

TD 5/TD 6 – RMI

François LEPAN

28 mars 2013

1

2 Election sur un anneau d'objets RMI

- 2.1 Quel est le test d'arrêt de l'élection i.e. du "tour de l'anneau" ?
- 2.2 Quelle information faut-il inclure dans le message qui fait le "tour de l'anneau" pour pouvoir réaliser ce test ?
- 2.3 Représenter sur un diagramme les échanges de messages engendrés par l'exécution d'une élection avec l'objet 2 comme initiateur
- 2.4 Quel est le test de décision de l'élu ?
- 2.5 Quelle information faut-il inclure dans le message qui fait le "tour de l'anneau" pour pouvoir réaliser ce test ?
- 2.6 À la fin d'un tour, quel(s) objet(s) connaît(ssen)t l'identifiant de l'élu ? Pourquoi ?
- 2.7 Pour chacune des solutions proposées à la question précédente, définir l'interface de la classe Java correspondant à un objet de l'anneau

```
public interface NoeudElection extends Remote {  
    // Bleu  
    // On ajoute l'initiateur si il y a plusieurs vague lancer en mm temps  
    // afin de ne pas avoir de concurrence entre les vagues  
    // appellant -> pour savoir de quel cote propager  
    public void electionChef(NoeudElection appellant, NoeudElection chef) throws RemoteExcepti  
  
    // Rouge  
    public void propagationChef(NoeudElection appellant, NoeudElection chef) throws RemoteExce  
  
    // en passant la valeur de puissance dans la 1ere fonction on peut enlever cette fction -  
    int getPuissance();  
}
```

2.8 Pour chacune des solutions proposées à la question précédente, donner en Java l'implantation de la méthode election

```
public class Noeud implements NoeudElection {

    // cette valeur est supposée initialisé par un joli Math.random()
    private int myVal;

    private NoeudElection chef;
    private NoeudElection gauche;
    private NoeudElection droite;

    public void electionChef(NoeudElection appellant, NoeudElection chef, int chefVal) throws

        // Bleu
        if (chef == null) {
            if (this.myVal > chefVal) {
                this.chef = this;
            } else {
                this.chef = chef;
                this.myVal = chefVal;
            }

            if (this.gauche == appellant) {
                this.droite.election(this, this.chef, this.myVal)
            } else {
                this.gauche.election(this, this.chef, this.myVal)
            }
        }

        // Rouge
    } else {
        if (this.myVal < chefVal) {
            this.myVal = chefVal;
            this.chef = chef;
        }

        appellant.propagationChef(this, this.chef);
    }
}
}
```