CSCI 344/715
Lecturer: Dr. Simina Fluture

Lecture #6

Topics:

Signal-and-Continue Monitor
- **Savage-Cook problem (if not completed)**

Condition variables – Java notification objects

Readings: **Web and class lectures**
[SH] ch5 (5.7)

Java Monitors - Notification Objects

It is possible to use a notification object that inside the monitor will play the role of a named condition variable.
Shared object: **Object obj = new Object( );**

In one thread: **synchronized (obj) { …**
(waiting) **if (!condition)**
try { obj.wait( );}
catch (InterruptedException e) { }
…..
}

In another thread:
(signaling) **synchronized (obj) { …**
if (condition) obj.notify( );
……
}

Reader/Writer monitor with notification objects
A thread calls a non-synchronized service method in the monitor, enters a synchronized block on a notification object, and then calls a private synchronized method. The thread might block on the notification object waiting for a signal.

To avoid deadlock, the thread must leave the synchronized method before waiting inside the notification object. After a signal, the thread reenters the monitor via a service method call.