



Développement WEB : JEE

TP 5 : Les JavaBeans

Exercice 1 : Utilisation de la JSTL

- 1- Développer une page JSP en utilisant la JSTL, qui affiche la liste des nombres de 2 à 30 en indiquant pour chaque nombre s'il est premier ou pas.
- 2- Développer une page JSP qui prend en paramètre un nombre et calcule s'il est premier ou non.
- 3- Reprendre (1) en remplaçant les valeurs écrites en dur dans le code (2 et 30) par des variables entrées par l'utilisateur.

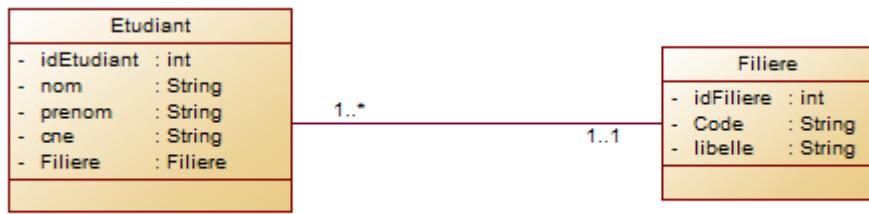
Exercice 2 : Utilisation des JavaBeans

Dans cet exercice, nous allons faire une authentification avec un login et un mot de passe en utilisant un JavaBean représenté par la classe **User**.

- 1- Créer la classe User (login, password).
- 2- Créer un formulaire d'authentification
- 3- Au clic sur le bouton "Envoyer" les données seront transmises vers la page "load.jsp", dans cette page créer un objet user et enregistrer le dans une session.
- 4- Créer une page web "affiche.jsp" permettant d'afficher le login et le mot de passe que l'utilisateur a saisis.

Exercice 3 : Architecture JEE multi-tiers (Séparation des couches)

On souhaite créer une application pour la gestion des filières et des étudiants, pour ce faire nous proposons le diagramme de classe suivant :



Gestion des Filières

Ajout d'une Filière

ID	<input type="text"/>
Code	<input type="text"/>
Libellé	<input type="text"/>
<input type="button" value="Ajouter"/>	

ID	Code	Libelle	Modifier	Supprimer
2	GINF	DCA GINFORMATIQUE		
3	GE	DCA G.ELECTRIQUE		
14	GM	DCA G.MECANIQUE		

- 1- Développer les couches : **entités**, accès aux données **dao**, **service**, **servlets** et **beans** :
- Créer la base de données SMI2020 sous MySQL
 - Créer les entités Filiere et Etudiant (*package entities*)
 - Créer la classe ConnectionTool permettant de établir une connexion *statique* vers la base de données et qui contient un getter (*statique*) permettant de retourner cette connexion (*package dao*)
 - Créer les interfaces FiliereDAO et EtudiantDAO héritant de l'interface *Idao<T>* et contenant les méthodes des CRUD (create, update, delete, getById et getAll) (*package dao*)
 - Créer les classes FiliereDAOImpl et EtudiantDAOImpl implémentant les interfaces FiliereDAO et EtudiantDAO (*package dao.impl*)
 - Créer les classes FiliereService et EtudiantService contenant les traitements métier des entités Filiere et Etudiant (*package service*)
 - Créer les classes FiliereBean et EtudiantBean contenant toutes les méthodes nécessaires pour la gestion des formulaires Filiere et Etudiant ainsi que les tables d'affichage des données (*package beans*)

h. Créer toutes les servlets nécessaires pour le traitement des formulaires Filiere et Etudiant ainsi que les tables d'affichage des données (*package servlets*) :

- i. Configurer en parallèle le descripteur de déploiement web.xml (ou bien utiliser les annotations @WebServlet)

2- Développer la couche présentation :

a. Créer les pages filiere.jsp et etudiant.jsp permettant de saisir/modifier/supprimer/afficher toutes les informations des filières et des étudiants. Pour des raisons de sécurité :

- i. Il faut créer ces pages dans le **dossier privé WEB-INF**. Ceci permet d'empêcher l'accès aux pages via URL
 - ii. Tous les accès à ces pages doit se faire par des redirections *coté serveur* (RequestDispatcher) au niveau des servlets
- b. Créer la page home.jsp permettant de réaliser un menu de navigation entre les deux pages filiere.jsp et etudiant.jsp
- c. Il faut ajouter, bien entendu, la page index.jsp au niveau du dossier public WebContent. Elle sera la page d'accueil de l'application et fera les redirections nécessaires pour accéder à l'application suite à une authentification.