

课程回顾

1 新增业务流程

MVC设计模式：jsp负责数据收集，提交Servlet调用Service层代码处理最终数据库实现

2 修改业务流程

BeanUtils简化form提交数据后的封装

课程目标

1 多条件模糊查询

2 分页

课程实施

1 多条件模糊查询

如图-1 所示。**必做, 该功能不做, 后面功能零分)**

图书借阅系统

图书分类: 全部

图书名称:

是否借阅: 请选择

查询

book_type =? and book_name like ? and is_borrow = ? 添加

图书编号	图书分类	图书名称	作者	出版社	操作	详细	删除	修改
GBK_0091	少儿	java基础	老王	人民教育出版社	已借阅	详细	删除	修改
GBK_0092	少儿	java基础1	老王	人民教育出版社	已借阅	详细	删除	修改
GBK_0093	少儿	sql基础1	老王2	人民教育出版社	已借阅	详细	删除	修改

首页 上一页 下一页 末页 1/3页 GO

1-1 domain+dao+Service

```
1 // sql需求
2 SELECT book_info.*,book_type.type_name,book_type.id
3 FROM book_info LEFT JOIN book_type
4 ON book_info.book_type=book_type.id
5 -- 思考：每次用户在网页上面，都会一次性提供三个条件吗？
6 -- 举例：只给了图书名称一部分核心词语
7 WHERE 1=1
8
9 AND book_info.book_type=?
10
11 AND book_info.book_name LIKE ?
12
13 AND book_info.is_borrow=?
14
15
```

1-2 代码实现

daoのBookInfoDaoImpl多条件查询

```
1      @Override
2      public List<BookInfo> selectAll(Integer bookType, String bookName,
Integer isBorrow) {
3          //执行SQL语句
4          StringBuilder sb=new StringBuilder();
5          sb.append(" SELECT book_info.book_id id,book_info.book_code, ");
6          sb.append(" book_info.book_name,book_info.book_type, ");
7          sb.append(" book_info.book_author,book_info.publish_press, ");
8          sb.append(" book_info.publish_date,book_info.is_borrow, ");
9          sb.append(" book_type.type_name FROM book_info ");
10         sb.append(" LEFT JOIN book_type ON book_info.book_type=book_type.id
");
11         //where 1=1作用：条件的拼接关键字and设置进去
12         sb.append(" where 1=1 ");
13         //String判断两个值：null ""
14         //将所有的? 对应的实参，使用集合保存
15         List params=new ArrayList();
16         if(!Objects.isNull(bookType)){
17             sb.append(" AND book_info.book_type=? ");
18             params.add(bookType);
19         }
20         if(!Objects.isNull(bookName)&&!bookName.isEmpty()){
21             sb.append(" AND book_info.book_name LIKE ? ");
22             params.add("%"+bookName+"%");
23         }
24         if(!Objects.isNull(isBorrow)){
25             sb.append(" AND book_info.is_borrow=? ");
26             params.add(isBorrow);
27         }
28         //qr.query(String 要执行的sql语句，
29         // BeanHandler BeanListHandler ScalarHandler 转换的类型，
30         // Object... sql语句? 对应的实参)
31         try {
32             return qr.query(sb.toString(),
33                             new BeanListHandler<>(BookInfo.class),
34                             // Object... 底层就是数组，所以集合必须转换为数组，底层才能正常解
析
35                             params.toArray());
36         } catch (SQLException e) {
37             throw new RuntimeException(e);
38         }
39     }
```

serviceのBookInfoServiceImpl多条件模糊查询

```
1      @Override
2      public List<BookInfo> getAll(Integer bookType, String bookName, Integer
isBorrow) {
3          return dao.selectAll(bookType,bookName,isBorrow);
4      }
```

