

classe Socket

□ douille ou prise logicielle pour la communication entre client et serveur

- `public Socket (String host, int port)`
throws `UnknowHostException, IOException`
- `public InputStream getInputStream()`
throws `IOException`
- `public OutputStream getOutputStream()`
throws `IOException`
- `public void close()`
throws `IOException`

Exemple de client

```
Socket bs; //branchement au Serveur
BufferedReader in,
    clavier = new BufferedReader(
        new InputStreamReader(System.in)
    );
PrintStream out, ecran;
String lu; boolean fini;
try {
    bs = new Socket("127.0.0.1", 3000);
    in = new BufferedReader(
        new InputStreamReader(bs.getInputStream())
    );
    out = new PrintStream(bs.getOutputStream());
    //dialogue avec le serveur
    ... //voir vue suivante
    bs.close();
} catch (IOException e) { System.err.println(e); }
```

Dialogue avec le serveur

```
//dialogue avec le serveur
do {
    ecran.print("? ");
    lu = clavier.readLine();
    out.println(lu); // "envoyer" au serveur
    fini = lu.equals(".");
    if (!fini) {
        lu = in.readLine(); // la réponse du serveur
        ecran.println(lu);
    }
} while(!fini);
```

classe ServerSocket

□ pour la gestion des connexions des clients

- public **ServerSocket**(int port)
throws UnknowHostException, IOException
- public Socket **accept**()
throws IOException
- public void **close**()
throws IOException

Exemple de serveur

```
ServerSocket serveur;  
try {  
    serveur = new ServerSocket(3000);  
    do {  
        Socket bc = serveur.accept(); //branchement au client  
        (new CommClient(bc)).start();  
    } while(true);  
    serveur.close();  
} catch (IOException e) { System.err.println(e); }
```

Communication avec client

```
Class CommClient extends Thread {  
    protected Socket cc;  
    protected BufferedReader in;  
    protected PrintStream out;  
    public CommClient(Socket sc) {  
        cc = sc;  
        try {  
            in = new BufferedReader(  
                new InputStreamReader(cc.getInputStream())  
            );  
            out = new PrintStream(cc.getOutputStream());  
        } catch(Exception e) { System.err.println(e); }  
    }  
    public void run() {  
        ...  
    }  
}
```

Communication avec client...

```
Class CommClient extends Thread {  
    ...  
    public void run() {  
        String lu = "";  
        try {  
            do {  
                lu = in.readLine();  
                if (lu!=null && !lu.equals(".")) out.println(lu);  
            } while(lu!=null && !lu.equals("."));  
            cc.close();  
        } catch(Exception e) { System.err.println(e); }  
        finally {  
            try { cc.close(); } catch (IOException e) {}  
        }  
    }  
}
```