



jsp 9강-쇼핑몰 장바구니, 주문

양 명 속

[now4ever7@gmail.com]



목차

- 장바구니 담기
- 장바구니 목록
- 장바구니 수정, 삭제

- 주문하기
- 주문완료 페이지



장바구니 테이블

```
CREATE TABLE Cart
```

```
(  
  cartNo          number    Primary Key NOT NULL,          --일련번호  
  customerId      VarChar2(20)  
    references member2(userid) on delete cascade not null, --고객구분번호 : 고유접속번호(세션ID)  
  productNo       number  
    references products(productNo) NOT NULL,               --상품고유번호(Product테이블의 ProductNo필드)  
  quantity        number NOT NULL,                         --수량  
  regdate         Date    Default sysdate                  --생성일  
);
```

```
create sequence cart_seq  
start with 1  
increment by 1  
nocache;
```

주문 테이블

```
--drop table orders cascade constraint;
```

```
CREATE TABLE Orders
```

```
(  
  orderNo          number    NOT NULL Primary Key,  --주문번호  
  customerId       varchar2(20)  
    references member2(userid) NOT NULL ,           --고객고유번호(회원아이디)(FK)  
  totalPrice       number Null,                     --주문 총금액  
  deliveryStatus   VarChar2(20) default '입금확인중', --배송상황(1:입금확인중, 2:결제완료, 3:배송중,  
    (배송준비중) 4:배송완료)  
  orderDate        Date default sysdate ,           --주문일자  
  deliveryDate     Date  NULL,                      --배송일자  
  paymentEndDate   Date Null,                      --결제완료일  
  message          varchar2(150),                  --남기고 싶은 말  
  
  customerName     VarChar2(50),                   --받는 사람 이름, 배송 정보  
  hp               VarChar2(20) Null,              --휴대폰번호  
  zipcode          VarChar2(7) Null,               --우편번호  
  address          VarChar2(100) Null,             --주소  
  addressDetail    VarChar2(50) Null              --상세주소  
);
```

```
create sequence orders_seq  
start with 1  
increment by 1  
nocache;
```



주문상세, 배송지 정보 테이블

--[5] 주문상세

CREATE TABLE OrderDetails

(

 orderNo number

 references orders(orderNo) on delete cascade NOT NULL , --주문번호(고유일련번호)

 productNo number

 references products(productNo) NOT NULL,

--상품번호

 quantity number NOT NULL,

--주문수량

 constraint orderdetails_pk primary key(orderNo, productNo)

);



장바구니 view

--장바구니 View 작성

create or replace view cartView

as

```
select p.imageurl, p.productname, p.sellprice, c.quantity,  
       p.sellprice*c.quantity as sum, c.cartno, c.customerid  
from cart c, products p  
where c.productno=p.productno
```

with read only;

--뷰 생성 권한 부여하기

grant create view to herb;



CartViewVO

```
public class CartViewVO {  
    //멤버변수  
    private int cartNo;  
    private String customerId;  
    private int productNo;  
    private int quantity;  
    private Timestamp regdate;  
  
    //장바구니 목록에 필요한 멤버변수  
    private String imageUrl;  
    private String productName;  
    private int sellPrice;  
    private int sum; //금액
```



productDetail.jsp

```
function cartFunc(mode){
    if(!frmPd.quantity.value){
        alert("수량을 입력하세요");
        frmPd.quantity.focus();
        return;
    }
    frmPd.action="<c:url value='/shop/cart/cartAdd.do?mode="+mode+" ' />";
    frmPd.submit();
}
```

```
<form name="frmPd" method="post">
```

```
.....
```

```
<input type="text" name="quantity" id="quantity" value="1" >
```

```
<input type="hidden" name="productNo" value="${pdBean.productNo}" >
```

```
<input type="button" value="바로구매" onclick="cartFunc('order')">
```

```
<input type="button" value="장바구니담기" onclick="cartFunc('cart')">
```

```
</form>
```


