Threads:

Thread is like a process or request which is used for execution of the code.

Multi-threading:

It is like multiple threads accessing the resource.

Thread life cycle:

* New 🡺 Thread is get created
* Runnable 🡺 Thread start() method will be called run() method
* Running 🡺 once run() method is called thread will be executed.
* Non-Runnable 🡺 sleep(1000) whenever thread got sleep and not running this stage is non-runnable stage.
* Terminated 🡺 once thread got task complete then it will be get terminated automatically.

Thread can be created in 2 ways

1. By extending Thread class
2. By implementing Runnable Interface

Methods:

Sleep() 🡺 if you want to make a thread to sleep we need to use Sleep()

Start() 🡺 Starting of execution of thread

Run() 🡺 once start() method calls run() method will be executed and logic will be created inside run() method

Join(): if you want to run a thread after completion of another thread tasks,

Daemon Thread:

It is like a service provider thread for the user Thread. This thread will be active as long as user thread is active and once user thread got terminates daemon thread will be terminated by the JVM.

setDaemon(true); 🡺 Will set thread to Daemon

isDaemon() 🡺 checks is it Daemon Thread.

Thread Pool:

Thread pool is like group of working thread that are reusable many times.

10 tasks 🡺 10 threads

10 tasks 🡺 5 threads 🡺 each thread will have 2 tasks

ExecutorService 🡺 using this we will be creating the thread pool.

1. newFixedThreadPool(int num) 🡺 it will create fixed number of threads.
2. newCachedThreadPool(): this method create new thread pool and cahed it and used same thread multiple times.
3. newSingleThreadExecutor() : this method will creates only single new thread.

Deadlaock:

Diagram

Description automatically generated

Synchronization:

* It is a Thread safe.
* Each thread will access at once and other threads will wait to complete the task of 1st thread.
* 3types:
* Synchronization method level
* Synchronization block
* Static synchronization