



EJERCICIO THREADS 8

Implementaremos un modelo cliente servidor con buffers y zonas sincronizadas. Simulará el típico juego de construcciones.

Hay un almacén donde almacenamos hierro y otro donde almacenamos herramientas. Los almacenes tienen una capacidad máxima de 10 unidades. Los almacenes empiezan a 0.

Para ello tendremos 3 threads:

El minero, recolecta una unidad de hierro cada 100-1000 ms. Es un bucle que va produciendo y añade +1 al almacén de hierro cada vez que recolecta una unidad de hierro. Si no hay sitio en el almacén de hierro se para y espera a que haya sitio para seguir trabajando.

El herrero, produce 1 herramienta a partir de dos hierros. Es un bucle que va produciendo y añade +1 al almacén de herramientas y -2 al almacén de hierro cada vez que fabrica una herramienta. Tarda de 200 a 2000 en producirla. Si no hay sitio en el almacén de herramientas o no queda hierro se para y espera a que pueda seguir trabajando.

El tren, se lleva 5 herramientas cada vez que viene. Tarda de 500 a 5000 en volver a por otra carga. Si no hay carga de herramientas, se espera.

El acceso de los 3 threads a los dos almacenes debe ser con sincronización, para que no se produzcan errores. Usar wait y notify para bloquear y desbloquear threads, asociándolos a la condición.

El sistema empezará preguntando cuantos mineros y cuantos herreros hay, lo ideal es el doble de mineros que herreros. Siempre hay un tren. Inicializa los almacenes a 0. El programa debe ir sacando por pantalla el valor de los dos almacenes.

Opcionalmente se puede ir sacando lo que hacen los trabajadores.

Entregar el código fuente bien comentado, y ejemplos de ejecución.

```
// MAIN //
```

```
package threads;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```
import javax.swing.JOptionPane;
```

```
public class Main {
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
static List<Minero> mineros = new ArrayList<>();
static List<Herrero> herreros = new ArrayList<>();

static Tren t = new Tren("TREN");
static Almacen almacenHierro = new Almacen("Hierro");
static Almacen almacenHerramientas = new Almacen("Herramientas");

static int numMineros, numHerreros, numTren;

/*
 * Main que pide los trabajadores que habran y los ejecuta 1 a 1 en un for
 */
public static void main(String[] args) {
    // TODO Auto-generated method stub

    // Se preguntan los mineros que habra
    numMineros = Integer.parseInt(JOptionPane.showInputDialog("Cuantos
mineros habra?"));
    for(int i = 0 ; i < numMineros; i++) {
        mineros.add(new Minero("MINERO " + i));
    }

    // Se preguntan los herreros que habra
    numHerreros = Integer.parseInt(JOptionPane.showInputDialog("Cuantos
herreros habra?"));
    for(int i = 0 ; i < numHerreros; i++) {
        herreros.add(new Herrero("HERRERO " + i));
    }

    /*
     * Se ejecutan los threads
     */
    for (Minero minero : mineros) {
        minero.start();
    }
    for (Herrero herrero : herreros) {
        herrero.start();
    }
    t.start();
}

}

// MINERO //

package threads;

public class Minero extends Thread {

    public Minero(String nombre) {
        this.setName(nombre);
    }

    public void run() {
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
do {
    try {
        // Se calcula random el tiempo de espera de cada
ejecucion
        int pausa = (int) (Math.random() * 1000) + 100;
        // El thread se espera lo calculado anteriormente
        Thread.sleep(pausa);
        /*
        * Llamamos a los metodos sincronizados para añadir o
quitar elementos de los almacenes
        */
        Main.almacenHierro.añadirStock(1);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
} while (true);
}

// HERRERO //

package threads;

public class Herrero extends Thread {

    public Herrero(String nombre) {
        this.setName(nombre);
    }

    public void run() {
        do {
            try {
                // Se calcula random el tiempo de espera de cada
ejecucion
                int pausa = (int) (Math.random() * 200) + 2000;
                // El thread se espera lo calculado anteriormente
                Thread.sleep(pausa);
                /*
                * Llamamos a los metodos sincronizados para añadir o
quitar elementos de los almacenes
                */
                Main.almacenHierro.sacarStock(2);
                Main.almacenHerramientas.añadirStock(1);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } while (true);
    }
}

// TREN //

package threads;
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
public class Tren extends Thread {

    public Tren(String nombre) {
        this.setName(nombre);
    }

    public void run() {
        // Se calcula random el tiempo de espera de cada ejecucion
        int pausa = (int) (Math.random() * 5000) + 500;
        do {
            try {
                // El thread se espera lo calculado anteriormente
                Thread.sleep(pausa);
                /*
                 * Llamamos a los metodos sincronizados para añadir o
quitar elementos de los almacenes
                 */
                Main.almacenHerramientas.sacarStock(5);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } while (true);
    }
}

// ALMACEN //

package threads;

public class Almacen {

    public static final int maxCapacidad = 10;
    private int stock;
    private String nombreAlmacen;

    public Almacen(String nombreAlmacen) {
        this.nombreAlmacen = nombreAlmacen;
        this.stock = 0;
    }

    // Metodo sincronizado de añadir elementos al almacen tanto hierro o
herramientas.
    public synchronized void añadirStock(int in) throws InterruptedException {

        /*
         * Si el stock que va a introducir el herrero o el minero supera la
capacidad del almacen este se espera.
         */
        if((this.stock + in) > maxCapacidad) {
            //this.stock = maxCapacidad;
            System.err.println(Thread.currentThread().getName() + " se
espera");
            wait();
        }
    }
}
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals
Esteve Terradas i Illa
Departament d'Informàtica

