1.3 Java EEによるWebアプリケーション開発

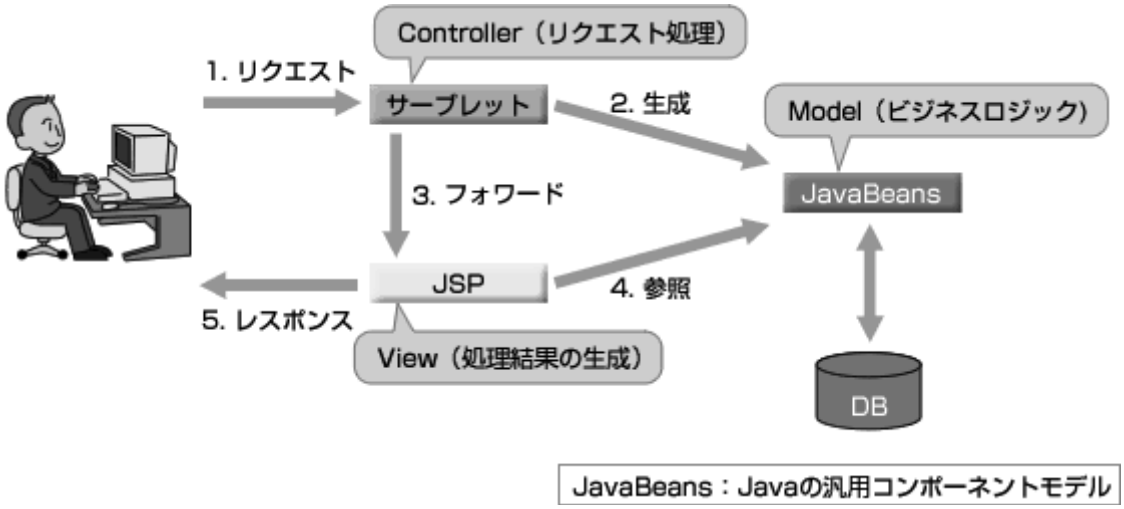
Java EEを用いて開発するWebアプリケーションは、既定のフォルダ構成にしたがって構築します。ここでは、Java EEにおけるWebアプリケーションのフォルダ構成と構成要素を説明します。

1.3.1 Java EEにおけるWebアプリケーション

Java EEにおけるWebアプリケーションとは、サーブレット、JSP、Webコンテンツ（HTML、PDF、GIF、Javaアプレット）など、Java EEのWeb層に置くファイルをパッケージ化したものです。

Web層に置くファイルの中でもJavaのプログラムであるサーブレットやJSP、ビジネスロジックなどは、保守性や効率性を考え、3つの役割（Model、View、Controller）に分けて実装を行います。この方法はそれぞれの頭文字をとって「MVCパターン」と呼ばれ、元はGUI機能を持つアプリケーションを効率的に開発できるように考案されたデザインパターン[1]です。

サーブレットやJSPの詳細、連携の実装方法は後述します。

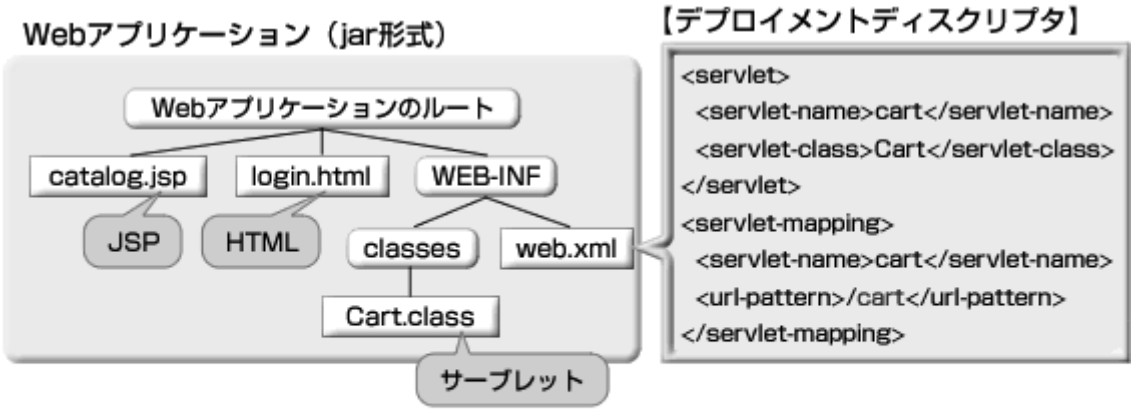


構成要素	説明
Model	アプリケーションのデータとその入出力を担う。 Java EEではJavaBeansコンポーネントやEJBコンポーネントで実装。
View	ユーザー操作の受け付け、Modelの表示を担う。 Java EEではJSPで実装。
Controller	ViewとModelの制御を担う。Viewからの入力に応じて、Modelにデータの変更を依頼し、その結果の表示をViewに依頼する。 Java EEではサーブレットで実装。

