

## **CONTRÔLE FINAL**

### **DÉVELOPPEMENT N-TIERS J2EE**

#### **Questions de cours (5 pts)**

1. Qu'est-ce qu'une transaction ?
2. Que contient une livraison EAR ?
3. Qu'est-ce qu'un EJB entité ?
4. Que doit-on utiliser pour transmettre une variable de la servlet à la vue ?
5. Dans la méthode `doPost()` d'une servlet, nous avons :

```
HttpSession session = request.getSession();  
session.setAttribute("var1", 10);
```

Expliquez ce que fait ce code.

#### **Problème (15 pts)**

Soit une base de données d'un hôpital appelée « ADMISSON » contenant la table « patient » avec les attributs (id, nom, prenom, date\_naiss, adr).

Et soit le projet J2EE créé dans Eclipse appelé « Exam\_2016\_2017 » contenant le module Web.

1. Créer une classe appelée **DBManager.java** qui permet de gérer la base de données. Cette classe contient deux méthodes :
  - Une méthode de connexion à la base `getConnection()`, on suppose qu'elle existe.
  - Une méthode qui récupère les données de la table Patient `getPatients()`.
2. Créer une servlet de contrôle appelée « liste\_patients\_servlet.java » qui fait le lien entre l'utilisateur et la vue. L'utilisateur accède à la servlet par l'url « /admission\_patient », et la servlet transmet la requête à la vue « liste\_patients.jsp ».
  - a. Donnez le contenu du fichier « web.xml » ;
  - b. Donnez le contenu de la classe « liste\_patients\_servlet.java » (`doGet()` et `doPost()`) ;
3. Dans le fichier « liste\_patients.jsp » de la vue, montrer comment on récupère les données de la table « patient » pour les afficher.

*Bonne chance...*

**NB:** Le corrigé type vous le trouverez sur le site :

<http://www.larbiguezouli.com>

# CORRECTION DU CONTRÔLE FINAL

## DÉVELOPPEMENT N-TIERS J2EE

### Questions de cours (5 pts)

1. Qu'est-ce qu'une transaction ?

Une transaction est définie comme une unité de code indivisible.

2. Que contient une livraison EAR ?

Une livraison EAR englobe tous les modules du projet J2EE : module Web, module client, module EJB, module intégration.

3. Qu'est-ce qu'un EJB entité ?

Un EJB entité représente un objet métier persistant. Il peut être utilisé par plusieurs clients en même temps, puisque il existe en permanence.

4. Que doit-on utiliser pour transmettre une variable de la servlet à la vue ?

Pour transmettre une variable à la vue, on utilise la méthode `setAttribute()` de l'objet `HttpServletRequest` dans la méthode `doGet()`.

5. Dans la méthode `doPost()` d'une servlet, nous avons :

```
HttpSession session = request.getSession();
session.setAttribute("var1", 10);
```

Expliquez ce que fait ce code.

```
HttpSession session = request.getSession();
```

Récupérer l'objet session à partir de la requête.

```
session.setAttribute("var1", 10);
```

Envoyer la valeur 10 à la vue dans une variable appelée "var1".

### Problème (15 pts)

Soit une base de données d'un hôpital appelée « ADMISSION » contenant la table « patient » avec les attributs (id, nom, prenom, date\_naiss, adr).

Et soit le projet J2EE créé dans Eclipse appelé « Exam\_2016\_2017 » contenant le module Web.

1. Créer une classe appelée **DBManager.java** qui permet de gérer la base de données. Cette classe contient deux méthodes :

- Une méthode de connexion à la base `getConnection()`, on suppose qu'elle existe.
- Une méthode qui récupère les données de la table Patient `getPatients()`.

