STIWK3014 REAL TIME PROGRAMMING
Tutorial / Exercise 9: Safe Lock and ReentrantLock Method

```
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"
                    + " him.%n",
                    this.name, bower.getName());
            }
        }

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

    static class BowLoop implements Runnable {
        private Friend bower;
        private Friend bowee;

        public BowLoop(Friend bower, Friend bowee) {
            this.bower = bower;
            this.bowee = bowee;
        }
```

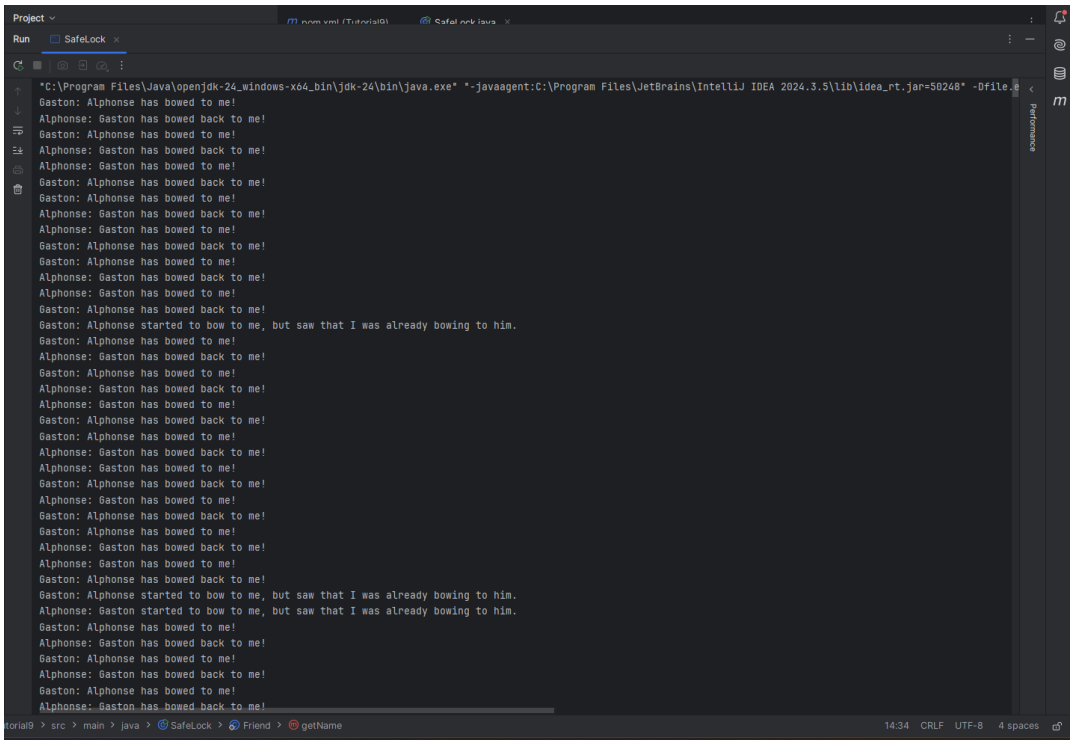
```
        public void run() {
            Random random = new Random();
            for (;;) {
                try {
                    Thread.sleep(random.nextInt(10));
                } catch (InterruptedException e) {}
                bowee.bow(bower);
            }
        }
    }

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

Coding

pom.xml (Tutorial9) × SafeLock.java ×

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

public class SafeLock {
    static class Friend {  11 usages
        private final String name;  5 usages
        private final Lock lock = new ReentrantLock();  6 usages

        public Friend(String name) {  2 usages
            this.name = name;
        }

        public String getName() {  3 usages
            return this.name;
        }

        public boolean impendingBow(Friend bower) {  1 usage
            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) {  1 usage
            if (impendingBow(bower)) {
                try {
                    System.out.format("%s: %s has bowed to me!%n", this.name, bower.getName());
                    bower.bowBack( bower: 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 him.%n",
                        this.name, bower.getName());
            }
        }

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

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

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

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

    public static void main(String[] args) {
        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();
    }
}
```

Output


Project
Run
SafeLock
"C:\Program Files\Java\openjdk-24_windows-x64_bin\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.5\lib\idea_rt.jar=50248" -Dfile.e
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!
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!
Alphonse: Gaston has bowed back to me!
Alphonse: Gaston has bowed to me!
Gaston: Alphonse has bowed back to me!
Gaston: Alphonse started to bow to me, but saw that I was already bowing to him.
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!
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 started to bow to me, but saw that I was already bowing to him.
Alphonse: Gaston started to bow to me, but saw that I was already bowing to him.
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!
Gaston: Alphonse has bowed to me!
Alphonse: Gaston has bowed back to me!
Performance
src
main
java
SafeLock
Friend
getName
14:34
CRLF
UTF-8
4 spaces

Submission:

Platform: 1. Online Learning - Sample Coding & Output in PdF form
2. GitHub – Upload the coding file to your GitHub account.

Date: 21 May 2025 (Wednesday, before 12.30 noon)