

Série de révision - JEE  
Servlet – JSP – JSTL - JDBC

### Exercice : Gestion des produits

On souhaite créer une application Web JEE basée sur le pattern DAO qui permet de gérer des produits informatiques. Cette application permet de :

- Consulter tous les produits
- Consulter les produits par mot clé.
- Ajouter un produit.
- Supprimer un produit
- Modifier un produit

### Recherche Produit

Mot clé:

Ref Produit:

Désignation:

Prix:

Quantité:

Ref. Produit	Désignation	Prix	Quantité		
100	PC HP	2000.0	10	<a href="#">Supprimer</a>	<a href="#">Edit</a>
200	PC Dell	1000.0	5	<a href="#">Supprimer</a>	<a href="#">Edit</a>
300	Imp epson	500.0	10	<a href="#">Supprimer</a>	<a href="#">Edit</a>

interface IGestionProduit {

list<Produit> getProduitByMc (String Mc);

list<Produit> getAllProduit();

void addProduit (Produit P);

void updateProduit (Produit P);

void deleteProduit (int id);

Produit getProduitById (int id); }

Model :

Public class Produit {

Private int Ref;

Private String Desg;

Private Double Prix;

Private int Cste;

// constructeur vide

// constructeur

// getter + setter. }

DAO :

Public class GestionProduitImp implements IGestionProduit {

connection con;

Public GestionProduitImp () {

try { class.forName ("com.mysql.jdbc.Driver");

con = DriverManager.getConnection (URL, login, Pass);

} catch (Exception e) {

e.printStackTrace (); } }

list<Produit> getProduitByMc (String Mc) {

try { list<Produit> lp = new ArrayList ();

String req = "select \* from Produit where Desg like ?";

PreparedStatement ps = con.prepareStatement (req); ~~X~~ getAllProduit ();

ps.setString (1, "%" + Mc + "%"); ~~X~~

ResultSet Rs = ps.executeQuery ();

while (Rs.next ()) {

Produit P = new Produit (Rs.getInt (1), Rs.getString (2), Rs.getDouble (3),

Rs.getInt (4));

lp.add (P); }

} catch (Exception e) { }

return lp; }



Public

madhabina  
bi  
construction

best = new GestionProduit ( ) ;

denizge

→ z.B. liste die produkte hoch  
in plan  
→ spage aus  
liste

gette qu'il s'agit de  
redirection de page 2

1) - непереносимый

req. set Attribute ("p", p);

repa-se. 8-nd medicine  
by cot-client  $\neq$  ne  
series.

9. nep); { motche  
mawjondia

La cotă servem

→ no some request how to (???)

Page  $\longrightarrow$

port model: Produkt  
mit  $\frac{1}{2}$   $\frac{1}{2}$

```
m.method = "POST" action = "sample"
```

```
type "Text" name "Hc" value = "1/1 Hc 1/2"
```

4 yaf, a. le mol af.  
dabellu ms. gl.

$\sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \ln n + \gamma + o(1)$

Def: Product  $f(x) = \sum_{i=1}^n f_i(x)$  has  $n$  min.  $\Rightarrow$  Per/product  $f_i(x)$

$$n(i=0; i < \text{fp.size}(); i++) \{ \} \} \Rightarrow \text{for (product fp : p)}$$

Ans)  $\angle A + \angle C = \frac{1}{2} \cdot \frac{360^\circ}{P} \cdot \frac{1}{2} \cdot \frac{360^\circ}{P} = \frac{360^\circ}{P}$

$$L1 = \langle p.get(u).getDom(); \rangle \quad \langle 110 \rangle$$

ligne 21d <? = \$p.get('i').getPrix(); ? > </td>

$$\Delta\% = \frac{p_i \cdot \text{get}(i) \cdot \text{get}(\text{Ste}())}{\%} \cdot \Delta\% / \text{td}$$

indem sie selbst  
S action = "de Rele"

