

## TP n°3

### Protocoles : decouverte

Dans ce TP, toutes les transmissions seront faites en mode connecte TCP.

En Java, il est possible d'ouvrir une connexion a un service de *port* donne sur une *machine* donnee en utilisant la classe `java.net.Socket`. Par exemple :

```
Socket s = new Socket(machine,port);
```

Les methodes `getOutputStream()` et `getInputStream()` permettent de retrouver des flux d'entrees/sorties (respectivement de type `OutputStream` et `InputStream`) pour ecrire et lire sur cet objet de communication. De facon a manipuler ces objets, regardez comment les utiliser avec les classes `PrintStream`, `InputStreamReader` et `BufferedReader`.

#### Exercice 1 [Client daytime]

Ecrivez un programme Java *jdaytime* qui se connecte au service *daytime* d'une *machine* donnee en parametre et recupere la date pour l'afficher a l'ecran.

#### Exercice 2 [Client/Serveur]

1. Ecrivez un client et un serveur en Java. Le serveur choisira un entier aleatoirement entre 1 et 100. Et repondra "+" si son entier est plus grand que celui envoye par le client, "-" s'il est plus petit et s'il est egal il repondra "=" avant de fermer la connexion. Le client affichera ce qu'il recoit du serveur.
2. Sans changer le serveur, modifier votre client pour que ce soit plus lisible pour l'utilisateur.
3. Tester votre client avec le serveur d'un de vos camarades, et inversement.

*Indications* : La classe `InetAddress` vous permet d'avoir des informations sur la machine ou vous executez votre programme, par exemple :

```
InetAddress.getLocalHost().getHostName();
```

Pour creer le serveur utilisez la classe `ServerSocket`.

#### Exercice 3 [Thread]

A l'aide des classes `Thread` et `Runnable`, modifiez votre serveur afin qu'il accepte plusieurs connections simultanees.