# Java Week Sixteen

**public** **class** LockResourceManager **extends** BasicResourceManager

{

//new lock

**final** Lock lock = **new** ReentrantLock();

//create our array of conditions for each priority

**final** Condition[] conditions = **new** Condition[11]; //lock.newCondition();

**boolean** inUse = **false**;

**public** LockResourceManager(Resource resource, **int** maxUses)

{

**super**(resource, maxUses);

//initialise our conditions array

**for**(**int** i = 0; i < conditions.length; i++)

{

conditions[i] = lock.newCondition();

}

}

**public** **void** requestResource(**int** priority) **throws** ResourceError

{

lock.lock();

**try**

{

//if our resource is inuse

**if**(inUse)

{

//add to the number waiting

increaseNumberWaiting(priority);

//wait until resource is free

conditions[priority].await();

}

// resource is in use

inUse = **true**;

} **catch** (InterruptedException e) {

e.printStackTrace();

}

**finally**

{

lock.unlock();

}

}

**public** **int** releaseResource() **throws** ResourceError

{

lock.lock();

**try**

{

**for**(**int** i = 10; i >= 0; i--)

{

//if anything is waiting

**if**(getNumberWaiting(i) > 0)

{

//decrease the number waiting

decreaseNumberWaiting(i);

//signal to tell user it can stop waiting

conditions[i].signal();

//resource is no longer in use

inUse = **false**;

//return our user

**return** i;

}

}

//not in use

inUse = **false**;

//no one is waiting

**return** ***NONE\_WAITING***;

}

**finally**

{

lock.unlock();

}

}

}

**public** **void** test4\_1() **throws** ResourceError

{

ResourceSystem resource1 = **new** ResourceSystem();

// The resource - may be used up to 20 times

resource1.addResource("A", 20);

// User 1 uses the resource for up to 1/10 second each time

resource1.addUser("1",0.1);

// User 2 uses the resource for up to 1/10 second each time

resource1.addUser("2",0.1);

// User 3 uses the resource for up to 1/5 second each time

resource1.addUser("3",0.2);

// User 4 uses the resource for up to 1/5 second each time

resource1.addUser("4",0.2);

resource1.run();

}

Results:

Starting Process "1" (priority: 0)

Starting Process "4" (priority: 0)

Starting Process "2" (priority: 0)

Starting Process "3" (priority: 0)

Process "1" (priority: 5) is requesting resource "A"

Process "4" (priority: 6) is requesting resource "A"

Process "1" (priority: 5) gained access to resource "A"

Process "2" (priority: 2) is requesting resource "A"

1 is using resource "A"

Process "3" (priority: 1) is requesting resource "A"

1 has finished using resource "A"

resource "A" has 5 uses left

Process "1" (priority: 5) released resource "A", to a process with priority 6

Process "4" (priority: 6) gained access to resource "A"

4 is using resource "A"

4 has finished using resource "A"

resource "A" has 4 uses left

Process "4" (priority: 6) released resource "A", to a process with priority 2

Process "2" (priority: 2) gained access to resource "A"

2 is using resource "A"

Process "4" (priority: 3) is requesting resource "A"

2 has finished using resource "A"

resource "A" has 3 uses left

Process "2" (priority: 2) released resource "A", to a process with priority 3

Process "4" (priority: 3) gained access to resource "A"

4 is using resource "A"

Process "2" (priority: 4) is requesting resource "A"

Process "1" (priority: 0) is requesting resource "A"

4 has finished using resource "A"

resource "A" has 2 uses left

Process "4" (priority: 3) released resource "A", to a process with priority 4

Process "2" (priority: 4) gained access to resource "A"

2 is using resource "A"

2 has finished using resource "A"

resource "A" has 1 uses left

Process "2" (priority: 4) released resource "A", to a process with priority 1

Process "3" (priority: 1) gained access to resource "A"

3 is using resource "A"

3 has finished using resource "A"

resource "A" has 0 uses left

Process "3" (priority: 1) released resource "A", to a process with priority 0

Process "1" (priority: 0) gained access to resource "A"

Process "1" (priority: 0) cannot use resource "A" as the resource is exhausted

resource "A" has 0 uses left

Process "1" (priority: 0) released resource "A", there were no waiting processes

Process "1" (priority: 0) has finished

Process "4" (priority: 3) has finished

Process "2" (priority: 4) has finished

Process "3" (priority: 1) has finished

All processes finished