Portafolio barbero

Los objetos de condicion usados son:

barbero: despierta al barbero cuando entra un cliente y cuando no hay cliente en la sala y en la silla, se duerme.

Silla: se usa para indicar que el barbero esta afeitando a un cliente, por lo que los clientes nuevos que llegan deben de esperarse en la sala de espera.

sala_de_espera: es la cola de clientes que esta esperando al barbero, se desbloquea cuando llama el metodo siguienteCliente().

Codigo fuente:

```
import java.util.Random;
import monitor.*;
class Barberia extends AbstractMonitor
{
	private Condition barbero=makeCondition();
	private Condition silla=makeCondition();
	private Condition sala_de_espera=makeCondition();

	public void CortarPelo(){
		enter();
		if(silla.isEmpty()){
			System.out.println("barbero empieza a afeitar cliente");
			barbero.signal();
			silla.await();
		}
		else{
			System.out.println("hay cliente afeitando, silla ocupada");
			sala_de_espera.await();
		}
		leave();
	}

	public void siguienteCliente(){
		enter();
		if(silla.isEmpty() && sala_de_espera.isEmpty()){
			barbero.await();
		}
		else{
			System.out.println("entra el siguiente cliente de la sala de espera");
			sala_de_espera.signal();
		}
		leave();
	}
```

```
	public void finCliente(){
		enter();
		System.out.println("el barbero ha terminado de afeitar al client ");
		silla.signal();
		leave();
	}
}

class Cliente implements Runnable{
	public Thread thr;
	private Barberia barberia;

	public Cliente(Barberia miBarberia){
		barberia=miBarberia;
		thr=new Thread(this, "Cliente");
	}
	public void run(){
		while(true){
			barberia.CortarPelo();
			aux.dormir_max(2000);
		}
	}
}

class Barbero implements Runnable{
	public Thread thr;
	private Barberia barberia;

	public Barbero(Barberia miBarberia){
		barberia=miBarberia;
		thr=new Thread(this,"Barbero");
	}
	public void run(){
		while(true){
			barberia.siguienteCliente();
			aux.dormir_max(2500);
			barberia.finCliente();
		}
	}
}

class BarberoDurmiente{
	public static void main(String[] args){
		final int NUM=6;
		Barberia barberia=new Barberia();
		Barbero barbero = new Barbero(barberia);
		Cliente[] clientes=new Cliente[NUM];

		for(int i=0;i<NUM;i++){
			clientes[i]=new Cliente(barberia);
		}
```

```
                barbero.thr.start();
                for(int i=0;i<NUM;i++){
                        clientes[i].thr.start();
                }

        }
}
```

Salida del programa:

```
yang@yang-VirtualBox:~/Escritorio/p2$ java BarberoDurmiente
barbero empieza a afeitar cliente
hay cliente afeitando, silla ocupada
hay cliente afeitando, silla ocupada
hay cliente afeitando, silla ocupada
hay cliente afeitando, silla ocupada
hay cliente afeitando, silla ocupada
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
hay cliente afeitando, silla ocupada
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
hay cliente afeitando, silla ocupada
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
hay cliente afeitando, silla ocupada
hay cliente afeitando, silla ocupada
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
hay cliente afeitando, silla ocupada
el barbero ha terminado de afeitar al cliente
entra el siguiente cliente de la sala de espera
barbero empieza a afeitar cliente
```