普通高等教育“十三五”规划教材

新工科建设之路·计算机类专业规划教材

Java EE 开发简明教程

——基于 Eclipse+Maven 环境的 SSM 架构

吴志祥　钱　程　王晓锋　鲁屹华　编著

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材

Java EE 开发简明教程

——基于Eclipse+Maven环境的SSM架构

吴志祥　钱　程　王晓锋　鲁屹华　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本教材系统地介绍了 Java EE 的基础知识及框架开发，共有 8 章，依次包括 Java EE 概述及开发环境搭建、使用 JSP 开发 Web 项目、使用 Servlet 开发 Web 项目、ORM 框架 MyBatis、Spring MVC 框架、Spring 框架、SSM 架构和当今流行的 Spring Boot 项目开发。

本教材结构合理，内容从简单到复杂、循序渐进、逻辑性极强，重要的知识点都配有使用案例，配套的课程网站包括相关软件下载、上机实验指导（含项目案例）、课件下载和课程档案文件下载等，可作为高等院校开设“Java EE 开发技术”课程的教材和编程爱好者的参考读物。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Java EE 开发简明教程：基于 Eclipse+Maven 环境的 SSM 架构/吴志祥等编著.
—北京：电子工业出版社，2020.2
ISBN 978-7-121-36549-2

Ⅰ. ①J… Ⅱ. ①吴… Ⅲ. ①JAVA 语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 092391 号

责任编辑：张小乐　　文字编辑：底　波
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1 092　1/16　印张：14.25　字数：364 千字
版　　次：2020 年 2 月第 1 版
印　　次：2020 年 6 月第 2 次印刷
定　　价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254462，zhxl@phei.com.cn。

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

Java 是当今流行的面向对象程序设计语言，因此使用 Java 语言的 Web 开发框架不断涌现。目前，市场上关于 Java EE 开发的书籍比较多，但真正从零基础开始、内容简明、系统且紧跟公司开发步伐的教材并不多见。为此，我们编写了这本符合高校应用型人才培养需要的 Java EE 教材。

本教材前三章分别介绍了 Java EE 概述及开发环境搭建、使用 JSP 开发 Web 项目和使用 Servlet 开发 Web 项目。接着，介绍了 ORM 框架 MyBatis、Spring MVC 框架和 Spring 框架的基本使用及其三大框架的整合（SSM 架构）。最后，介绍了当今流行的 Spring Boot 项目开发。本教材强调项目式教学，所包含的主要案例项目如下：

1．会员管理系统 MemMana1：单纯地使用 JSP 技术开发。

2．会员管理系统 MemMana2：使用 JSP 和 JavaBean 技术开发。

3．会员管理系统 MemMana3：使用 Servlet 实现的 MVC 模式开发。

4．案例 TestServletFileDownloadAndUpload：使用 Servlet 实现文件的上传与下载。

5．案例 TestMybatis1、TestMybatis2 和 TestMybatis3：MyBatis 框架的使用。

6．案例 TestSpringMVC1 和 TestSpringMVC2：Spring MVC 框架的使用。

7．案例 SpringMVCFileUpload：Spring MVC 文件的上传。

8．会员管理系统 MemMana4_5：使用 MyBatis 和 Spring MVC 实现的 MVC 框架开发（未整合），后台功能使用了插件 PageHelper 和 jQuery Ajax。

9．案例 TestSpringDI：测试 Spring 依赖注入功能。

10．案例 TestSpringAOP1 和 TestSpringAOP2：测试 Spring 面向切面编程功能。

11．案例 SpringIntegratedMybatis：测试 Spring 对 MyBatis 的整合。

12．会员管理系统 MemMana5：三大框架整合开发。

13．案例 springbootdemo1_web：测试 Spring Boot Web 功能。

14．案例 springbootdemo2_mysql：测试 Spring Boot 的数据库访问功能。

15．案例 springbootdemo3_thymeleaf：以 Thymeleaf 作为视图模板引擎。

16．会员管理系统 memmana6：在 IDEA 环境下，使用 Spring Boot 开发。

本教材力求做到结构合理、逻辑性和实用性强，除了设计单元学习的小案例，还设计了若干使用不同技术实现的会员管理项目，以凸显不同技术的差异。此外，教材还通过图解的方式，清晰地反映软件包里类（或接口）的成员属性（方法）。

