Threading in JAVA

* Context Switching: Switching between task
* Thread: Light weight Process.
* Context switching is faster in Thread than process.
* Thread uses shared memory space.
* Could be performed in java using these :
  + Thread Class 🡪 Lang Package
  + Runnable Interface
* run() method is used and to call run , start method is used.

Multithreading in JAVA

* Thread is a light weight process.
* All threads share the common memory area.
* Threads are independent of each other.
* In thread, Context switching is faster, than processes.
* Multithreading is a process to run multiple threads concurrently.

* To Implement Multithreading in Java.
  + 1. By Extending Thread Class.
    2. By Implementing Runnable Interface.

NOTE:🡪 java.lang.Thread package is used to implement Multithreading.

Life cycle of Thread (States):



1. New
2. Runnable



1. Running



1. Wating or Blocked
2. Terminate



Getting the name and priority of thread and setting the priority.

* To get the name of thread we use getName() method of Thread class.
* To get the priority of thread we use getPriority() method of Thread class.
* To set the priority of thread we set setPriority(int) method of Thread class
* Priority Lies in the range of 1 to 10. (More the number, More its priority). This could be true, it is all dependent on JVM and Algorithms like First come first serve ,etc.
  + - Three Variables are used for Priority🡪

1. MIN\_PRIORITY 🡪 1
2. MAX\_PRIORITY 🡪 10
3. NORM\_PRIORITY 🡪 5 (Default Priority)
   * To make object into thread currentThread(); is used
   * Main method acts like a single thread.

Join() Method in Thread Class:

* Join() method allows to wait the thread until the other thread completes its execution.
* When other thread completes its execution then first thread starts execution.

Synchronization in Java :

* `synchronized` keyword is used to synchronize the processes

Synchronization Block in Java :

* `synchronized(this){…lines….}` is used to synchronize the few lines of code.

isAlive() method in Thread:

class isAlivedemo extends Thread{

    public void run(){

        System.out.println("Hi");

    }

    public static void main(String[] args) {

        isAlivedemo al = new isAlivedemo();

        System.out.println("Before Starting :"+ al.isAlive());

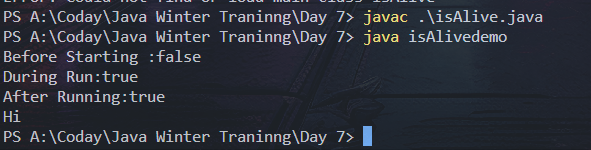
        al.start();

        System.out.println("During Run:"+ al.isAlive());

        // System.out.println("After Running:"+ al.isAlive());

    }

}



Package 🡪 java.util.\*

1. Calender class
2. Date class
3. java.time.LocalTime
4. java.time.LocalDate