



EXAMEN JEE

Automne 2015

Sébastien Rufiange, Besma Zeddini
ING-2, Génie Informatique

Durée de l'examen : 2h

Conditions d'examen :

- L'accès à une machine est permis seulement dans le cadre de cet examen.
- L'accès à Internet n'est pas toléré. AREL demeure accessible.
- Les échanges et communications sous toute forme sont interdits.
- Documents électroniques non acceptés.
- Documentation papier non acceptée.
- Clés USB non acceptées.

Remise de votre travail :

Vous pouvez vous servir des logiciels à votre disposition (eclipse, idea, gedit,...) afin de réaliser cet examen. Essayez de faire le moins d'erreurs possibles ; en cas de doute, commentez afin de préciser vos intentions même si le code ne compile pas ou ne fonctionne pas. Assurez-vous que votre travail est remis dans le dossier de rendu avant de quitter la salle.

1. Application de Login (7 points) [Servlet, JSP]

Concevez une appli JEE pour pouvoir effectuer un login utilisateur à l'aide d'une Servlet.

- Lorsque la Servlet (LoginServlet.java) est accédée directement (URL = /LoginServlet), sans paramètres, cette servlet doit rediriger l'utilisateur vers la page login.jsp.
- La page de login (login.jsp) doit permettre de saisir un nom d'utilisateur, un mot de passe côté client via un formulaire HTML.
- Une fois le formulaire soumis, la servlet doit valider les données et effectuer les opérations ci-dessous (LoginServlet.java) :

(1) Rediriger l'utilisateur vers une page JSP nommée admin.jsp si le login et le mot de passe sont "admin". Cette page admin.jsp doit simplement afficher "Menu admin".

(2) Rediriger vers une page d'accueil (user.jsp) si le login est adéquat ; cette page user.jsp doit simplement afficher "Menu utilisateur". Supposez qu'une fonction checkLogin(String user, String pass) existe et qu'elle retourne VRAI en cas de bon login.

(3) Dans les autres cas, afficher un message d'erreur ("Login incorrect").

Instructions : Utilisez un RequestDispatcher pour rediriger l'utilisateur aux bonnes adresses. Utilisez les notions de Servlet et JSP afin de répondre à cette question. Placez tous les fichiers de cette question dans un dossier nommé **Q1-Login**.

Mise en situation : Votre soeur et vous avez accumulé une grande quantité d'objets au fil du temps et souhaitez vous en débarrasser. Vous vous êtes proposés pour développer un petit site transactionnel en JEE afin de faciliter la vente de ces divers objets. Dans ce cadre, un membre de votre famille ou un ami peuvent proposer la vente de leurs biens.

2. Échange de Beans (6 points) [Java Bean, Scope]

- a. Proposez un Java Bean pour modéliser un panier d'achat (PanierBean.java). Un panier peut comprendre un certain nombre d'items achetés (Item.java). Chaque item comprend un nom, une description, un état (mauvais, passable, bon, excellent) et un prix (en euros ou zéro pour un don).
- b. Expliquez clairement dans les commentaires du fichier PanierBean.java pourquoi il s'agit bien d'un Java Bean.
- c. Créez une servlet AchatPanierServlet.java. Lorsqu'un visiteur se rend à la page /AchatPanier?nom=PRODUIT&desc=DESCRIPTION, une servlet doit ajouter un item au panier (à enregistrer dans le scope Session) et rediriger vers une vue panier.jsp.
- d. La page panier.jsp doit afficher les noms des items présents dans le panier via une liste HTML.

Instructions : Vous devez utiliser les notions de Java Bean et Scope afin de répondre à cette question. Placez tous les fichiers de cette question dans un dossier **Q2-Bean**.

3. Panier en Hibernation (4 points) [Hibernate]

Vous souhaitez que le panier puisse être sauvegardé dans une base de données. Vous décidez donc de créer une nouvelle classe ItemPersistant.java et d'utiliser Hibernate. Pour cette question, vous devez fournir (au choix) un fichier de mapping xml (ItemPersistant.hbm.xml) **ou bien** annoter et commenter votre classe ItemPersistant.java. Négligez les autres détails, comme la configuration avec la base de données.

Instructions : Votre solution doit être cohérente avec la mise en situation. Vous devez utiliser les notions d'Hibernate afin de répondre à cette question.

Pour simplifier, vous n'avez seulement qu'à modéliser un item de panier (classe ItemPersistant), sachant qu'en pratique vous auriez aussi à faire le mapping pour PanierPersistant. Uniquement la solution pour ItemPersistant est demandée ici.

Placez tous les fichiers de cette question dans un dossier nommé **Q3-Hibernate**.

4. Sonic (3 points) [Spring]

Le site prend de l'ampleur et afin de gagner en productivité, vous décidez d'utiliser le Framework JEE Spring. Ajoutez un contrôleur (PanierController.java) supportant les requêtes ci-dessous.

Note : Supposez pour cette question que votre panier (Objet de type PanierBean) existe et est chargé dans une variable nommée "PanierBean" au niveau de la Session. Ajoutez au besoin des méthodes à cette classe afin d'ajouter, de retirer et de lister les items du panier.

`/PanierController/ajouterItem/<nom du produit>/<description>/<état>/<prix>`

Cette requête doit créer, s'il n'existe pas déjà, l'item correspondant (en fonction des paramètres dans l'URL) puis l'ajouter au panier.

Instructions : Votre solution doit être cohérente avec la mise en situation. Vous devez utiliser les notions de Spring et de MVC afin de répondre à cette question. Négligez les détails de configuration Spring à part les mappings et autres éléments demandés.

Placez tous les fichiers de cette question dans un dossier nommé **Q4-Spring**.

FIN DE L'EXAMEN