课后练习与实验是教学的一个重要环节。本教材每章后均配有习题及实验。此外，通过第 8 章综合案例的设计与分析，可使学生综合掌握 Java 的各个知识点。

本教材有配套的上机实验网站，包括实验目的、实验内容、在线测试（含答案和评分）和素材的提供等，以及教学大纲、实验大纲、各种软件的下载链接、课件和案例源代码下载、在线测试等内容，极大地方便了教与学。

本教材由吴志祥、钱程、王晓锋和鲁屹华共同编著，其分工如下：吴志祥负责整体

构思并制作了精美的 PPT 课件，主要编写第 1 章和第 8 章，钱程编写第 2 章和第 3 章，王晓锋编写第 4 章和第 5 章，鲁屹华编写第 6 章和第 7 章。

本教材既可以作为高等院校计算机专业和相关专业学生学习“Java EE 架构”“Java 企业级应用技术”等课程的教材，也可以作为 Web 开发者的参考书。

获取本教材配套的课件、案例源代码等教学资料，可访问本课程网站 http://www.wustwzx.com/javaee/index.html。

由于编者水平有限，错漏之处在所难免，在此真诚欢迎读者多提宝贵意见，通过访问网站http://www.wustwzx.com 可与作者 QQ联系，以便再版时更正。

编著者

2019 年 9 月于武汉

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章 Java EE 概述及开发环境搭建

计算机的应用经历了从桌面型（安装在本机上运行的桌面软件，即单机版本）到多用户型（一台主机带若干终端，即多用户版本）再到 Web 型（采用 B/S 体系的网站系统）。Web 应用使人们超越了时间、地理位置的限制，可以方便地处理各种各样的信息。

作为 Web 应用开发的基础，本章主要介绍了 B/S 体系的含义、搭建 Java Web 应用开发环境和使用 JDBC 方式访问 MySQL 数据库；此外，还介绍了 Java 单元测试和动态调试方法。学习要点如下：

- 理解 Web 应用与传统桌面应用方式的不同；
- 掌握静态网页的基础知识及编辑器 HBuilder 的使用；
- 掌握 Java Web 服务器的运行环境；
- 掌握在 Eclipse 里配置 Tomcat 和 Maven 的方法；
- 掌握 Eclipse 常用快捷键的使用；
- 掌握使用 JDBC 访问 MySQL 数据库的方法；
- 掌握 Java 单元测试和动态调试的使用方法。

1.1 网站与网页基础

1.1.1 Web 应用体系与 B/S 模式

在 Internet 网站中，存放着许多服务器，其中最重要的是 Web 服务器，客户端通过浏览器等软件来访问 Web 服务器里的网站。

访问网站是对网站里网页的访问。通过访问网页，人们能够查询所需要的信息，也能提交信息并将其保存在数据库服务器里。

网页分为静态网页与动态网页两种。静态网页采用 HTML 的标签语言编写，动态网页除了包含静态的 HTML 代码，还包含了只能在服务器端解析的服务器代码。动态网页是与静态网页相对应的，通常以.aspx、.jsp、.php 等作为扩展名，而静态网页通常以.html 作为扩展名。

