# 1.为了是我们的额程序能够简洁，像韩老师一样，做出了一个SqlHelper.java，下面的是它的代码（ArrayList）

**public** **class** SqlHelper {

**private** Connection ct=**null**;

**private** ResultSet rs=**null**;

**private** PreparedStatement ps=**null**;

//下面这个是通过 select选择得到的东西（比如我们挑选了其中的）

**public** ArrayList executeQuery(String sql,String paras[]){

ArrayList al = **new** ArrayList();

**try**{

ct = SqlHelper1.*getConnection*();

ps = ct.prepareStatement(sql);

**for**(**int** i=0;i<paras.length;i++){

ps.setString(i+1,paras[i]);

}

rs=ps.executeQuery();

ResultSetMetaData rsmd=rs.getMetaData();

//通过上上面的rsmd得到 该结果有多少列

**int** columnNum=rsmd.getColumnCount();

**while**(rs.next()){

Object objects[] = **new** Object[columnNum];

**for** (**int** i = 0; i < objects.length; i++) {

objects[i]=rs.getObject(i+1); }

al.add(objects); }

}**catch**(Exception e){

e.printStackTrace();

}**finally**{

SqlHelper1.*close*(rs, ps, ct);

}

**return** al;

}

}

1.上面的的制作完成之后，我们对于用户登录，首先想到的就是用户登录用到的验证用户是不是正确的，我们制作一个java来进行验证

**public** **class** UserService {

**public** **boolean** CheckUser(User user){

**boolean** a = **false**;

String sql ="select \* from lu where user = ? and password = ?";

String paras[] = {user.getUser(),user.getPassword()};

SqlHelper sqlHelper = **new** SqlHelper();

ArrayList arrayList = sqlHelper.executeQuery(sql, paras);

//这里需要注意的是下面的不是空的，而是

/\*if(arrayList!=null){

a = true;

}\*/

**if**(arrayList.size()!=0){

a =**true**;

}

**return** a ;

}

}

1.2上面的那个只是对用户进行了验证我们，有的时候还需要的到用户的所有的信息，这个将来会用到

**public** **boolean** CheckUser(User user){

**boolean** a = **false**;

String sql ="select \* from lu where user = ? and password = ?";

String paras[] = {user.getUser(),user.getPassword()};

SqlHelper sqlHelper = **new** SqlHelper();

ArrayList arrayList = sqlHelper.executeQuery(sql, paras);

**if**(arrayList.size()!=0){

//使用对象将我们想要得到的东西

Object objects[] = (Object[])arrayList.get(0);

user.setEmil((String)objects[4]);

user.setGrade((String)objects[5]);

user.setPhone((String)objects[3]);

a =**true**;

}

**return** a ;

}

}

## 2.用户检查完成之后的下一步就是跳转到我们想要跳转的（购物的）界面，要分页显示，所以我们需要得到数据库中的东西，我们制作一个函数来得到这些记录

//得到所有书籍的信息

**public** ArrayList getAllBook(){

String sql = "selete \* from lu ";

SqlHelper sqlHelper = **new** SqlHelper();

ArrayList arrayList = sqlHelper.executeQuery(sql, **null**);

ArrayList<Book> books = **new** ArrayList<Book>();

**if**(arrayList.size()!=0){

**for** (**int** i = 0 ; i< arrayList.size() ; i++){

//下面的这个arrayList.get(0); 是得到我们的第一条记录，

Object objects[] = (Object[])arrayList.get(i);

Book book = **new** Book();

//object 是一个项目我们这里通过的到里面的饿东西，然后进行转化到字符串

book.setBookName(objects[1].toString());

book.setPublishHouse(objects[2].toString());

book.setAuthor(objects[3].toString());

//这个操作有点麻烦，但是我们一定要记住

book.setPrice(Integer.*parseInt*(objects[4].toString()));

book.setNum(Integer.*parseInt*(objects[5].toString()));

books.add(book);

}

}

**return** **null**;

}

## 3.然后我们到了验证用户的jsp文件中，进行回收这个东西，然后发送给购物界面yan.jsp中

<%

String name = request.getParameter("user");

String password = request.getParameter("password");

User user = **new** User();

user.setUser(name);

user.setPassword(password);

UserService userService = **new** UserService();

**if**(userService.CheckUser(user)){

BookService bookService = **new** BookService();

//下面这个不需要加<Book>

ArrayList a = bookService.getAllBook();

request.setAttribute("book", a);

//上面的这个是为了让我们能够 到了下一个界面进行应用

request.getRequestDispatcher("/shop.jsp").forward(request, response);

}

**else** {

request.getRequestDispatcher("/Login.jsp").forward(request, response);

}

# 4. .然后我们到了显示购物界面的jsp文件（shop.jsp）非常重要

## 4.1 <% 可以分开使用

## 4.2 对于我们得到的序列，我们可以按照他的本来面目，进行识别如下面的绿色字体，之前我们学习的都是 Object

<table border=*"1"*cellspacing=*"0"* width=*"600"*height=*"250"*>

<tr align=*"center"*>

<td>书名</td>

<td>价格</td>

<td>出版社</td>

<td>点击购买</td>

</tr>

<%

response.setContentType("text/html;character=gbk");

