Trabajo Grupal

Nombre: Erick Rodas, Andy Veintimilla, María Ángel Rojas, Jordy Betancourth, Jeimy Torres.

Prompts Utilizadas:

1. Problema 1 (Race Condition):

Analiza este código y genera una explicación de su funcionamiento.

```
package ec.edu.utpl.carreras.computacion.troublea;

/**
 * Antes de ejecutar el programa debe:
 * - Analizarlo y comprender su funcionamiento. Responder a preguntas
así:
 *      - ¿Cuántos hilos se crean?
 *      - Se crea un total de 3 hilos
 *      - ¿Qué hace cada hilo?
 *      - Ambos hilos ejecutran la misma tarea, "task"
 * - Determinar cuál es el valor del atributo counter al finalizar la
ejecución del programa. Registrar las respuestas.
 * -14895
 * -11736
 * -14465
 * -14839
 * -13176
 * Ejecute el programa al menos 5 veces y responda a las preguntas:
 * - En cada ejecución se imprimió el mismo valor para counter
 * - No, en cada ejecución se imprimió un valor diferente para counter
 * - Existe diferencia entre el valor que imprime el programa y los
valores que registraron anteriormente.
 * - Si existe una diferencia, nuestros valores previstos esperaban un
valor fijo y que no fluctué
 *
 * Ahora use IA generativa, para solicitar explicación del propósito
de alguna sentencia o del funcionamiento del programa.
 * Además, para que le explique el problema (suministre el código en
el prompt), obtener el nombre del problema
 * y una solución al mismo. No olvide pedirle una versión del código
con la solución.
 *
 */

public class RCndtnExample {
    private static int counter = 0;

    public static void main(String[] args) throws InterruptedException
{
      Runnable task = () -> {
          for(var i = 0; i < 10_000; i ++) {
              counter ++;
          }
      };
      var thread1 = new Thread(task);
      var thread2 = new Thread(task);

      thread1.start();
      thread2.start();

      thread1.join();
```

```
        thread2.join();

        System.out.printf("Counter: %d%n", counter);
    }
}
```

Indaga sobre el problema y dame el nombre, explicacion y solucion de este.

Este programa presenta un problema de programacion concurrente, dame la solucion a este problema.

2. **Problema 2 (DeadLock)**

Para este programa, por el momento solo generame la explicacion del codigo en un solo bloque de texto

```
package ec.edu.utpl.carreras.computacion.troubleb;

/**
 * Ejecute el programa varias veces (al menos 5) y responda a las
pregutas:
 * - En alguna ejecución ¿el programa concluye?
 * - No, el programa no concluye en ninguna ejecucion
 * - El orden de presentación de los mensajes ¿se mantiene?
 * - El orden de los mensajes se mantiene a lo largo de estas 5
ejecuciones
 *
 * Ahora use IA generativa, para solicitar explicación del propósito
de alguna sentencia o del funcionamiento del programa.
 * Además, para que le explique el problema (suministre el código en
el prompt), obtener el nombre del problema
 * y una solución al mismo. No olvide pedirle una versión del código
con la solución.

 */

public class DdlckExample {
    private static final Object lock1 = new Object();
    private static final Object lock2 = new Object();

    public static void main(String[] args){
      Thread thread1 = new Thread(() -> {
          synchronized (lock1) {
              System.out.println("Thread 1: Acquired lock1");
              try{ Thread.currentThread().sleep(100);} catch
(InterruptedException _){}
              synchronized (lock2) {
                  System.out.println("Thread 1: Acquired lock2");
              }
          }
      });

      Thread thread2 = new Thread(() -> {
          synchronized (lock2) {
              System.out.println("Thread 2: Acquired lock2");
              try{ Thread.currentThread().sleep(100);} catch
(InterruptedException _){}
              synchronized (lock1) {
                  System.out.println("Thread 2: Acquired lock1");
              }
```

```
            }
        });
        thread1.start();
        thread2.start();
    }
}
```

Identifica este problema de programacion concurrente, dame el nombre, una explicacion breve y un codigo con la solución.