注意：

（1）动态网页与网页上的各种动画、滚动字幕等视觉上的动态效果没有直接关系，动态网页是采用动态网站技术生成的网页，它可以是纯代码的；

（2）动态网页需要使用某种运行于服务器端的脚本语言编写。脚本分为客户端脚本与服务器端脚本两大类。

1.1.2 常用 HTML 标记及其使用

HTML（HyperText Markup Language，超文本标记语言）用于描述 Web 页面的显示格式。在 HTML 中，所有的标记符都是用一对尖括号括起来的，绝大部分标记符是成对出现的，包括开始标记符和结束标记符。开始标记符和相应的结束标记符定义了该标记符作用的范围。结束标记符与开始标记符的区别是结束标记符在“<”之后有一个斜杠。例如，定义一个向上滚动的新闻 HTML 代码为：

```
<marquee width="300" height="280" direction="Up">滚动新闻文本</marquee>
```

除了<marquee>标记，常用的 HTML 标记如下。

- 超链接标记<a>：用于设计超链接。
- 区隔标记：用于修饰特定的文本。
- 区块标记 ：具有 float、padding 和 margin 等 CSS 样式属性，这些 CSS 样式属性是不具备的。
- 图像标记<img>：用于引入图像。
- 段落标记<p>：可以对一个段落应用 CSS 样式。
- 换行标记
：起换行作用，单标记名后的斜杠表示自闭。
- 列表标记<ul>或<ol>：需要配合<li>标记使用。
- 表格标记<table>、<tr>、<td>和<th>：常用于数据显示。
- 页内框架标记<iFrame>：定义页内框架。
- 表单标记<form>：需要内嵌若干<input>标记。

注意：在客户端中，页面呈现的过程就是浏览器程序解释 HTML 标记的过程。

表单常用来制作客户端的信息录入界面或登录界面。当用户单击“提交”按钮后，浏览器地址栏将出现一个新的 HTTP 请求，跳转至表单处理页面，接收用户提交的信息并进行相应的处理，表单定义的示例代码如下：

```
<form name="表单名称" method="post" action="表单处理程序" >
        …  <!--定义接收用户数据输入的表单元素-->
        <input type="submit" value="提交">
 </form>
```

如果不指定表单的 action 属性值，则默认由本页面自处理。表单自处理的 JSP 页面，参见第 2.4.2 节项目 MemMana1 里的会员登录页面 mLogin.jsp 等。

对于文件上传表单使用标记<input type="file" name="wjy"/>时，必须对表单使用属性 enctype="multipart/form-data"。页面浏览时，单击“浏览”按钮后出现“选择文件”对话框。文件上传表单，可参见项目 MemMana4 的后台管理功能。

在表单内，还可以用命令来响应客户端的单击事件，其定义方法如下：

```
<input type="button" value="提交" onClick="客户端脚本方法()" >
```

注意：

（1）表单的 method 属性值一般指定为“post”，为默认值；

（2）提交按钮/重置按钮，只能作为表单里的最后元素；

（3）提交按钮通常是表单必需的，而重置按钮则不然；

（4）定义表单元素时，一般要使用 name 属性，因为客户端脚本和服务器脚本是按元素名称来获取表单元素的提交值；

（5）在网站开发实务中，对表单提交的数据进行有效性验证的方式有两种：一种是定义表单的 onSubmit 事件来实现客户端脚本进行验证；另一种是在服务器程序中验证。显然，客户端验证可以减轻 Web 服务器的压力，值得推荐。

页内框架是指页面里的一块区域，使用 Div 布局页面时，可以将某个 Div 定义为页内框架，其方法是使用成对的 HTML 标记<iFrame>及</iFrame>，其定义格式如下：

```
<div><iFrame src="预载页面" name="框架名"  width=""  height="" ></iFrame></div>
```

其中，src、name、width 和 height 是 iFrame 标记的四个常用属性，但 src 不是必填属性。

页内框架应用于超链接中，将链接页面的内容输出到指定的页内框架中，而不是打开一个新窗口，其引用方法如下：

```
<a href="目标页面" target="页内框架名" >
```

