

Examen Final (CAGL M2 IL - 22/01/2024)

Questions :

1. Donner le nom d'un serveur d'application, d'un serveur web et d'un serveur de base de données.

| | | |
|-----------------------------|---------------|----------------------------|
| WildFly ou bien JBoss | Apache Tomcat | MySQL ou bien Oracle |
|-----------------------------|---------------|----------------------------|

2. Donner les types des composants EJB3 session avec les annotations associées.

| |
|---|
| EJB3 Session sans état (@stateless) et EJB3 Session avec état (@stateful) |
|---|

3. Généralement le code d'une application peut être regroupé dans deux catégories. Lesquelles ?

| | |
|------------------------------|----------------------------------|
| Les traitements fonctionnels | Les traitements non-fonctionnels |
|------------------------------|----------------------------------|

4. Expliquer la différence entre la Servlet et JSP.

| |
|---|
| JSP= Java injecté dans le Html, Servlet= Html injecté dans le code Java |
|---|

5. Donner trois méthodes de la classe EntityManager.

| | | |
|-----------|--------|---------------|
| persist() | find() | createQuery() |
|-----------|--------|---------------|

6. Dans une communication RMI, le programme client a besoin de trois informations pour établir une communication avec un programme serveur. Quelles sont ces informations.

| | | |
|------------|------|---|
| Adresse IP | Port | Nom de la méthode ou bien références, |
|------------|------|---|

7. Citer les éléments qui permettent de créer un composant EJB3 Entity.

| | | | |
|----------|---------|-----------|-----------------------------------|
| L'entité | package | interface | classe qui implémente l'interface |
|----------|---------|-----------|-----------------------------------|

8. Donner trois méthodes bloquantes.

| | | |
|-----------|--------------|----------|
| receive() | readObject() | accept() |
|-----------|--------------|----------|

9. Donner les paramètres du constructeur de la classe DatagrammePacket.

| | | | |
|------------------|----------------------|------------|------|
| Tableau de byte, | la taille du tableau | Adresse IP | port |
|------------------|----------------------|------------|------|

10. Donner les versions de la technologie JAVA

| | | |
|------|------|------|
| J2SE | J2EE | J2ME |
|------|------|------|

11. Quel est le rôle de la méthode trim()

| |
|--|
| Supprimer les blanc au début et à la fin d'une chaine de caractères. |
|--|

12. Quelle sont les annotations utilisées pour: (i) déclarer une propriété d'une classe comme clé primaire, (ii) changer le nom de la colonne au niveau de la base des données, (iii) définir une EJB3 entité.

| | | |
|-----|---------|---------|
| @Id | @Column | @Entity |
|-----|---------|---------|

13. Donner les types de transformation des modelés.

| | |
|------------------------|-------------------------|
| Transformation exogène | Transformation endogène |
|------------------------|-------------------------|

14. Quels sont les principaux niveaux de modèles définit par MDA

| | |
|----------------------------------|-------------------------------|
| PIM : Platform Independent Model | PSM : Platform Specific Model |
|----------------------------------|-------------------------------|

15. Donner les concepts de base d'une architecture logicielle à base de composants.

| | | | |
|-----------|------------|---------------|------|
| Composant | Connecteur | Configuration | Port |
|-----------|------------|---------------|------|

16. Donner trois concepts utilisés dans la programmation orientée aspect.

| | |
|-------------------------------|--|
| Point de jonction (joinpoint) | Point d'action (pointcut) ou bien Greffon (advice) |
|-------------------------------|--|

17. Donner un exemple d'une transformation des modèles (préciser le modèle source et le modèle cible).

| |
|--|
| modèle orienté objet -> modèle orienté composant (JavaBean -> Entreprise Java Bean). |
|--|

18. Donner deux exemples des utilisations possibles d'AOP.

| | |
|---------------------------|------------|
| La gestion des exceptions | Le logging |
|---------------------------|------------|

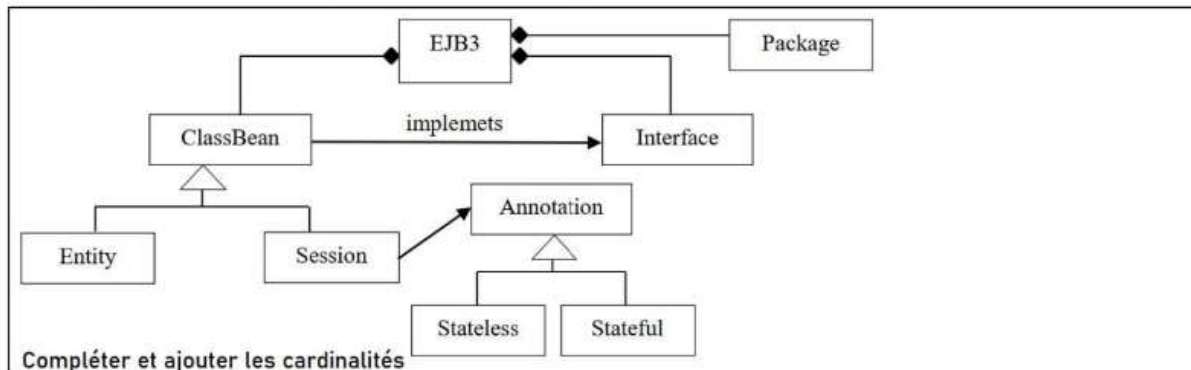
19. Donner les principes de la POA

| | |
|----------------|---------------|
| Quantification | Obliviousness |
|----------------|---------------|