cart.xml

```
<mapper namespace="config.mybatis.mapper.oracle.cart">
  <select id="getCartPdNoCount" parameterType="cartBean" resultType="int">
    select count(*) from cart
    where customerid=#{customerid}
    and productNo=#{productNo}
  </select>

  <update id="cartUpdateByPdNo" parameterType="cartBean">
    update cart set quantity=quantity+#{quantity}
    where customerid=#{customerid}
    and productNo=#{productNo}
  </update>

  <insert id="cartInsert" parameterType="cartBean">
    <selectKey keyProperty="cartNo" resultType="int" order="BEFORE">
      select cart_seq.nextval from dual
    </selectKey>
    insert into cart(cartno, customerid, productno,quantity)
    values(#{cartNo}, #{customerid}, #{productNo}, #{quantity})
  </insert>

  <select id="cartList2" parameterType="string" resultType="hashmap">
    select * from cartView
    where customerid=#{customerid}
  </select>
```

```
<update id="cartUpdate" parameterType="cartBean">  
    update cart set quantity=#{quantity}  
    where cartNo=#{cartNo}  
</update>
```

cart.xml

```
<delete id="cartDelete" parameterType="int">  
    delete from cart  
    where cartNo=#{cartNo}  
</delete>
```

```
</mapper>
```



CartDAOMybatis

@Repository

```
public class CartDAOMybatis extends SqlSessionDaoSupport implements CartDAO{
    private String namespace="config.mybatis.mapper.oracle.cart.";

    //메서드
    public int getCartPdNoCount(CartBean bean){
        return getSqlSession().selectOne(namespace+"getCartPdNoCount", bean);
    }

    public int cartUpdateByPdNo(CartBean bean){
        return getSqlSession().update("namespace+"cartUpdateByPdNo", bean);
    }

    public int insertCart(CartBean bean){
        return getSqlSession().insert("namespace+"cartInsert", bean);
    }
}
```



CartDAOMybatis

```
public List<Map<String, Object>> listCart2(String userid) {  
    return getSqlSession().selectList(namespace+"cartList2", userid);  
}  
  
public int updateCart(CartBean bean){  
    int n= getSqlSession().update(namespace+"cartUpdate", bean);  
    return n;  
}  
  
public int deleteCart(int cartNo){  
    int n= getSqlSession().delete(namespace+"cartDelete", cartNo);  
    return n;  
}  
}
```



CartService

```
public int insertCart(CartBean bean){
    //장바구니 담기
    // 해당 회원의 해당 상품이 이미 장바구니에 들어 있으면
    // update, 처음 넣는 상품이면 insert 하자

    int cnt = cartDao.getCartPdNoCount(bean);

    int n=0;
    if(cnt>0){ //해당 상품이 이미 있는 경우
        n=cartDao.cartUpdateByPdNo(bean);
    }else{
        n = cartDao.insertCart(bean);
    }

    return n;
}
```

```

@Controller
@RequestMapping("/shop/cart/")
public class CartController {
    @Autowired CartService cartService;
    private static final Logger logger = LoggerFactory.getLogger(CartController.class);

    @RequestMapping("/cartAdd.do")
    public String cartAdd(CartBean bean, HttpServletRequest request, String mode, Model model){
        //로그인된 경우에만 장바구니 담기
        //1. 파라미터
        String userid = (String) request.getSession().getAttribute("userid");
        logger.debug("파라미터: userid={}, cartBean={}",userid,bean);
        logger.debug("mode={}",mode);

        bean.setCustomerId(userid);

        //2. db작업 - update, insert
        int n=cartService.insertCart(bean);
        logger.debug("장바구니 담기 결과, n={}",n);

        //3. 결과 저장, 리턴
        String viewName="";
        if(mode.equals("cart")){ //장바구니 담기
            viewName="redirect:/shop/cart/cartList.do";
        }else{ //바로 구매
            viewName = "redirect:/shop/order/orderSheet.do";
        }

        return viewName;
    }
}

