

# TD 3C – Programmation Socket en Java

## 1 - Un premier programme en java

Le code ci-dessous correspond a un serveur TCP/IP en Java.

**class applic**

```
{  
  
    public void main (String args[]) throws Exception  
    { ServerSocket sockserv=null; // ?  
      DataInputStream in ; // ?  
      DataOutputStream out; // ?  
      byte mess[];  
      sockserv = new ServerSocket (1234);  
      try { while (true)  
          { try  
              { Socket sockcli = sockserv.accept(); // ?  
                in = new DataInputStream (sockcli.getInputStream()); // ?  
                out = new DataOutputStream (sockcli.getOutputStream()); // ?  
                mess = new byte[80];  
                in.read(mess,0,80); // ?  
                out.write(mess);  
                sockcli.close(); // ?  
              } catch (IOException ex) { }  
            }  
          } finally { try { sockserv.close(); // ?  
                      } catch (IOException ex) { }  
                    }  
        }  
      }  
    }  
}
```

1 – Mettez en évidence les parties équivalentes entre les deux programmes

2 – Ce code correspond à une application serveur , quels éléments nous permettent de dire cela ?

3 - Utilisez Telnet pour dialoguer avec ce programme

## 2 - Le client java

1 – Sachant qu'il existe un objet socket(url, port) permettant à un client de se connecter avec un serveur et que cet objet possède une méthode (ou fonction) connexion() qui établit la connexion, écrire le pseudo code d'une application qui communique avec celle-ci.

2 – L'application ci-dessus ne peut pas communiquer avec plusieurs clients, pourquoi ? Que faudrait-il faire pour que cela fonctionne ? (ne pas la faire encore ...)

## 3 - Un "chat" en java

1 - Modifier les programmes pour en faire un chat

2 - Lancer maintenant l'application serveur java et l'application cliente C. Ca marche ???

3 - On inverse, lancer l'application serveur C et l'application java client. Ca marche toujours ????