

(AddressRegistration.xhtml 中第 64 行)。该方法检查姓和名是否不为空 (AddressRegistrationJSFBean.java 中第 106 ~ 111 行)。如果不为空, 则返回字符串 "ConfirmAddress", 这将导致 ConfirmAddress 的 JSF 页面被显示。

ConfirmAddress 这个 JSF 页面显示用户输入的数据 (ConfirmAddress.xhtml 第 12 行)。getInput() 方法 (AddressRegistrationJSFBean.java 中第 120 ~ 133 行) 得到输入。

ConfirmAddress 这个 JSF 页面中 Confirm 按钮的动作是 storeStudent() (ConfirmAddress.xhtml 中第 15 行)。该方法将地址存储到数据库中 (AddressRegistrationJSFBean.java 中第 157 ~ 177 行), 并返回字符串 "AddressStoredStatus", 这将导致显示 AddressStoredStatus 页面。状态信息显示在该页面中 (AddressStoredStatus.xhtml 中第 12 行)。

ConfirmAddress 页面中 Go Back 按钮的动作是 "AddressRegistration" (ConfirmAddress.xhtml 中第 17 行)。这将导致显示 AddressRegistration 页面, 让用户可以重新输入。

受管 bean 的范围是会话 (AddressRegistrationJSFBean.java 中第 9 行), 因此多个页面可以共享同一个 bean。

注意, 该程序显式装载数据库驱动 (AddressRegistrationJSFBean.java 中第 139 行)。有时候, 诸如 NetBeans 这样的 IDE 无法找到一个合适的驱动。显式装载一个驱动可以避免这个问题。

关键技术语

application scope (应用范围)

JavaBean (JavaBean 组件)

request scope (请求范围)

scope (范围)

session scope (会话范围)

view scope (视图范围)

本章小结

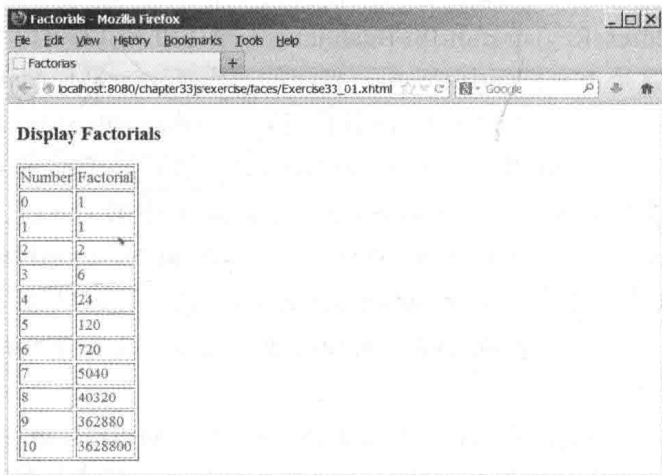
1. JSF 使得 Java 代码和 HTML 完全分离。
2. facelet 是混合使用了 JSF 标签和 XHTML 标签的 XHTML 页面。
3. JSF 应用使用模型 - 视图 - 控制器 (MVC) 架构实现, 这样将应用程序数据 (包含在模型中) 和图形表示 (视图) 进行了分离。
4. 控制器是负责协调视图和模型之间交互的 JSF 框架。
5. JSF 中, facelet 是表现数据的视图。数据从 Java 对象处得到。使用 Java 类定义对象。
6. JSF 中, 从 facelet 访问的对象是 JavaBean 对象。
7. JSF 表达式可以通过属性名字, 或者调用方法来获得当前时间。
8. JSF 提供许多元素用来显示 GUI 组件。以 h 为前缀的标签在 JSF HTML 标签库中。以 f 为前缀的标签在 JSF 核心标签库中。
9. 可以指定 JavaBean 对象的范围在应用范围、会话范围、视图范围或者请求范围内。
10. 只要停留在一个视图上, 视图范围将保持 bean 有效。视图范围介于会话范围和请求范围之间。
11. JSF 提供了一些方便且强大的方式来进行输入验证。可以使用 JSF 核心标签库中的标准验证器, 也可以创建自定义验证器。