ArrayList books = (ArrayList)request.getAttribute("book");

**for**(**int** i = 0 ; i < books.size(); i++){

//下面这个一定要谨记

Book book = (Book)books.get(i);

%>

<tr align=*"center"*>

<td style="width: *123px*; "><%out.println(book.getBookName()); %></td>

<td><%out.println(book.getPrice()); %></td>

<td><%out.println(book.getPublishHouse()); %></td>

<td><a href=*"#"*>点击购买</a></td>

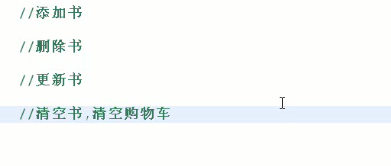
</tr>

<%

}

%>

# 5.到了这里书籍已经有了，下面的步奏就是我们购买东西了，购买就会涉及到



### 1.我们新建立一个java文件专门用来处理有与购物出相关的类

**public** **class** MyCar {

HashMap<String, Book> hm= **new** HashMap<String, Book>();

//添加书籍

**public** **void** addBooks(String id, Book book){

hm.put(id,book);

}

//删除书籍

**public** **void** delBook(String id){

hm.remove(id);

}

//更新书籍

**public** **void** updatebooks(String id,String nums,Book book){

}

//清空购物车

**public** **void** CleanBooks(){

hm.clear();

}

}

### 2.新建一个servlet用来接收用户购买的请求（在这个之前我们应该先传一个id进入）

<tr align=*"center"*>

<td style="width: *123px*; "><%out.println(book.getBookName()); %></td>

<td><%out.println(book.getPrice()); %></td>

<td><%out.println(book.getPublishHouse()); %></td>

<td><a href=*"shopServlet?id="*+<%= book.getId()%>>点击购买</a></td>

</tr>

### 3.然后到了shopServlet中进行接收，接收之后势必会和我们的购物出产生一定联系（这个购物车应该是登录之后就会存在们，也就是创建一个session，先到了验证中添加购物车）

**if**(userService.CheckUser(user)){

BookService bookService = **new** BookService();

//下面这个不需要加<Book>

**try**{

ArrayList a = bookService.getAllBook();

// out.println(a.get(0));

// out.println("woshi");

MyCar car = **new** MyCar();

request.getSession().setAttribute("car",car);

### 4.这个时候我们到了shopservlet界面进行接收我们购买的id已经和我们拥有的购物车进行连接，接收id的时候我们就要将我们点击购买的id所代表的书添加到购物出里去，所以需要通过id先找到我们的书（到了BoolService）中得到书

//根据书的编号返回书的信息

**public** Book getBook(String id){

Book book = **new** Book();

String sql = "select \* from book whrer id = ? ";

String paras[] = {id};

SqlHelper sqlHelper = **new** SqlHelper();

ArrayList a = sqlHelper.executeQuery(sql, paras);

**if**(a.size()==1){

Object object[] = (Object[])a.get(0);

book.setId(Integer.*parseInt*((String)object[0]));

book.setBookName((String)object[1]);

book.setPublishHouse((String)object[2]);

book.setAuthor((String)object[3]);

book.setPrice(Integer.*parseInt*((String)object[4]));

book.setNum(Integer.*parseInt*((String)object[5]));

}

**return** book;

}

### 5然后将我们的书籍在购物得到购物车（跳转到我的购物车去）

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

//接收用户想购买的id

String id = request.getParameter("id");

out.println(id);

//接收来自yan的session

MyCar myCar = (MyCar)request.getSession().getAttribute("car");

//将用户的购买的id假如购物车

myCar.addBooks(id, **new** BookService().getBook(id));

request.getRequestDispatcher("/showmyCar").forward(request, response);

}

## 6.这个时候就会跳到了我们的购物车的界面，然后让这个界面显示我们的购物车里面的东西了，所以我们首先需要得到数据库中购物车中的动习让它以表格的形式显示出来

//显示购物车中的所有的信息

**public** ArrayList showMyCar(){

ArrayList<Book> books = **new** ArrayList<Book>();

//要想展示我们的所有的信息，这个就是需要遍历我们的hm

Iterator iterator = hm.keySet().iterator();

**while** (iterator.hasNext()) {

//去除key

String id = (String)iterator.next();

//去除book

Book book = hm.get(id);

books.add(book);

}

**return** books;

}

### 7.然后就是把我们要显示的东西放到我们准备好的request中，传递出去，准备显示

**public** **class** shopServlet **extends** HttpServlet {

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

//接收用户想购买的id

String id = request.getParameter("id");

out.println(id);

//接收来自yan的session

MyCar myCar = (MyCar)request.getSession().getAttribute("car");

//将用户的购买的id假如购物车

myCar.addBooks(id, **new** BookService().getBook(id));

request.setAttribute("booklist", myCar.showMyCar());

request.getRequestDispatcher("/showmyCar").forward(request, response);