[1] デザインパターンとは、よく使われるクラス設計のパターンのことです。Web アプリケーションに関連するデザインパターンの詳細は「Web アプリケーション詳細設計(JavaEE 編)」コースで取り上げます。

1.3.2 Webアプリケーションの構成

Webアプリケーションにパッケージ化するファイルは、規定の階層構造に従って配置します。Webコンテンツ とJSPファイルはWebアプリケーションのルートディレクトリ以下に配置し、サーブレットなどのクラスファイルは”WEB-INF/classes”ディレクトリに配置します。



【各ファイルを呼び出すURL】

JSP	‥ http://サーバーのホスト名:ポート番号/WebアプリケーションのURL/catalog.jsp
HTML	‥ http://サーバーのホスト名:ポート番号/WebアプリケーションのURL/login.html
サーブレット	‥ http://サーバーのホスト名:ポート番号/WebアプリケーションのURL/cart

Webアプリケーションの構成情報は「デプロイメントディスクリプタ（web.xml）」に定義します。このファイルには、サーブレットのクラス名、サーブレットを呼び出すURL、アクセス制御の情報、ユーザー認証の方法などを定義します。

Webコンテンツ とJSPファイルは、そのファイル名を示すURLをブラウザで指定して呼び出します。サーブレットはURLをweb.xml[1]に定義し、そのURLをブラウザで指定して呼び出します。

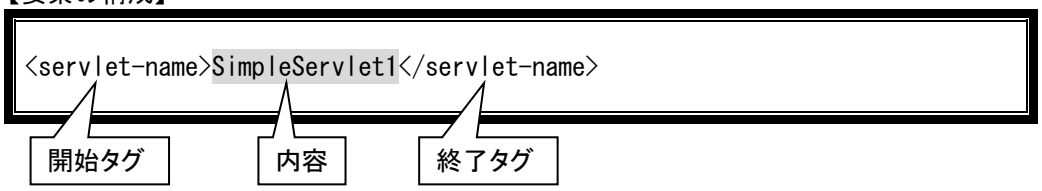
[1] JavaEE5 ではメタデータアノテーションの使用により、標準の web.xml デプロイメント記述子を省略することも可能です。

1.3.3 デプロイメントディスクリプタ(web.xml)

web.xmlはXML文書です。XMLの構造を表すスキーマはXMLスキーマ[1]を採用しています。

XML文書は、開始タグ、内容、終了タグによって構成される「要素」の組み合わせによって記述します。XMLは階層化された構造となっており、ルート要素を1つ持ちます。web.xmlのルート要素は「web-app」です。

【要素の構成】



■ デプロイメントディスクリプタ(記述例)

web.xml

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2
 3 <web-app xmlns="http://java.sun.com/xml/ns/Java EE"
 4          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 5          xsi:schemaLocation="http://java.sun.com/xml/ns/Java EE
 6          http://java.sun.com/xml/ns/Java EE/web-app_2_5.xsd"
 7 version="2.5" >
 8
 9    <context-param>
10        <param-name>message</param-name>
11        <param-value>welcome</param-value>
12    </context-param>
13
14    <session-config>
15        <session-timeout>20</session-timeout>
16    </session-config>
17
18    <servlet>
19        <description> SimpleServlet1</description>
20        <servlet-name>SimpleServlet1</servlet-name>
21        <servlet-class>servlet.SimpleServlet1</servlet-class>
22        <init-param>
23            <param-name>servletParam</param-name>
24            <param-value>false</param-value>
25        </init-param>
26    </servlet>
27
28    <servlet-mapping>
29        <servlet-name>SimpleServlet1</servlet-name>
30        <url-pattern>/Servlet1</url-pattern>
31    </servlet-mapping>
32
```

[1] 以前は DTD(Document Type Definition)が採用されていました。