测试题

回答位于网址 www.cs.armstrong.edu/liang/intro10e/quiz.html 的本章测试题。

编程练习题

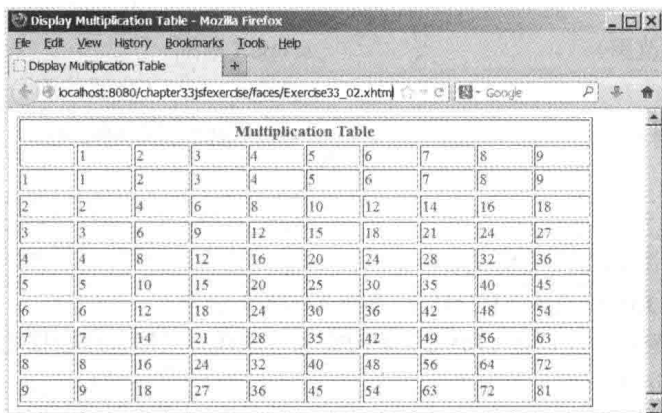
- *33.1 (JSF 中的阶乘表) 编写一个 JSF 页面, 如图 33-25 显示一个阶乘页面。在一个 `h:outputText` 组件中显示表格。将其 `escape` 属性设置为 `false`, 从而将其显示为 HTML 内容。



Number	Factorial
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880
10	3628800

图 33-25 JSF 页面为 0 到 10 之间的数字在表格中显示阶乘

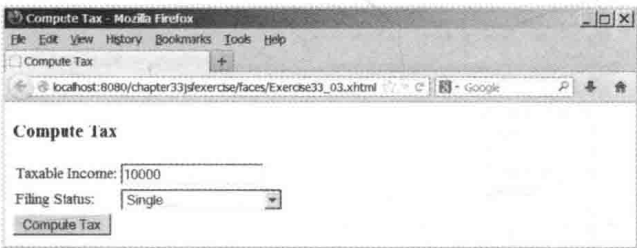
- *33.2 (乘法表) 编写一个 JSF 页面, 如图 33-26 显示一个乘法表。



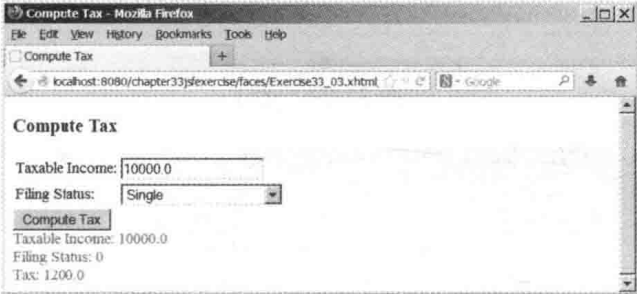
	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

图 33-26 JSF 页面显示乘法表

- *33.3 (计算税) 编写一个 JSF 页面, 让用户输入需要缴税的收入以及婚姻状况, 如图 33-27a 所示。点击 `Compute Tax` 按钮计算并显示缴税, 如图 33-27b 所示。使用程序清单 3-5 中引入的 `computeTax` 方法来计算税。
- *33.4 (计算贷款) 编写一个 JSF 页面, 让用户输入贷款额度、利率以及年数, 如图 33-28a 所示。点击 `Compute Loan Payment` 按钮计算并显示每个月以及整个的贷款支付, 如图 33-28b 所示。使用程序清单 10-2 中给出的 `Loan` 类来计算每个月以及整个的支付。
- *33.5 (加法测试) 编写一个 JSF 页面, 可以随机生成加法测试题, 如图 33-29a 所示。当用户回答完所有的题目, 如图 33-29b 显示结果。
- *33.6 (大的阶乘) 重写编程练习题 33.1, 使之可以处理大的阶乘。使用 10.9 节中介绍的 `BigInteger` 类。
- *33.7 (猜测生日) 程序清单 4-3 给出了一个猜测生日的程序。编写一个 JSF 程序, 显示 5 个数据集, 如图 33-30a 所示。用户勾选合适选项并且点击 `Guess Birthday` 按钮后, 程序如图 33-30b 显示生日。

