

TP n°3

Protocoles : découverte

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

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

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

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

Exercice 1 [Client daytime]

Écrivez un programme Java *jdaytime* qui se connecte au service `daytime` d'une *machine* donnée en paramètre et récupère la date pour l'afficher à l'écran.

Exercice 2 [Client/Serveur]

1. Écrivez un client et un serveur en Java. Le serveur choisira un entier aléatoirement entre 1 et 100. Et répondra "+" si son entier est plus grand que celui envoyé par le client, "-" s'il est plus petit et s'il est égal il répondra "=" avant de fermer la connexion. Le client affichera ce qu'il reçoit 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 où vous exécutez votre programme, par exemple :

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

Pour créer le serveur utilisez la classe `ServerSocket`.

Exercice 3 [Thread]

À l'aide des classes `Thread` et `Runnable`, modifiez votre serveur afin qu'il accepte plusieurs connexions simultanées.