

Nom, Prénom : _____

Médian JEE

Documents autorisés : Aucun

Durée : 1h30.

QUESTIONS DE COURS

Question 1 : Quelle est la différence entre un serveur Web, un serveur d'applications Web Java et un serveur d'applications JEE ?

(3 points / 5 minutes)

Question 2 : De quoi doit au minimum être constituée une application Web Java pour être prise en charge par un conteneur de Servlets ?

(1 point / 3 minute)

Question 3 : Décrivez en quelques mots comment le conteneur de Servlets prend en charge une Servlet et le cycle de vie de celle-ci.

(4 points / 10 minutes)

Question 4 : Imaginons une servlet invoquée à la suite d'une requête en POST. Si celle-ci effectue un "forward" vers une autre servlet, dans quelle méthode de cette seconde servlet la requête sera t-elle traitée (doGet / doPost) ? Pourquoi ?

Que se serait-il passé avec une redirection plutôt qu'un forward ? Pourquoi ?

(2 point / 5 minutes)

Question 5 : Décrivez la manière dont un serveur d'application Web Java prend en charge le mécanisme de session utilisateur.

(2 points / 5 minutes)

Question 6 : Que signifie être "Thread safe" pour une Servlet ou une page JSP ?

(2 points / 5 minutes)

Question 7 : Quelle est la différence entre la portée « request », la portée « session » et la portée « application » ?

(2 points / 3 minutes)

Question 8 : Dans une page JSP, quelle est la différence entre l'inclusion à la compilation et l'inclusion à la traduction ? Quels sont les impacts lorsque l'on choisit l'une ou l'autre solution ?

(2 points / 3 minutes)

Question 9 : Pourquoi écrire des applications Web selon le design pattern MVC ?

(2 points / 4 minutes)

Question 10 : Pourquoi est-il plus difficile de mettre en oeuvre le concept de DataSource introduit par JDBC 2 dans une application Java traditionnelle que dans une application

Web Java ?

(2 points / 5 minutes)

Question 11 : Que permet la classe `CallableStatement` ?

(2 point / 4 minutes)

Question 12 : Pourquoi modifie-t-on rarement la forme du jeu de résultats et la nature de l'accès au jeu de résultats d'une requête JDBC ?

(2 point / 5 minutes)

Question 13 : Quel est l'intérêt d'introduire une couche d'accès aux données (DAO) dans une application ?

(2 point / 5 minutes)

EXERCICES

Question 1 : La classe suivante a été conçue pour être un JavaBean. Ses propriétés `marque` et `taille` sont prévues pour être "ouvertes" à réutilisation. Corrigez les multiples erreurs de ce code. Expliquez pourquoi il s'agit d'erreurs.

```
package com.jnesis.x.bean;

public class Chaussure implements java.io.Serializable{
    String marque;
    int taille;

    public Chaussure(String marque){
        this.marque = marque;
    }

    public String getmarque() {
        return marque;
    }
    public void setmarque(String marque) {
        this.marque = marque;
    }
    public int gettaille() {
        return taille;
    }
    public void settaille(String taille) {
        this.taille = Integer.parseInt(taille);
    }
}
```

(2 points / 5 minutes)

Question 2 : Expliquer chaque ligne de l'extrait de code suivant :

...

```
<%@page import="tp.Film"%>
<%@page import="tp.Acteur"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>

<c:if test="{param.etape==1}">
<c:choose>
```

```

<c:when test="\${!empty erreur}">
    <p>Erreur de saisie : <c:out value="\${erreur}"/></p>
</c:when>
<c:otherwise>
    <ul>
        <c:forEach var="acteur" items="\${film.acteurs}">
            <li><c:out value="\${acteur.nom}"/></li>
        </c:forEach>
    </ul>
</c:otherwise>
</c:choose>
<c:set var="etape-prec" scope="request" value="\${param.etape}" />
</c:if>

```

...

(3 points / 8 minutes)

Question 3 : La banque X possède un duplicata Excel (.xls) de chacun des relevés de compte de ses clients. Elle souhaite créer une application qui permette aux clients de télécharger ces documents.

Cependant, pour des questions de sécurité, ces documents ne doivent pas être accessibles directement sur Internet.

Après étude du problème, le service informatique propose de placer un serveur applicatif Java entre la zone de stockage des duplicata (sécurisée) et le réseau Internet.

Une servlet **ProxyDuplicataServlet** située dans le package `com.xbank.duplicata` permettra de répondre à toute URL de type `.xls` et d'envoyer dans le flux de réponse le contenu du document voulu. La servlet joue ainsi le rôle de proxy.

Quel va selon vous être le contenu du descripteur de déploiement, pour ce qui concerne la servlet (balises et contenu) ?

(2 points / 5 minutes)

Question 4 : La banque souhaite évaluer l'impact de ce nouveau service sur les clients en effectuant un comptage du nombre de téléchargements.

Pour ce faire, l'application devra insérer en base de données le nom et le numéro de compte de chaque client qui aura utilisé le service.

Malheureusement, les tests de l'application montrent que la méthode qui doit effectuer l'opération ne fonctionne pas et, pire encore, finit par rendre la base de données indisponible.

Ci-dessous le code de la méthode défaillante. Proposez une correction aux multiples erreurs de ce code et indiquez en quoi le code initial est fautif.

```

public void historise(String nomDuClient, long numeroDeCompte){
    Connection con=null;
    try{
        Class.forName("org.gjt.mm.mysql.Driver").newInstance();
        String dataServerURL=new
String("jdbc:mysql://localhost:3306/DUPLICATA");
        Connection con=DriverManager.getConnection(dataServerURL,
"route","tgy23g");
        con.setAutoCommit(true);
    }
}

```

```

        String query="INSERT INTO HISTORIQUE (NUMERO, NOM) VALUES
        (+numeroDeCompte+", "+nomDuClient+" )";
        Statement stmt=con.createStatement();
        stmt.executeQuery(query);
        con.commit();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

```

(3 points / 5 minutes)

Question 5 : Lors d'une requête en lecture sur une base de données via JDBC 1.0, quelles sont parmi les classes listées ci-dessous celles qui sont obligatoirement mises en jeu, celles qui ne le sont jamais et celle qui peuvent l'être. Indiquez en quelques mots à quoi servent chacune de ces classes.

Exemple:

Datasource : Jamais mise en jeu en JDBC 1.0. Cette classe représente une source de données selon la norme JDBC 2.0.

ResultSet :

PreparedStatement :

java.sql.Types :

RequestDispatcher :

(2 points / 5 minutes)