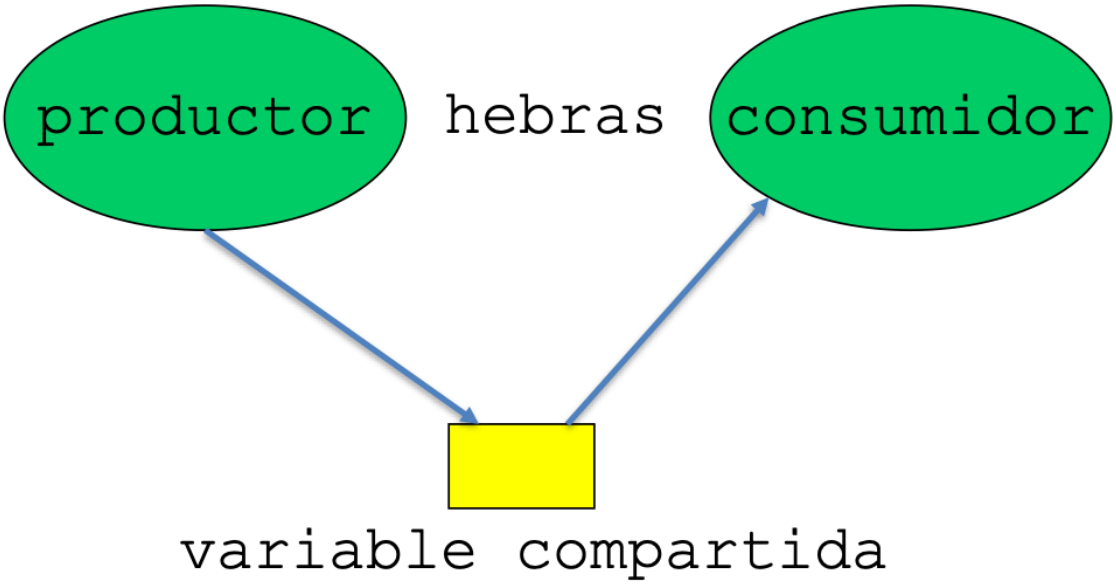
UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación

Programación de Sistemas y Concurrencia
Tema 6 – Semáforos-1

1. (jardines) Tenemos un jardín con N puertas en un jardín por el que entran visitantes, pero nunca salen, ¡muahahah! Hay que contar el número de visitantes que han entrado al final de la jornada. Clases:
    - **Contador**: Recurso compartido que usan las hebras para notificar que ha entrado una persona. Los tipos primitivos se definen como volátiles para que no se haga cache y tengamos problemas de actualización.
    - **Puerta**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a iterar e incrementar en 1 el número de personas que hay en **Contador**.
    - **Principal**: Clase que contiene el *main* y que crea las hebras, las ejecuta y espera a que finalice para mostrar el valor total que tiene Contador.
2. (productorconsumidor) Tenemos una hebra productor que está escribiendo números y otra hebra consumidor que los lee. El productor no puede producir otro dato hasta que no se ha consumido. El consumidor no puede leer el dato hasta que no se ha producido. Clases:
    - **RecursoCompartido**: Lo usan las hebras para intercambiar el recurso (volatile int).
    - **Productor**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir produciendo datos (10 en concreto).
    - **Consumidor**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir consumiendo datos (10 en concreto).
    - **Principal**: Clase que contiene el *main* y que crea las hebras y las ejecuta y termina.



3.

4. (sensores) En un sistema industrial existen tres sensores que realizan mediciones del nivel de temperatura, humedad y luz respectivamente. Cuando se han recogido mediciones de los tres sensores, existe un dispositivo "trabajador" encargado de realizar ciertas tareas según las mediciones realizadas.
El dispositivo **no puede** comenzar a realizar sus tareas hasta que se han recogido mediciones de los tres sensores, y los sensores **no pueden** volver a realizar mediciones hasta que el dispositivo trabajador finaliza sus tareas. El proceso se repite de forma indefinida de manera que cuando el dispositivo finaliza sus tareas, volverá a esperar a que haya mediciones de los tres sensores.

Realizar utilizando semáforos generales el modelado de dicho sistema. Modelar el dispositivo trabajador y cada sensor como una hebra (con lo cual habrá un total de 4 hebras). Modelar el proceso de realizar mediciones y las tareas del dispositivo con retrasos aleatorios y valores de tipo entero. Inicialmente puede suponerse que los sensores pueden comenzar haciendo peticiones. Considera el número de permisos como los sensores que han terminado su lectura. Recuerda que puedes hacer reléase y acquire con un número mayor que 1. Clases:

- **Mediciones**: Lo usan las hebras para sincronizarse.
- **Sensor**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir realizando mediciones de forma **infinita (nunca para)**.
- **Trabajador**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a leer las mediciones de los sensores y finalizando sus tareas, dando pie a otra iteración **(nunca para).**
- **Principal**: Clase que contiene el *main* y que crea las hebras y las ejecuta y termina, dejando a las hebras iterando infinitamente.