}

# 8.然后就是到了我的购物车的界面，开始显示，这里的显示是去除了购物车中的东西，所以我们需要在

//显示购物车中的所有的信息

**public** ArrayList showMyCar(){

ArrayList<Book> books = **new** ArrayList<Book>();

//要想展示我们的所有的信息，这个就是需要遍历我们的hm

Iterator iterator = hm.keySet().iterator();

**while** (iterator.hasNext()) {

//去除key

String id = (String)iterator.next();

//去除book

Book book = hm.get(id);

books.add(book);

}

**return** books;

}

### 9.开始显示购物车中的东西了，看看里面的东西，很重要的

、

<%

ArrayList arrayList = (ArrayList)request.getAttribute("booklist");

**if**(arrayList.size()!=0){

**for**(**int** i = 0 ; i< arrayList.size() ; i++){

Book book = (Book)arrayList.get(i); %>

<tr align=*"center"*><td>1</td>

<td style="width: *123px*; "><%out.println(book.getBookName()); %></td>

<td><%=book.getPrice()%></td>

<td><%=book.getPublishHouse() %></td><td><input type=*"text"* name=*"booknum"value =’1’*/></td>

<td><a href=*"#"*>删除</a></td>

</tr>

### 10.这个时候我们可以直接购买很多种书了，但是我们还不能购买几本一样的书（）这也是需要我们到了book.java的中进行修改

**public** **class** Book {

**int** id;

String bookName;

String publishHouse;

String author;

**int** price;

**int** num;

**int** shopnum = 1;//这个是购买的数量

**public** **int** getShopnum() {

**return** shopnum;

}

### 11这个有了之后我们就可以到了购物车的java文件中进行设置了

//添加书籍

**public** **void** addBooks(String id, Book book){

**if**(hm.containsKey(id)){

//上面验证过已经有了id， 所以我们先要通过id先找到这个 书 ，然后我们得到所需要的东西

book = hm.get(id);

//如果这本书购买过了，就将我们我们在book中的购买的数量 +1

**int** shopnum = **new** Book().getNum();

book.setShopnum(shopnum+1);

//上面已经改变了这个boo的属性，下面这个就是继续讲这个数放到我们的购物车里面去}{

hm.put(id,book);

}

}

### 12.上面的是一张方法我但是一定程度上是浪费了资源了

**public** **class** MyCar {

HashMap<String, Book> hm= **new** HashMap<String, Book>();

//第二种添加书籍的方法

**public** **void** addbook(String id){

**if**(hm.containsKey(id)){

Book book = hm.get(id);

**int** shopnum = book.getShopnum();

book.setShopnum(shopnum+1);

}**else**{

hm.put(id, **new** BookService().getBook(id));

}

}

12.然后就是到了我们需要的地方将它显示出来

## 13.然后我们到了显示购物车的界面进行改动

<%

ArrayList arrayList = (ArrayList)request.getAttribute("booklist");

**if**(arrayList.size()!=0){

**for**(**int** i = 0 ; i< arrayList.size() ; i++){

Book book = (Book)arrayList.get(i);

%>

<tr align=*"center"*>

<td>1</td>

<td style="width: *123px*; "><%out.println(book.getBookName()); %></td>

<td><%=book.getPrice()%></td>

<td><%=book.getPublishHouse() %></td>

<td><input type=*"text"* name=*"booknum"* values =*""*+<%=book.getShopnum() %>/></td>

<td><a href=*"#"*>删除</a></td>

</tr>

13.

14.然后我们想到的就是在界面中显示购物车的总价，回到我们的购物车的java文件中

//得到总价格,其实这就是这个就是一个遍历的过程

**public** **float** getAllPrice(){

ArrayList<Book> books = **new** ArrayList<Book>();

Iterator iterator = hm.keySet().iterator();

**float** a = 0 ;

**while** (iterator.hasNext()) {

//去除key

String id = (String)iterator.next();

//取出book

Book book = hm.get(id);

a = a+ book.getPrice()\*book.getShopnum();

}

**return** a;

}

15，然后我们到了shop的中转站，进行传送shopServlet

**public** **class** shopServlet **extends** HttpServlet {

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

//接收用户想购买的id

String id = request.getParameter("id");

out.println(id);

//接收来自yan的session

MyCar myCar = (MyCar)request.getSession().getAttribute("car");

//将用户的购买的id假如购物车

myCar.addBooks(id, **new** BookService().getBook(id));

//先让我们的id 通过购物车 然后通过

request.setAttribute("booklist", myCar.showMyCar());

request.setAttribute("allPrice", **new** MyCar().getAllPrice()+“”);

request.getRequestDispatcher("/showmyCar").forward(request, response);

}

<%

ArrayList arrayList = (ArrayList)request.getAttribute("booklist");

**if**(arrayList.size()!=0){

**for**(**int** i = 0 ; i< arrayList.size() ; i++){

Book book = (Book)arrayList.get(i);

%>

…………………………………………

<tr align=*"center"*>

<td colspan=*"6"*>购物总价格${allPrice}元 </td>

</tr>

# 16然后就是到了显示购物车的界面进行接收，这里有一个高级的功能用来显示