

Protocole client-serveur

On souhaite implanter un protocole qui permette à un « client » de consulter à distance les données stockées dans une liste sur un « serveur ». Dans un premier temps, le client et le serveur s'exécuteront dans des activités séparées (*Thread*) et la communication se fera grâce à des implantations de `BlockingQueue` dans la JDK (nous utiliserons les méthodes bloquantes).

Le protocole de départ est le suivant :

- le client envoie un indice pour recevoir la donnée correspondante dans la liste
- le client envoie -2 pour recevoir la taille de la liste
- le client envoie -1 pour mettre fin à la communication

Questions

1. Implanter le protocole ci-dessus, partie serveur et partie client. Pour la partie client, un implémentera une API avec deux méthodes `get` et `size`. On supposera d'abord pour simplifier que les données stockées sont de type `Integer`.
2. Peut-on facilement stocker des éléments de type `String` dans la liste au lieu du type `Integer` ?
3. Améliorer le protocole en définissant les types suivants :

```
sealed interface Request {}  
record GetSize() implements Request {}  
record GetValue(Integer index) implements Request {}  
record Stop() implements Request {}
```

```
sealed interface Answer {}  
record Size(Integer index) implements Answer {}  
record Value(Integer element) implements Answer {}
```

4. Modifier le protocole et l'implantation pour une liste qui stocke des `String`.
5. Généraliser le protocole et l'implantation pour une liste ayant un type d'élément générique.
6. Modifier la partie client pour implanter un « objet » liste (pas juste une API) qui représente « localement » chez le client la liste « distante » stockée sur le serveur. Ce « mandataire » (*proxy*) devra implanter l'interface `List` (on pourra utiliser la classe `AbstractList`).
7. Étendre le protocole pour rendre la liste mutable (méthodes `set`, `add` et `remove`).