3. Problema 3 (Starvation)

Dame solamente la explicación de este programa, sin indagar el problema ni la solución

```
package ec.edu.utpl.carreras.computacion.troublec;

import java.util.concurrent.locks.ReentrantLock;

/**
 * Ejecute el programa por 5 segundos y responda a la preguta:
 * - ¿Qué grupo de hilos aparentemente se repiten más los pares o los
impares?
 * - Los pares
 * - ¿Por qué?
 * - Porque los pares tienen prioridad mayor que los impares
 *
 * Ahora use IA generativa, para solicitar explicación del propósito
de alguna sentencia o del funcionamiento del programa.
 * Además, para que le explique el problema (suministre el código en
el prompt), obtener el nombre del problema
 * y una solución al mismo. No olvide pedirle una versión del código
con la solución.

 */

public class StrvtnExample {
    private static final ReentrantLock lock = new ReentrantLock();

    public static void main(String[] args){
      Runnable task = () -> {
        while(true) {
            lock.lock();
            try{
                System.out.printf("%s acquired the lock.%n",
Thread.currentThread().getName());
            } finally {
                lock.unlock();
            }
        }
      };

      for(var i = 0; i < 10; i++) {
          Thread t = new Thread(task, "Hilo-" + i);
          if( i % 2 == 0) t.setPriority(Thread.MAX_PRIORITY);
          else t.setPriority(Thread.MIN_PRIORITY);
          t.start();
      }
    }
}
```

Ahora si, indaga sobre este problema de programacion concurrente y danos el nombre, la explicacion, la solucion y el codigo solucionado.

este problema deberia ejecturase indefinidamente?

4. **Problema 4 (LiveLock)**

Generame una explicacion enteramente de este programa

```
package ec.edu.utpl.carreras.computacion.troubled;

import java.util.concurrent.locks.ReentrantLock;

/**
 * Ejecute el programa por lo menos 5 veces y por cada ejecución copie
todos los mensajes de salida. Responda a la pregunta:
 * - ¿En qué número de línea aparecen los mensajes: Thread-1 adquirió
segundo lock, va a proceder y Thread-2 adquirió segundo lock, va a
proceder?
 * - Primer mensaje: 12, 10, 6, 6, 8
 * - Segundo mensaje: 14, 8, 3, 3, 10
 * - ¿Es el mismo número de línea o cambia?
 * - El numero de linea si cambia
 * Ahora use IA generativa, para solicitar explicación del propósito
de alguna sentencia o del funcionamiento del programa.
 * Además, para que le explique el problema (suministre el código en
el prompt), obtener el nombre del problema
 * y una solución al mismo. No olvide pedirle una versión del código
con la solución.
 */


public class LvlckExample {
    private static final ReentrantLock lock1 = new ReentrantLock();
    private static final ReentrantLock lock2 = new ReentrantLock();

    public static void main(String[] args) {
        Runnable r1 = () -> attemptBoth(lock1, lock2, "Thread-1");
        Runnable r2 = () -> attemptBoth(lock2, lock1, "Thread-2");

        new Thread(r1).start();
        new Thread(r2).start();
    }

    private static void attemptBoth(ReentrantLock first, ReentrantLock
second, String name) {
        while (true) {
            // intenta coger el primer lock
            if (!first.tryLock()) {
                Thread.yield();
                continue;
            }
            System.out.println(name + " adquirió primer lock");

            // corta espera para simular trabajo
            try { Thread.sleep(50); } catch (InterruptedException
ignored) {}

            // intenta el segundo lock
            if (!second.tryLock()) {
                // no lo consiguió: libera el primero y vuelve a
```

```
intentar
                first.unlock();
                System.out.println(name + " libera primer lock y
cede");
                Thread.yield();
            } else {
                // ¡ambos locks conseguidos! sale del bucle
                System.out.println(name + " adquirió segundo lock, va
a proceder");
                break;
            }
        }

        // sección crítica…
        // al terminar, liberar ambos
        second.unlock();
        first.unlock();
    }
}
```

Indaga sobre el problema y retorname el nombre, explicacion y solucion de este, luego dame un codigo con la solucion

Pero estos hilos seguirian ejecutandose simultaneamente?