

Ateliers java - séance 9

Objectifs :

- Comprendre l'introspection en Java.
- Utiliser la réflexivité pour comprendre la structure d'une classe.

Thèmes abordés :

- introspection



La société **ABBA** souhaite fournir une application qui propose de fournir un diagramme UML lorsqu'on soumet une classe (bytecode).

À la première connexion, la page suivante s'affiche :



Lorsqu'on soumet une classe, son uml est proposé :



Pour le moment, ni l'identité ni les propriétés ne s'affichent ; cela sera votre job de cette semaine !
Au finish vous devrez avoir comme résultat :



Le projet a déjà été initié en java EE. Les parties servlets et JSP sont opérationnelles.

Installation

1. Importez le projet AJ_2017_seance_9 dans votre workspace (via import existing project).
2. Créez un répertoire **fichiers** dans le Z (de telle sorte que le chemin **Z:/fichiers** soit opérationnel)
3. Il faut ensuite démarrer votre serveur (flèche verte) comme un projet « normal ». Une classe Main existe dans le source folder.
4. Constatez ensuite dans votre browser préféré (**Firefox** only !) l'affichage de l'application lorsque vous accédez à l'url : <http://localhost:8080/>

Compréhension

Comprenez bien la structure dans laquelle vous allez devoir travailler. Ouvrez votre projet Eclipse.

Dans le dossier **web** de votre projet se trouve une seule page **index.jsp**.

index.jsp contient 2 articles :

- le premier contient un formulaire permettant d'introduire un fichier. Ce formulaire envoie ensuite l'information à **soumettre.html**.
- le second s'occupe de l'affichage de l'UML du fichier. Il renferme un certain nombre d'expressions qui permettent l'affichage des informations transmises. Observez les noms d'**attributs** utilisés car ce sont ceux que vous devrez transmettre !

Lorsque le formulaire est soumis, l'action de celui-ci est **soumettre.html**. Pour comprendre à quoi réfère celui-ci, votre instinct vous guide vers le descripteur de déploiement (**web.xml**) qui vous renseigne qu'il s'agit de l'url de votre servlet **Controleur**. Ce nom de servlet correspond à votre classe **gimme.servlets.Controleur.class**.

Constatez dans le **web.xml** que les **inputs** de type « **multipart** » sont stockés dans un endroit déterminé (**Z:/fichiers**). C'est dans ce répertoire que seront chargés les fichiers uploadés. Il faut que ce répertoire existe (créez-le ou alors changez-en la valeur).

Dans les **src** du projet, au sein du répertoire **gimme.servlets** vous retrouvez bien entendu la servlet **Controleur** mais aussi une classe **MyClassLoader** (spécialisation de **ClassLoader**) qui permet de redéfinir le **findClass** pour le fichier téléchargé.

Dans les **src** du projet, au sein du répertoire **gimme.domaine** vous trouvez deux classes (**Propriete** et **Visibilite**) qui sont les attributs que l'on va placer dans la requête (**request**) afin de permettre l'affichage dans la jsp.

Enfin, il y a une classe **Test** qui, comme son nom l'indique, permettra de tester les différents affichages UML.

Implémentation

Votre travail dans ce projet consiste à récupérer les informations sur les diverses propriétés de la classe soumise **par réflexivité**. Ces informations doivent être stockées dans les types fournis dans le domaine afin que la page jsp puisse s'afficher correctement.

Complétez le **Controlleur.java** et traitez dans l'ordre les éléments suivants :

- ☐ l'identité de la classe
- ☐ les attributs de la classe
- ☐ les méthodes de la classe
- ☐ les constructeurs de la classe
- ☐ les exceptions lancées éventuelles

En bonus

- ☐ l'ordonnancement des méthodes (constructeurs, getter, setter, méthodes surchargées d'Object puis autres par ordre alphabétique)
- ☐ les types génériques des collections
- ☐ les valeurs par défaut éventuelles