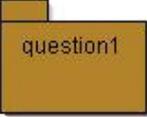


TP 7

Java Management eXtensions

- Usage des composants JMX, jconsole
- Documentation [JMX chez Sun](#)

- Visualisez le sujet en ouvrant `index.html` du répertoire qui a été créé à l'ouverture de `tp7.jar` par BlueJ; vous aurez ainsi accès aux différents liens qui sont proposés pour vous aider.
- Soumettez chaque question à l'outil d'évaluation `jnews`.

question1

Un serveur TCP "administrable" à distance

Soit l'implémentation "expérimentale" d'un serveur en mode TCP; ce serveur se contente de retourner au client une forme "lisible" de l'instance de la classe `java` transmise en paramètre. Toutes les classes sont connues du serveur.

Un extrait du serveur (classe `ExperimentalTCPServer`), ici le source du traitement de chaque requête:

```
while(!local.isInterrupted()) {
    Socket socket = serveur.accept();

    // en entrée
    ObjectInputStream ois= new ObjectInputStream( socket.getInputStream());
    Object obj = ois.readObject();

    // en sortie
    ObjectOutputStream oos= new ObjectOutputStream( socket.getOutputStream());
    oos.writeObject(obj.toString());

    socket.close();
}
```

Or, ce serveur est "très" expérimental, il tombe en panne fréquemment: les mesures effectuées montrent que celui-ci s'arrête définitivement après une dizaine de secondes d'utilisation !!! La méthode de l'instance implémentant l'interface `ExceptionListener`, transmise lors de la création de ce "serveur", est alors exécutée. (*bien entendu, c'est une panne volontaire pour les besoins du tp et il n'est pas souhaitable de corriger cette "erreur"...*)

Proposer un composant compatible `MBean`, permettant de gérer ce serveur à distance, et de notifier tous les abonnés inscrits en cas de "panne".

L'interface à respecter du composant `MBean` (interface `TCPServerMBean`) :

```

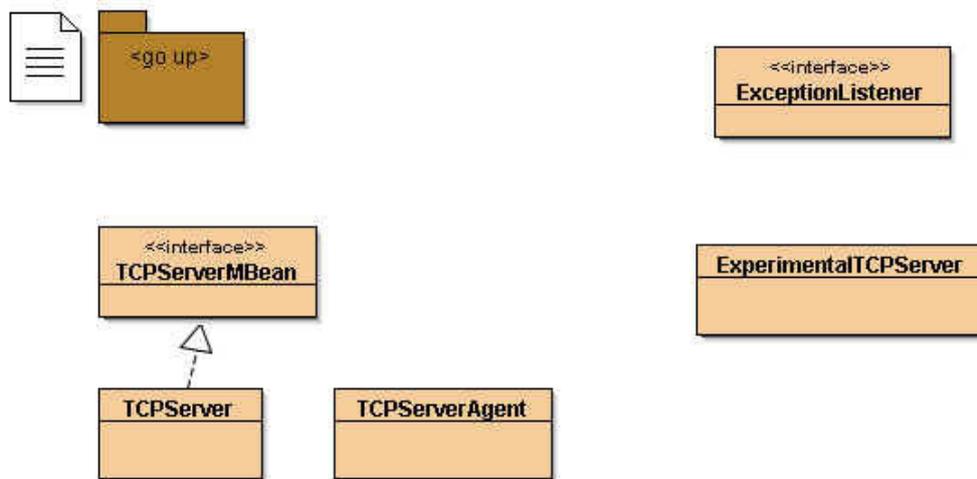
public interface TCPServerMBean {

    // opérations
    public void create(int port) throws Exception;
    public void start();
    public void stop();

    // lecture seulement
    // la durée d'activité de ce serveur (entre 2 pannes...)
    public long getMaxActiveTime();
    public long getPort();
    public boolean isActive();
}

```

Proposez une implémentation (classe TCPServer) de ce composant MBean (implémentant cette interface) :



Proposez un agent de ce composant (classe TCPServerAgent) qui enregistre le composant TCPServer et installe un connecteur RMI ainsi qu'un adaptateur HTTP; voir les méthodes setRMIConnecteur et setHTTPAdapter.

```

public class TCPServer extends NotificationBroadcasterSupport implements TCPServerMBean, Seriali
....

```

Un exemple d'utilisation en utilisant jconsole en "local".

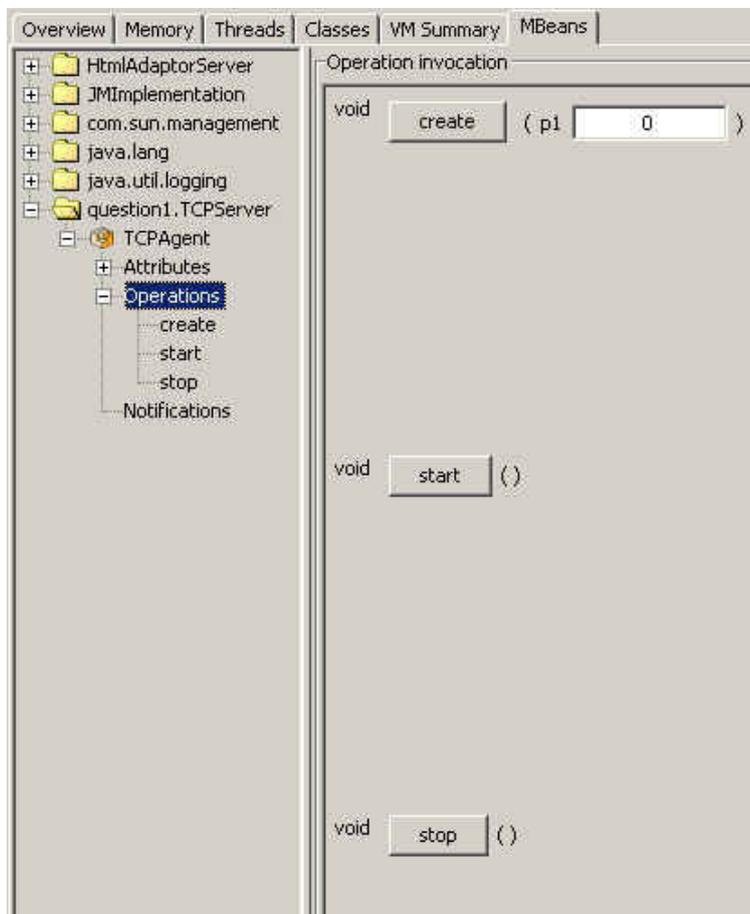
Démarrage de l'agent :

```

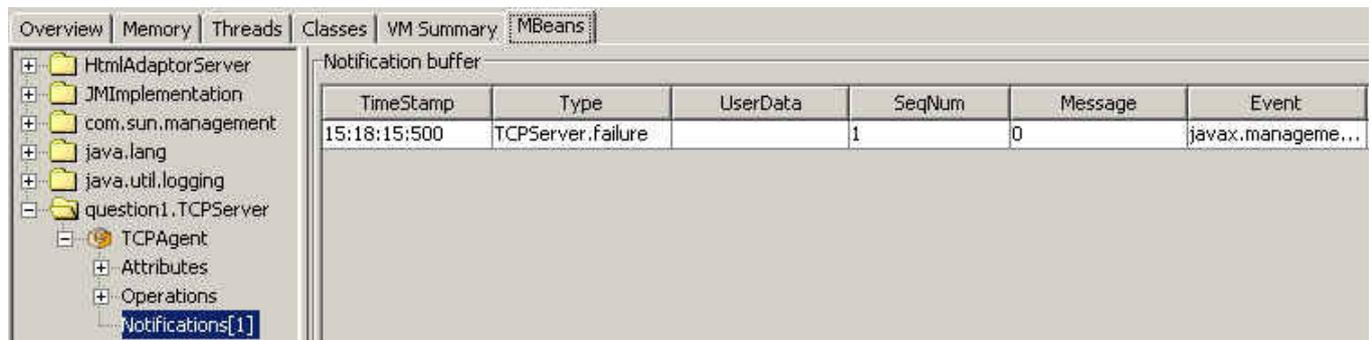
H:\NSV102\tp_jmx_correction>start rmiregistry 9999
H:\NSV102\tp_jmx_correction>java -cp .;jdmkrt.jar question1.TCPServerAgent
TCPServerAgent is running...

```

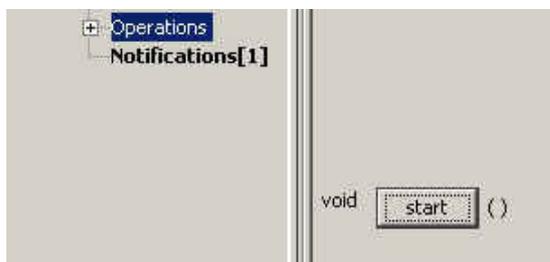
Utilisation du composant depuis jconsole :



Une panne s'est produite à environ 15h 18mn 15s :



Après avoir déclenché la méthode start de ce composant (il suffit de cliquer) :

