Instituto Tecnológico Superior de Jerez – ITSJ

Estudiante: Miguel Angel Bazán garduño.

mabg211299@hotmail.com

3er Semestre Carrera: Ingeniería en sistemas computacionales (ISC).

Materia: Tópicos avanzados De Programación.

Actividad: Mapa Conceptual.

Docente: I.S.C. Salvador Acevedo Sandoval

Jerez de García Salinas Zacatecas

Exercises

1. Fill in the blanks in each of the following statements:

a) A thread enters the *terminated* state when _ it successfully completes its task or otherwise terminates _.

b) To pause for a designated number of milliseconds and resume execution, a thread should call method _sleep_ of class _ Thread _.

c) Method _signal_ of class Condition moves a single thread in an object's *waiting* state to the *runnable* state.

d) Method _signalAll_ of class Condition moves every thread in an object's *waiting* state to the *runnable* state.

e) A(n) _runnable_ thread enters the _terminated_ state when it completes its task or otherwise terminates.

f) A *runnable* thread can enter the _ timed waitings _ state for a specified interval of time.

g) At the operating-system level, the *runnable* state actually encompasses two separate states, _ready_ and _running_.

h) Runnables are executed using a class that implements the _Executor_ interface.

i) ExecutorService method _shutdonw_ ends each thread in an ExecutorService as soon as it finishes executing its current Runnable, if any.

j) A thread can call method _await_ on a Condition object to release the associated Lock and place that thread in the _waiting_ state.

k) In a(n) _consumer/producer_ relationship, the _producer_ generates data and stores it in a shared object, and the _consumer_ reads data from the shared object.

l) Class _ArrayBlockingQueue_ implements the BlockingQueue interface using an array.

m) Keyword _synchronized_ indicates that only one thread at a time should execute on an object.

2. State whether each of the following is *true* or *false*. If *false*, explain why.

a) A thread is not *runnable* if it has terminated. TRUE

b) Some operating systems use timeslicing with threads. Therefore, they can enable threads to preempt threads of the same priority. FALSE. TIME SLICING ALLOWS A THREAD TO EXECUTE UNTIL ITS TIME SLICE EXPIRES.

c) When the thread's quantum expires, the thread returns to the *running* state as the operating system assigns it to a processor. TRUE

d) On a single-processor system without timeslicing, each thread in a set of equal-priority threads (with no other threads present) runs to completion before other threads of equal priority get a chance to execute. TRUE

3. *(True or False)* State whether each of the following is *true* or *false*. If *false*, explain why.

a) Method sleep does not consume processor time while a thread sleeps. TRUE

b) Declaring a method synchronized guarantees that deadlock cannot occur. FALSE. DEADLOCKS CAN OCCUR IF THE LOCK ON AN OBJECT IS NEVER RELEASED.

c) Once a ReentrantLock has been obtained by a thread, the ReentrantLock object will not allow another thread to obtain the lock until the first thread releases it. TRUE

d) Swing components are thread safe. FALSE. SWING COMPONENTS ARE NOT THREAD SAFE.

4. *(Multithreading Terms)* Define each of the following terms.

a) Thread: An individual execution context of a program

b) Multithreading: The ability of more than one thread to execute concurrently.

c) *Runnable* state: A state in which the thread is capable of running

d) *Timed waiting* state: A state in which the thread cannot use the processor because it is waiting for a time interval to expire or a notification from another thread.

e) Preemptive scheduling: A thread of higher priority enters a running state and is assigned to the processor.

f) Runnable interface: only declare a specific member function execute, which have been defined by the classes that implement this interface

g) notifyAll method: Transitions all threads waiting on an object's monitor to the runnable state.

h) producer/consumer relationship: A relationship in which a producer and a consumer share common data.

i) Quantum: A small amount of processor time, also called a time slice

5. *(Multithreading Terms)* Define each of the following terms in the context of Java's threading mechanisms:

a) Synchronized: it is the concept of monitor, which controls access to an object.

b) Producer: A thread that writes data to a shared memory resource

c) Consumer: A thread that reads data from a shared memory resource

d) Wait: Places a thread in the waiting state until another thread call notify or notifyAll

e) Notify: Wake a thread currently waiting on the given object

f) Lock: An interface implemented by objects that control access to a resource shared among multiple threads.

g) Condition: Objects of this interface represent condition variables that can be used with Locks to manage access to a shared resource

Mapa Conceptual: