

Java et le multithreading

Mis à jour le 30/11/2016

6 heures

Difficile

Licence



TP : simulez une conversation

Connectez-vous ou inscrivez-vous gratuitement pour bénéficier de toutes les fonctionnalités de ce cours !

Ce TP va vous permettre de vous assurer que vous avez bien compris le principe des verrous. Il va peut-être vous paraître simple au premier abord, mais vous allez avoir besoin d'un minimum de réflexion pour trouver une solution. Allez, au travail !:)

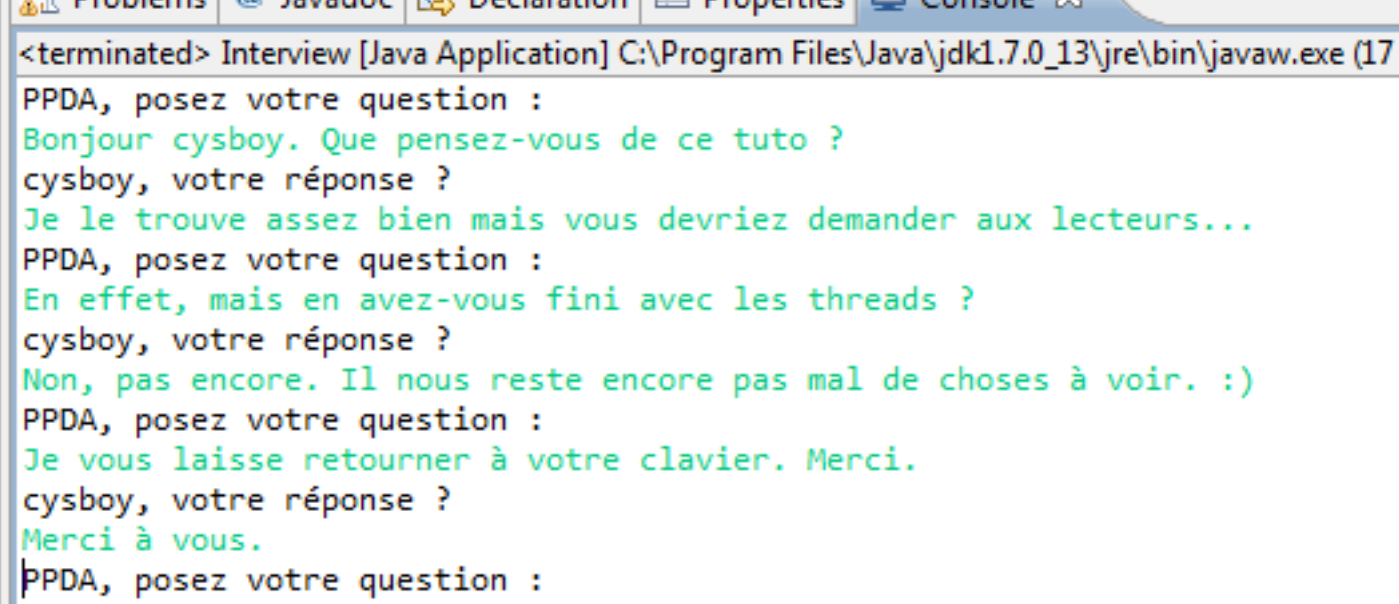
Votre mission

Comme le nom du TP l'indique, nous allons simuler une simple conversation en utilisant des threads.

Le but est de créer un objet `Journaliste` contenant une méthode `question()` et un objet `Personne` contenant une méthode `reponse()`, tous deux héritant de la classe `Thread`.

Le but est simple : au lancement du code, un journaliste pose une question et la personne attend la fin de la question. Une fois celle-ci posée, c'est le journaliste qui attend la réponse et ainsi de suite...

Voici un résultat d'exécution possible :



Exemple de résultat

Je ne vous donnerai pas plus d'indications, sinon autant vous donner la solution tout de suite... ^^

Vous êtes prêt ? Alors à vos claviers !

Correction

Voici une correction possible pour ce TP.

Le but de ce TP étant de vous faire pratiquer les threads, j'en ai utilisé dans cette correction. Mais sachez qu'il est également possible de simuler la conversation sans threads...

Journaliste.java

```
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;

public class Journaliste extends Thread{
    private String nom;
    private Lock verrou;
    private Condition question, reponse;
    private Scanner sc = new Scanner(System.in);

    public Journaliste(String pNom, Lock pVerrou, Condition pQuestion, Condition pReponse){
        nom = pNom;
        verrou = pVerrou;
        question = pQuestion;
        reponse = pReponse;
    }

    public void question(){
        verrou.lock();
        try{
            System.out.println(nom + ", posez votre question : ");
            sc.nextLine();
            //On libère le thread de réponse
            reponse.signalAll();
            //On bloque ce thread
            question.await();
        }catch (InterruptedException e) {
            e.printStackTrace();
        }finally{
            //On n'oublie pas de libérer le verrou !
            verrou.unlock();
        }
    }

    public void run(){
        while(true){
            question();
        }
    }
}
```

Personne.java

```
import java.util.Scanner;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;

public class Personne extends Thread {

    private String nom;
    private Lock verrou;
    private Condition question, reponse;
    private Scanner sc = new Scanner(System.in);

    public Personne(String pNom, Lock pVerrou, Condition pQuestion, Condition pReponse){
        nom = pNom;
        verrou = pVerrou;
        question = pQuestion;
        reponse = pReponse;
    }

    public void reponse(){
        try{
            verrou.lock();
            System.out.println(nom + ", votre réponse ?");
            sc.nextLine();
            //On informe redemande une question
            question.signalAll();
            //On bloque la réponse
            reponse.await();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }finally{
            //et on libère le verrou !
            verrou.unlock();
        }
    }

    public void run(){
        while(true){
            reponse();
        }
    }
}
```

Interview.java

```
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

public class Interview {
    public static void main(String[] main){
        Lock lock = new ReentrantLock();
        Condition question = lock.newCondition();
        Condition reponse = lock.newCondition();
        Journaliste j = new Journaliste("PPDA", lock, question, reponse);
        Personne p = new Personne("cysboy", lock, question, reponse);

        j.start();
        p.start();
    }
}
```

Alors, vous avez réussi ? Pas tout à fait ? N'hésitez pas à reprendre les chapitres précédents au calme ou à demander de l'aide sur le [forum Java](#).

Je vous propose ensuite de poursuivre sur votre lancée en voyant comment lancer vos threads différemment.

← PROTÉGEONS NOS VARIABLES

AMÉLIOREZ LA GESTION DE VOS THREADS →

Le professeur

Cyrille Herby

Spécialiste en développement Java et curieux insatiable d'informatique et de programmation web. Actuellement auditeur en sécurité.

< Notions de bases >

1. Avant toutes choses

2. Protégeons nos variables

▶ 3. TP : simulez une conversation

ACCÉDER AU FORUM

Créé par

OpenClassrooms, Leading E-Learning Platform in Europe

