Blogs

Core java

PageNo: Go << >>

--Tag-- Search

IO - Externalizable and Serializable

Q: What are differences between Externalizable and Serializable interfaces ?
A:

SERIALIZABLE uses default implementation for reading and writing the object you want to persist. You just have to implement SERIALIZABLE interface for your class and rest will be taken care.

EXTERNALIZABLE you will have to implement the readExternal() and writeExternal() methods. You can specify your own way of storing the information and retrieving the information of the object.

Unless you have very specific requirements one wouldn't use EXTERNALIZABLE.

SERIALIZABLE is the most common form of using.

Threads : What is monitor (key)

It is lock

that will be used to sequentially access an object or a class. There are two monitors (keys)

(1) Class Monitor : It is applied on synchronized static methods

(2) Object Monitor : It is applied on synchronized instance methods

Threads : wait() and join()

wait() : In case of wait method(),

1. calling

   Thread will release the monitor (lock) and go in BLOCKED

   state.
2. When other thread come into same monitor
3. and call notify() method
4. then waiting thread will be activated and go in ready queue.

join() : In case for join() method,

1. calling Thread will NOT release the monitor (lock),
2. but will give chance to Joint thread to be execute first.
3. As soon as Joint thread execution is finished,
4. calling thread again acquires same lock and start execution.

Suppose **T1** thread is calling **T2.join()** then **T1** will wait until T2 is finished then it will start back its processing

In other words

When a thread enter into racing condition then thread need monitor (lock) for execution.

wait() :
1. When wait() method of current thread is called then thread stops execution, releases monitor and go in waiting (BLOCKED) state. Released monitor (lock) will be used by some other thread for execution.
2. When other tread of same monitor calls notify() method then waiting-thread will be activated, go in ready queue, and wait for same monitor for execution.

join() :
1. When join() method of current thread is called then thread stops execution, DO NOT release monitor, and give chance to joint thread for execution.
2. As soon as execution of joint-thread is finished, current thread again gets same monitor and start execution.

Threads : wait(), notify() and notifyAll()

Similarities:

- Threads can communicates to each other using **wait( )**, **notify( )**, and **notifyAll( )** methods.
- These methods are final in java.lang.Object class.
- These methods are called only from **synchronized** method or blocks.

Differences:
wait( ) : calling Thread will release the Monitor (lock) and go in BLOCKED state. When other thread come into same monitor and call notify() method then waiting thread will be activated and go in ready queue.
notify( ) : wakes up the first thread that called **wait( )** on the same Monitor.
notifyAll( ) : wakes up all the threads that called **wait( )** on the same Monitor. The
highest priority thread will run first.

What is Collection Framwork ?

1. Collection is set of 4 basic interfaces Collection, List, Set and Map.
2. Set can contain unique objects whereas List can contain duplicate objects.
3. Map contains key-value pairs but Value and Key cannot be null.