

JSP 프로그래밍

(재)대덕인재개발원

22 Tiles

- ▶ 1. Composite View 패턴
- ▶ 2. Tiles 적용



22.1 Composite View 패턴?

- ▶ 레이아웃 코드의 중복을 피하기 위한 방법론. – 모듈화된 웹페이지에서 각 모듈에 해당하는 페이지의 내용이 다르기때문에 각 요청에 대한 응답페이지 내에서 레이아웃을 형성하는 코드의 중복이 불가피하고, 이를 해결하기 위한 방법으로 Composite View 패턴이 제시됨.
- ▶ Composite View 패턴은 레이아웃을 구성하는 정보를 담고 있는 템플릿을 생성하여 이용.
- ▶ Composite View 패턴 구현 라이브러리 : Tiles, SiteMesh등(Layout Framework / Layout Engine ... 등으로 불림.)



22.1 Composite View 패턴?

Layout 구성

```
<table>
  <tr>
    <td colspan="2">
      [헤더조각 삽입코드]
    </td>
  </tr>
  <tr>
    <td>
      [메뉴조각 삽입코드]
    </td>
    <td>
      [내용조각 삽입코드]
    </td>
  </tr>
  <tr>
    <td colspan="2">
      [풋터조각 삽입코드]
    </td>
  </tr>
</table>
```

22.2 Tiles 적용(1)

- ▶ Tiles 엔진 초기화
 - ▶ TilesListener 를 DD에 등록.
 - ▶ 컨텍스트 파라미터로 설정파일 등록.
- ▶ 레이아웃 작성.
- ▶ Tiles 설정파일 작성.
 - ▶ 레이아웃 목록이 기술된 파일.
 - ▶ 레이아웃은 definition으로 등록.
- ▶ Definition 을 사용하는 뷰파일 작성.



22.2 Tiles 적용(2)

- ▶ Tiles 엔진 초기화
- ▶ 레이아웃 작성.
- ▶ Tiles 설정파일 작성.
- ▶ Body 를 제외한 공통모듈의 내용이 상황에 따라 다를 경우 ,
ViewPreparer의 구현체를 생성하여 이용.
- ▶ TilesDispatcherServlet 을 등록하여 요청 uri와 definition 을 연결.