jspのbooklist.jsp 查询页面的设计

```
1  <script type="text/javascript">
2      function doSearch(){
3          //单击查询按钮，向BookInfoServlet发出请求
4          //发出请求时，需要带着查询条件过去
5          //form走POST  地址栏地址能变，就是发请求统一get
6          // window.open();
7          //js支持EL
8          //定义字符串，保存搜索条件
9          var query="bookType="+document.getElementById("type").value;///?应该是
      用户实际输入的数据
10         query+="&bookName="+document.getElementById("book_name").value;
11         query+="&isBorrow="+document.getElementById("borrow").value;
12
13         location.href="${pageContext.servletContext.contextPath}/BookInfoServlet?" +
      query;
14     }
15 </script>
16 <body>
17 <table border="1px" cellspacing="0px" align="center">
18     <tr>
19         <td align="center">
20             <h2 align="center">图书借阅系统</h2>
21             <!--<p align="center">-->
22             图书分类:
23             <select name="book_type" id="type">
24                 <option value="">全部</option>
25                 <!-- 从数据库查询的图书分类 --%>
26                 <c:forEach items="${types}" var="type">
27                     <!-- 要求: value指定主键 innerText:列值 汉字 --%>
28                     <option value="${type.id}">${type.type_name}</option>
29                 </c:forEach>
30             </select>
31             图书名称: <input type="text" id="book_name"/>
32             是否借阅:
33             <select name="is_borrow" id="borrow">
34                 <option value="">===请选择===</option>
35                 <option value="1">已借阅
36                 </option>
37                 <option value="0">未借阅
38                 </option>
39             </select>
40             <!--
41             submit和Image 提交表单按钮
42             reset 重置按钮
43             button 普通按钮
44             --%>
45             <input type="button" value="查询" id="search"
      onclick="doSearch()"/>
46             //略
```

ServletのBookInfoServlet实现查询

```
1 package cn.kgc.controller; /**
2  * @Author: lc
3  * @Date: 2022/5/17
4  * @Description: ${PACKAGE_NAME}
5  * @Version: 1.0
6  */
7
8 import cn.kgc.domain.BookInfo;
9 import cn.kgc.domain.BookType;
10 import cn.kgc.service.BookInfoService;
11 import cn.kgc.service.impl.BookInfoServiceImpl;
12 import cn.kgc.service.impl.BookTypeServiceImpl;
13
14 import javax.servlet.ServletException;
15 import javax.servlet.annotation.WebServlet;
16 import javax.servlet.http.HttpServlet;
17 import javax.servlet.http.HttpServletRequest;
18 import javax.servlet.http.HttpServletResponse;
19 import java.io.IOException;
20 import java.util.List;
21
22 @WebServlet("/BookInfoServlet")
23 public class BookInfoServlet extends HttpServlet {
24     @Override
25     protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
26         //请求发送的搜索条件中，图书名称可能是中文，GET必须处理中文
27         //tomcat8 9 10针对get没有乱码
28         //取 请求体或url地址传输的表单数据？ 略
29         String bookTypeStr = request.getParameter("bookType");//请求没有数据
时，bookTypeStr null
30         String bookName = request.getParameter("bookName");//bookName null
31         //解决get请求乱码问题 trim():去除字符串首尾的无效空格
32         if (bookName!=null && !bookName.trim().isEmpty()) {
33             bookName=new String(bookName.getBytes("iso8859-1"),"utf-8");
34         }
35         String isBorrowStr = request.getParameter("isBorrow");
36
37         //类型转换
38         Integer bookType=null;
39         try {
40             //valueOf(null)不能正常转换
41             bookType=Integer.valueOf(bookTypeStr);
42         } catch (Exception e) { }
43
44         Integer isBorrow=null;
45         try {
46             //valueOf(null)不能正常转换
47             isBorrow=Integer.valueOf(isBorrowStr);
48         } catch (Exception e) { }
49         //调 service层
50         BookInfoService service = new BookInfoServiceImpl();
51         //思考： 查询按钮提交请求，是不是可以使用这个Servlet来处理？？
52         List<BookInfo> books = service.getAll(bookType,bookName,isBorrow);
53         //查询图书分类信息
```

```

54     List<BookType> types = new BookTypeServiceImpl().getAllBookTypes();
55     //存 共享
56     request.setAttribute("list",books);//显示图书信息
57     request.setAttribute("types",types);//显示图书分类信息
58     //为了实现查询条件的回显，所以要将所有查询条件存入域
59     request.setAttribute("type",bookType);
60     request.setAttribute("bookName",bookName);
61     request.setAttribute("borrow",isBorrow);
62     //转
63     //jsp本质就是一个servlet
64
65     request.getRequestDispatcher("/booklist.jsp").forward(request,response);
66
67     @Override
68     protected void doPost(HttpServletRequest request, HttpServletResponse
69     response) throws ServletException, IOException {
70         doGet(request, response);
71     }