```

```
@RequestMapping("/cartEdit.do")
public String cartEdit(CartBean bean){
    //1. 파라미터
    logger.debug("파라미터: cartBean={}", bean);

    //2. db작업 - update
    int n=cartService.updateCart(bean);
    logger.debug("장바구니 수정 결과: n={}", n);

    //3. 결과 저장, 리턴
    return "redirect:/shop/cart/cartList.do";
}
```

```
@RequestMapping("/cartDelete.do")
public String cartDelete(int cartNo){
    //1. 파라미터
    logger.debug("파라미터: cartNo={}", cartNo);

    //2. db작업
    int n=cartService.deleteCart(cartNo);
    logger.debug("장바구니 삭제 결과 : n={}", n);

    //3. 결과 저장, 리턴
    return "redirect:/shop/cart/cartList.do";
}
```

```

@RequestMapping("/cartList.do")
public String cartList(HttpServletRequest request, Model model){
    //로그인한 회원의 장바구니 목록 조회하기
    //1. 파라미터
    String userid= (String) request.getSession().getAttribute("userid");

    logger.debug("파라미터: userid={}", userid);

    //2. db작업 - select
    List<Map<String, Object>> alist= cartService.listCart2(userid);
    logger.debug("장바구니 목록 조회 결과, alist.size()={}", alist.size());

    //3. 결과저장, 리턴
    model.addAttribute("cartList", alist);
    model.addAttribute("TOTAL_PRICE", cartService.BUY_PRICE);
    model.addAttribute("DELIVERY", cartService.DELIVERY);

    return "/shop/cart/cartList";
}
}
}

```


장바구니 리스트

Herb mall

(홍길동 님) 로그아웃 | 회원정보변경 | 회원탈퇴 | 장바구니 | 주문내역 | 마이페이지

베스트

자유게시판

이용후기

Q&A

FAQ

이벤트&쿠폰

허브

장바구니

허브차

아로마오일

허브향초

상품명	가격	수량	금액	삭제
 로즈마리	8,000원	<input type="text" value="1"/> <input type="button" value="수정"/>	8,000원	삭제
 라벤다허브차	7,000원	<input type="text" value="2"/> <input type="button" value="수정"/>	14,000원	삭제
			총 구매금액 :	22,000 원
			+ 배송비 :	3,000 원
			총 주문합계 :	25,000원

[주문하기] [계속 쇼핑하기]

```

<script type="text/javascript">
function edit(frm){
    if(!frm.qty.value){
        alert("수량을 입력하세요");
        frm.qty.focus();
        return false;
    }
    return true;
}
</script>
<h2>장바구니</h2>
<table class="cartListTbl box2"
    summary="장바구니 목록에 관한 표로써, 상품명, 가격, 수량, 금액 등의 정보를 제공합니다.">
    <caption>장바구니 목록</caption>
    <colgroup>
        <col width="40%" /><col width="13%" /><col width="20%" />
        <col width="13%" /><col width="*" />
    </colgroup>
    <thead>
        <tr><th scope="col">상품명</th><th scope="col">가격</th><th scope="col">수량</th>
            <th scope="col">금액</th><th scope="col">삭제</th>
        </tr>
    </thead>
    <tbody>
        <c:if test="${empty cartList }">
            <tr><td colspan="5" align="center">장바구니가 비었습니다</td></tr>
        </c:if>
        <c:if test="${!empty cartList }">
            <!--배송비 관련 변수 -->
            <c:set var="totalPrice" value="0" />
            <c:set var="delivery" value="0" />
            <c:set var="sumPrice" value="0" />

```

```

<!--반복 시작 -->
<c:forEach var="map" items="${cartList }">
    <c:set var="sum" value="${map['SELLPRICE'] * map['QUANTITY'] }" />
    <tr>
        <td><img src='<c:url value="/pd_images/${map['IMAGEURL']}' />'
            width="60" align="absmiddle">
            ${map['PRODUCTNAME']}</td>
        <td align="right">
            <fmt:formatNumber value="${map['SELLPRICE']}" pattern="#,###"/>원</td>
        <td align="center">
            <form name="frm1" method="post" onsubmit="return edit(this)"
                action='<c:url value="/shop/cart/cartEdit.do" />' >
                <input type="hidden" name="cartNo"
                    value="${map['CARTNO']}">
                <input type="text" name="quantity" size="5"
                    value="${map['QUANTITY']}">
                <input type="submit" value="수정">
            </form>
        </td>
        <td align="right"> <fmt:formatNumber value="${sum}"
            pattern="#,###"/>원</td>
        <td align="center">
            <a href='<c:url value="/shop/cart/cartDelete.do?cartNo=${map['CARTNO']}' />' >
                삭제</a>
        </td>
    </tr>
    <!-- 총 구매금액 구하기 -->
    <c:set var="totalPrice" value="${totalPrice + sum }" />
</c:forEach>
<!-- 반복 끝 -->
<!-- 배송비 구하기 - 총 구매금액이 30000원 미만이면 배송비 3000원 -->