注意：

（1）使用页内框架，避免频繁打开新窗口，可使浏览过程更加连贯，改善用户体验；

（2）使用页内框架的示例，可参见案例项目 MemManal 的主页 index.jsp。

1.1.3　流行的网页编辑器——HBuilder

HBuilder 是 DCloud（北京数字天堂公司）推出的一款支持 HTML5 的 Web 开发 IDE。HBuilder 的编写用到了 Java、C、Web 和 Ruby，其主体由 Java 编写，基于 Eclipse，可兼容 Eclipse 的插件。

访问官网 http://www.dcloud.io，可以下载 HBuilder 获取更多的功能介绍。

HBuilder 运行界面由菜单、工具栏、项目子窗口、编辑子窗口、预览窗口和控制台等组成。其中，控制台能显示 JavaScript 程序的运行状态信息，方便调试和修改源程序。

HBuilder 的常用快捷键如下。

- Ctrl+Shift+/：用于注释若干行代码或取消注释。
- Ctrl+D：用于删除光标所在行。
- Ctrl+Enter：用于在下面产生一个新的空白行。
- Ctrl+Shift+F：用于代码格式化。
- Ctrl+Shift+W：用于一次性关闭已经打开的文档。

HBuilder 在打开不同的页面时，预览窗口能自动切换。工具栏上的按钮（A^+和 A^-）能即时放大或缩小文档字体，如图 1.1.1 所示。

注意：

（1）本教材主要涉及的页面文档类型都是静态的 HTML；

（2）HBuilder 默认使用 utf-8 作为文档的编码；

（3）默认文档类型一般选择 HTML 4 或 HTML 5；

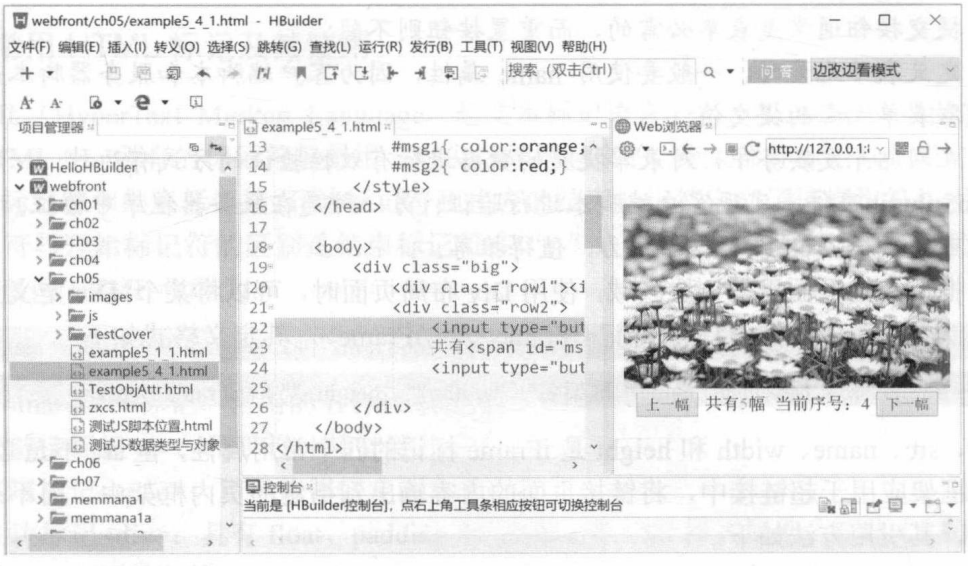

图 1.1.1　HBuilder 工作界面

（4）如果只是简单地修改文档中的某些文本，也可使用 NotePad 和 EditPlus 等较为小巧的编辑器。

1.1.4　CSS 样式与 Div 布局

1. CSS 样式技术

CSS 是 1996 年年底产生的新技术，是 Cascading Style Sheet 的缩写，译名为层叠样式表。CSS 是一组样式，它并不属于 HTML，把 CSS 样式应用到不同的 HTML 标记中，可扩展 HTML 功能，如调整字间距、行间距、取消超链接的下画线效果、多种链接效果等，这是原来的 HTML 标记+属性所无法实现的效果。