(5.25 pts)

```
public class DBManager { (0.25 pts)
    public static Connection getConnection() {
        Connection con = null;
        try {
            Context context = new InitialContext();
            DataSource ds = (DataSource)
context.lookup("java:jboss/datasources/Admission");
            con = ds.getConnection();
        } catch (NamingException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```

        return con;
    }

    public static ArrayList <Patient> getPatients() { (0.5 pts)
        ArrayList<Patient> patients = new ArrayList <Patient>();
        Connection con = getConnection(); (0.25 pts)
        if (con == null) {
            System.out.println("Problème de connection à la base de données");
        } else {
            try {
                String myRequest = "";
                myRequest = "SELECT * FROM patient"; (0.5 pts)
                // Afficher myRequest
                System.out.println("getPatients : " + myRequest);
                PreparedStatement pStmt = con.prepareStatement(myRequest); (0.25 pts)
                ResultSet rs = pStmt.executeQuery(); (0.25 pts)
                int j = 0;
                while (rs.next()) { (0.25 pts)
                    Patient patient = new Patient(); (0.25 pts)
                    String nom = rs.getString("nom"); (0.25 pts)
                    String prenom = rs.getString("prenom"); (0.25 pts)
                    String date_naiss = rs.getString("date_naiss"); (0.25 pts)
                    String adresse = rs.getString("adr"); (0.25 pts)
                    patient.nom = nom; (0.25 pts)
                    patient.prenom = prenom; (0.25 pts)
                    patient.date_naiss = date_naiss; (0.25 pts)
                    patient.adr = adresse; (0.25 pts)
                    patients.add(j, patient); (0.25 pts)
                    j++;
                }
            } catch (SQLException e) {
                e.printStackTrace();
            } finally {
                if (con != null) {
                    try {
                        con.close();(0.25 pts)
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            }
        }
        return patients; (0.25 pts)
    }
}

```

2. Créer une servlet de contrôle appelée « liste\_patients\_servlet.java » qui fait le lien entre l'utilisateur et la vue. L'utilisateur accède à la servlet par l'url « /admission\_patient », et la servlet transmet la requête à la vue « liste\_patients.jsp ». (4.75 pts)
- Donnez le contenu du fichier « web.xml » ; (2.25 pts)

```

<web-app> (0.25 pts)
  <servlet> (0.25 pts)
    <servlet-name>liste_patients_servlet</servlet-name> (0.25 pts)
    <servlet-class>org.liste_patients_servlet</servlet-class> (0.5 pts)
  </servlet>

```

```

<servlet-mapping> (0.25 pts)
  <servlet-name>liste_patients_servlet</servlet-name> (0.25 pts)
  <url-pattern>/admission_patient</url-pattern> (0.5 pts)
</servlet-mapping>
</web-app>

```

- b. Donnez le contenu de la classe « liste\_patients\_servlet.java » (doGet () et doPost ()) ; (2.5 pts)

```

public class liste_patients_servlet extends HttpServlet { (0.5 pts)
  public static final String VUE = "/WEB-INF/liste_patients.jsp"; (0.25 pts)
  public liste_patients_servlet() {
    super();
  }

  protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException { (0.25 pts)
    /* Traitement de la requête et récupération du bean en résultant */
    ArrayList <Patient> patients = DBManager.getPatients(); (0.5 pts)
    if (patients != null) {
      /* Stockage du bean dans l'objet request */
      request.setAttribute("var1", patients); (0.5 pts)
    }
    this.getServletContext().getRequestDispatcher(VUE).forward(request,
response); (0.25 pts)
  }

  protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException { (0.25 pts)
  }
}

```

3. Dans le fichier « liste\_patients.jsp » de la vue, montrer comment on récupère les données de la table « patient » pour les afficher. (5 pts)

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.Connection" %>
<%@ page import="java.sql.PreparedStatement" %>
<%@ page import="java.sql.ResultSet" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.sql.SQLException" %>
<%@ page import="org.*" %>
<!DOCTYPE html>
<html>
<head>
<title>Liste des patients</title>
</head>

<body>
<h2>Liste des patients</h2>
<table width="800" border="1" cellspacing="0.1"> (0.25 pts)
  <tr> (0.25 pts)
    <th bgcolor="#95D5FF">NOM</th>
    <th bgcolor="#95D5FF">PRENOM</th>
    <th bgcolor="#95D5FF">Date de naissance</th>

```

```

        <th bgcolor="#95D5FF">Adresse</th>
    </tr>
<%
ArrayList <Patient> patients = (ArrayList <Patient>)request.getAttribute("var1"); (0.5 pts)
Patient p = null;
if (patients != null) {
    for (int i=0; i<patients.size(); i++) (0.25 pts)
    {
        p = (Patient)(patients.get(i)); (0.25 pts)
%>
        <tr>
            <th> <%=p.nom%> </th> (0.25 pts)
            <th> <%=p.prenom%> </th> (0.25 pts)
            <th> <%=p.date_nais%> </th> (0.25 pts)
            <th> <%=p.adr%> </th> (0.25 pts)
        </tr>
    }
}%>
</table>
</body>
</html>

```

**Le code source peut être téléchargé à partir du lien suivant :**  
[Controle Final Correction 2016 2017 Modelisation](#)