

NUR AZYYATI BITNI ABU BAKAR | 291560

TUTORIAL9: SAFELOCK & REENTRANTLOCK

THE CODE:

SafeLock.java

```
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;
import java.util.Random;

public class SafeLock {
    static class Friend {
        private final String name;
        private final Lock lock = new ReentrantLock();

        public Friend(String name) {
            this.name = name;
        }

        public String getName() {
            return this.name;
        }

        public boolean impendingBow(Friend bower) {
            boolean myLock = false;
            boolean yourLock = false;
            try {
                myLock = lock.tryLock();
                yourLock = bower.lock.tryLock();
            } finally {
                if (!(myLock && yourLock)) {
                    if (myLock) {
                        lock.unlock();
                    }
                    if (yourLock) {
                        bower.lock.unlock();
                    }
                }
            }
            return myLock && yourLock;
        }

        public void bow(Friend bower) {
            if (impendingBow(bower)) {
                try {
                    System.out.format("%s: %s has bowed to me!\n",
                        this.name, bower.getName());
                    bower.bowBack(this);
                } finally {
                    lock.unlock();
                    bower.lock.unlock();
                }
            } else {
                System.out.format("%s: %s started to bow to me, but saw that I was already bowing to %s.\n",
                    this.name, bower.getName(), this.name);
            }
        }
    }
}
```

```

    public void bowBack(Friend bower) { 1 usage new *
        System.out.format("%s: %s has bowed back to me!\n",
            this.name, bower.getName());
    }
}

static class BowLoop implements Runnable { 2 usages new *
    private final Friend bower; 2 usages
    private final Friend bowee; 2 usages

    public BowLoop(Friend bower, Friend bowee) { 2 usages new *
        this.bower = bower;
        this.bowee = bowee;
    }

    public void run() { new *
        Random random = new Random();
        for (;;) {
            try {
                Thread.sleep(random.nextInt( bound: 10));
            } catch (InterruptedException e) {}
            bowee.bow(bower);
        }
    }
}

public static void main(String[] args) { new *
    final Friend alphonse = new Friend( name: "Alphonse");
    final Friend gaston = new Friend( name: "Gaston");
    new Thread(new BowLoop(alphonse, gaston)).start();
    new Thread(new BowLoop(gaston, alphonse)).start();
}
}

```

THE OUTPUT:

```

Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Gaston: Alphonse has bowed to me!
Alphonse: Gaston has bowed back to me!
Gaston: Alphonse has bowed to me!
Alphonse: Gaston has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Gaston: Alphonse has bowed to me!
Alphonse: Gaston has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Gaston: Alphonse has bowed to me!
Alphonse: Gaston has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Gaston: Alphonse has bowed to me!

```