

CONCURRENCY ISSUES AND SYNCHRONIZATION – II

SYNCHRONIZED METHODS

ANY METHOD IN JAVA
CAN BE MARKED AS
SYNCHRONIZED

DOING SO MEANS THAT
ONLY ONE THREAD CAN
BE EXECUTING THIS
MEMBER FUNCTION
ON THIS OBJECT AT A
GIVEN POINT IN TIME

```
public class SynchronizedCounter {  
    private int c = 0;  
  
    public synchronized void increment() {  
        c++;  
    }  
  
    public synchronized void decrement() {  
        c--;  
    }  
  
    public synchronized int value() {  
        return c;  
    }  
}
```

REMEMBER HOWEVER THAT
THE "ONLY-1-THREAD-AT-A-TIME"
ONLY APPLIES TO THE SAME
METHOD OF THE SAME OBJECT

SO, IF FOR INSTANCE THE METHOD
DOES SOMETHING TO A STATIC
CLASS VARIABLE (NOT AN OBJECT
VARIABLE), ERRORS CAN STILL RESULT

USED RIGHT, MARKING A METHOD
AS SYNCHRONIZED CAN HELP
ELIMINATE THREAD INTERFERENCE AND
MEMORY CONSISTENCY ERRORS

SYNCHRONIZED BLOCKS OF CODE

SINCE EVERY OBJECT IN JAVA HAS AN INTRINSIC LOCK ASSOCIATED WITH IT, IT IS ACTUALLY POSSIBLE TO LOCK ANY SECTION OF CODE BY MARKING IT AS SYNCHRONIZED

IN GENERAL, ANY OBJECT CAN BE USED AS A LOCK USING THE SYNCHRONIZED STATEMENT

```
public void addName(String name) {  
    synchronized(this) {  
        lastName = name;  
        nameCount++;  
    }  
    nameList.add(name);  
}
```

IN FACT MARKING A METHOD AS SYNCHRONIZED IS MERELY A SHORTCUT TO MARKING THE ENTIRE BODY OF THE METHOD AS SYNCHRONIZED ON 'THIS', I.E. ON THE OBJECT IN QUESTION

BTW, A THREAD NEVER GETS BLOCKED ON ITSELF, WHICH MEANS THAT ONE SYNCHRONIZED METHOD OF AN OBJECT CAN ALWAYS CALL ANOTHER SYNCHRONIZED METHOD OF THE SAME OBJECT WITHOUT BLOCKING

THREAD CONTENTION

SYNCHRONIZATION AND LOCKS ARE POWERFUL
AND ANY POWER CAN BE MISUSED

DEADLOCK

TWO THREADS, EACH IS BLOCKED
ON A LOCK HELD BY THE OTHER

LIVELOCK

TWO THREADS DON'T DEADLOCK,
BUT KEEP BLOCKING ON LOCKS
HELD BY EACH OTHER. NEITHER
REALLY CAN PROGRESS

STARVATION

SOME THREADS KEEP ACQUIRING
LOCKS GREEDILY, AND CAUSE
OTHER THREADS TO BE UNABLE
TO GET ANYTHING DONE

CORPORATE
CODE IS TRULY LIKE LIFE

