

Consigne : soyez bref. Si vous argumentez en délayant vos arguments et/ou vous n'êtes pas clair, vous perdez des points.

Soit une file d'attente « **jms/File_dattente** » dans un serveur d'application Java EE. Cette file d'attente a été préalablement remplie par la suite suivante de chaînes de caractères : "0" - "1" - "1" - "Stop" (le "0" à gauche est la 1^{ère} chaîne entrée et "Stop" la dernière).

Ces chaînes sont traitées comme suit dans un *Message-Driven Bean* :

```
@javax.ejb.MessageDriven(activationConfig = {
    @javax.ejb.ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/File_dattente"),
    @javax.ejb.ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
})
public class Distributeur implements javax.jms.MessageListener {
    private int _compteur = 0;
    private Utiliser_service _utiliser_service = null;

    @Override
    public void onMessage(javax.jms.Message message) {
        try {
            String s = ((javax.jms.TextMessage) message).getText();
            if (s.equals("1")) {
                _compteur++;
            } else {
                if (s.equals("0")) {
                    if (_utiliser_service == null) _utiliser_service = (Utiliser_service) (new
                    javax.naming.InitialContext()).lookup("java:global/Examen_CSL_2015_2016/Utiliser_service");
                    _utiliser_service.service(_compteur);
                }
            }
        } catch (Exception e) {
        }
    }
}
```

Le composant « **Utiliser_service** » est construit comme suit :

```
@javax.ejb.Stateful
public class Utiliser_service {
    private int _compteur = 0;

    @javax.ejb.EJB
    private Rendre_service _rendre_service;

    @javax.ejb.PrePassivate
    public void passivate() {
        _compteur--;
    }

    @javax.ejb.PostActivate
    public void activate() {
        _compteur++;
    }

    public void service(int i) {
        assert (_rendre_service != null);
        _compteur += i;
        _rendre_service.service(_compteur);
    }
}
```

Le composant « **Rendre_service** » est construit comme suit :

```
@javax.ejb.Stateless
public class Rendre_service {
    private int _compteur = 0;

    public void service(int i) {
        _compteur += i;
        System.out.println("Encore du travail ! Pas le choix... " + _compteur);
    }
}
```

Lors des tests, le développeur voit le message « **Encore du travail ! Pas le choix... 0** » dans le log du serveur Java EE alors qu'il espérait : « **Encore du travail ! Pas le choix... 2** ».

Question 1 (8 points) : d'où vient le problème ? Énoncez 1 ou 2 failles techniques irréfutables du programme global qui font qu'il est illusoire de voir afficher « **Encore du travail ! Pas le choix... 2** ».

Question 2 (8 points) : proposez un correctif qui permette de voir afficher « **Encore du travail ! Pas le choix... 2** ». Donnez des modifications simples à faire dans le code. Conservez l'architecture à 3 composants.

Question 3 (4 points) : proposez un correctif différent de celui de la question 2. Donnez des modifications simples (et différentes de la question 2) à faire dans le code. Conservez l'architecture à 3 composants.