|  |  |
| --- | --- |
| **Contrôle Continue**  Semestre : 1  Session : Principale Rattrapage |  |
| Module :Framework Spring  Enseignant(s) :Ben Khelifa Seddiki Rabeb daly  Classe(s) :5eme INFO M2 | |
| Documents autorisés  : OUI NON Nombre de pages : 5 | |
| Date :11/10/2012 Heure: 9H Durée :10 mn | |

**Partie A :QCM**

Cochez les bonnes réponses :



***Question1 :***

Un conteneur léger sert à contenir un ensemble d’objets instanciés et initialisés, formant un contexte initial (ou une hiérarchie de contextes) pour une application. Ce contexte initial est construit :

-**Réponse 1** : Par les objets eux-mêmes par programmation.

- **Réponse 2 :** A partir d’une description externe (xml)

***Question2 :***

Spring est considéré comme un conteneur dit « lourd », c'est-à-dire une infrastructure similaire à un conteneur EJB

**-Reponse1** :Vrai.

**-Reponse2** :Faux.

***Question3 :***

L’architecture de spring est basé sur l’approche par composant, elle encourage :

**-Reponse1 :** les bonnes pratiques de programmation par interface

**-Reponse2 :** les bonnes pratiques de programmation a faible couplage.

***Question4 :***

Le cycle de vie du conteneur Spring se compose des phases suivantes :

**-Réponse1**: Phases d'initialisation, d'utilisation et de destruction d'un contexte Spring

**-Réponse2** : Initialiser, utiliser, et détruire les phases

**-Réponse3**: Travailler avec les post-processeurs

**-Reponse4**: Externaliser des paramètres de configuration dans des fichiers propriétés.

**-Reponse5 :** Notion de portée (bean scope).

***Question 5 :***

Il existe dans Spring plusieurs implémentations de conteneurs :

- **Réponse1:** ClassPathXMLApplicationContext.

**-Réponse2**: FileSystemXMLApplicationContext.

-**Reponse3**: XMLBeanFactory.

-**Reponse4** :SessionFactory.

***Question 6:***

L’IOC entre deux Objets A et B s’implémente comme suit :

- **Reponse1**: Lorsqu’un objet A a besoin d’un objet B, on délègue à un objet C la

mise en relation de A avec B

- **Réponse2 :** Lorsqu’un objet A a besoin d’un objet B, on injecte l’instance de B dans A

***Question 7:***

Lors de l’injection si le nombre de dépendances est important ou si certaines sont optionnelles on utilise :

**-Reponse1 :** L’injection par constructeur

**-Réponse2 :** L’injection par setter.

***Question 8:***

Via les concepts de classes et d’interfaces (mécanismes d’héritage et de polymorphisme), la POO fournit:

- **Réponse1:** une bonne séparation des fonctionnalités verticales (métier)

-**Réponse2:** une bonne séparation des fonctionnalités transversales (techniques).

***Question 9:***

Parmi les Aspect techniques transverses on cite :

- **Réponse1:** Journalisation logs applicative, persistance, transaction, sécurité.

-**Réponse2**: Logique : Lecture du solde, débit ou crédit, demande d’une carte.

**-Réponse3 :** Domaine : Compte bancaire, chéquier, cartes.

***Question 10:***

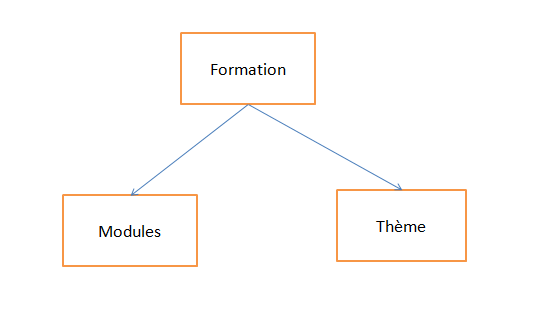
La programmation Orientée Aspect (POA) permet de modulariser une 'préoccupation transversale' qui assure les fonctionnalités suivantes :

- **Réponse1 :** Éliminer toute duplication de code

- **Réponse2 :** Centraliser les séquences d’instructions répétitives.

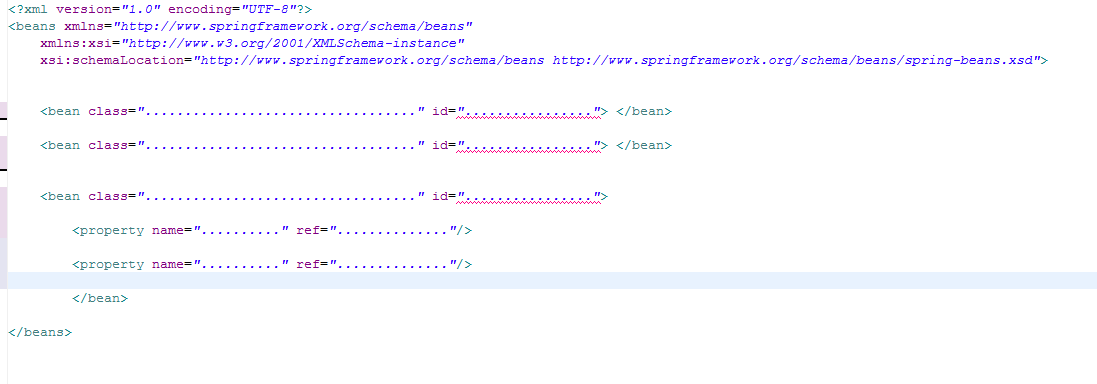
- **Réponse3 :** Réduire le besoin d’omniscience.

**Partie B :**

****

On se propose d’implémenter le diagramme ci dessus relatif à la « **gestion d’une formation** ».L’injection des beans du diagramme au niveau du fichier **applicationContext.xml** de spring est comme suit, Complétez le fichier :

NB : Toutes les classes java(Formation,Modules,Theme) sont créés sous le package **com.esprit.spring.gestionFormation**.



***Bon Travail…***