CC60H : Nuevos Lenguajes de Programación
Control 2–Semestre Primavera'2001

Prof.: Luis Mateu.

Sin apuntes, 1 hora 30 minutos

Pregunta 1

Se tienen N threads que en un instante de la ejecución deben votar para poder proseguir sus respectivos cálculos. El conteo lo realiza un objeto escrutador que es compartido por todos los threads. Un thread vota de la siguiente manera :

```
int voto= decidirVoto(); // 0 o 1
int ganador= escrutador.votar(voto);
```

El voto es 0 o 1. El método `votar` sólo retorna cuando se haya decidido el resultado de la votación y retorna quién es el ganador (-1 en caso de empate). La elección queda decidida cuando alguna de las opciones obtiene $N/2+1$ votos.

El objeto escrutador se crea especificando el número de threads que participarán en la votación. Además el método `resultados` sirve para esperar hasta que todos los procesos hayan votado, entregando un arreglo de 2 elementos con el resultado de la votación. Ejemplos :

```
Escrutador escrutador= new Escrutador(N);
...
int[] resultados= escrutador.resultados();
```

Programe completamente la clase Escrutador.

Pregunta 2

Se necesita implementar un sistema de votación interactiva usando RMI. El diseño contempla los siguientes comandos :

i. `java Escrutar URL N` : Crea una votación. El URL especifica la máquina en donde corre el registro de RMI y la identificación que se usará para el objeto escrutador. N corresponde al número de usuarios que deben votar. Este programa debe terminar cuando hayan votado los N usuarios (termine con System.exit o borrando el escrutador del registro de RMI). Ud. debe mostrar en pantalla la cantidad de votos para 0 y para 1. Ejemplo:

```
% java Escrutar //anakena.dcc.uchile.cl/escrutador 5
1 4      <- votos para 0 y para 1
```

ii. `java Votar URL VOTO` : Emite un voto (0 o 1), espera a que la votación se decida y muestra en pantalla el ganador de la votación.

```
% java Votar //anakena.dcc.uchile.cl/escrutador 0
1         <- ganador
```

Programe los métodos main de las clases Escrutar y Votar. Además programe todo otro método o clase adicional que requiera. Esto incluye reescribir la clase Escrutador de modo que pueda ser referenciada en forma remota. Sin embargo, si hay trozos de código que no cambian puede indicarlo sin reescribirlos completamente. No se preocupe de las excepciones.

Pregunta 3

Programe el siguiente método :

```
void apropos(Class cl, String txt);
```

Este método debe mostrar todas las variables de instancia definidas en la clase que posean un nombre que contiene `txt` como substring. Se debe desplegar el nombre del método y el tipo de los campos encontrados. Por simplicidad ignore los campos definidos en la clase base.

Torpedo

```
class Integer {
  static int parseInt(String num); // convierte de string a int
}

class String {
  int indexOf(String txt); // -1 si txt no es substring
}

class Naming {
  static void rebind(String url, Remote obj);
  static Remote lookup(String url);
  static void unbind(String url); // borra del registro RMI
}

interface Remote { ... }
class UnicastRemoteObject { ... }

class Class {
  String getName();
  Field[] getDeclaredFields();
}

class Field {
  String getName();
  Class getType();
}
```