20. Dans le développement logiciel, l'entrecroisement de concepts est relié à deux symptômes. Lesquelles ?

| | |
|---------------------------|---------------------------|
| Dispersion de code source | Croisement de code source |
|---------------------------|---------------------------|

21. Proposer un méta modèle pour le modèle des composants EJB3.

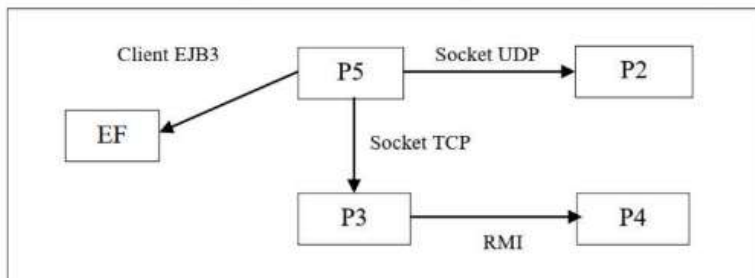


22. Donner deux techniques d'adaptation des applications.

| | |
|----------------|---|
| La réflexivité | Les langages de programmation par aspects |
|----------------|---|

23. Compléter les tableaux ci-dessous (ordre d'exécution : EF, P2, P3, P4, P5)

| | | | | |
|-------------------------|-----------------------|------------------------|---------------|-----------------------------|
| P2 (Affichage) M2RSD | P3(Affichage) M2IL | P4(Affichage) M2SSI | P5(Affichage) | EF (EJB3) (Affichage) 24 |
|-------------------------|-----------------------|------------------------|---------------|-----------------------------|



| Programme source | Message(s) envoyé(s) | Programme destination |
|------------------------|----------------------|-----------------------|
| P5 | M2IL | P3 |
| P5 | M2RSD | P2 |
| P5 | 12, 12 | EF (EJB3) |
| P3 | M2SSI | P4 |
| Type de l'architecture | | Distribuée |
| Programme initiateur | | P5 |

| | |
|---|---|
| <pre> public class P5 { public static void main(String[] args) { try {Properties props = new Properties(); props.put("java.naming.factory.url.pkgs", "org.jboss.ejb.client.naming"); props.put("jboss.naming.client.ejb.context", true); InitialContext context = new InitialContext(props); DatagramSocket c1 = new DatagramSocket(); IEF v = (IEF)context.lookup("ejb:/EF/EFBean!Pack.IEF?stateful"); Socket c = new Socket("localhost", 2004); ObjectOutputStream out = new ObjectOutputStream(c.getOutputStream()); out.writeObject("M2IL"); v.setA(12); v.setB(12);v.m(); String str = "M2RSD"; byte[] T = new byte[30]; T=str.getBytes(); DatagramPacket p = new DatagramPacket(T,T.length, InetAddress.getByAddress("localhost"), 9876); c1.send(p); c.close();c1.close();out.close(); }catch(Exception e) {System.out.println(e.toString());}} </pre> | |
| <pre> public class P3 { public static void main(String[] args) { try { ServerSocket s = new ServerSocket(2004); Socket c = s.accept(); ObjectInputStream in = new ObjectInputStream(c.getInputStream()); String ch = (String) in.readObject(); System.out.println(ch); Registry r = LocateRegistry.getRegistry("localhost", 1099); IP4 i = (IP4) r.lookup("psy"); i.m("M2SSI"); s.close();in.close();c.close(); }catch(Exception e) {System.out.println(e.toString());}} </pre> | <pre> public class P2 { public static void main(String[] args) { try {DatagramSocket ss = new DatagramSocket(9876); byte[] dr = new byte[60]; DatagramPacket ch = new DatagramPacket(dr, dr.length); ss.receive(ch); System.out.println(new String(ch.getData())); ss.close(); }catch(Exception e) {System.out.println(e.toString ());}} </pre> |
| <pre> public interface IP4 extends Remote {public void m(String ch) throws RemoteException;} public class P4 extends java.rmi.server.UnicastRemoteObject implements IP4 { public P4() throws RemoteException {} public void m(String ch) throws RemoteException {System.out.println(ch);} public static void main(String[] args) { try {P4 p = new P4(); Registry r = LocateRegistry.createRegistry(1099); r.rebind("psy", p); }catch(Exception e) {System.out.println(e.toString());}} </pre> | |
| <pre> package Pack; @Stateful public class EFBean implements IEF { int a;int b; public void setA(int a) {this.a = a;} public void setB(int b) {this.b = b;} public EFBean() {} public void m() {System.out.println(a+b);}} </pre> | <pre> package Pack; @Remote public interface IEF { public void setA(int a); public void setB(int b); void m();} </pre> |