

NOM Lefranc
Prénom Thomas
Promo 2015
Date 09/07/2015



1 5 7 5



LEFRANC Thomas
M1 - 2014

MATIÈRE

1°) JSP (Java ^{Servlet} ~~Servlet~~ Pages) : Page qui sert de vue pour une application web, contient généralement du code HTML (0.25)

Servlet : Constructeur d'une application web, on gère les requêtes dans le Servlet. (0.5)

(0.5) EJB (Enterprise Java Bean) : Composant métier réutilisable, une structure de données correspondant à un cahier de charges Java EE précis.

JPA (Java P A) : API permettant la gestion de base de données et persistance d'un service. (0.25)

(1.5) 2°) EL, Expression Language, JSTL (Java ^{Standard} ~~Servlet~~ T ^{of} ~~Language~~ ^{Library}) sont des outils visant à simplifier le code des pages JSP en abolissant le code Java à l'intérieur de celle-ci.

(0.75) 3°) La request, la donnée migre d'une page à une autre.
La session, la donnée est conservée durant toute la durée d'utilisation du service par un utilisateur.

Le contexte : les données sont liées au service lui-même et toujours présentes et disponibles (On peut les voir dans le web.xml)

(2) 4°) Un JavaBean est une classe Java respectant certaines normes Java EE, construction vide, méthodes getter et setter seulement et ce afin de pouvoir être extrait et réutilisé dans n'importe quel environnement Java EE.

On peut représenter un utilisateur par un Java Bean :

```
public class Utilisateur {  
    private String nom
```

```
    public getNom()  
    {  
        return nom;  
    }  
}
```

```
public setNom(String s)  
{  
    this.nom = s;  
}
```

```
public Utilisateur ()  
{  
}
```

```
}
```

60) Il faut alors les récupérer : `request.getParameter("fond")` pour récupérer le texte de la première ~~et~~ ~~suivre~~ ~~en~~ stocker ces données en string :

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

70) La méthode la plus souvent utilisée est l'héritage de la classe

~~HttpServlet~~ `HttpServlet`, car la méthode `service` est déjà définie et il suffit juste d'implémenter quelques méthodes telles que `doGet`, `doPost`, etc.

8°) Les méthodes envoient les objets ^{Servlet} http request et ^{Servlet} http response.

9°) Il existe les Session Bean, stateless ou statefull selon les besoins, qui gèrent les transactions avec l'utilisateur.

Les messages - Driven Bean basé sur le module SMS

Les classes Entity, anciennement Entity Bean, qui gèrent les persistences de données dans les bases de données.

Les Enterprise Bean.

10°) Il est vrai que les classes Entity sont très similaires dans leur déclaration mais ils ne remplissent pas le même rôle, un Java Bean est plus proche d'une simple classe Java tandis qu'une classe Entity est pensée pour la persistance et le RPD.

11°) L'insertion se situe au niveau de l'attribut id, qui doit jouer le rôle de clé primaire pour la base, il faut indiquer qu'il est généré automatiquement.

12°) Le code erreur une erreur à la ligne em. persist (monlora);

En effet, dans la déclaration de notre classe lors, nous avons indiqué concernant l'attribut isbn, @ nullable = false, la colonne ne peut pas être à null, signifiant l'attribut doit obligatoirement avoir une valeur.

Il suffit de rajouter une ligne ~~code~~ attribuant une valeur à isbn monlora - set isbn ("1234"); par exemple.