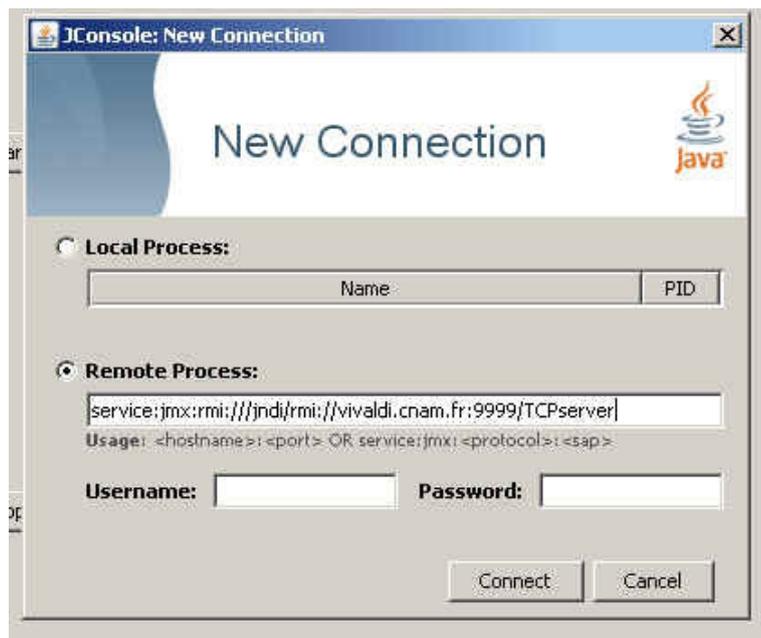


Une seconde "panne" a été notifiée environ 12 secondes plus tard (valeur du champ Message) :

Overview Memory Threads Classes VM Summary MBeans					
-Notification buffer-					
TimeStamp	Type	UserData	SeqNum	Message	
15:24:16:250	TCPServer.failure		2	12281	jav.
15:21:31:312	TCPServer.failure		1	0	jav.

Un autre exemple d'utilisation, avec jconsole depuis une machine distante, et à l'aide du connecteur rmi

(URL `service:jmx:rmi:///jndi/rmi://vivaldi.cnam.fr:9999/TCPserver`) :



Encore un autre exemple d'utilisation en utilisant un navigateur et l'adaptateur HTTP (URL `http://vivaldi.cnam.fr:8088/`) :

- **MBean Name:** question1.TCPServername=TCPAgent
- **MBean Java Class:** question1.TCPServer

[Back to Agent View](#)

Reload Period in seconds:

MBean description:

Information on the management interface of the MBean

List of MBean attributes:

Name	Type	Access	Value
Active	boolean	RO	false
MaxActiveTime	long	RO	UNAVAILABLE: javax.management.RuntimeMBeanException wraps [java.lang.NullPointerException]
Port	long	RO	UNAVAILABLE: javax.management.RuntimeMBeanException wraps [java.lang.NullPointerException]

List of MBean operations:

[Description of stop](#)

void stop()

[Description of start](#)

void start()

```
// un extrait possible de l'installation de l'adaptateur
HtmlAdaptorServer adapter = new HtmlAdaptorServer();
adapter.setPort(8088);
name = new ObjectName("HtmlAdaptorServer:name=html,port=8088");
mbs.registerMBean(adapter, name);
```

question2

MBean "de démarrage et redémarrage"

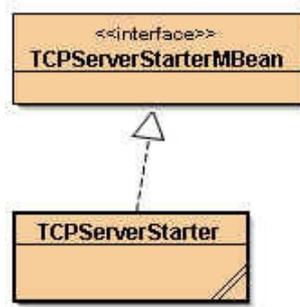
Proposez un composant TCPServerStarterMBean chargé de :

- créer et démarrer le Bean TCPServer de la question1
- redémarrer automatiquement le serveur TCP associé au composant TCPServer en cas de panne,
- fournir quelques mesures comme le nombre de pannes depuis la mise en service de ce composant,
- ...

```
public interface TCPServerStarterMBean{
    // activité du ce composant
    public boolean getActive();
    public void setActive(boolean active);

    // le nombre de notifications, émises par TCPServer, depuis le démarrage
    public long getNotifications();

    // arrêt du TCPServer, et désinstallation de ce composant
    public void stop() throws Exception;
}
```



Proposez la possibilité de charger dynamiquement ce composant : l'agent "m-let" et son fichier de description .

Notez que les soumissions avec JNEWS durent environ 20 secondes à chaque question (au moins la durée d'une "panne" ...)