# Java Week Sixteen

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

{

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

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

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

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

{

**super**(resource, maxUses);

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

{

conditions[i] = lock.newCondition();

}

}

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

{

lock.lock();

**try**

{

**if**(inUse)

{

increaseNumberWaiting(priority);

conditions[priority].await();

}

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**(getNumberWaiting(i) > 0)

{

decreaseNumberWaiting(i);

conditions[i].signal();

inUse = **false**;

**return** i;

}

}

inUse = **false**;

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

}

**finally**

{

lock.unlock();

}

}