```

JSP的booklist.jsp查询条件的回显

```

1  <%@ page contentType="text/html;charset=UTF-8" language="java" %>
2  <!--导入jstl依赖
3  prefix设置值 自定义，一般建议以导入库单词首字母
4  uri: 使用标签官方提供的一个访问路径
5  --%>
6  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
7  <html>
8  <head>
9      <title>Title</title>
10 </head>
11 <script type="text/javascript">
12     function doSearch(){
13         //单击查询按钮，向BookInfoServlet发出请求
14         //发出请求时，需要带着查询条件过去
15         //form走POST 地址栏地址能变，就是发请求统一get
16         // window.open();
17         //js支持EL
18         //定义字符串，保存搜索条件
19         var query="bookType="+document.getElementById("type").value;//?应该是
    用户实际输入的数据
20         query+="&bookName="+document.getElementById("book_name").value;
21         query+="&isBorrow="+document.getElementById("borrow").value;
22
23         location.href="${pageContext.servletContext.contextPath}/BookInfoServlet?"
+
    query;
24         // location.assign();
25     }
26 </script>
27 <body>
28 <table border="1px" cellspacing="0px" align="center">
29     <tr>
30         <td align="center">
31             <h2 align="center">图书借阅系统</h2>
32             <!--<p align="center">-->

```

```

32      图书分类:
33      <select name="book_type" id="type">
34          <option value="">全部</option>
35          <!--      从数据库查询的图书分类  -->
36          <c:forEach items="${types}" var="t">
37              <!--      要求: value指定主键  innerText:列值  汉字  -->
38              <option value="${t.id}"
39                  <c:if test="${type==t.id}">
40                      selected
41                  </c:if>
42              >${t.type_name}</option>
43          </c:forEach>
44      </select>
45      图书名称: <input type="text" id="book_name" value="${bookName}"/>
46      是否借阅:
47      <select name="is_borrow" id="borrow">
48          <option value="">===请选择===</option>
49          <option value="1"
50              <c:if test="${borrow==1}">
51                  selected
52              </c:if>
53          >已借阅
54          </option>
55          <option value="0"
56              <c:if test="${borrow==0}">
57                  selected
58              </c:if>
59          >未借阅
60          </option>
61      </select>
62      <!--
63      submit和Image 提交表单按钮
64      reset 重置按钮
65      button 普通按钮
66      --%>
67      <input type="button" value="查询" id="search"
68          onclick="doSearch()"/>
69  </td>
70 </tr>
71 </table>
72 </body>
</html>

```

2 分页+多条件模糊查询的实现

2-1 分页的概述

分页优点: 分页减少数据显示的条数, 进而提升页面响应速度。

页面保留有限数据显示, 页面会更加易于操作、浏览

SSM使用第三方分页插件。分页变得非常简单。

但是U2, 使用原生态代码, 自己写, 但是使用分页类, 可以Page直接使用。

侧重: 理解分页的实现流程

Page类代码直接使用

```
1 package cn.kgc.domain;
2
3 import java.util.List;
4
5 /**
6  * @Author: lc
7  * @Date: 2022/5/21
8  * @Description: 保存jsp需要显示分页信息
9  * @Version: 1.0
10 */
11 public class Page<T> { //new Page<BookInfo>
12     /**
13      * 分页后需要显示的查询结果集合信息
14      * 即jsp表格需要显式图书信息
15      */
16     private List<T> list;
17     //总页数 字段 数据库列名称为字段
18     //private Integer totalPages;
19     //查询结果总行数
20     private Integer totalResults;
21     //页大小
22     private Integer pageSize;
23     //页码,当前显示的页码
24     private Integer pageNum;
25     //是否有下一页
26     //private boolean hasNextPage;
27     //是否有上一页
28     //private boolean hasPrevPage;
29
30     //属性: getter 只读属性
31     public List<T> getList() {
32         return list;
33     }
34     //属性: setter 只写属性
35     public void setList(List<T> list) {
36         this.list = list;
37     }
38     //e1读取数据${totalPages}
39     public Integer getTotalPages() {
40         //举例 13/3==>总页码: 4+1 15/3==>总页码5页
41         return getTotalResults()/getPageSize()!=0?
42         getTotalResults()/getPageSize()+1:getTotalResults()/getPageSize();
43     }
44
45     /*public void setTotalPages(Integer totalPages) {
46         this.totalPages = totalPages;
47     }*/
48
49     public Integer getTotalResults() {
50         return totalResults;
51     }
52
53     public void setTotalResults(Integer totalResults) {
54         this.totalResults = totalResults;
55     }
56 }
```

```

55
56     public Integer getPageSize() {
57         return pageSize;
58     }
59
60     public void setPageSize(Integer pageSize) {
61         this.pageSize = pageSize;
62     }
63
64     public Integer getPageNum() {
65         return pageNum;
66     }
67
68     public void setPageNum(Integer pageNum) {
69         this.pageNum = pageNum;
70     }
71
72     public boolean hasNextPage() {
73         return getPageNum() != getTotalPages();
74     }
75
76     //public void setHasNextPage(boolean hasNextPage) {
77     //    this.hasNextPage = hasNextPage;
78     //}
79
80     public boolean hasPrevPage() {
81         return getPageNum() != 1;
82     }
83
84     //public void setHasPrevPage(boolean hasPrevPage) {
85     //    this.hasPrevPage = hasPrevPage;
86     //}
87 }
88