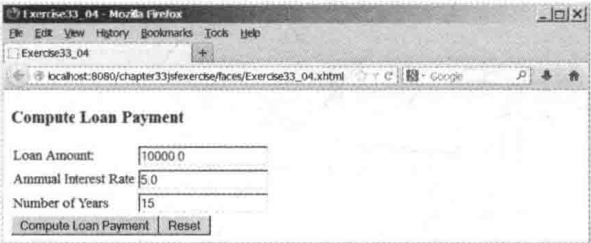


a)



b)

图 33-27 JSF 页面计算税

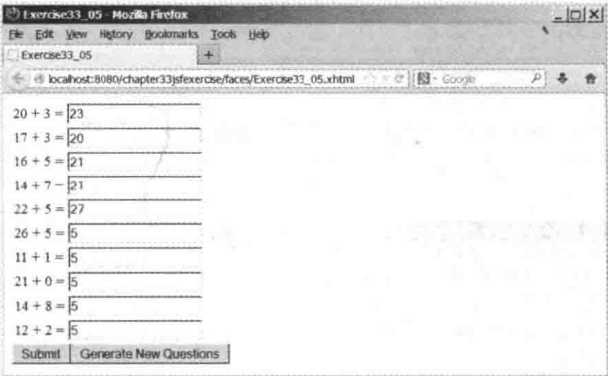


a)



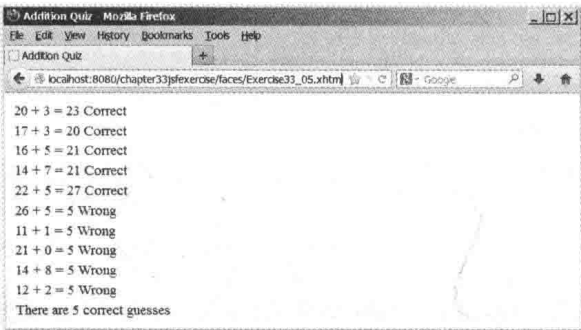
b)

图 33-28 JSF 页面计算借贷支付



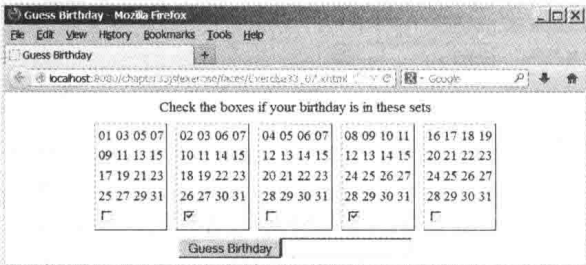
a)

图 33-29 程序在 a 中显示加法问题，在 b 中显示答案

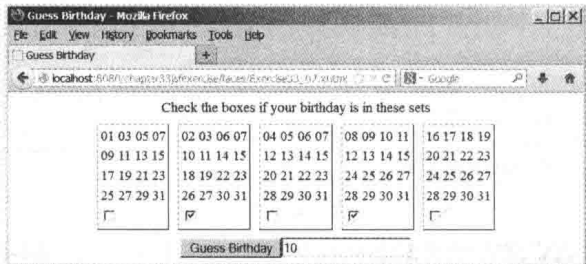


b)

图 33-29 (续)



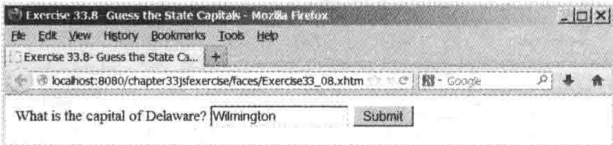
a)



