

## R309 – TP3

- 1 Proposer une classe abstraite `ClientServeur` qui offre deux méthodes abstraites : `envoyerMsg` et `recieveMsg` qui permettent d'envoyer et de recevoir des messages textes (chaîne de caractères). Attention à ce niveau on ne précise pas le protocole de transport employé.
- 2 En réutilisant la classe `ClientServeur`, proposer une classe abstraite `ClientServeurUDP` qui permet d'envoyer et de recevoir de messages texte
- 3 En réutilisant la classe `ClientServeurUDP` proposer une classe `EchoUDPServer` qui implémente un serveur d'écho qui utilise le protocole UDP. Le fonctionnement d'un serveur echo consiste à renvoyer aux clients qui contacte le serveur le message reçu précédé par le préfixe : Echo de *adresse du serveur numéro du port*
- 4 En réutilisant la classe `ClientServeurUDP`, proposer une classe `EchoUDPClient` qui permet d'envoyer un message texte à un serveur echo et affiche la réponse du serveur.
- 5 En réutilisant la classe `ClientServeurUDP` proposer une classe abstraite `ServeurUDPMT` qui implémente un serveur UDP multi-thread : Chaque message reçu par un client sera traité par un thread de traitement spécifique.
- 6 En réutilisant la classe `ServeurUDPMT` , proposer une classe `EchoUDPServerMT` qui implémente un serveur echo multi-thread. La réponse obtenu par le client est composé du message initialement envoyé précédé du nom du thread qui le traite.