Bai Tap ve Thread trong Java - Code va Giai Phap

Bai 1: Race Condition

Yeu cau:

Viet mot chuong trinh Java trong do hai luong cung tang gia tri cua mot bien counter len 1000 lan.

Kiem tra xem ket qua cuoi cung co phai 2000 khong? Neu khong, hay sua loi.

Code loi (Race Condition):

```java
class Counter {
    int count = 0;

    public void increment() {
        count++;
    }
}

public class RaceConditionExample {
    public static void main(String[] args) throws InterruptedException {
        Counter counter = new Counter();

        Thread t1 = new Thread(() -> {
            for (int i = 0; i < 1000; i++) counter.increment();
        });

        Thread t2 = new Thread(() -> {
            for (int i = 0; i < 1000; i++) counter.increment();
        });

        t1.start();
        t2.start();
        t1.join();
        t2.join();

        System.out.println("Final count: " + counter.count);
    }
}
```

Cach sua loi: Dung synchronized hoac AtomicInteger.

Code fix:

```java
class Counter {
    private int count = 0;

    public synchronized void increment() {
        count++;
    }

    public synchronized int getCount() {
        return count;
    }

}
```

Bai 2: Deadlock

Yeu cau:

Viet mot chuong trinh co hai luong, moi luong co gang khoa hai tai nguyen theo thu tu nguoc nhau, gay ra deadlock.

Sau do sua loi.

Code loi (Deadlock):

```java
class DeadlockExample {
    static final Object resource1 = new Object();
    static final Object resource2 = new Object();

    public static void main(String[] args) {
        Thread t1 = new Thread(() -> {
            synchronized (resource1) {
                System.out.println("Thread 1: Locked resource 1");
                try { Thread.sleep(100); } catch (InterruptedException e) {}
                synchronized (resource2) {
                    System.out.println("Thread 1: Locked resource 2");
                }
            }
        });

        Thread t2 = new Thread(() -> {
            synchronized (resource2) {
                System.out.println("Thread 2: Locked resource 2");
                try { Thread.sleep(100); } catch (InterruptedException e) {}
                synchronized (resource1) {
                    System.out.println("Thread 2: Locked resource 1");
                }
            }
        });

        t1.start();
        t2.start();
    }
}
```

Cach sua loi: Giu thu tu khoa nhat quan hoac dung tryLock().

Bai 3: Starvation

Yeu cau:

Viet mot chuong trinh su dung Thread Priority, trong do mot luong co do uu tien cao lien tuc chiem CPU, lam cho luong co do uu tien thap khong bao gio chay.

Code loi (Starvation):

```java
public class StarvationExample {
    public static void main(String[] args) {
        Thread highPriority = new Thread(() -> {
            while (true) System.out.println("High priority thread running...");
        });

        Thread lowPriority = new Thread(() -> {
            while (true) System.out.println("Low priority thread running...");
        });

        highPriority.setPriority(Thread.MAX_PRIORITY);
        lowPriority.setPriority(Thread.MIN_PRIORITY);

        highPriority.start();
        lowPriority.start();
    }
}
```

Cach sua loi: Dung Thread.yield() hoac Fair Lock.

Bai 4: Thread Interruption

Yeu cau:

Viet chuong trinh co mot luong thuc thi vong lap vo han. Sau 3 giay, dung luong dung cach bang interrupt().

Code:

```java
class InterruptExample {
    public static void main(String[] args) throws InterruptedException {
        Thread worker = new Thread(() -> {
            while (!Thread.currentThread().isInterrupted()) {
                try {
                    System.out.println("Worker is running...");
                    Thread.sleep(500);
                } catch (InterruptedException e) {
                    System.out.println("Worker thread interrupted!");
                    Thread.currentThread().interrupt();
                }
            }
            System.out.println("Worker stopped.");
        });

        worker.start();
        Thread.sleep(3000);
        worker.interrupt();
    }
}
```

Bai 5: Memory Visibility Issue

Yeu cau:

Tao mot bien flag trong mot luong, va thay doi gia tri trong mot luong khac. Kiem tra xem luong dau tien co nhin thay su thay doi khong.

Code loi (Memory Visibility Issue):

```java
```

```
class VisibilityExample {
    private static boolean flag = false;

    public static void main(String[] args) {
        new Thread(() -> {
            while (!flag) {}
            System.out.println("Flag changed!");
        }).start();

        try { Thread.sleep(2000); } catch (InterruptedException e) {}

        flag = true; // Luong khac co the khong thay su thay doi nay
    }
}
```

Cach sua loi: Dung tu khoa volatile.

```java
private static volatile boolean flag = false;
```