```

2-2 分页的业务分析

实现分页的主要步骤就是在：查询数据的sql后面使用limit限制select语句返回行数。

```

1  USE exam;
2
3  SELECT book_info.*,book_type.type_name,book_type.id
4  FROM book_info LEFT JOIN book_type
5  ON book_info.book_type=book_type.id
6  -- 思考：每次用户在网页上面，都会一次性提供三个条件吗？
7  -- 举例：只给了图书名称一部分核心词语
8  WHERE 1=1
9
10 AND book_info.book_type=?
11
12 AND book_info.book_name LIKE ?
13
14 AND book_info.is_borrow=?
15 -- 限制查询结果在jsp页面显示行数
16 -- 翻页：默认从第一行开始显示，每页显示总行数应该是固定 3条数据
17 -- 如果请求第一页数据，开始行数从1行，对应下标0
18 -- 如果请求第二页数据，开始行数从4行，对应下标3

```



```

19 -- 如果请求第N页数据，开始行数从N行，对应下标(N-1)*3
20 -- ?:(页码-1)*3  ? :3
21 LIMIT ?,?

```

2-3 分页的代码实现

图书借阅系统

图书分类: 图书名称: 是否借阅:

分页在jsp需要回显的数据:
查询结果的图书信息

[新增图书](#)

当前页码 总页码

编号	图书编号	图书名称	图书分类	图书作者	出版社	出版日期	借阅状态	查看详情	操作
1	GBK_001	唉唉唉格林童话123	格林123	文学	人民教育出版社哦哦	2021-08-04	未借阅	查看详情	删除 修改

[首页](#)
[上一页](#)
[下一页](#)
[尾页](#)

当前页码/总页码

dao的BookInfoDaoImpl实现分页sql

```

1  @Override
2  public List<BookInfo> selectAll(Integer bookType, String bookName, Integer
   isBorrow,
3                                     Integer pageNum,Integer pageSize) {
4      //执行SQL语句
5      StringBuilder sb=new StringBuilder();
6      sb.append(" SELECT book_info.book_id id,book_info.book_code, ");
7      sb.append(" book_info.book_name,book_info.book_type, ");
8      sb.append(" book_info.book_author,book_info.publish_press, ");
9      sb.append(" book_info.publish_date,book_info.is_borrow, ");
10     sb.append(" book_type.type_name FROM book_info ");
11     sb.append(" LEFT JOIN book_type ON book_info.book_type=book_type.id
12     ");
13     //where 1=1作用：条件的拼接关键字and设置进去
14     sb.append(" where 1=1 ");
15     //String判断两个值： null ""
16     //将所有的? 对应的实参，使用集合保存
17     List params=new ArrayList();
18     if(!Objects.isNull(bookType)){
19         sb.append(" AND book_info.book_type=? ");
20         params.add(bookType);
21     }
22     if(!Objects.isNull(bookName)&&!bookName.isEmpty()){
23         sb.append(" AND book_info.book_name LIKE ? ");
24         params.add("%"+bookName+"%");
25     }
26     if(!Objects.isNull(isBorrow)){
27         sb.append(" AND book_info.is_borrow=? ");
28         params.add(isBorrow);
29     }
30     //添加分页语句
31     sb.append(" LIMIT ?,? ");
32     //limit语句第一个? 对应的实参
33     params.add((pageNum-1)*pageSize);
34     //limit语句第二个? 对应的实参

