springframework > spring-context
Spring Context
Last Release on Oct 27, 2020

Maven Repositoryの利用

- 利用するバージョンを選び、dependency要素をコピー
⇒ pom.xmlに貼り付ける



① 利用するバージョンを選択

The screenshot shows a code editor with tabs for 'Maven', 'Gradle', 'SBT', 'Ivy', and 'Grape'. The 'Maven' tab is selected. A red box highlights the following XML code:

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc/5.2.10.RELEASE -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.2.10.RELEASE</version>
</dependency>
```

② コピーして、pom.xmlに貼り付ける

groupId : 開発者／開発組織を示すID
artifactId : プロダクトのID
version : プロダクトのバージョン