```
        }else {
            this.stock += in;
            System.out.println("Se han añadido " + in + " unidades de " +
nombreAlmacen + " al almacen. Hay un total de " + stock + " en el almacen" );
            // Al añadir elementos a los almacenes este se suma y se
notifica a todos los demas thread que ha habido movimiento.
            notifyAll();
        }
    }

    // Metodo sincronizado de sacar elementos al almacen tanto hierro o
herramientas.
    public synchronized void sacarStock(int out) throws InterruptedException {

        /*
         * Si el stock que va a retirar el herrero o el tren no es suficiente
para generar una herramienta o para que el tren se lleve las herramientas este se
espera.
        */
        if((this.stock - out) < 0) {
            //this.stock = 0;
            System.err.println(Thread.currentThread().getName() + " se
espera");
            wait();
        }else {
            this.stock -= out;
            System.out.println("Se han sacado " + out + " unidades de " +
nombreAlmacen + ".Quedan " + stock + " unidades.");
            // Al poder quitar elementos de los almacen este se resta y se
notifica a todos los demas thread que ha habido movimiento.
            notifyAll();
        }
    }

}

// FOTOS //
```



PRUEBA CON 4 MINEROS Y 2 HERREROS, EL HERRERO NO SALE PORQUE LOS MINEROS GENERAN MUCHO HIERRO Y SIEMPRE TIENE TRABAJO.

```
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 1 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 2 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 3 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 4 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 5 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 6 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 7 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 8 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 6 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 1 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 7 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 8 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 6 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 2 en el almacen
TREN se espera
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 7 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 8 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 9 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 10 en el almacen
MINERO 0 se espera
MINERO 2 se espera
MINERO 1 se espera
Se han sacado 2 unidades de Hierro.Quedan 8 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 3 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 9 en el almacen
```

PRUEBA HECHA CON 1 MINERO Y 2 HERREROS. LOS HERREROS Y EL TREN ESPERAN AMENUDO PORQUE SOLO HAY 1 MINERO GENERANDO HIERRO.

```
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 1 en el almacen
TREN se espera
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 2 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 0 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 1 en el almacen
HERRERO 0 se espera
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 1 en el almacen
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 2 en el almacen
TREN se espera
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 2 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 3 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 1 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 3 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 2 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 0 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 4 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 1 en el almacen
TREN se espera
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 2 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 3 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 4 en el almacen
Se han añadido 1 unidades de Hierro al almacen. Hay un total de 5 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 3 unidades.
Se han añadido 1 unidades de Herramientas al almacen. Hay un total de 5 en el almacen
Se han sacado 2 unidades de Hierro.Quedan 1 unidades.
```



Generalitat de Catalunya
Departament d'Educació
Institut d'Educació Secundària i Superior
d'Ensenyaments Professionals

Esteve Terradas i Illa
Departament d'Informàtica

LOS THREAD SE EJECUTAN CORRECTAMENTE Y ESPERAN SU TURNO PARA ENTRAR AL METODO, A DEMAS DE CONTROLAR SI PUEDEN O NO RECOGER O AÑADIR ELEMENTOS A LOS ALMACENES Y ESPERAN A QUE UN THREAD VAYA HACIENDO MOVIMIENTOS.