```

```

<c:if test="\${totalPrice< TOTAL_PRICE}">
    <c:set var="delivery" value="\${DELIVERY }" />
</c:if><!-- 총 주문합계 구하기 -->
<c:set var="sumPrice" value="\${totalPrice + delivery }" />
<tr>
    <td colspan="4" align="right" style="border-right:none">
        총 구매금액 : <br>
        + 배송비 : <br>
        총 주문합계 :
    </td>
    <td align="right" style="border-left:none">
        <fmt:formatNumber value="\${totalPrice}" pattern="#,###"/> 원<br>
        <fmt:formatNumber value="\${delivery}" pattern="#,###"/> 원<br>
        <fmt:formatNumber value="\${sumPrice}" pattern="#,###"/>원
    </td>
</tr>
</c:if>
</tbody>
</table>

```

```

<div class="cartListDiv1">
    <c:if test="\${!empty cartList }">
        <a href='<c:url value="/shop/order/orderSheet.do" />' >[주문하기]</a>
    </c:if>

    <a href='<c:url value="/index.do" />'>[계속 쇼핑하기]</a>
</div>

```



주문하기

주문하기-orderSheet.jsp

주문서 작성

상품명	가격	수량	금액
 알로에캔들	18,000원	1	18,000원
 자스민허브차	9,000원	1	9,000원
총 구매금액 :			27,000 원
+ 배송비 :			3,000 원
총 주문합계 :			30,000원

주문하시는 분

이름	홍길동
주소	121-760 서울 마포구 서교동 서교타워오피스텔 150
연락처	016 -2222 -3333
이메일	h2@gmail.com

주문하기

> 상품 받으시는 분

배송지 선택

☒ 주문고객과 동일 주소 ☐ 새로운 주소 입력

성명

홍길동

주소

121-760

우편번호찾기

서울 마포구 서교동 서교타워오피스텔

150

연락처

016-2222-3333

배송시 요청사항

> 결제 정보

결제금액

30,000원

결제하기

주문 완료 페이지-orderComplete.jsp

주문 완료

주문과 결제가 정상적으로 완료되었습니다 [주문번호 : 32]

> 주문하신 상품

상품명	판매가격	수량	합계
 자스민허브차	9,000원	1	9,000원
 알로에캔들	18,000원	1	18,000원

> 배송 정보

보내시는 분 홍길동

받으시는 분 홍길동

배송지 주소 서울 마포구 서교동 서교타워오피스텔 150

휴대폰 번호 016-2222-3333

> 결제 정보

결제금액 30,000원



주문 관련 View

```
create or replace view orderDetailsView
as
select o.orderNo, o.productNo, o.quantity,
p.productName, p.sellPrice, p.imageUrl
from orderDetails o, products p
where o.productNo = p.productNo;
```

```
-----

create or replace view ordersView
as
select o.orderNo, o.customerId, o.totalPrice,
o.deliveryStatus, o.orderDate, o.deliveryDate,
o.paymentEndDate, o.message, o.customerName,
o.hp, o.zipcode, o.address, o.addressDetail,
m.name
from orders o, member2 m
where o.customerId=m.userid;
```



OrdersViewVO

```
public class OrdersViewVO {  
    //orders  
    private int orderNo;  
    private String customerId;  
    private int  totalPrice;  
    private String  paymentStatus;  
    private String  deliveryStatus;  
    private Timestamp orderDate;  
    private Timestamp  deliveryDate;  
    private Timestamp  paymentEndDate;  
    private String  message;  
  
    //delivery  
    private String customerName; //받는 사람 이름  
    private String hp;  
    private String zipcode;  
    private String address;  
    private String addressDetail;  
  
    //member  
    private String name; //보내는 사람 이름
```



