



Programación de Sistemas y Concurrencia

Control Bloque 2-Semáforos

Informática A + Software A

Curso 2023-2024

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO/TITULACIÓN _____

Descripción del sistema

Un grupo de **J** jugadores han ganado un torneo de fútbol y la organización del evento les ha concedido un trofeo. Como son muchos jugadores, y solo uno de ellos puede quedarse con el trofeo, inventan un juego que les permite escoger de forma aleatoria al jugador depositario del trofeo. El juego consta de **J-1** jugadas en las que juegan un número **decreciente** de jugadores haciendo uso de una **urna** con varias **bolas** de color **blanco** y **una** de color **negro** del modo siguiente: “en la jugada *i* (con $i=1..J-1$), juegan $J-i+1$ jugadores y la urna tiene $J-i$ bolas blancas y 1 bola negra”. Es decir,

- en jugada **1**, juegan los **J** jugadores con una urna con **J-1** bolas blancas y **una** bola negra
- en jugada **2**, juegan **J-1** jugadores con una urna con **J-2** bolas blancas y **una** bola negra
- ...
- en la jugada **J-1**, juegan **2** jugadores y la urna tiene **una** bola blanca y **una** bola negra.

Así, en cada jugada, se descarta a un jugador, hasta que solo queda uno, que es el ganador. Para ir descartando jugadores, en cada jugada, los jugadores que todavía no han salido del juego sacan una bola de la urna. Si la bola que saca un jugador es blanca, continúa en el juego; si la bola es negra, ha perdido y sale del juego. El jugador que saca la bola blanca en la última jugada es el ganador y se lleva el trofeo.

Se pide implementar el juego descrito **utilizando sólo semáforos binarios para sincronizar las hebras jugadores**. En el campus virtual, se encuentran los esqueletos de la solución con las siguientes clases:

- **Color**: es una clase enumerada con los dos colores posibles de cada bola: **blanca** y **negra**.
- **Urna**: es la clase del objeto **urna** que se utiliza en cada jugada. La urna está implementada con un array de objetos de tipo **Color**, cada uno representando una bola que está en la urna. La clase tiene tres métodos (ya implementados):
 - o **public void** llenoUrna(int NBolas) que llena la urna con NBolas-1 bolas blancas y 1 bola negra.
 - o **public Color** sacoBola() que extrae una bola de la urna y devuelve su color.
 - o **public String** toString() que devuelve un string con el contenido de la urna.

Nota: Observa que los métodos de la clase Urna no están sincronizados.

- **Juego**: es la clase que gestiona el juego y contiene los métodos que hay que implementar.
 - o **public Juego**(int NBolas) es el constructor de la clase. Recibe el número (NBolas) que tendrá la urna inicialmente. En el constructor se crea la primera urna con la que se va a jugar, se llena con el número de bolas indicado y se imprime por pantalla.
 - o **public Color** sacarBola(int id): este método es utilizado por el jugador id que quiere sacar una bola de la urna. A partir de la segunda jugada, el jugador debe esperar a que la jugada anterior haya terminado completamente antes de sacar la bola de la urna, es decir, debe cumplirse la condición de sincronización **CS1_Jugador** descrita más adelante. La función devuelve el color de la bola que ha sacado el jugador.
 - o **public boolean** esperaTodos(int id): este método es utilizado por el jugador id que acaba de sacar una bola de la urna. Debe esperar a que todos los demás jugadores de la jugada saquen su bola para poder continuar con la siguiente jugada, si no es la última jugada y no ha sacado la bola negra. El método devuelve un booleano indicando si es la última jugada (es decir, si

la urna actual tenía solo dos bolas). Debe cumplirse la condición de sincronización **CS2_Jugador** descrita más adelante.

Observa que antes de que empiece cada nueva jugada debe llenarse la urna con un número nuevo de bolas blancas (salvo en la primera jugada, ya que la urna se llena en el constructor del juego).

- **Jugador**: es una clase que extiende a Thread y recibe como parámetros en el constructor un identificador (id) y una referencia al juego (j).
 - o **public void run()** que ya está implementado contiene el comportamiento de cada jugador:
 - saca una bola de la urna actual
 - espera a que todos los demás jugadores saquen su bola de la urna
 - (1) si la bola que ha sacado es blanca y no es la última jugada, vuelve a jugar; si la bola que ha sacado es blanca y es la última jugada, es el ganador; (3) si ha sacado la bola negra, ha perdido y sale del juego
- **Principal**: contiene el método **main** que define la constante J, crea una urna, un juego, J instancias de jugadores y las pone en funcionamiento.

Las condiciones de sincronización que hay que implementar son:

- (1) CS1_Jugador: Un jugador no puede participar en una nueva jugada hasta que la jugada anterior no ha terminado completamente (todos los jugadores saben si deben continuar en el juego). Inicialmente, participan todos los jugadores.
- (2) CS2_Jugador: Un jugador que ha sacado una bola, tiene que esperar a que todos los demás jugadores de la misma jugada saquen su bola antes de decidir si continúa en el juego, ha perdido o es el ganador

Una posible salida de la ejecución del juego con 6 jugadores podría ser:

```
Jugada: 1
Urn=[blanca, blanca, blanca, blanca, negra, blanca]
```

```
Jugador 0 ha sacado una bola blanca
Jugador 2 ha sacado una bola negra. Ha perdido
Jugador 1 ha sacado una bola blanca
Jugador 3 ha sacado una bola blanca
Jugador 4 ha sacado una bola blanca
Jugador 5 ha sacado una bola blanca
Jugador 5 termina jugada
Jugador 0 termina jugada
Jugador 2 termina jugada
Jugador 1 termina jugada
Jugador 3 termina jugada
Jugador 4 termina jugada
*****
```

```
Jugada: 2
Urn=[negra, blanca, blanca, blanca, blanca]
```

```
Jugador 4 ha sacado una bola blanca
Jugador 5 ha sacado una bola blanca
Jugador 0 ha sacado una bola blanca
Jugador 1 ha sacado una bola blanca
Jugador 3 ha sacado una bola negra. Ha perdido
Jugador 3 termina jugada
Jugador 4 termina jugada
Jugador 5 termina jugada
Jugador 0 termina jugada
Jugador 1 termina jugada
*****
```

```
Jugada: 3
Urn=[blanca, blanca, negra, blanca]
```

```
Jugador 1 ha sacado una bola blanca
Jugador 4 ha sacado una bola negra. Ha perdido
Jugador 5 ha sacado una bola blanca
Jugador 0 ha sacado una bola blanca
Jugador 0 termina jugada
Jugador 1 termina jugada
Jugador 4 termina jugada
Jugador 5 termina jugada
*****
```

```
Jugada: 4
Urn=[negra, blanca, blanca]
```

```
Jugador 5 ha sacado una bola blanca
Jugador 0 ha sacado una bola blanca
Jugador 1 ha sacado una bola negra. Ha perdido
Jugador 1 termina jugada
Jugador 5 termina jugada
Jugador 0 termina jugada
*****
```

```
Jugada: 5
Urn=[negra, blanca]
```

```
Jugador 0 ha sacado una bola blanca
Jugador 5 ha sacado una bola negra. Ha perdido
Jugador 5 termina jugada
Jugador 0 termina jugada
*****
```

Jugador 0 es el GANADOR y se lleva el trofeo a casa!!!

Fin del programa

FIN DEL ENUNCIADO