使用 CSS 技术，除了可以在单独网页中应用一致的格式，对于大网站的格式设置和维护更具有重要意义。将 CSS 样式定义到样式表文件中，然后在多个网页中同时应用就能确保多个网页具有一致的格式，并且能够随时更新（只需更新样式表文件），从而可大大降低网站的开发和维护的工作量。

由于 CSS 样式的引入，HTML 新增了<style>和两个标记，对所有产生页面实体元素的 HTML 标记都可以使用属性 style、class 或 id 来应用 CSS 样式。

常用的 CSS 选择器的特性如下。

- 类选择器：在<style>标记内定义时，样式名前缀为“.”，由用户决定哪些对 HTML 标记使用 class 属性来应用该样式。
- ID 选择器：在<style>标记内定义时，样式名前缀为“#”，对 HTML 标记使用 id 属性来应用该样式，且要求应用本 ID 样式的页面元素是唯一的。
- 标签选择器：在<style>标记内，以 HTML 标记作为样式名（无前缀），用来重新定义 HTML 标记的外观（自动应用于相应的 HTML 标记）。

- 伪类选择器：对超链接不同状态的样式进行定义，包括 a:hover（鼠标位于超链接上时）等。

注意：

（1）当不涉及 JavaScript 脚本（含 jQuery）时，ID 样式与类样式可以互换。ID 选择器的唯一性是指应用 ID 样式的页面元素应当是唯一的；

（2）对一个页面元素同时应用多种样式时，其选择器名称之间应使用空格隔开；

（3）上面的伪类选择器是复合内容选择器（组合选择器）的一种使用形式。

编辑网页时，在页面里使用成对标签<style>和</style>定义为当前页面使用的 CSS 样式，其示例代码如下：

```
<style type="text/css">
    .zw {   /*定义类选择器*/
        font-size: 12px;    /*字体大小，像素为单位*/
        color: #F00;      /*颜色为红色*/
        line-height:20   /*文字的行高*/
    }
</style>
```

多个选择器之间使用空格分隔时，表示需要根据文档的上下文关系来确定 HTML 标记应用或者避免的 CSS 样式，即通过 CSS 后代选择器来实现准确定位。后代选择器的示例代码如下：

```
.faces .face1{   /*定义后代选择器*/
    /*定义类选择器 face1 的 CSS 样式属性*/
}
.menu {
    /*定义类选择器 menu*/
}
.menu ul{
    /*定义作为后代的标签选择器 ul 的 CSS 样式属性*/
}
.menu ul li{
    /*定义辈分更低的后代标签选择器*/
}
.menu ul li a{
    /*定义辈分更低的后代标签选择器*/
}
```

每个 HTML 标记所生成的页面元素，都有其默认的外观。例如，HTML 标记<a>所产生的超链接，在默认情况下，存在下画线。在实际进行网站开发时，通过重新定义 HTML 标记<a>样式，可以取消默认的下画线，例如：

```
<style type="text/css">
    a {
        text-decoration: none;   /*取值 none 时无下画线，取值为 underline 时有下画线*/
        font-size: 18px;   /*设置链接文字的大小*/
```

```
}
</style>
```

与伪类选择器对应的样式称为伪类样式。如除了a:hover，还有 a:active（超链接被选中时）、a:visited（超链接被访问时）和 a:link（没有被访问时）都是伪类样式。

内联样式是通过 style 属性把 CSS 样式属性键值对引入到定义对象的 HTML 标记中，例如：

```
<span style="font-size: 24px; color: red;">文字</span>
```

