Tarea 2.1 PSP

Para el siguiente ejercicio crearemos las siguientes clases de java:

Buffer.java

Consumidor.java

Productor.java

Programa.java (main.class)

## Clase Buffer:

Este código implementa un buffer con operaciones sincronizadas para la producción y el consumo de caracteres. El buffer tiene un tamaño específico y se asegura de que las operaciones de producción y consumo sean seguras en un entorno multihilo.

* Declaración de variables:
  + bufferLleno y bufferVacio indican si el buffer está lleno o vacío, respectivamente.
  + buffer es un arreglo de caracteres que actúa como el buffer.
  + next es un índice que apunta al próximo elemento del buffer.
* Constructor:
  + Inicializa las variables de instancia, crea un nuevo arreglo de caracteres con el tamaño especificado y establece el índice inicial.
* Método consumir():
  + Permite consumir un elemento del buffer.
  + Espera activamente si el buffer está vacío.
  + Decrementa el índice y actualiza el estado del buffer.
  + Notifica a los demás hilos que están esperando.
* Método producir():
  + Permite producir un elemento en el buffer.
  + Espera activamente si el buffer está lleno.
  + Agrega el nuevo elemento, actualiza el estado y el índice.
  + Si el buffer está lleno, actualiza el estado.
  + Notifica a los demás hilos que están esperando.

## Clase Productor:

Este código representa una clase Productor que extiende la clase Thread y se encarga de producir caracteres y depositarlos en un objeto Buffer. Aquí tienes un resumen:

* Variables de instancia:
  + buffer: Objeto Buffer en el que se depositarán los caracteres producidos.
  + letras: String que contiene todas las letras minúsculas del alfabeto.
  + produced: Contador de la cantidad de caracteres producidos.
  + LIMIT: Límite de caracteres que el productor debe producir (15 en este caso).
* Constructor:
  + Inicializa el contador produced y guarda el objeto Buffer proporcionado.
* Método run():
  + Método que se ejecuta cuando el hilo del productor es iniciado.
  + Utiliza un bucle while para continuar produciendo hasta alcanzar el límite (LIMIT).
  + Genera un carácter aleatorio de la cadena letras.
  + Llama al método producir del objeto Buffer para depositar el carácter.
  + Incrementa el contador de caracteres producidos y muestra un mensaje en la consola.
  + Introduce una pausa aleatoria para simular la producción intermitente.
  + Maneja excepciones de interrupción.

En resumen, esta clase representa un productor que genera caracteres aleatorios y los deposita en un buffer compartido. La producción es controlada por un límite, y se introduce una pausa aleatoria para simular una producción intermitente.

## Clase Consumidor:

La clase Consumidor extiende Thread y está diseñada para consumir caracteres de un objeto Buffer. En su método run, utiliza un bucle para consumir caracteres hasta alcanzar un límiteVariables de instancia:

buffer: Objeto Buffer del cual se consumirán los caracteres.

consumed: Contador de la cantidad de caracteres consumidos.

LIMIT: Límite de caracteres que el consumidor debe consumir (en este caso, 15).

Constructor:

Inicializa el contador consumed y guarda el objeto Buffer proporcionado.

Método run():

Método que se ejecuta cuando el hilo del consumidor es iniciado.

Utiliza un bucle while para continuar consumiendo hasta alcanzar el límite (LIMIT).

Llama al método consumir del objeto Buffer para obtener un carácter.

Incrementa el contador de caracteres consumidos y muestra un mensaje en la consola.

Introduce una pausa aleatoria para simular un consumo intermitente.

Maneja excepciones de interrupción.

En resumen, la clase Consumidor representa un consumidor que retira caracteres de un buffer compartido. El consumo se realiza hasta alcanzar un límite, y se introduce una pausa aleatoria para simular un consumo intermitente. predefinido

## Clase Programa(main)

La clase Programa realiza lo siguiente:

* Creación de objetos:
  + Se crea un objeto Buffer con capacidad para 6 elementos.
  + Se crean instancias de las clases Productor y Consumidor pasándoles el mismo objeto Buffer.
* Iniciación de hilos:
  + Se inicia el hilo del productor (Productor) y se espera 3 segundos.
  + Luego, se inicia el hilo del consumidor (Consumidor).
* Espera de finalización de hilos:
  + Se espera a que ambos hilos (Productor y Consumidor) finalicen antes de continuar.
* Mensaje de finalización:
  + Se imprime en la consola un mensaje indicando que el programa ha terminado.

En resumen, el programa principal crea un buffer compartido, inicia un hilo productor y espera brevemente antes de iniciar un hilo consumidor. Luego, espera a que ambos hilos terminen y finaliza el programa.