***NB : 0 ou plusieurs réponses peuvent être correcte***

***Question1 :***

JEE désigne

Un ensemble de spécifications

Un framework

Un langage

***Question2 :***

Les types d’EJB pour la version 3.x sont

Entity

Message Driven Bean – MDB

Session Bean

Stateless session Bean et Stateful Session Bean

***Question3 :***

Les types de session bean sont :

Entity

Singleton

Stateless

MDB (message driven bean)

Stateful

***Question4 :***

Parmi les serveurs d’application certifiés JEE6 :

JBOSS 5.x

JBOSS 7.x

Tomcat 6.x

Glassfish

WebSphere

***Question5 :***

ORM désigne

Object Relational Model

Object Relational Messaging

Object Remote Method

***Question6 :***

Indiquez les ORM possible parmi cette liste

JPA

Toplink

JDBC

Hibernate

***Question7 :***

L’ORM intégré par défaut avec JBOSS est

JPA

Toplink

JDBC

Hibernate

***Question8 :***

Quelle(s) relation(s)existe (ent) entre JPA et Hibernate

JPA utilise Hibernate

Hibernate implémente JPA

Aucune relation

***Question9 :***

Quel est l’ordre correct pour la définition de ces niveaux d’abstraction

Driver, DB, ORM, JDBC, JPA

DB, Driver, ORM, JDBC, JPA

DB, JPA, ORM, JDBC, Driver

NB : BD: Base De Données

***Question10 :***

JPA désigne

Java Persistence Annotations

Java Project Application

Java Persistence API

***Question11 :***

La configuration de la data source s’effectue au niveau du :

Fichier persistence.xml

Fichier standalone.xml

***Question12 :***

Le rôle de l’entity manager

Assure le mapping objet relationnelle

Gère les entités au niveau du contexte de persistance

Gère les entités au niveau de la base de données

***Question13 :***

On peut déclarer un EJB session bean

Avec état avec l’annotation @Stateful

Sans état avec l’annotation @Singleton

Sans état avec l’annotation @Stateless

***Question14 :***

Le type de session bean qui garantie le fait d’avoir qu’une seule instance est

Entity

Singleton

Stateless

MDB

Stateful

***Question15 :***

Pour invoquer un session bean depuis un client distant on utilise

Une interface Local ou une interface Remote

Une interface Local et une interface Remote

Une interface Remote

Aucune interface

***Question16 :***

Le rôle du fichier ***jndi.properties***

La configuration du data source

Fournir des informations concernant l’adresse IP et le port RMI du serveur

La configuration de l’entity manager

***Question17 :***

Pour créer un EJB3 session bean sans état puis l’invoquer depuis un client distant, on a besoin de

Déclarer une interface annotée par @Local

Déclarer une interface annotée par @Remote

Déclarer deux interface annotées par @local et @Remote

Déclarer une classe d’implémentation annotée par @Session

Déclarer une classe d’implémentation annotée par @SessionBean

Déclarer une classe d’implémentation annotée par @Stateless

Déclarer une classe d’implémentation annotée par @Stateful

***Question18 :***

Le JNDI binding permet de

Récupérer un objet à partir d’un nom unique

Associer un nom unique à un objet

Invoquer un session bean à distance

***Question19 :***

Le JNDI lookup permet de

Renvoyer une instance de la classe d’implémentation du session bean

Renvoyer une instance de l’interface distante

Récupérer une image de la classe d’implémentation du session bean dans le serveur

Récupérer une image de la classe d’implémentation du session bean dans l’application cliente

***Question20 :***

Quelle sont les annotations minimales pour définir une entité JPA

@Table

@Entity

@PersistenceContext

@Id

@Column

@GeneratedValue

***Question21 :***

Ce schéma modélise une relation bidirectionnelle de type un à plusieurs entre les deux entités A et B

***Question21-A :***

L’attribut du lien dans la classe A est de type

Simple

Collection

***Question21-B :***

L’annotation qui doit précéder l’attribut de lien déclaré dans la classe A est

@OneToMany

@ManyToOne

@OneToMany(mappedBy= ‘’a’’ )

***Question21-C :***

Du côté de la base de données

Une clé étrangère va être générer dans la table A

Une clé étrangère va être générer dans la table B

Pas de clé étrangère

***Question22 :***

La méthode entityManager.persist(a) permet d’ajouter un enregistrement dans la base de données

Vraie

Faux (il ajoute dans la persistence)

***Question23 :***

Complétez ce code pour assurer l’implémentions de la relation un à plusieurs bidirectionnelle de la classe A vers la classe B (comme indiquer dans la question 23)