1. > " > delete /a>  
 2. > " > Edit /a>  
 3. > & action = "update"

$$2:1:3:1 \rightarrow 4 \text{ parts}$$



```
public class GestionProduitImp {
```

```
public void addProduit(Produit p) {
```

```
try {
    String req = "Insert into values (?, ?, ?, ?)";
```

```
PreparedStatement ps = con.prepareStatement(req);
```

```
ps.setInt(1, p.getId());
```

```
ps.setString(2, p.getDescription());
```

```
ps.setDouble(3, p.getPrix());
```

```
ps.setInt(4, p.getQuantite());
```

```
ps.executeUpdate();
```

```
} catch (Exception e)
```

```
{ e.printStackTrace(); }
```

Servet :

```
public class ServletAjout extends HttpServlet {
```

```
GestionProduitImp Gest = new GestionProduitImp();
```

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws {
```

```
try {
    Integer ref = Integer.parseInt(req.getParameter("Ref"));
```

```
String desc = req.getParameter("desc");
```

```
Double Prix = Double.parseDouble(req.getParameter("Prix"));
```

```
Integer qte = Integer.parseInt(req.getParameter("qte"));
```

```
try {
    if (Gest.getProduitById(ref) != null) {
        Gest.updateProduit(p);
    }
```

```
else
    Gest.addProduit(p);
}
```

```
catch (Exception e) {
    String s = e.getMessage();
    resp.sendRedirect(req.getHeader("Referer") + "index.jsp?msg=" + s);
}
```

```
List<Produit> lp = Gest.getAllProduit();
```

```
request.setAttribute("lp", lp);
```

```
req.getRequestDispatcher("index.jsp").forward(req, resp);
```

Dao :

```
Public Product getProduitById(int id) {
```

```
    Product p = null;
```

```
    try {
```

```
        String req = "Select * from Product where id like? ";
```

```
        PreparedStatement ps = con. prepareStatement (req);
```

```
        ps.setInt(1, id);
```

```
        ResultSet res = ps.executeQuery();
```

```
        if (res == NULL)
```

```
            return p;
```

```
        while (res.next()) {
```

```
            P = newProduct (res.getInt(1), res.getString(2), res.getDouble(3),
```

```
            res.getInt(4)); } }
```

```
        catch (Exception e) {
```

```
            e.getMessage(); }
```

```
        return P; }
```

Jsp :

```
<form method = "get" action = "Servletajsp">
```

```
<p>
```

```
RefProduct <input type = "text" name = "Ref" value = "<?p.getRef() ?>"
```

```
Designation <input type = "text" name = "Desg">
```

```
Prix <input type = "text" name = "Prix">
```

```
Quantité <input type = "text" name = "Qte">
```

```
<input type = "submit" , name = "ajouter"
```

```
</form>
```



```
Public void deleteProduct (int id) {
```

```
try { String req = "Delete from Product where id = ?";
    PreparedStatement ps = con.prepareStatement(req);
    ps.setInt(1, id);
    ps.executeUpdate();
} catch (Exception e) { e.printStackTrace(); }
}
```

```
Public void updateProduct (Product p) {
```

```
try { String req = "Update products Set Desg = ?, Prix = ?, Ste = ? where Ref = ?";
    PreparedStatement ps = con.prepareStatement(req);
    ps.setString(2, p.getDesg());
    ps.setDouble(3, p.getPrix());
    ps.setInt(4, p.getSte());
    ps.setInt(1, p.getRef());
    ps.executeUpdate();
} catch (Exception e) { e.printStackTrace(); } }
```

Servlet

```
Public class ServletDelete extends HttpServlet {
    GestionProductImp Gest = new GestionProductImp();
```

```
doGet (req, resp) {
    try { int Ref = Integer.parseInt(req.getParameter(Ref));
        if (Gest.getProductById(Ref) != null)
            Gest.deleteProduct(p);
```

```
(else
    out.println("Le produit n'a pas été trouvé.");)
} catch ( ) { }
//Afficher la liste de tous les produits
req.setAttribute("lp", lp);
```



