SPRING

Objectif : Réduire les dépendances fortes, faire en sorte que le code soit indépendant des futurs classes d’implémentation

Spring :   
1) Config : jar dans lib. Eventuellement SpringIDE in Eclipse (donnera : click droit, add « Spring Nature »)

2) XML : applicationContext.xml (sinon Annotations, et dernièrement Java Config)

3) API Java Spring : ApplicationContext.

Objectifs TP :

1. ID
2. Prog par aspect
3. Config avec Hibernate (faz cuidado caralho !)

(Note : Lors du lancement de l’appli, ou pendant la soutenance, drop database. Création du lancement, hibernate crée, etc …. Et 2-3 contacts doivent être automatiquement (ou un bouton « peupler la base ») inséré, en utilisant Spring.  
Note 2 : si un bean dépend d’un autre, qui dépend d’un autre, etc … Pas de problème avec contrscuteurs, mais setter va merder. Il faut mettre un sens dans le xml.)

Ajouter élément « init-method » ou impléments InitializeBean pour faire une action après la création.

Dans les servlets : ApplicationContext context = WebApp

**Programmation par Aspect (AOP)**

Mots-clefs :   
 - Target : la classe sur laquelle on va se baser  
 - Advice : La classe qui va se tisser sur le target (la « copie »)  
 - JoinPoint : Point d’execution du programme (Constructeur, method invoc, etc…)  
 -Pointcut : « Si j’ai ça, alors je lance ce pointcut, sinon methode normale »

Si un object est tissé par un aspect, alors getBean risque de rendre pas l’objet XX mais ProxyXX, donc on ne pourra pas insérer l’objet dans la base.

-----------------------------------------------------------------------------------------------------------------------------------

Question de cours :  
- Qu’est-ce qu’un conteneur lourd ? Spring légé, EJB lourd.  
1) Spring léger, pas besoin d’un serveur d’application (tomcat). EJB non (JBOSS, Websphere,…)  
2) Dans spring, nos classes sont de simples POJO.

* Schéma (à retenir) :

Servlet -> Service -> DAO -> Hibernate

S P R I N G  
(en général, le Service appel Spring)

* Interaction avec le conteneur : Possible de modifier dynamiquement le contenu du fichier xml ? OUI !
* A quoi correspond le parametre pjp (ProccedingJoinPoint) : permet de revenir à la méthode initialement appelée.