5. (pajaritosbinarios) Considera un nido con **n pájaros bebés y dos pájaros padres** (el papá y la mamá). Todos los pájaros comparten un plato común que puede contener a lo sumo **B bichitos**. Cada pájaro padre (papá o mamá) da una vuelta volando, atrapa un bichito, vuelve al nido, espera a que haya sitio en el plato, deposita en él el bichito capturado, y repite todas las acciones de nuevo.

Cada pájaro bebé pía un ratito, espera a que el plato tenga algún bichito, lo coge, se lo come y repite de nuevo todas las acciones.

Implementa este sistema utilizando semáforos binarios, suponiendo un comportamiento infinito para cada uno de los procesos. Clases:

- **Nido**: Lo usan las hebras para sincronizarse y controlar el número de bichitos que hay en el nido.
- **Adulto**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir introduciendo bichitos, si puede, en el nido, de forma **infinita (nunca para)**.
- **Bebe**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a leer comer si hay bichitos **(nunca para).**
- **Principal**: Clase que contiene el *main* y que crea las hebras y las ejecuta y termina, dejando a las hebras iterando infinitamente.

6. (fumadores) Considera un sistema formado por tres hebras fumadores que se pasan el día liando cigarros y fumando. Para fumar un cigarro necesitan tres ingredientes: tabaco, papel y cerillas. Cada fumador dispone de un surtido suficiente (para el resto de su vida) de uno de los tres ingredientes. Cada fumador tiene un ingrediente diferente, es decir, un fumador tiene una cantidad infinita de tabaco, el otro de papel y el otro de cerillas. Hay también una hebra agente que pone dos de los tres ingredientes encima de una mesa. El agente dispone de unas reservas infinitas de cada uno de los tres ingredientes y escoge de forma aleatoria cuáles son los ingredientes que pondrá encima de la mesa. Cuando los ha puesto, el fumador que tiene el otro ingrediente puede fumar (los otros dos no). Para ello coge los ingredientes, se lía un cigarro y se lo fuma. Cuando termina de fumar vuelve a repetirse el ciclo. En resumen, el ciclo que debe repetirse es:

"agente pone ingredientes - fumador hace cigarro - fumador fuma - fumador termina de fumar - agente pone ingredientes - ... "

Es decir, en cada momento a lo sumo hay un fumador fumando un cigarrillo. Clases:

- **Mesa**: Lo usan las hebras para sincronizarse y controlar el número de bichitos que hay en el nido.
- **Agente**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir introduciendo nuevos ingredientes (valor 0, 1 o 2) y esperar a que le toque de nuevo poner **(nunca para).**
- **Fumador**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a intentar fumar, cuando tenga su ingrediente (corresponde con el id), fumará, y luego llamara a finfumar **(nunca para).**
- **Principal**: Clase que contiene el *main* y que crea las hebras y las ejecuta y termina, dejando a las hebras iterando infinitamente.

7. (sensoresnbinarios) En un sistema industrial existen tres sensores que realizan mediciones del nivel de temperatura, humedad y luz respectivamente. Cuando se han recogido mediciones de los tres sensores, existe un dispositivo "trabajador" encargado de realizar ciertas tareas según las mediciones realizadas.

El dispositivo **no puede** comenzar a realizar sus tareas hasta que se han recogido mediciones de los tres sensores, y los sensores **no pueden** volver a realizar mediciones hasta que el dispositivo trabajador finaliza sus tareas. El proceso se repite de forma indefinida de manera que cuando el dispositivo finaliza sus tareas, volverá a esperar a que haya mediciones de los tres sensores.

Realizar utilizando semáforos **binarios** el modelado de dicho sistema. Modelar el dispositivo trabajador y cada sensor como una hebra (con lo cual habrá un total de 4 hebras). Modelar el proceso de realizar mediciones y las tareas del dispositivo con retrasos aleatorios y valores de tipo entero. Inicialmente puede suponerse que los sensores pueden comenzar haciendo peticiones. Clases.

- **Nido**: Lo usan las hebras para sincronizarse y controlar el número de bichitos

que hay en el nido.

- **Adulto**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a ir introduciendo bichitos, si puede, en el nido, de forma **infinita (nunca para)**.
- **Bebe**: Clase que implementa la interfaz *Runnable* que cuando se ejecute va a leer comer si hay bichitos **(nunca para).**
- **Principal**: Clase que contiene el *main* y que crea las hebras y las ejecuta y termina, dejando a las hebras iterando infinitamente.