```

```

34     params.add(pageSize);
35     //qr.query(String 要执行的sql语句,
36     // BeanHandler BeanListHandler ScalarHandler 转换的类型,
37     // Object... sql语句? 对应的实参)
38     try {
39         return qr.query(sb.toString(),
40             new BeanListHandler<>(BookInfo.class),
41             // Object... 底层就是数组, 所以集合必须转换为数组, 底层才能正常解
析
42             params.toArray());
43     } catch (SQLException e) {
44         throw new RuntimeException(e);
45     }
46 }
47 //查询分页数据的总行数
48 @Override
49 public Object selectCount(Integer bookType, String bookName, Integer
isBorrow) {
50     //执行SQL语句
51     StringBuilder sb=new StringBuilder();
52     sb.append(" SELECT count(*) FROM book_info ");
53     sb.append(" LEFT JOIN book_type ON book_info.book_type=book_type.id
");
54     //where 1=1作用: 条件的拼接关键字and设置进去
55     sb.append(" where 1=1 ");
56     //String判断两个值: null ""
57     //将所有的? 对应的实参, 使用集合保存
58     List params=new ArrayList();
59     if(!Objects.isNull(bookType)){
60         sb.append(" AND book_info.book_type=? ");
61         params.add(bookType);
62     }
63     if(!Objects.isNull(bookName)&&!bookName.isEmpty()){
64         sb.append(" AND book_info.book_name LIKE ? ");
65         params.add("%"+bookName+"%");
66     }
67     if(!Objects.isNull(isBorrow)){
68         sb.append(" AND book_info.is_borrow=? ");
69         params.add(isBorrow);
70     }
71
72     //qr.query(String 要执行的sql语句,
73     // BeanHandler BeanListHandler ScalarHandler 转换的类型,
74     // Object... sql语句? 对应的实参)
75     try {
76         return qr.query(sb.toString(),
77             new ScalarHandler<>(),
78             // Object... 底层就是数组, 所以集合必须转换为数组, 底层才能正常解
析
79             params.toArray());
80     } catch (SQLException e) {
81         throw new RuntimeException(e);
82     }
83 }

```

serviceのBookInfoServiceImpl实现分页方法调用

```
1  @Override
2  public Integer getCount(Integer bookType, String bookName, Integer isBorrow)
3  {
4      Object o = dao.selectCount(bookType, bookName, isBorrow);
5      Integer count=0;
6      //Number是Integer Long Double Float公共的父类
7      if(o instanceof Number){
8          Number num=(Number)o;
9          //提取number保存数据的整数格式
10         count=num.intValue();
11     }
12     return count;
13 }
14
15 @Override
16 public List<BookInfo> getAll(Integer bookType, String bookName, Integer
17 isBorrow,Integer pageNum,Integer pageSize) {
18     /**
19      * 业务逻辑层处理页面和页大小的默认值
20      * jsp页面不给我pageNum的值，默认查询第一页
21      * jsp页面不给我pageSize的值，默认查询每页显示3条
22      */
23     if(Objects.isNull(pageNum) || pageNum==0){
24         pageNum=1;
25     }
26     if(Objects.isNull(pageSize) || pageSize==0){
27         pageSize=3;
28     }
29     return dao.selectAll(bookType,bookName,isBorrow,pageNum,pageSize);
30 }
```