CSS 滤镜是 CSS 样式的扩展，它能将特定效果应用于文本容器、图片或其他对象。CSS 滤镜通常作用于 HTML 控件元素，如 img、td 和 div 等。

在 CSS 样式中，通过关键字 filter 引入滤镜。如对于空间文字，应用 shadow 滤镜可以实现文字的阴影效果，其 CSS 样式的属性如下：

```
filter:shadow(color=cv,direction=dv)
```

其中，滤镜参数 color 表示阴影的颜色；cv 值可使用代表颜色的英文单词，如 red、blue、green 等，也可以使用色彩代码；参数 direction 表示阴影的方向；dv 取值为 0～360。

注意：不同的浏览器对滤镜的支持是有区别的。如 Shadow 滤镜只有 IE 浏览器支持，而其他浏览器则不支持。

外部样式是指将样式定义在一个单独的文件里，该样式文件以.css 作为扩展名。建立外部样式文件后，就可以在网站的每个页面里引用它，用于统一网站风格。

在页面里引用外部样式之前，需要使用<link>标签引入外部样式文件，其示例代码如下：

```
<link rel="stylesheet" type="text/css" href="带路径的样式文件名.css">
```

2．CSS+Div 布局

设计页面时，通常先将页面按功能划分为若干个小区域，每个小区域使用标签 来表示。一个 表示的区域，可以进一步划分，就形成了 的嵌套。

每个 表示的区域，可以通过 class、id 或 style 属性来应用 CSS 样式，达到设置 div 区域外观和位置关系等目的。

设置 div 区域外观的示例代码如下：

```
<div class="yangshi">演示</div>
```

标签 是块级元素，显示属性为 display，以 block 作为默认值，使用该值将为对象添加新行，取值 none 时将隐藏对象（不保留其物理空间）；可见属性 visibility 以 visible 作为默认值（表示可见），取值为 hidden 表示不可见，但保留着占用的物理空间。

在使用 div 布局的页面里，通常情况下，需要将页面里最外面的那个 div 设置成水平居中，CSS 样式属性的应用如下：

```
margin:0 auto;    /*margin-right 与 margin-left 属性值为 auto*/
```

当 div 嵌套时，同一级别的多个 div，其默认位置关系是上下关系。要想改变成左右关系，

只需要对同一级别的多个 div 设置 CSS 样式属性即可，其示例代码如下：

```
float:left;    /*并排多个 div*/
```

div 常用的 CSS 样式属性如表 1.1.1 所示。

表 1.1.1　div 常用的 CSS 样式属性

CSS 属性名	功 能 描 述
position	定位属性，常用取值为 absolute、relative，默认值为 static
left 和 top	定义左上角点，适用于 absolute 和 relative 两种定位方式，相对父 div
right 和 bottom	定义右下角点，适用于 absolute 和 relative 两种定位方式，相对父 div
width 和 height	定义 div 的宽度和高度，以像素为单位
text-align	定义 div 内容的对齐方式
border	定义 div 的边框，以像素为单位
background	定义 div 背景图片
float	浮动，取值 left 或 right，常用于实现 div 的并排方式
margin	外填充，用于设置 div 之间的间距，可按“上右”“下左”的顺序分别设置
padding	内填充，用于设置 div 与其内部元素的间距，也可分别设置
overflow	取值为 hidden 时，隐藏超出 div 尺寸的内容，且不破坏整体布局
z-index	定义层叠加的顺序，取值整数，值越大，就越靠上

div 除了通过 style 属性应用内联 CSS 样式，还可以通过 class 属性应用类样式或 id 属性应用 ID 样式。

为了分析页面里各元素应用的 CSS 样式与页面布局，建议读者使用 Google 浏览器。右键单击页面元素，在弹出的快捷菜单中选择“审查元素”命令或按功能键 F12，就会出现图 1.1.2 所示的效果。

图 1.1.2　使用 Google 浏览器分析页面元素应用的 CSS 样式

此为试读，需要完整PDF请访问：www.ertongbook.com