

Questions de cours (7pts) :

1-Quelles sont les différentes méthodes pour implémenter les threads en java ? Justifier quand est ce qu'on utilise chaque méthode (2pts)

2- Pour chacune de ces questions, cocher les affirmations qui sont vraies (0,5 point par bonne réponse, -0,5 point par mauvaise réponse) (5 pts)

Pour chaque question, cocher les affirmations qui sont vraies (0,5 point par bonne réponse, -0,5 point par mauvaise réponse)

1- Quelle est la méthode qui permet de démarrer un thread :

- ☐ **init() ;**
☐ **start() ;**
☐ **resume() ;**
☐ **run() ;**

2- Pour assurer la synchronisation du point de rendez-vous suivant, déterminer les solutions correctes Pour Pt de rdv (init(SA,0), init (SB,0)):

Processus A	Processus B
...	...
I1;	J1;
Pt de rdv	Pt de rdv
I2;	J2;
...	...

[]

Processus A	Processus B
...	...
I1;	J1;
I2;	J2;
...	...

[]

Processus A	Processus B
I1;	J1;
V(SA);	V(SB);
P(SB);	P(SA);
I2;	J2;

[]

Processus A	Processus B
I1;	J1;
P(SB);	V(SB);
V(SA);	P(SA);
I2;	J2;

[]

Processus A	Processus B
I1;	J1;
P(SB);	P(SA);
V(SA);	V(SB);
I2;	J2;

3- Qu'implémente la classe Semaphore :

- ☐ Object ;
- ☐ Thread ;
- ☐ Runnable ;
- ☐ Serializable ;

4- Parmi les propositions suivantes déterminer celles qui sont correctes :

- ☐ Le problème d'accès concurrent aux variables peut être résolu en déclarant ces variables `synchronized` ;
- ☐ Les méthodes `synchronized` sont remplacées par les méthodes `lock()`, `unlock()` etc. de l'interface `Lock` ;
- ☐ Dans une classe ayant une partie du code synchronisé, plusieurs threads peuvent accéder à l'autre partie du code non synchronisé ;
- ☐ Pour une barrière utilisée une seule fois, il vaut mieux utiliser `CountDownLatch` ;
- ☐ `notify()` et `notifyAll()` peuvent être appelées dans un contexte non-synchronisé ;
- ☐ Les variables à conditions sont gérées par le paquetage `java.util.concurrent.lock` ;
- ☐ La barrière est cyclique mais pas réutilisable ;
- ☐ Les sémaphores en java sont gérées par les méthodes `acquire()` et `signal()` de la classe `Semaphore` ;
- ☐ Une méthode statique ne peut pas être synchronisée ;
- ☐ Une barrière permet de bloquer plusieurs thread en un point jusqu'à ce qu'un nombre prédéfini de threads atteigne ce point ;
- ☐ Une barrière permet de bloquer un thread en un point jusqu'à ce qu'un nombre prédéfini de threads atteigne ce point ;
- ☐ Quand le thread est endormi, il libère ses verrous ;

5- Que va afficher ce code ?

```
class Test extends Thread{
    public void run(){
        for(int i = 0; i < 30; i++){
            System.out.println("LMD GL");
        }
    }
    public static void main(String[] args){
        Test t = new Test();
    }
}
```

- ☐ Rien du tout.
- ☐ 30 fois "LMD GL".
- ☐ Ce code ne compilera pas !
- ☐ 29 fois "LMD GL"

6- Quand un thread est-il considéré comme mort ?

- a. Lorsqu'il est mis en attente.
- b. Lorsqu'il est bloqué par un autre thread.
- c. Lorsqu'il a dépilé la méthode `run()` de sa pile d'exécution.

```

Demande d'accès à la voie par la VILLE2 () {
.....
.....
.....
.....
.....
.....
.....
}

Sortie de la voie par la VILLE1 () {
.....
.....
.....
.....
.....
.....
.....
}

```