ServletのBookInfoServlet实现分页数据的存储

```
1  package cn.kgc.controller; /**
2      * @Author: lc
3      * @Date: 2022/5/17
4      * @Description: ${PACKAGE_NAME}
5      * @Version: 1.0
6      */
7
8  import cn.kgc.domain.BookInfo;
9  import cn.kgc.domain.BookType;
10 import cn.kgc.domain.Page;
11 import cn.kgc.service.BookInfoService;
12 import cn.kgc.service.impl.BookInfoServiceImpl;
13 import cn.kgc.service.impl.BookTypeServiceImpl;
14
15 import javax.servlet.ServletException;
16 import javax.servlet.annotation.WebServlet;
17 import javax.servlet.http.HttpServlet;
18 import javax.servlet.http.HttpServletRequest;
19 import javax.servlet.http.HttpServletResponse;
20 import java.io.IOException;
21 import java.util.List;
```

```

22
23 @WebServlet("/BookInfoServlet")
24 public class BookInfoServlet extends HttpServlet {
25     @Override
26     protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
27         //请求发送的搜索条件中，图书名称可能是中文，GET必须处理中文
28         //tomcat8 9 10针对get没有乱码
29         //取 请求体或url地址传输的表单数据？ 略
30         String bookTypeStr = request.getParameter("bookType");//请求没有数据
时，bookTypeStr null
31         String bookName = request.getParameter("bookName");//bookName null
32         String pageNumStr=request.getParameter("pageNum");
33         Integer pageSize=3;
34         //解决get请求乱码问题 trim():去除字符串首尾的无效空格
35         if (bookName!=null && !bookName.trim().isEmpty()) {
36             bookName=new String(bookName.getBytes("iso8859-1"),"utf-8");
37         }
38         String isBorrowStr = request.getParameter("isBorrow");
39
40         //类型转换
41         Integer pageNum=1;
42         try {
43             //valueOf(null)不能正常转换
44             pageNum=Integer.valueOf(pageNumStr);
45         } catch (Exception e) { }
46         Integer bookType=null;
47         try {
48             //valueOf(null)不能正常转换
49             bookType=Integer.valueOf(bookTypeStr);
50         } catch (Exception e) { }
51
52         Integer isBorrow=null;
53         try {
54             //valueOf(null)不能正常转换
55             isBorrow=Integer.valueOf(isBorrowStr);
56         } catch (Exception e) { }
57         //调 service层
58         BookInfoService service = new BookInfoServiceImpl();
59         //思考：查询按钮提交请求，是不是可以使用这个servlet来处理？？
60         List<BookInfo> books =
service.getAll(bookType,bookName,isBorrow,pageNum,pageSize);
61         //查询结果封装成Page对象
62         Page<BookInfo> page=new Page<>();
63         //设置属性，以便jsp显示
64         page.setList(books);
65         page.setPageNum(pageNum);//1
66         page.setPageSize(pageSize);//3
67         //books是分页查询结果集合，books.size作为总行数使用吗？ 不行
68         //查询结果的总行数： dao+ select count(*) from ...
69         Integer count = new BookInfoServiceImpl().getCount(bookType,
bookName, isBorrow);
70         page.setTotalResults(count);//3
71
72         //查询图书分类信息
73         List<BookType> types = new BookTypeServiceImpl().getAllBookTypes();
74         //存 共享
75         request.setAttribute("list",books);//显示图书信息

```

```

76     request.setAttribute("page", page);
77     request.setAttribute("types", types); //显示图书分类信息
78     //为了实现查询条件的回显，所以要将所有查询条件存入域
79     request.setAttribute("type", bookType);
80     request.setAttribute("bookName", bookName);
81     request.setAttribute("borrow", isBorrow);
82     //转
83     //jsp本质就是一个servlet
84
85     request.getRequestDispatcher("/booklist.jsp").forward(request, response);
86
87     @Override
88     protected void doPost(HttpServletRequest request, HttpServletResponse
89     response) throws ServletException, IOException {
90         doGet(request, response);
91     }
92 }

```

jsp的booklist.jsp分页代码实现

```

1  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2  <!-- 导入jstl依赖
3  prefix设置值 自定义，一般建议以导入库单词首字母
4  uri: 使用标签官方提供的一个访问路径
5  --%>
6  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
7  <html>
8  <head>
9      <title>Title</title>
10 </head>
11 <script type="text/javascript">
12     function doSearch(){
13         //单击查询按钮，向BookInfoServlet发出请求
14         //发出请求时，需要带着查询条件过去
15         //form走POST 地址栏地址能变，就是发请求统一get
16         // window.open();
17         //js支持EL
18         //定义字符串，保存搜索条件
19         var query="bookType="+document.getElementById("type").value; //应该
20         query+="&bookName="+document.getElementById("book_name").value;
21         query+="&isBorrow="+document.getElementById("borrow").value;
22
23         location.href="${pageContext.servletContext.contextPath}/BookInfoServlet?"
24         +query;
25         // location.assign();
26     }
27 </script>
28 <body>
29 <table border="1px" cellspacing="0px" align="center">
30     <tr>
31         <td align="center">
32             <h2 align="center">图书借阅系统</h2>
33             <!--<p align="center">-->
34             图书分类:
35             <select name="book_type" id="type">