OrderDetailsViewVO

```
public class OrderDetailsViewVO {  
    //orderDetails  
    private int orderNo;  
    private int productNo;  
    private int quantity;  
  
    //product  
    private String productName;  
    private int sellPrice;  
    private String imageUrl;  
}
```

order.xml

```
<mapper namespace="config.mybatis.mapper.oracle.order">
    <insert id="orderInsert" parameterType="orderBean">
        <selectKey keyProperty="orderNo" resultType="int" order="BEFORE">
            select orders_seq.nextval from dual
        </selectKey>
        insert into orders(orderNo, customerId, totalPrice,
            message, customerName, hp, zipcode, address, addressDetail)
        values(#{orderNo}, #{customerId}, #{totalPrice},
            #{message}, #{customerName}, #{hp}, #{zipcode}, #{address}, #{addressDetail})
    </insert>

    <update id="orderDetailsInsert" parameterType="orderBean">
        insert into orderDetails(orderNo, productNo, quantity)
        select #{orderNo}, productNo, quantity from cart
        where customerId=#{customerId}
    </update>

    <delete id="orderCartDelete" parameterType="string">
        delete from cart where customerId=#{customerId}
    </delete>

    <select id="orderDetailsViewSelect2" parameterType="int" resultType="hashmap">
        select * from orderDetailsView
        where orderNo=#{orderNo}
    </select>

    <select id="ordersViewSelect2" parameterType="int" resultType="hashmap">
        select * from ordersView
        where orderNo=#{orderNo}
    </select>
```

@Repository

```
public class OrderDAOMybatis extends SqlSessionDaoSupport implements OrderDAO{  
    private String namespace="config.mybatis.mapper.oracle.order.";
```

OrderDAOMybatis

```
    public int insertOrder(OrderBean bean){  
        int cnt = getSqlSession().insert(namespace+"orderInsert", bean);  
        return cnt;  
    }
```

```
    public void insertOrderDetails(OrderBean bean){  
        getSqlSession().update(namespace+"orderDetailsInsert", bean);  
    }
```

```
    public int deleteOrderCart(OrderBean bean){  
        int n= getSqlSession().delete(namespace+"orderCartDelete", bean.getCustomerId());  
        return n;  
    }
```

```
    public Map<String, Object> getOrdersView2(int orderNo){  
        return getSqlSession().selectOne(namespace+"ordersViewSelect2", orderNo);  
    }
```

@Override

```
    public List<Map<String, Object>> listOrderDetailsView2(int orderNo) {  
        return getSqlSession().selectList(namespace+"orderDetailsViewSelect2", orderNo);  
    }
```

@Service

@Transactional

OrderServiceImpl

```
public class OrderServiceImpl implements OrderService{
    @Resource(name="orderDAOMybatis") OrderDAO orderDao;

    public int insertOrder(OrderBean bean){
        int n =orderDao.insertOrder(bean);

        orderDao.insertOrderDetails(bean); //orderNo가 bean에 세팅된 상태

        orderDao.deleteOrderCart(bean);

        return n;
    }
    public List<Map<String, Object>> listOrderDetailsView2(int orderNo) {
        return orderDao.listOrderDetailsView2(orderNo);
    }

    public Map<String, Object> getOrdersView2(int orderNo) {
        return orderDao.getOrdersView2(orderNo);
    }

    public List<OrdersAllViewBean> listOrderListAll(DateSearchBean dsBean){
        return orderDao.listOrderListAll(dsBean);
    }

    @Override
    public int getTotalRecord(DateSearchBean dsBean) {
        return orderDao.getTotalRecord(dsBean);
    }
}
```

```
@Controller
@RequestMapping("/shop/order")
public class OrderController {
    @Autowired MemberService memberService;
    @Autowired CartService cartService;
    @Autowired OrderService orderService;

    private static final Logger logger = LoggerFactory.getLogger(OrderController.class);
    @RequestMapping(value="/orderSheet.do", method=RequestMethod.GET)
    public String orderGet(HttpSession session, Model model){
        //주문하기 페이지 보여주기
        //=> userid에 해당하는 장바구니 내역 조회, 회원정보 조회
        //1. 파라미터
        String userid = (String) session.getAttribute("userid");
        logger.debug("파라미터: userid={}", userid);

        //2. db작업 - select
        List<Map<String, Object>> alist = cartService.listCart2(userid);
        logger.debug("주문화면 - 장바구니 내역 조회결과, alist.size()={}, alist.size());

        MemberBean bean = memberService.selectMember(userid);
        logger.debug("주문화면 - 회원정보 조회결과, bean={}", bean);

        //3. 결과저장, 리턴
        model.addAttribute("cartList", alist);
        model.addAttribute("memBean", bean);
        model.addAttribute("TOTAL_PRICE", CartService.BUY_PRICE);
        model.addAttribute("DELIVERY", CartService.DELIVERY);

        return "/shop/order/orderSheet";
    }
}
```

```

@RequestMapping(value="/orderSheet.do", method=RequestMethod.POST)
public String orderPost(HttpSession session, OrderBean bean, Model model){
    //결제하기- 주문 처리
    //=> orders 테이블에 insert, orderDetails 테이블에 insert
    //=> cart 테이블에서 해당 회원의 내역 삭제
    //1. 파라미터
    String userid=(String) session.getAttribute("userid");
    bean.setCustomerId(userid);

    logger.debug("파라미터: userid={}, orderBean={}", userid , bean);

    //2. db작업
    int n=orderService.insertOrder(bean);
    logger.debug("주문처리 결과: n={}", n);

    //mybatis - orderNo 추가
    int orderNo=bean.getOrderNo();

    //3. 결과 저장, 리턴
    return "redirect:/shop/order/orderComplete.do?orderNo=" + orderNo;
}

