

1. Se tiene una impresora que recibe trabajos de 100 terminales. La clase Impresora está representada por el siguiente código:

```
class Impresora {
    int mId;
    void dormir() {
        try {
            long tiempoEspera = (long)(Math.random() * 2000.0);
            Thread.sleep(tiempoEspera);
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
    public Impresora(int id) {
        mId = id;
    }
    public void imprimir(String texto) {
        System.out.println("Impresora " + mId + ": imprimiendo: " + texto);
        dormir();
        System.out.println("Impresora " + mId + ": Texto impreso " + texto);
    }
}
```

Los terminales son hilos que invocan el método imprimir() de la clase impresora y se duermen un tiempo aleatorio hasta que vuelven a solicitar el uso de la impresora. La ejecución de los hilos es constante y no se paran. Cuando terminan de usar la impresora y después de su siesta, volverán siempre a volver a solicitarla. Usando semáforos y sin modificar la clase Impresora haga que los 100 terminales usen la impresora de forma que sólo la pueda usar uno cada vez.

2. Modifique el ejercicio anterior, y la clase Impresora, de forma que la impresora también sea un hilo que está esperando que le lleguen trabajos.
3. Modifique el ejercicio anterior para que la clase Impresora tenga un método “boolean imprimiendo()” que devuelva true o false si la impresora se encuentra imprimiendo.
4. Modifique el ejercicio anterior para que haya 10 impresoras disponibles.

Pistas:

Tendrá que crear un array con las 10 impresoras que debe estar protegida por un semáforo. El método imprimir() de la clase Impresora también debe estar protegida por un semáforo. Los hilos deberán comprobar las impresoras de forma que deben encontrar una disponible. Si no hay impresoras disponibles, simplemente pierden el turno de imprimir y vuelven a dormirse repitiendo el proceso de nuevo cuando despierten.