

1. Sincroniza tres hilos de tal manera que escriban la secuencia infinita *uno, dos, tres, uno, dos, tres...* Cada uno de estos tres hilos se encarga un número de manera coordinada con el resto. La serie, indistintamente de cuál sea el hilo que comience, ha de empezar con *uno*. La clase *EscribeUnoDosTres* contendrá al menos tres métodos *escribeUno()*, *escribeDos()* y *escribeTres()* que serán invocados desde los Threads. El programa principal debe ser el siguiente:

```
public class Main
{
    public static void main(String[] args)
    {
        EscribeUnoDosTres escribeUnoDosTres = new EscribeUnoDosTres();
        Thread thUno = new Thread(new EscribirUno(escribeUnoDosTres));
        Thread thDos = new Thread(new EscribirDos(escribeUnoDosTres));
        Thread thTres = new Thread(new EscribirTres(escribeUnoDosTres));
        thTres.start();
        thDos.start();
        thUno.start();
    }
}
```

2. Se dispone de una caja fuerte con N lingotes de oro y acceden a robar dos ladrones (1 y 2) que deciden repartirse el botín de la siguiente manera:
 - o Empezando por el ladrón 1, tirarán por turnos un dado y extraerán de la caja si es posible, el número que haya salido.
 - o En el momento en el que el ladrón no pudiera extraer el número de lingotes indicado en el dado, deja de participar y el otro ladrón seguirá el mismo proceso pero en solitario.

Ejemplos de ejecución:

Número de lingotes iniciales en la caja fuerte: 10
El ladrón 1 ha sacado 3.
El ladrón 2 ha sacado 2.
El ladrón 1 no puede sacar 6 y finaliza.
El ladrón 2 ha sacado 4.
El ladrón 2 no puede sacar 6 y finaliza.

Número de lingotes iniciales en la caja fuerte: 6
El ladrón 1 ha sacado 6.
El ladrón 2 no puede sacar 5 y finaliza.
El ladrón 1 no puede sacar 5 y finaliza.