```
33      <filter>
34          <filter-name>Filter1</filter-name>
35          <filter-class>filter.Filter1</filter-class>
36      </filter>
37
38      <filter-mapping>
39          <filter-name>Filter1</filter-name>
40          <url-pattern>/Filter1</url-pattern>
41      </filter-mapping>
42
43      <filter-mapping>
44          <filter-name>Filter1</filter-name>
45          <servlet-name>SimpleServlet1</servlet-name>
46      </filter-mapping>
47
48      <welcome-file-list>
49          <welcome-file>index.html</welcome-file>
50          <welcome-file>index.jsp</welcome-file>
51          <welcome-file>default.jsp</welcome-file>
52      </welcome-file-list>
53  </web-app>
```

web.xmlに記述する主要な要素を以下に示します。

- servlet

  サーブレットの宣言などを指定します。

要素	説明
<description>	サーブレットの説明文。
<servlet-name>	サーブレット名を定義。 web.xml 内でサーブレット定義を参照する際に使用。
<servlet-class>	サーブレットのクラス名。完全修飾クラス名を指定する。
<init-param>	サーブレット初期化パラメータの名前と値の組み合わせを指定。

- servlet-mapping

  サーブレットとリクエストURLの間のマッピングを指定します。

要素	説明
<servlet-name>	サーブレット名を定義。 servlet 要素内の servlet-name 要素で定義した名前に対応。
<url-pattern>	サーブレットを呼び出す URL を定義。 対応するサーブレットが呼び出される。

- context-param

  webアプリケーションの初期化属性の名前や値の組み合わせを指定します。

要素	説明
<param-name>	属性名を定義。
<param-value>	属性値(文字列)を定義。

- init-param

  servlet内の要素。サーブレットの初期化属性の名前や値の組み合わせを指定します。

要素	説明
<param-name>	パラメータ名を定義。
<param-value>	パラメータ値(文字列)を定義。

- welcome-file-list

URL指定の際、ファイル名が省略された場合に自動的に該当ディレクトリから検索されるデフォルトページを指定します。

要素	説明
<welcom-file>	デフォルトのウェルカムファイルとして使用するファイル名を定義。

- session-config

セッションを期限切れにするまでの時間を指定します。

要素	説明
<session-timeout>	セッションが期限切れになるまでの分数を定義。

- error-page