```



```

@RequestMapping("/orderComplete.do")
public String orderComplete(int orderNo, Model model){
    //주문 완료 페이지
    //=> orderDetailView에서 orderNo에 해당하는 상세 주문내역 조회
    //=> ordersView에서 orderNo에 해당하는 주문내역 조회
    //1. 파라미터
    logger.debug("파라미터: orderNo={}", orderNo);

    //2. db작업 - select
    Map<String, Object> orderMap = orderService.getOrdersView2(orderNo);
    logger.debug("주문완료 - 주문내역 조회 결과, orderMap={}", orderMap);

    List<Map<String, Object>> alist = orderService.listOrderDetailView2(orderNo);
    logger.debug("주문완료 - 주문상세조회 결과, alist.size()={}", alist.size());

    //3. 결과 저장, 리턴
    model.addAttribute("oViewMap", orderMap);
    model.addAttribute("oDetailList", alist);

    return "/shop/order/orderComplete";
}

}
}

```

```
<script type="text/javascript">
$(document).ready(function(){
    $.setAddress();
    $("#customerName").focus();
    $("#or_submit").click(function(){
        if($("#customerName").val().length<1){
            alert("이름을 입력하세요");
            $("#customerName").focus();
            return false;
        }else if($("#zipcode").val().length<1){
            alert("주소를 입력하세요");
            $("#zipcode").focus();
            return false;
        }else if($("#hp").val().length<1){
            alert("휴대폰번호를 입력하세요");
            $("#hp").focus();
            return false;
        }
    });

    $("#btnZipcode").click(function(){ //우편번호 찾기 창 띄우기
        window.open("${pageContext.request.contextPath}/member/zipcode.do",
            "zipWin", "left=50, top=20, width=500, height=500, scrollbars=yes,resizable=yes");
    });

    $("input[name=delivery]").click(function(){
        $.setAddress();
    });
}); //document.ready
```

```

$.setAddress = function(){
    if($('#delivery1').is(':checked')){    //주문고객과 동일 주소가 체크된 경우
        $('#customerName').val($("#oName").text());
        $('#zipcode').val($("#oZipcode").html());
        $('#address').val($("#oAddress1").html());
        $('#addressDetail').val($("#oAddress2").html());

        var oHp1= $("#oHp1").html();
        if(oHp1!=null && oHp1!=""){
            $("#hp").val($("#oHp1").html() + "-" + $("#oHp2").html() + "-" + $("#oHp3").html());
        }
    }else{    //새로운 주소 입력 체크
        $('#customerName').val('');
        $('#zipcode').val('');
        $('#address').val('');
        $('#addressDetail').val('');
        $('#hp').val('');
    }
}

</script>
<p class="titleP">
    
    <span style="font-size:13pt;font-weight:bold">주문서 작성</span></p>
<div>
    <table class="cartListTbl box2"
        summary="장바구니 목록에 관한 표로써, 상품명,가격, 수량, 금액 등의 정보를 제공합니다.">
        <caption>장바구니 목록</caption>
        <colgroup><col width="49%" /><col width="17%" /><col width="17%" /><col width="*" /></colgroup>35

```

```

<thead><tr><th scope="col">상품명</th><th scope="col">가격</th>
      <th scope="col">수량</th><th scope="col">금액</th></tr>