```
req.getRequestDispatcher("index.jsp").forward(req, resp); }
```

```
Public class ServletUpdate extends HttpServlet {
    GestionProduitImp Gest = new GestionProduitImp();
    doGet( req, resp ) {
try {
    List<Produit> Products = (List<Produit>) req.getAttribute("lp");
    for (Produit p: Products) {
        int id = p.getId();
        String msg = request.getParameter("msg");
        double prix = Double.parseDouble(request.getParameter("Prix"));
        int qte = Integer.parseInt(request.getParameter("Qte"));
        Gest.updateProduit(p);
    }
    catch ( ) { }
    List<Produit> p = Gest.getAllProduit();
    req.setAttribute("lp", p);
    req.getRequestDispatcher("index.jsp").forward(req, resp); }
}
```

Stack :

La href = "serv1?"

```
doGet( )
{
    String action = request.getParameter("action");
    if (action.equals("delete"))
    {
        else
    }
}
```

@ Entity

```
Public class Product {
```

@Id

@GeneratedValue (strategy = GenerationType.IDENTITY)

private int id;

private String des;

private double Price;

private int Quantity;

DAO :

```
Public class GestProduct implements IGestProduct {
```

```
* List<Product> getProductByMc (String Mc) {
```

```
Query q = em.createQuery ("select p from Product p where Des = :Mc");  
q.setParameter ("Mc", Mc);
```

```
List<Product> lp = q.getResultList();
```

```
return lp; }
```

```
* List<Product> getAllProduct() {
```

```
Query q = em.createQuery ("select p from Product p");
```

```
List<Product> lp = q.getResultList();
```

```
return lp; }
```

```
* void addProduct (Product p) {
```

```
em.getTransaction().begin();
```

```
em.persist (p);
```

```
em.getTransaction().commit(); }
```

```
* void updateProduct (Product p) {
```

```
Product pp = em.find (Product.class, p);
```

```
em.getTransaction().begin();
```

```
em.merge (pp);
```

```
em.getTransaction().commit(); }
```



```
* void deleteProduit (int id) {
    Produit P = em.find(Produit.class, id produit.getId());
    em.getTransaction().begin();
    em.remove(P);
    em.getTransaction().commit(); }

```

```
* Produit getProduitById (int id) {
Produit P = em.getTransaction().
    Produit P = em.find(Produit.class, id);

    return P; }

```

\* Servlet:

```
Public class ServletP extends HttpServlet { GestionProduit gest = new gestionProduit();
    doGet(— request, — response) {
        List<Produit> lp = gest.getAllProduit();
        request.setAttribute("lp", lp);
        request.getRequestDispatcher("index.jsp").forward(request, resp—);
    }
    doPost(— request, — response) {
        List<Produit> lp = gest.getProduitByMc(String Mc);
        String Mc = request.getParameter("Mc");
        request.setAttribute("lp", lp);
        request.getRequestDispatcher("index.jsp").forward(req, resp); }
}

```

\* JSTL:

```
<form method="post" action="ServletP">
    modèle : <input type="text" name="Mc" value="<%= Mc %>" />
              <input type="submit" value="valider" />

```

```
RefProduit : <input type="text" name="Ref-Produit" />

```

```
Des : <input type="text" name="lp" />
      <input type="text" name="..." /> // Prix Quantité / submit

```



```

</th>
<th> Ref. Produit </th>
<th> Désignation </th>
<th> Prix </th>
<th> Quantité </th>

```

```

<c:foreach items = "{% %}" var = "p" % varStatut = index >

```

```

<tr>
<td> {% p. Ref %} </td>
<td> {% p. Des %} </td>
<td> {% p. Prix %} </td>
<td> {% p. Quant %} </td>
<td> <a href = "supprimer ? id = index " > supprimer </td>
<td> <a href = "seupdate ? id = index " > Edit </td>

```

```

</tr>
</c:foreach >
</table>

```