HTTPのエラーコードや例外のタイプとwebアプリケーションにあるリソースのパスとのマッピングを指定します。

要素	説明
<error-code>	有効な HTTP エラーコードを定義。
<exception-type>	Java の例外クラスの完全修飾クラス名を定義。
<location>	エラー時に表示されるリソース配置場所を定義。

- filter

複数のサーブレットなどで使用するリクエストやレスポンス[1]の加工などの処理をサーブレットとは分離したクラスで用意しフィルタとして指定します。

要素	説明
<filter-name>	フィルタ名を定義。 デプロイメント記述子内でフィルタ参照する際に使用される。
<filter-class>	フィルタの完全修飾クラス名。

- filter-mapping

フィルタクラスと適用するサーブレットの間のマッピングを指定します。

要素	説明
<filter-name>	フィルタ名を定義。
<url-pattern>	マップされたフィルタを呼び出す URL。
<servlet-name>	フィルタを実行するサーブレット名。

[1] リクエストに対するフィルタは「事前処理」、レスポンスに対するフィルタは「事後処理」として機能します。

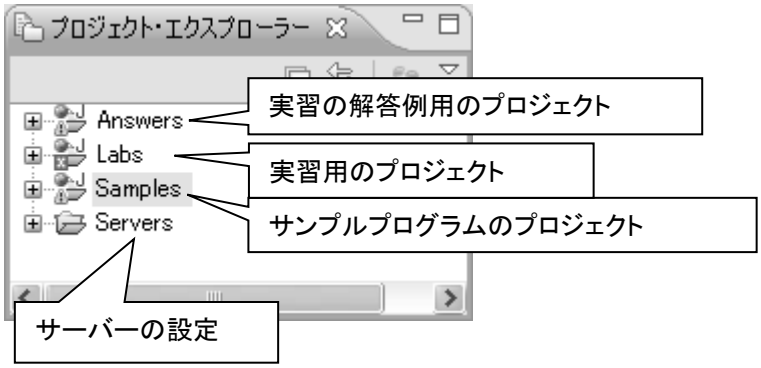
（参考）実習環境のディレクトリ構成

本コースでは実習環境としてEclipseとWTPプラグインを利用します。

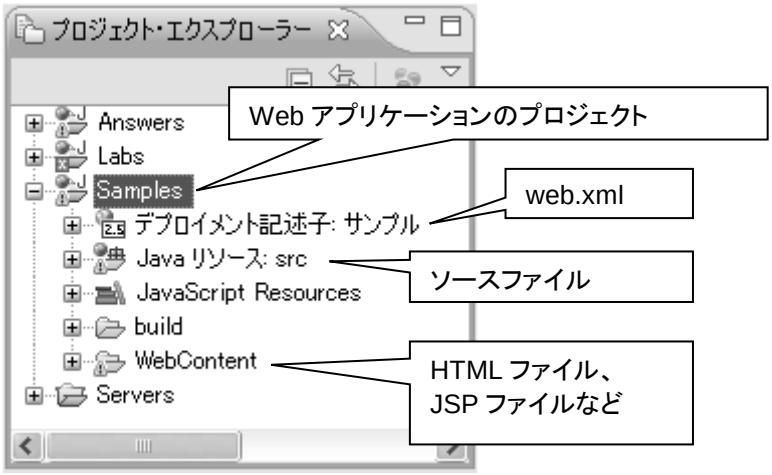
Eclipseは、オープンソースの統合開発環境です。開発に必要なコンパイラ、デバッガなどが標準で同梱されていますが、プラグインを組み込むことでさらに機能拡張ができます。ここでは、WTP（Web Tools Platform）プラグインを利用します。WTPプラグインは、サーブレットやJSPなどサーバサイド開発に必要な機能を提供するプラグインです。

実習用のWebアプリケーションは、Eclipseの動的Webプロジェクトとして提供しています。プロジェクトとは、ソースファイルやクラスファイル、設定ファイルなどアプリケーションを構成するファイルを1つにまとめたものです。

実習環境のプロジェクト[1]は次のとおりです。



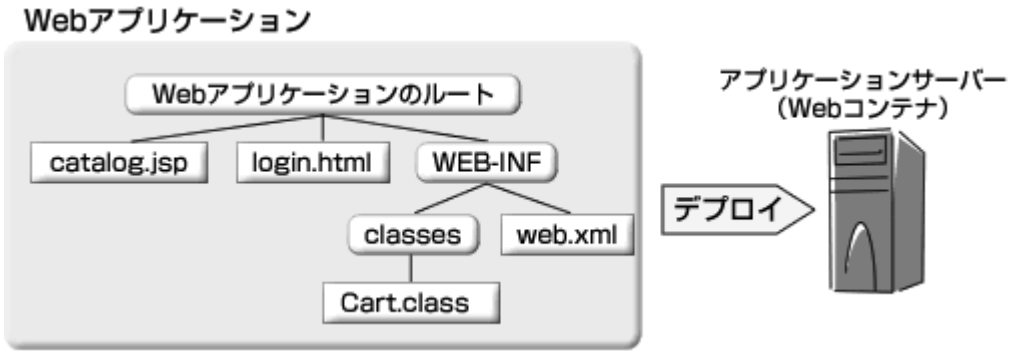
プロジェクトの階層構造は次のとおりです。プロジェクト・エクスプローラーで表示される階層構造は、Webアプリケーションの規定の階層構造とは異なっていますが、実際にサーバーにデプロイされるファイルは規定に則った階層構造になっています。Javaファイルは、「Javaリソース：src」フォルダに格納します。HTMLファイルやJSPファイルは「WebContent」フォルダに格納します。



[1] Labs プロジェクトには、未完成のプログラムが含まれているため、エラーマークが付いています。

1.3.4 Webアプリケーションのデプロイ

WebアプリケーションをWebコンテナに登録することを「デプロイ」といいます。Webアプリケーションをコンテナにデプロイすると、Webアプリケーションにパッケージ化されているコンテンツが公開され、サーブレットやJSPの実行が可能になります。



具体的なデプロイの方法は、使用するアプリケーションサーバーによって異なります。

Tomcatでは、Webアプリケーションを「Tomcatをインストールしたディレクトリ¥webapps」ディレクトリにコピーする事でデプロイが自動的に行われます。