```

```

34         <option value="">全部</option>
35         <!-- 从数据库查询的图书分类 -->
36         <c:forEach items="${types}" var="t">
37             <!-- 要求: value指定主键 innerText:列值 汉字 -->
38             <option value="${t.id}"
39                 <c:if test="${type==t.id}">
40                     selected
41                 </c:if>
42                 >${t.type_name}</option>
43         </c:forEach>
44     </select>
45     图书名称: <input type="text" id="book_name"
value="${bookName}"/>
46     是否借阅:
47     <select name="is_borrow" id="borrow">
48         <option value="">===请选择===</option>
49         <option value="1"
50             <c:if test="${borrow==1}">
51                 selected
52             </c:if>
53         >已借阅
54         </option>
55         <option value="0"
56             <c:if test="${borrow==0}">
57                 selected
58             </c:if>
59         >未借阅
60         </option>
61     </select>
62     <!--
63     submit和Image 提交表单按钮
64     reset 重置按钮
65     button 普通按钮
66     -->
67     <input type="button" value="查询" id="search"
onclick="doSearch()"/>
68     <!--</p>-->
69     <p align="right">
70         <!--
71         <a
href="${pageContext.servletContext.contextPath}/servlet或jsp"></a>
72         -->
73         <a
href="${pageContext.servletContext.contextPath}/GetBookTypesServlet">新增图
书</a>
74     </p>
75     <!--
76     写地址如何区分是否带web项目名称
77     看浏览器地址是否会发生变化
78     带web路径
79     转发不用带项目名
80     -->
81
82     <!-- dreamweaver -->
83     <table width="100%" border="1" cellspacing="0" cellpadding="0">
84         <caption>
85             图书信息一览表
86         </caption>

```

[illegible]

```

136         <a
href="${pageContext.servletContext.contextPath}/BookInfoServlet?pageNum=1">
    首页</a>
137         <a
href="${pageContext.servletContext.contextPath}/BookInfoServlet?
pageNum=${page.pageNum-1}">上一页</a>
138         <a
href="${pageContext.servletContext.contextPath}/BookInfoServlet?
pageNum=${page.pageNum+1}">下一页</a>
139         <a
href="${pageContext.servletContext.contextPath}/BookInfoServlet?
pageNum=${page.totalPages}">尾页</a>
140         1/3页
141         <input type="text" id="txtPageNo" size="2" />
142         <input type="button" value="Go" name="go"/>
143     </td>
144 </tr>
145 </table>
146 </td>
147 </tr>
148 </table>
149 </body>
150 </html>

```

常见JSP页面的异常

```

38:         <option value="${type.id}"
39:             <c:if test="${type==type.id}">
40:                 selected
41:             </c:if>
42:         >${type.type_name}</option>

```

Stacktrace:

```

org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:568)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:470)
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
cn.kgc.controller.BookInfoServlet.doGet(BookInfoServlet.java:71)
javax.servlet.http.HttpServlet.service(HttpServlet.java:621)
javax.servlet.http.HttpServlet.service(HttpServlet.java:728)

```

root cause

```

javax.el.ELException: Cannot convert BookType(id=1, type_name='文学') of type class cn.kgc.domain.BookType to class java.lang.Long
    org.apache.el.lang.ELSupport.coerceToNumber(ELSupport.java:294)
    org.apache.el.lang.ELSupport.equals(ELSupport.java:143)
    org.apache.el.parser.AstEqual.getValue(AstEqual.java:40)
    org.apache.el.ValueExpressionImpl.getValue(ValueExpressionImpl.java:185)
    org.apache.jasper.runtime.PageContextImpl proprietaryEvaluate(PageContextImpl.java:1026)
    org.apache.jsp.booklist_jsp._jspx_meth_c_005fif_005f0(booklist_jsp.java:262)
    org.apache.jsp.booklist_jsp._jspx_meth_c_005fforEach_005f0(booklist_jsp.java:227)
    org.apache.jsp.booklist_jsp._jspService(booklist_jsp.java:106)
    org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
    org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:432)
    org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
    org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)

```

课程总结

1 掌握：多条件查询

2 掌握：分页

