**Estacionamiento con hilos**

**Productor-Consumidor**

Deberá de implementar un programa concurrente que simule el comportamiento de un estacionamiento con **n** entradas y **m** salidas, tomando en cuenta las siguientes consideraciones.

**Datos que introduce el usuario**:

1. Número de entradas.
2. Número de salidas.
3. Capacidad del estacionamiento.
4. Número de hilos generadores de autos por entrada

**Condiciones:**

1. Las filas de entrada de autos al estacionamiento son llenadas por los hilos generadores (productores) de automóviles:
2. Si el usuario indica que se necesitan n hilos generadores de autos por entrada, entonces cada fila de autos será llenada por n hilos productores de autos. Es decir, se crearan n hilos productores para la entrada 1, n hilos productores para la entrada 2, etc.
3. Los autos se generan de forma aleatoria (0-100), si el hilo productor genera un número aleatorio entre 0-30 entonces el hilo produce un automóvil caso contrario NO se genera automóvil.
4. Deberá mandar a dormir al hilo un tiempo aleatorio entre 1-300, esto es para darle tiempo a los hilos (alentar el programa). Recuerda que esta no es una sincronización.
5. A cada entrada se le asocia una fila de autos:

Esta fila se comporta como una **cola** implementada con una **lista simplemente ligada** de objetos tipo automóvil, cada vez que se inserta o extrae un auto de la lista se deberá de listar su contenido indicando a que entrada está asociada, por ejemplo:

“el hilo x genero el automóvil w en la fila de la entrada z” y se lista el contenido de la misma.

1. El estacionamiento se modela con:
   1. Nivel 1: **variable** que funciona como un contador de lugares, si entra un automóvil se incrementa la variable, si sale un automóvil se decrementa la variable.
   2. Nivel 2: una **cola circular estática** que almacena objetos tipo Automóvil. Si entra un automóvil, se inserta a la estructura cola, si sale u automóvil se elimina de la estructura cola.
2. Cada entrada es un solo hilo (consumidor-productor):

Por lo que si se tienen M entradas se crearan M hilos, uno para cada entrada.

El hilo deberá de extraer un auto de la cola de entrada de autos (roll-consumidor) y si hay espacio en el estacionamiento insertarlo (roll-productor) en la cola circular estática. Indicando:

“Por la entrada z entra el automóvil w, quedan # lugares en el estacionamiento” y listar el contenido de la fila.

1. Las salidas son hilos consumidores:

Cada salida es un hilo consumidor, para determinar si sale o no un auto deberá de generar un numero aleatorio entre 0-100, si el número es < 30 entonces sale un auto del estacionamiento por esta salida.

Cada vez que sale un automóvil del estacionamiento (nivel 1: decrementar la variable, nivel 2: extraer un auto de la cola circular estática) deberá de indicar por que salida (nombre del hilo) y que auto sale, así como la nueva capacidad disponible del estacionamiento.

Nivel 1:

“Salida x: sale el automóvil w, y quedan # lugares disponibles en el estacionamiento”

Nivel 2:

“Salida x: sale el automóvil w, y quedan # lugares disponibles en el estacionamiento” y listar el estacionamiento.

Nivel 3:

A las salidas también se les asocia una fila de salida representada por una **cola dinámica** (nodos enlazados). Estos hilos además generan un número aleatorio que indica cuantos autos se formarán en esta fila para salir del estacionamiento.

* Listar la fila de la salida x
* Imprimir el mensaje “Salida x: sale el automóvil w, y quedan # lugares disponibles en el estacionamiento”
* Listar el estacionamiento.

**Criterios de evaluación:**

1. Recuerda **No usar los TAD (listas, colas, etc..) de java, pues ya están sincronizados**
2. Los códigos con región crítica deben de estar sincronizados.
3. Los códigos NO deberán de ir documentados. Si tu proyecto esta con código documentado su calificación será cero.
4. Se entrega el examen en clase, la evaluación es individual y alumnos que no se presenten con su equipo se les asignará cero de calificación.
5. Se debe de enviar a la plataforma para evidencia y evaluación.