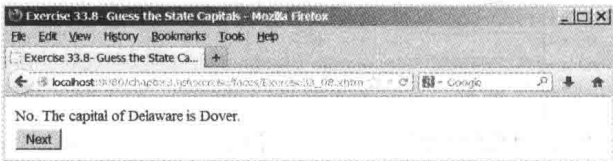
b)

图 33-30 a) 程序显示 5 组数字，让用户勾选；b) 程序显示日期

*33.8 (猜测首府) 编写一个 JSF 页面，提示用户输入一个州的首府，如图 33-31a 所示。得到用户输入后，程序报告答案是否正确，如图 33-31b 所示。可以点击 Next 按钮显示另外一个问题。可以使用二维数组来存储州和首府，如编程练习题 8.37 所提示。从数组创建一个线性表，并引用 shuffle 方法来重新对线性表排序从而问题将以随机顺序显示。



a)



b)

图 33-31 a) 程序显示一个问题；b) 程序显示问题的答案

*33.9（访问和更新 staff 表）编写一个 JSF 程序，可以观看、插入以及更新保存在数据库中的员工信息，如图 33-32 所示。View 按钮显示一个指定 ID 的记录。Staff 表如下创建：

```
create table Staff (  
    id char(9) not null,  
    lastName varchar(15),  
    firstName varchar(15),  
    mi char(1),  
    address varchar(20),  
    city varchar(20),  
    state char(2),  
    telephone char(10),  
    email varchar(40),  
    primary key (id)  
);
```

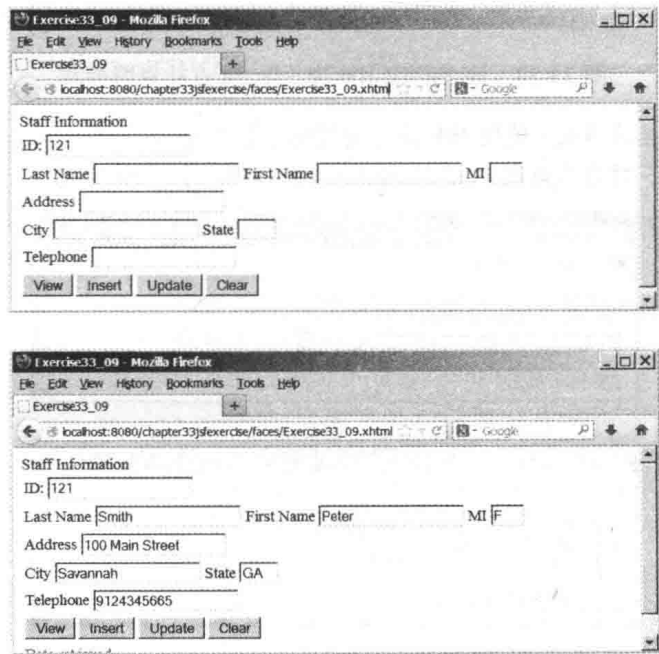


图 33-32 Web 页面让你观看、插入以及更新员工信息

*33.10（随机扑克牌）编写一个 JSF 程序，显示一叠 52 张扑克牌中的 4 张随机扑克牌如图 33-33 所示。当用户点击 Refresh 按钮时候，4 张新的随机扑克牌被显示。

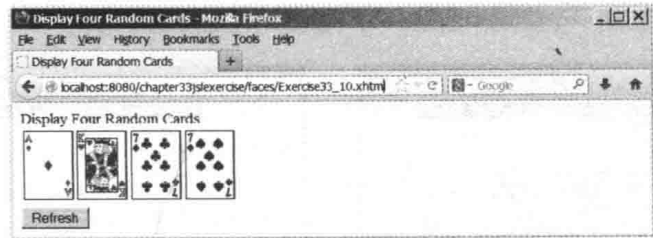


图 33-33 该 JSF 应用显示 4 张随机扑克牌

***33.11（游戏：24 点扑克牌游戏）使用 JSF 重写编程练习题 20.13，如图 33-34 所示。当点击 Refresh 按钮时，程序显示 4 张随机扑克牌，并且显示一个 24 点的解答，如果该解答存在。否则，显示 No solution。