```

```

</thead>

```

```

<tbody>

```

```

<c:if test="${empty cartList }">

```

```

    <tr><td colspan="4" align="center">장바구니가 비었습니다</td></tr>

```

```

</c:if>

```

```

<c:if test="${!empty cartList }">

```

```

    <!--배송비 관련 변수 -->

```

```

    <c:set var="totalPrice" value="0" />

```

```

    <c:set var="delivery" value="0" />

```

```

    <c:set var="sumPrice" value="0" />

```

```

    <!--반복 시작 -->

```

```

    <c:forEach var="map" items="${cartList }">

```

```

        <c:set var="sum" value="${map['SELLPRICE'] * map['QUANTITY'] }" />

```

```

        <tr><td><img src='<c:url value="/pd_images/${map['IMAGEURL']}' />'
                                width="50" align="absmiddle">

```

```

                                ${map['PRODUCTNAME']}</td>

```

```

        <td align="right">

```

```

        <fmt:formatNumber value="${map['SELLPRICE']}" pattern="#,###"/> 원</td>

```

```

        <td align="center">${map['QUANTITY']}</td>

```

```

        <td align="right"><fmt:formatNumber value="${sum}" pattern="#,###"/> 원</td>

```

```

        </tr>

```

```

    <!-- 총 구매금액 구하기 -->

```

```

    <c:set var="totalPrice" value="${totalPrice + sum }" />

```

```

</c:forEach>

```

```

    <!-- 반복 끝 -->

```

```

    <!-- 배송비 구하기 - 총 구매금액이 30000원 미만이면 배송비 3000원 -->

```

```

    <c:if test="${totalPrice< TOTAL_PRICE}">

```

```

        <c:set var="delivery" value="${DELIVERY }" />

```

```

    </c:if>

```

orderSheet.jsp

```

<!-- 총 주문합계 구하기 -->
<c:set var="sumPrice" value="\${totalPrice + delivery }" />
<tr>

```

```

    <td colspan="3" align="right" style="border-right:none">

```

```

        총 구매금액 : <br>

```

```

        + 배송비 : <br>

```

```

        총 주문합계 :

```

```

    </td>

```

```

    <td align="right" style="border-left:none">

```

```

        <fmt:formatNumber value="\${totalPrice}" pattern="#,###"/>원 <br>

```

```

        <fmt:formatNumber value="\${delivery}" pattern="#,###"/> 원 <br>

```

```

        <fmt:formatNumber value="\${sumPrice}" pattern="#,###"/> 원

```

```

    </td>

```

```

</tr>

```

```

</c:if>

```

```

</tbody>

```

```

</table></div> <br />

```

```

<div class="divForm">

```

```

    <form name="frm1" method="post" action='<c:url value="/shop/order/orderSheet.do"/>' >

```

```

        <fieldset>

```

```

            <legend>상품 받으시는 분</legend>

```

```

            <div class="titleP">

```

```

                

```

```

                <span class="title">주문하시는 분</span>

```

```

            </div>

```

```

        <div><span class="sp1">이름</span><span id="oName" >\${memBean.name }</span> </div>

```

```

        <div>

```

```

            <span class="sp1">주소</span>

```

```

            <span id="oZipcode">\${memBean.zipcode }</span>

```

```

            <span id="oAddress1">\${memBean.address }</span>

```

```

            <span id="oAddress2">\${memBean.addressDetail }</span>

```

```

        </div>

```

```

        <div> <span class="sp1">연락처</span>

```

orderSheet.jsp

```

<c:if test="${!empty memBean.hp1}">
    <span id="oHp1">${memBean.hp1 }</span>-<span id="oHp2">${memBean.hp2 }</span>
    -<span id="oHp3">${memBean.hp3 }</span>
</c:if>
</div>
<div>
    <span class="sp1">이메일</span>
    <span>
        <c:if test="${!empty memBean.email1}">
            ${memBean.email1 }@${memBean.email2 }
        </c:if>
    </span>
</div><br />
<div class="titleP">
    
    <span class="title">상품 받으시는 분</span>
</div>
<div>
    <span class="sp1">배송지 선택</span>
    <input type="radio" name="delivery" id="delivery1" checked>
    <label for="delivery1" class="lbl">주문고객과 동일 주소</label>
    <input type="radio" name="delivery" id="delivery2" >
    <label for="delivery2" class="lbl">새로운 주소 입력</label>
</div>
<div>
    <label for="customerName">성명</label>
    <input type="Text" name="customerName" id="customerName">
</div>
<div>
    <label for="zipcode">주소</label>
    <input type="Text" name="zipcode" id="zipcode" size="15" title="우편번호" readonly="readonly">

