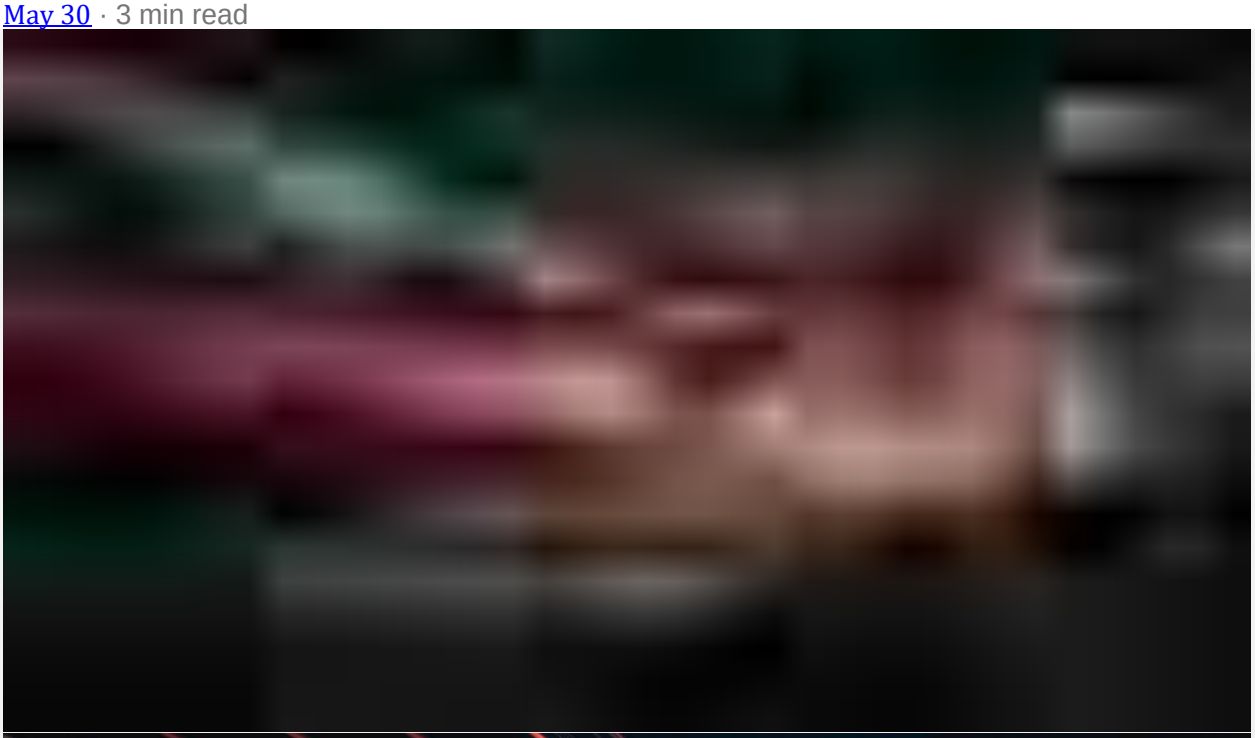
Threads in Java

Pasindu Ukwatta

May 30 · 3 min read



src :https://images.app.goo.gl/CGEFfk775MHXiYn46

“It is far easier to design a class to be thread-safe than to retrofit it for the thread safety later”

(Brian Goetz - Java language Architect,Oracle Corporation)

The lightweight, series of statements which facility to allow multiple activities within a single process called Thread. Each java program has at least one Thread called main ( ) thread . The job of the main thread is to start the main Thread. Then the other Threads will start and run according to the Program. There are two types of Threads. They are User Threads and Daemon Threads. User Threads are which have Highest Priority and the thread will terminate after completing the job. JVM terminate the Thread. Daemon threads have the lowest Priority and they provide the services for the user Threads. Threads are running parallelly and work independently. So, it helps maximize the utilization of the CPU. Threads are used for crating Animations, Video editing and Game Developing .

Threads in java can be created using Two Ways .

Thread Creation

I.)Extending java.lang.Thread Class

```java
public class ThreadExample extends Thread{
    public static void main(String[] args) {
        ThreadExample t = new ThreadExample();
        t.start();
    }

    public void run(){
        System.out.println("Thread created from extending thread class");
    }

}
```

Thread creation using Thread Class

II.)Implementing java.lang.Runnable Interface

```java
public class ThreadExample implements Runnable{
    public static void main(String[] args) {
      Thread t = new Thread(new ThreadExample());
      t.start();
    }

    public void run(){
       System.out.println("Thread created from implementing Runnable Interface");
    }

}
```

Thread creation using Runnable Interface

But most recommended way to create Thread in java is using the Runnable Interface . Because it helps to extend it to another class . If we use thread class we cannot extend it to another class. Because in java we can only extend in to one class. The job of the thread is override by the programmer itself in the run( ) method .

Thread Life Cycle

src:https://images.app.goo.gl/myxib54XaDPGWyMK6

i) New — Thread object creation is done by using new key word .

Thread t = new Thread( );

ii.) Ready/Runnable — when start( ) method called ,then the thread changed to the Runnable state and waiting in the thread pool .

iii.) Running — run ( ) method called in the thread ,So the thread is running the job of the thread.

iv.) Blocked/Waited — Threads are waiting in the thread pool for the Execution .

v.) Dead — when the thread job is completed or the stop( ) method called ,then the thread will be terminated.

Thread Synchronization

When the many threads are trying to perform action on same object , then their may be a chance of a data inconsistency problem . Because every thread is trying to access the same object . So we need to synchronized keyword to this . when the method or the block declared as synchronized , then at a given time only one

thread is allowed to execute the method on the object .This is ensure the Thread safe but this increase the waiting time .

Clear understanding about the thread will help to make a high performance program . For that knowledge in thread is needed.

Src :https://docs.oracle.com/javase/specs/jls/se14/jls14.pdf