|  |  |
| --- | --- |
| **实验**  **目的** | 1. 掌握session特性和使用； |
| **实验条件** | JDK8、IntelliJ IDEA、Maven |
| **实验**  **内容** | 1. 开发一个综合项目Web应用：购物车功能；  2. 实现统计购买的书的数量和类别。 |
| **实验**  **过程** | 实现购物车功能的数据统计和书本点击时进行累计数量的代码  HttpSession session = req.getSession();  String id = req.getParameter("id");  //  System.out.println(id);  //session中暂存了书籍信息  HashMap<String, Integer> bookMap = (HashMap<String, Integer>) session.getAttribute("books");  // bookMap 只能初始化一次  if (bookMap == null) {// 初始化  bookMap = new HashMap<String, Integer>();  bookMap.put("1001", 0);  bookMap.put("1002", 0);  bookMap.put("1003", 0);  }  if (id != null) {  int cnt = bookMap.get(id);  bookMap.put(id, cnt + 1);  }  session.setAttribute("books", bookMap);  int bookCnt = 0;  // 图书总数  if (session.getAttribute("books") != null) {  Set<String> keys = bookMap.keySet();  for (String key : keys) {  bookCnt += bookMap.get(key);  }  String html =  "<head>\n" +  "<meta http-equiv=\"content-type\" content=\"text/html;charset=UTF-8\">\n" +  "<meta charset=\"UTF-8\">\n" +  "<title>购物车</title>\n" +  "</head>\n" +  "<body>\n" +  "<h1>图书列表</h1>\n" +  "<hr>\n" +  "<img src=\"imgs/cart.jpg\" height=\"100\"/><a href=\"" + "cart.view" + "\">已采购图书" + bookCnt + "本</a>\n" +  "<p>\n" +  " <img src=\"imgs/book1.jpg\" height=\"200\"/> <a href=\"?id=1001\"> 采购此书</a>\n" +  " <img src=\"imgs/book2.jpg\" height=\"200\"/><a href=\"?id=1002\"> 采购此书</a>\n" +  " <img src=\"imgs/book3.jpg\" height=\"200\" /><a href=\"?id=1003\"> 采购此书</a>\n" +  "\n" +  "</body>";  执行效果    实现对最终结果的统计关键代码  // 从请求中获取书籍ID参数  String id = req.getParameter("id"); // 获取id参数  System.out.println(id); // 打印获取到的id值，用于调试  // 获取HTTP会话对象  HttpSession session = req.getSession(); // 获取当前请求的会话，如果不存在则创建一个新的会话  // 从会话中获取存储的书籍信息HashMap  HashMap<String, Integer> bookMap = (HashMap<String, Integer>) session.getAttribute("books"); // 获取名为"books"的属性  // 初始化书籍数量变量  int b1 = 0, b2 = 0, b3 = 0; // 初始化三本书的数量    // 更新书籍数量  if (id != null && bookMap != null) { // 如果id不为空且bookMap不为空  int cnt = bookMap.getOrDefault(id, 0); // 获取当前书籍的数量，如果没有则为0  bookMap.put(id, cnt + 1); // 更新书籍数量  }  // 设置书籍信息回会话，这一步是多余的，因为上面的代码已经设置过了  session.setAttribute("books", bookMap);  // 计算总书籍数量  int bookCnt = 0; // 初始化总书籍数量  if (session.getAttribute("books") != null) { // 如果会话中存在书籍信息  Set<String> keys = bookMap.keySet(); // 获取所有书籍ID  for (String key : keys) { // 遍历所有书籍ID  switch (key) { // 根据书籍ID分类统计数量  case "1001":  b1 += bookMap.get(key);  break;  case "1002":  b2 += bookMap.get(key);  break;  case "1003":  b3 += bookMap.get(key);  break;  }  bookCnt += bookMap.get(key); // 累加总书籍数量  }  }  // 输出HTML页面内容  out.println("<head>\n" +  "<meta http-equiv=\"content-type\" content=\"text/html;charset=UTF-8\">\n" +  "<meta charset=\"UTF-8\">\n" +  "<title>购物车</title>\n" +  "</head>\n" +  "<body>\n" +  "<h1>图书列表</h1>\n" +  "<hr>\n" +  "<img src=\"imgs/cart.jpg\" height=\"100\"/><a href=\"cart.view\">已采购图书" + bookCnt + "本</a>\n" +  "<p>\n" +  " <img src=\"imgs/book1.jpg\" height=\"200\"/> <a href=\"shopping?id=1001\"> 采购此书</a>\n" +  " <img src=\"imgs/book2.jpg\" height=\"200\"/><a href=\"shopping?id=1002\"> 采购此书</a>\n" +  " <img src=\"imgs/book3.jpg\" height=\"200\" /><a href=\"shopping?id=1003\"> 采购此书</a>\n" +  "\n" +  "\n" +  "</body>");  执行结果 |
| **实验**  **总结** | 我们首先设置了购物车的基本框架，包括书籍的展示和链接到Servlet的采购操作。  通过使用HttpSession对象，我们在会话中存储了一个HashMap，用于记录每本书的购买数量。  实现了点击书籍链接时，更新对应书籍数量的逻辑，并在页面上显示总购买数量。  ****技术应用与创新****：  我们使用了Java Servlet技术来处理HTTP请求，并动态生成HTML页面。  通过HashMap数据结构，我们有效地管理了购物车中的书籍信息，并通过session实现了跨请求的数据共享。  ****问题与解决方案****：  在实验过程中，我们遇到了session数据管理的挑战，特别是在多用户环境下保持数据的独立性。  通过仔细设计session的生命周期和存储结构，我们成功解决了这些问题。  ****实验反思****：  实验让我们意识到了Web开发中状态管理的重要性和复杂性。  我们也认识到了在实际开发中需要考虑用户体验和数据一致性的重要性。  ****未来展望****：  我们计划在未来的学习中，进一步探索更高级的Web开发技术和框架，如Spring MVC和React。  我们还希望将本次实验的成果应用到更复杂的电子商务项目中，以实现更丰富的功能。 |