Threads Observations

1.Multi Tasking

2.Multi Threading

3.How many ways we can create threads

4.Can we overload run()

5.diff betn t1.start() and t1.run();

6.with out overriding the run() what will happen

7.Can we overide start()==>

ans:-start() will be called like a normal method further it wont call run().Thread will not be created

It is not recommended to override start method.In this case main thred only execute all these things

8.inside of start() override method by adding super.start()

Possiblities:-

#1

  start()

  run()

main thread

#2

main thread

  start()

  run()

#3

  run()

  start()

main thread

#4

start()

main thread

run()

9.Thread states

# MyThread t=new MyThread()---> new Born/ready

#t.start()--->start state

#t.run()---->run state If therad schedular allocates processor then it is in run state

#if t.run() not run then it is in death state.If run() completes then it will goes to dead state

#from run state to it will go to some other states wait by calling wait or sleep

10.After starting the thread once again if you are trying to reastrt the same thared then we will get run time exception IllegalState exception

MultiThreading

==============================

MyRunnable r=new MyRunnable();

Thread t1=new Thread();

Thread t2=new Thread(r);

#case1

t1.start();

#case2

t2.start();

#case3

t1.run();

#case4

t2.run()

Q:-which approach is best for creation of a therd?

implements Runnanble is best.

In first approach our class extends Therad.There is no chance of extending any other class.hence we are missing Inherting Benfit

But in the second approach Implementing Runnable wed can extend any other class.Hence we will not miss any inheritance benfit

========Thread class constructors=========

Thread t=new Thread()

Thread t=new Thread(Runnable r)

Thread t=new Thread(String nameOftheThread)

MAX_PRIORITY=10

MIN_PRIORITY=1;

NORMAL_PRIORITY=5

LOW_PRIORITY=0//invalid

#If you set thread proiority more than 10 then we will get Illegal Argument exception

=========Default Priority========

The default priority only for the main therd is 5 .But for all remaing therads default priorityis will be inherited from parent to child.

i.e what ever priority parent therad has the same priorty will be there for child for thread.

Note-

Some platforms wont provide proper support for therd priorties

===========Pause thread===========

we can prevent a thread execution by using the foloowing methods

yield()

sleep()

join()

========Yield======================

->yield() causes to pause the current executiing thread to give the chance for waiting threads of same priority.

->If there is no waiting thread or all waiting threads have low prioroty then same thered can coninue it's execution

->If multiple threads are waiting with same priority then which waiting thread will get the chance

we can't expect it depends on thread schedular

->

The therad which is yielded,When it will get the chance once again.

It depends on the threads schedular and we can't expect exactly

==>signature

public static native void yield()

Note:-

Some platforms wont provide proper support for yield method.

=====================join()==================

If a thread wants to wait for untill completion of other therad then we should go for join() method

ex:=

If a thread t1 wants to wait untill completion of t2 then we can use t1 has to call t2.join

If t1 executes t2.join then immediately t1 will be entered into t1 will be entered into untill t2 completes

once t2 completes then t1 can continue it's execution

join(1000)

->Note:-Every join() throws Interrupted Exception which is checked exception hence compulsury we should handle this exception eirther by using try catch or by using throws key word othe wise we will get compile time error.

=====================sleep=========

If a thread dont want to perform any operations for a particualr amount of time then we can use sleep()

ex:-class slide rotater is a example for sleep

==========================interrupt==

A thread can interrupt a sleeping therad or waiting thread by using interrupt() of thread class

public void interrupt()

********Note**********

when ever we are callling intterupt() if the target therad not in sleeping state or waiting state then there is no impact of interrupt call immediately.Interrupt call wil be waited until target thread entered into sleeping or waiting state

If the tareget thread enter into sleep or waiting state then immediately intterupt call will interrupt the target thread

If the target therad never entered into sleep or waiting state in it's life time then there is no impact of intterupt call this is the only case where interrupted call will be wasted

==============Callable and Future=====

In the case of runnable thread wont return after completing job

If a thread requires to return some result after execution, then we should go for callable

Callable interface contains only one method

Public Object call () throws Exception

If we submit callable object to executor after completing the job thread returns an object of type Future

Future objects can be used to retrieve the results from the callable job