```

```

    &nbsp;<input type="Button" value="우편번호찾기" id="btnZipcode" title="새창열림"/><br />
    <span class="sp1">&nbsp;</span>
    <input type="Text" name="address" id="address" size="60" title="주소" readonly> <br />
    <span class="sp1">&nbsp;</span>
    <input type="Text" name="addressDetail" id="addressDetail" size="60" title="상세주소">
</div>
<div>
    <label for="hp">연락처</label><input type="Text" name="hp" id="hp" size="17">
</div>
<div>
    <label for="message">배송시 요청사항</label>
    <textarea name="message" id="message" cols="82" rows="3" ></textarea>
</div> <br />
<div class="titleP">
    
    <span class="title">결제 정보</span>
</div>
<div>
    <span class="sp1">결제금액</span>
    <span><fmt:formatNumber value="{sumPrice}" pattern="#,###"/>원</span>
</div>
<div class="center">
    <input type="submit" id="or_submit" value="결제하기" />
</div>

<!-- 주문 총 금액 hidden field -->
<input type="hidden" name="totalPrice" value="{sumPrice}" >
</fieldset>
</form>
</div>

```

orderSheet.jsp

```

<h2>주문 완료</h2>
<p style="font-size:13pt;font-weight:bold">
    주문과 결제가 정상적으로 완료되었습니다
    [주문번호 : ${oViewMap['ORDERNO']} ]</p> <br><br>
<p class="titleP">
    <img src='<c:url value="/images/dot7.JPG"/>' align="absmiddle" />
    <span style="font-size:12pt;font-weight:bold">주문하신 상품</span>
</p>
<div>
<table class="cartListTbl box2"
    summary="주문 상품 목록에 관한 표로써, 상품명,가격, 수량, 금액 등의 정보를 제공합니다.">
    <caption>주문 상품 목록</caption>
    <colgroup><col width="49%" /><col width="17%" /><col width="17%" /><col width="*" /></colgroup>
    <thead>
        <tr><th scope="col">상품명</th><th scope="col">판매가격</th>
            <th scope="col">수량</th><th scope="col">합계</th></tr>
    </thead>
    <tbody>
    <!-- 데이터가 있는 경우 -->
    <!-- 반복문 시작 -->
    <c:forEach var="map" items="${oDetailList }">
        <tr align="center">
            <td>
                <img src='<c:url value="/pd_images/${map['IMAGEURL']}' />' border="0"
                    align="middle" width="50">
                    ${map['PRODUCTNAME']}</td>
            <td align="right">
                <fmt:formatNumber value="${map['SELLPRICE']}" type="number" pattern="#,###" />원</td>
            <td>${map['QUANTITY']}</td>
            <td align="right"><fmt:formatNumber value="${map['SELLPRICE'] * map['QUANTITY']}"
                type="number" pattern="#,###" />원</td>
        </tr>
    </c:forEach> <!-- 반복문 끝 --> </tbody> </table> </div> <br>

```



```

<div class="divForm">
  <div class="titleP">
    <img src='<c:url value="/images/dot7.JPG"/>' align="absmiddle" />
    <span class="title">배송 정보</span>
  </div>
  <div>
    <span class="sp1">보내시는 분 </span>
    <span>${oViewMap['NAME']}</span>
  </div>
  <div>
    <span class="sp1">받으시는 분 </span>
    <span>${oViewMap['CUSTOMERNAME']}</span>
  </div>
  <div>
    <span class="sp1">배송지 주소</span>
    <span>${oViewMap['ADDRESS']} ${oViewMap['ADDRESSDETAIL']}</span>
  </div>
  <div>
    <span class="sp1">휴대폰 번호</span>
    <span>${oViewMap['HP']}</span>
  </div><br>
  <div class="titleP">
    <img src='<c:url value="/images/dot7.JPG"/>' align="absmiddle" />
    <span class="title">결제 정보</span>
  </div>
  <div>
    <span class="sp1">결제금액</span>
    <span><fmt:formatNumber value="${oViewMap['TOTALPRICE']}" type="number" pattern="#,###" />원
  </span>
  </div>
</div>

```