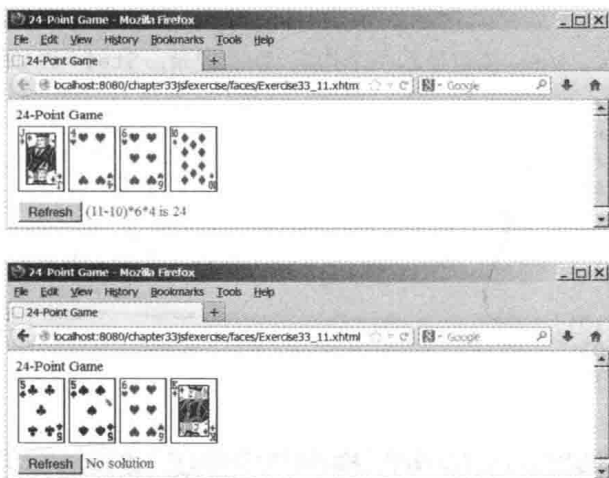


图 33-34 该 JSF 应用解决一个 24 点扑克牌游戏

***33.12 (游戏: 24 点扑克牌游戏) 使用 JSF 重写编程练习题 20.17, 如图 33-35 所示。程序让用户输入 4 张扑克牌的值, 并且当点击 Find a Solution 按钮时找到一个解决方案。

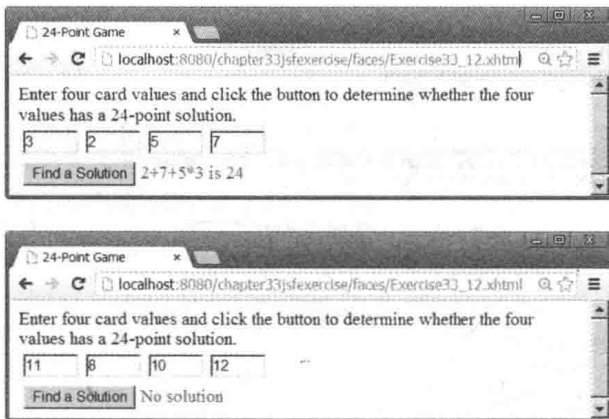


图 33-35 用户输入 4 个数字, 程序找到一个解答

*33.13 (一个星期中的周几) 编写一个程序, 对于一个给定的年、月、日, 显示是一个星期中的周几, 如图 33-36 所示。程序让用户选择一个日期、月份以及年份, 然后点击 Get Day of Week 按钮来显示一个星期中的周几。如果这是将来的某一天则 Time 字段显示 Future, 否则显示 Past。使用蔡勒公式来找到一个星期中的某一天 (参见编程练习题 3.21)。

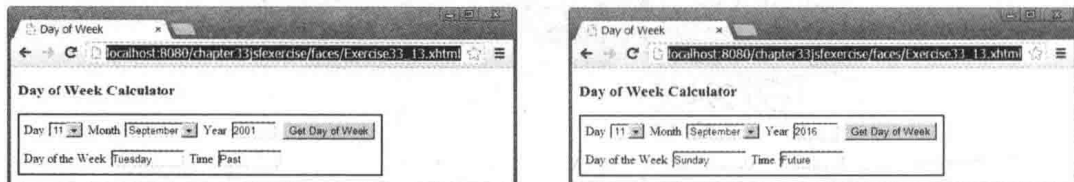


图 33-36 用户输入一个年、月、日, 程序找到属于一个星期中的周几