

1. Présentez, de façon claire et concise, chacune des briques suivantes de l'architecture Java EE : JSP, Servlet, EJB, JPA (Pensez à préciser la signification de chaque acronyme)? (2 pts)

JSP = Java Server Pages : Technologie de la plateforme Java EE qui permet de créer des pages de texte contenant des balises (HTML, XHTML,..) et du code Java, et pouvant être interprétées par un browser.

Servlet : Classe Java s'exécutant côté serveur dans un conteneur Web et qui permet de créer dynamiquement des données. Elle sert à traiter et à aiguiller les requêtes http. Elle joue le rôle de contrôleur dans le paradigme MVC.

JPA Java Persistence API est une API qui permet de gérer la persistance des données.

Les instances de ces classes peuvent être "mappées" sur une table de base de données et en plus contenir des méthodes de manipulation de données.

EJB = Enterprise Java Beans, spécifications permettant de développer des composants métiers dans le cadre d'applications distribuées tournant sur la plateforme Java EE. En suivant les normes EJB, ces composants pourront être exécutés sur un serveur d'application.

2. La technologie JSP comporte d'autres composantes, notamment JSTL et EL. Indiquez la signification de chaque acronyme, puis expliquez la raison principale de la création de JSTL et d'EL. (2 pts)

JSTL = Java Standard Tag Library

EL : Expression Library

Réduire le code Java dans les pages JSP.

3. Quels sont les différents types de portée des variables dans les applications Web ? (1 pt)

Ref Support de cours, p 66

4. Qu'est-ce qu'un Java Bean ? Donnez un exemple. (2 pts)

Un Java Bean est une classe java respectant les critères suivants :

a) doit être une classe publique ;

b) doit avoir au moins un constructeur par défaut, public et sans paramètres;

c) peut implémenter l'interface Serializable,

d) ne doit pas avoir de champs publics ;

e) peut définir des propriétés (des champs non publics), qui doivent être accessibles via des méthodes publiques getter et setter.

```
public class MonBean{
    private String proprieteNumero1;
    private int proprieteNumero2;

    public String getProprieteNumero1() {
        return this.proprieteNumero1;}

    public int getProprieteNumero2(){
        return this.proprieteNumero2;}

    public void setProprieteNumero1( String proprieteNumero1 ) {
        this.proprieteNumero1 = proprieteNumero1;}

    public void setProprieteNumero2( int proprieteNumero2 ) {
        this.proprieteNumero2 = proprieteNumero2;}
}
```

5. Soit un bean **Membre** avec un attribut **nom**. Indiquez deux façons différentes de récupérer la valeur de nom, dans une page JSP. (2 pts)

<jsp:getProperty name="Membre" property="nom"/>

<%= membre.getNom() %> (membre étant une instance de Membre)

6. Soit le code HTML suivant (2 pts)

```
<form name="form1">
  <input type="text" value="couleur" name="fond">
  <input type="button" value="changer couleur fond" name="fond2"><br/>
  <input type="text" value="couleur" name="texte">
  <input type="button" value="changer couleur texte" name="texte2">
</form>
```

Lorsqu'on envoie la requête de cette page à une servlet, quelles instructions (code) permettent de récupérer et manipuler les valeurs renseignées par l'utilisateur dans les différents champs?

```
String couleur = request.getParameter("fond");
String texte = request.getParameter("texte");
```

7. Il existe trois façons de créer une servlet. Quelle est celle que l'on utilise le plus souvent? Pourquoi? (1 pt)
On utilise plus fréquemment l'extension de la classe `javax.servlet.http.HttpServlet`, spécialisation des `GenericServlet` car cette méthode facilite grandement l'implémentation du protocole http
8. Les méthodes `doGet` et `doPost` reçoivent les mêmes paramètres en entrée. Lesquels ? (1 pt)
Ref Support de cours, p 51
9. Quels sont les différents types (et sous-types) d'EJB et quel est le rôle de chacun d'eux ? (2 pts)
Ref Support de cours, pp 91 à 100
10. Que pensez-vous de cette affirmation : " 'Classe Entity' est l'autre manière de désigner les Java Beans". (1 pt)
Faux. La mise en œuvre, l'exécution et l'utilisation d'une classe entity suit scrupuleusement les spécifications Java EE et permet, via JPA, de communiquer directement et de façon complète avec un SGBD. Un Java Bean est une classe Java simple (POJO) totalement indépendante des spécifications Java EE.
11. Dans le code suivant, indiquez, s'il y a une erreur (jusqu'à la ligne 12): Dans l'affirmative, expliquez la nature de l'erreur. (2 pts)
- ```
1. @Entity
2. public class Livre {
3. @Id @GeneratedValue
4. private Long id;
5. @Column(nullable = false)
6. private String titre;
7. private Float prix;
8. @Column(length = 2000)
9. private String description;
10. @Column(nullable = false)
11. private String isbn;
12. private Integer nbOfPage;
13. private Boolean illustrations;
14.
15. // Constructeurs, getters, setters
16. }
```
- Il manque l'annotation « @Id » (en rouge) pour définir la clé primaire de la table.*

12. Dans l'exemple suivant d'utilisation de notre classe "**Livre**" (strictement la même classe que la question précédente), il y a-t-il une erreur ? Si oui, corrigez pour que le code s'exécute sans lever d'exception. (2 pts)

```
1. public class Main {
2. public static void main(String[] args) {
3. Livre monLivre = new Livre();
4. monLivre.setTitre("L'intelligence pour les nuls");
5. monLivre.setPrix(12.5);
6. monLivre.setDescription("Développement personnel");
7. em.persist(monLivre);
8. }
```

*Il manque `monLivre.setIsbn("1234566-00")` car lors de la définition de la classe `Livre`, nous avons indiqué que l'attribut "`isbn`" ne pouvait avoir une valeur "null" (`@Column(nullable = false)`)*