**《Java EE Web组件开发》实验三**

**Thymeleaf+MyBatis**

**[实验任务**] 利用**Thymeleaf+MyBatis完成用户信息的增删改查**

**[任务介绍**]

1. **任务描述**

首先设置好数据库和数据表，利用Thymeleaf+MyBatis实现用户信息的增加、删除、修改和显示功能。

**运行结果**

**[任务目标]**

* 掌握Thymeleaf和MyBatis框架；
* 熟练应用MyBatis完成数据访问，会使用XML编写访问数据库代码。

**[实现思路]**

使用Thymeleaf框架实现前端展示，使用MyBatis框架编写XML编写访问数据库代码

**[实现代码以及运行效果]**

@Service  
public class UserServiceImpl implements UserService{  
 @Autowired  
 private UserRepository userRepository;  
  
 @Override  
 public String selectUser(Model model, int currentPage) {  
 //共多少个用户  
 int totalCount = userRepository.selectAllUser();  
 //计算共多少页  
 int pageSize = 5;  
 int totalPage = (int)Math.*ceil*(totalCount\*1.0/pageSize);  
 List<BUser> typeByPage = userRepository.selectUserByPage((currentPage-1)\*pageSize, pageSize);  
 model.addAttribute("allUsers", typeByPage);  
 model.addAttribute("totalPage", totalPage);  
 model.addAttribute("currentPage", currentPage);  
 return "admin/allUser";  
 }  
  
 @Override  
 public String deleteUser(Model model, int id) {  
 if(userRepository.selectCartUser(id).size() > 0   
 ||userRepository.selectOrderUser(id).size() > 0) {  
 return "no";  
 }else {  
 userRepository.deleteUser(id);  
 return "/selectUser?currentPage=1";  
 }  
 }  
  
}

@Controller  
public class UserController extends AdminBaseController{  
 @Autowired  
 private UserService userService;  
 @RequestMapping("/selectUser")  
 public String selectUser(Model model, int currentPage) {  
 return userService.selectUser(model, currentPage);  
 }  
 @RequestMapping("/deleteUser")  
 @ResponseBody  
 public String deleteUser(Model model, int id) {  
 return userService.deleteUser(model, id);  
 }  
}

****

图 1运行结果

**[总结或感悟]（**对运行结果所作的分析以及本次调试程序所取得的经验。如果程序未能通过，分析其原因。**）**

本次实验基于老师给的程序完成，花了一段时间看懂后各个函数之间的关联及其作用后并没有太大问题。基本了解了MyBatis框架使用XML代码访问数据库的实现方法