

Prénom/Nom :

Applications Web

Examen

Daniel Hagimont

Durée: 1h30, documents autorisés

Questions de cours (7 points)

Cochez la/les bonnes réponses.

Dans une page HTML, on veut écrire un lien "Ajout" qui, lorsqu'on clique dessus, passe un paramètre "operation" ayant pour valeur "ajout". Quelle est alors la balise HTML ?

- ☐ `Ajout`
- ☐ `Ajout`
- ☐ `Ajout`

Dans une page HTML, on a une balise `Ajout`

Cela signifie que :

- ☐ Il s'agit d'un passage de paramètre avec la méthode HTTP POST
- ☐ Il s'agit d'un passage de paramètre avec la méthode HTTP GET
- ☐ La servlet appelée lorsque l'on clique sur le lien est Servlet

Dans une page HTML, on a un formulaire :

```
<form method="POST" action="GET">  
login : <input type="text" name="login"><br>  
password : <input type="text" name="motdepasse"><br>  
<input type="submit" value="OK">  
<input type="hidden" name="password" value="true">  
</form>
```

Cela signifie que :

- ☐ Il s'agit d'un passage de paramètre avec la méthode HTTP POST
- ☐ Il s'agit d'un passage de paramètre avec la méthode HTTP GET
- ☐ Deux chaînes de caractères saisies seront passées dans des paramètres login et motdepasse
- ☐ Il faut soumettre le formulaire si l'utilisateur tape OK au clavier
- ☐ La saisie du paramètre password doit être cachée lors de la saisie

Quand un formulaire HTML avec la méthode POST est validé, les paramètres sont passés

- ☐ Dans l'URL présente dans l'entête de la requête sous la forme `URL?param1=value1¶m2=value2 ...`
- ☐ Dans le corps de la requête sous la forme `param1=value1¶m2=value2 ...`
- ☐ Dans le corps de la requête en format XML
- ☐ Dans le corps de la requête en format JSON

Dans un schéma MVC, la ligne de code suivante est utilisée

`Properties properties = (Properties)request.getAttribute("properties");`

- ☐ Dans la servlet pour récupérer les propriétés (comme l'émetteur ou la date) de la requête request
- ☐ Dans la JSP pour récupérer les propriétés (comme l'émetteur ou la date) de la requête request
- ☐ Dans la JSP pour récupérer un objet passé en paramètre par la servlet

Dans la Facade, la variable EntityManager em;

- ☐ Doit être initialisée avec `em = new EntityManager();`
- ☐ Doit être annotée avec `@EntityManager`
- ☐ Doit être annotée avec `@PersistenceContext`
- ☐ Permet de rechercher des objets à partir de leur clé primaire

Dans la Facade, pour créer un objet persistant dans la BD, on doit

- ☐ Instancier la classe d'un entity et faire `em.persist(p);`
- ☐ Instancier la classe d'un entity et faire `em.store(p);`
- ☐ Instancier la classe d'un entity et mettre devant cette ligne de code l'annotation `@Entity`

Devant la déclaration de la classe Facade, `@Singleton`

- ☐ Indique qu'on doit gérer un unique exemplaire
- ☐ Indique que c'est le container EJB qui doit instancier cette classe
- ☐ Indique que la Facade ne doit pas référencer d'autres objets

`em.createQuery("from Personne where age > 20", Personne.class).getResultList();`

- ☐ Charge les personnes depuis la BD, puis filtre celles qui ont plus de 20 ans
- ☐ Fait une requête SQL sur la BD pour ne charger que les personnes qui ont plus de 20 ans
- ☐ Crée et stocke cette requête dans la base de données
- ☐ Retourne une `Collection<Personne>`

Dans une servlet, pour avoir une référence à la Facade, on peut

- ☐ Déclarer une variable précédée de l'annotation `@Facade`
- ☐ Déclarer une variable précédée de l'annotation `@EJB`
- ☐ Déclarer une variable et instancier la classe Facade

Dans la ligne de code suivante :

`request.getRequestDispatcher(URL).forward(request, response);`

- ☐ URL peut être une page HTML
- ☐ URL peut être une page JSP
- ☐ La ligne peut être suivie d'autres lignes renvoyant du contenu supplémentaire
- ☐ Génère une servlet dont l'exécution génère la page retournée

Dans un schéma MVC, lors du passage d'un paramètre name entre une page HTML et une Servlet, on récupère la valeur du paramètre avec

- ☐ `String name = request.getAttribute("name");`
- ☐ `String name = request.getParameter("name");`
- ☐ `String name = request.getName("name");`

Si un entity bean est retourné par la Facade à la Servlet

- ☐ On dit que le bean est détaché
- ☐ Le bean est retiré de la BD
- ☒ Si le bean est repassé par la servlet à la Facade, la Facade peut synchroniser les beans avec `em.merge()`;

Dans un schéma MVC, le contrôleur effectue

- ☐ Des appels à la Facade
- ☐ Des appels à des entity
- ☐ Des appels à des JSP

Dans un schéma MVC, une JSP

- ☐ Est appelée depuis la Facade
- ☐ Génère la page retournée au client
- ☐ Reçoit en paramètre de la servlet les données à afficher
- ☐ Effectue des appels à la BD pour récupérer les données à afficher

Problème (13 points)

On utilise les EJB vu en cours.

Une application est composée de 3 entity beans

L'entity Ouvrier:

- possède une clé primaire : `int id`;
- possède 1 champs : `String nom`;

L'entity Entreprise:

- possède une clé primaire : `int id`;
- possède 1 champs : `String numero`;

L'entity Chantier:

- possède une clé primaire : `int id`;
- possède 1 champs : `String adresse`;

Une entreprise peut avoir plusieurs chantiers et un chantier est réalisé par une seule entreprise.

Un chantier peut nécessiter plusieurs ouvriers et un ouvrier ne travaille que sur un seul chantier.

Ces relations sont toutes bi-directionnelles.

On ne donnera pas les setter/getters ni les imports.

Question 1 : Donner l'implantation de ces entity (3 points)

Question 2 : donnez une possible instantiation en BD (précisez les PrimaryKeys et ForeignKeys de chaque table) (2 points)

Question 3 : Donner le code des deux méthodes de la Facade suivantes (4 points)

```
void ajouter_ouvrier(String nom, int id_chantier);  
// ajouter un nouvel ouvrier à un chantier existant
```

```
void changer_entreprise(int id_chantier, int id_entreprise);  
// change l'entreprise opérant sur un chantier
```

Question 4 : on veut maintenant qu'un chantier puisse être réalisé par plusieurs entreprises et qu'un ouvrier puisse travailler sur plusieurs chantiers.

Les relations de l'entity Chantier sont les suivantes :

```
@Entity
public class Chantier {
    @ManyToMany(mappedBy="chantiers")
    Collection<Entreprise> entreprises;
    @ManyToMany(mappedBy="chantiers")
    Collection<Ouvrier> ouvriers;
}
```

Donner la définition des deux autres entity et donner la nouvelle version de la méthode ajouter_ouvrier() de la Question 3 (2 points)

Question 5 : On veut afficher la liste des entreprises dans une JSP. Donner l'implantation de la méthode de la Facade permettant de récupérer cette liste (1 point)

```
Collection<Entreprise> getEntreprises();
```

Question 6 : Si dans la JSP, on veut pouvoir afficher pour chaque entreprise la liste de ses chantiers, que faut il ajouter pour être sûr que les données soient accessibles dans la JSP ? (1 point)