

26-05-2020

Today, according to last classes, we are now able to compare between Comparable and Comparator interfaces.

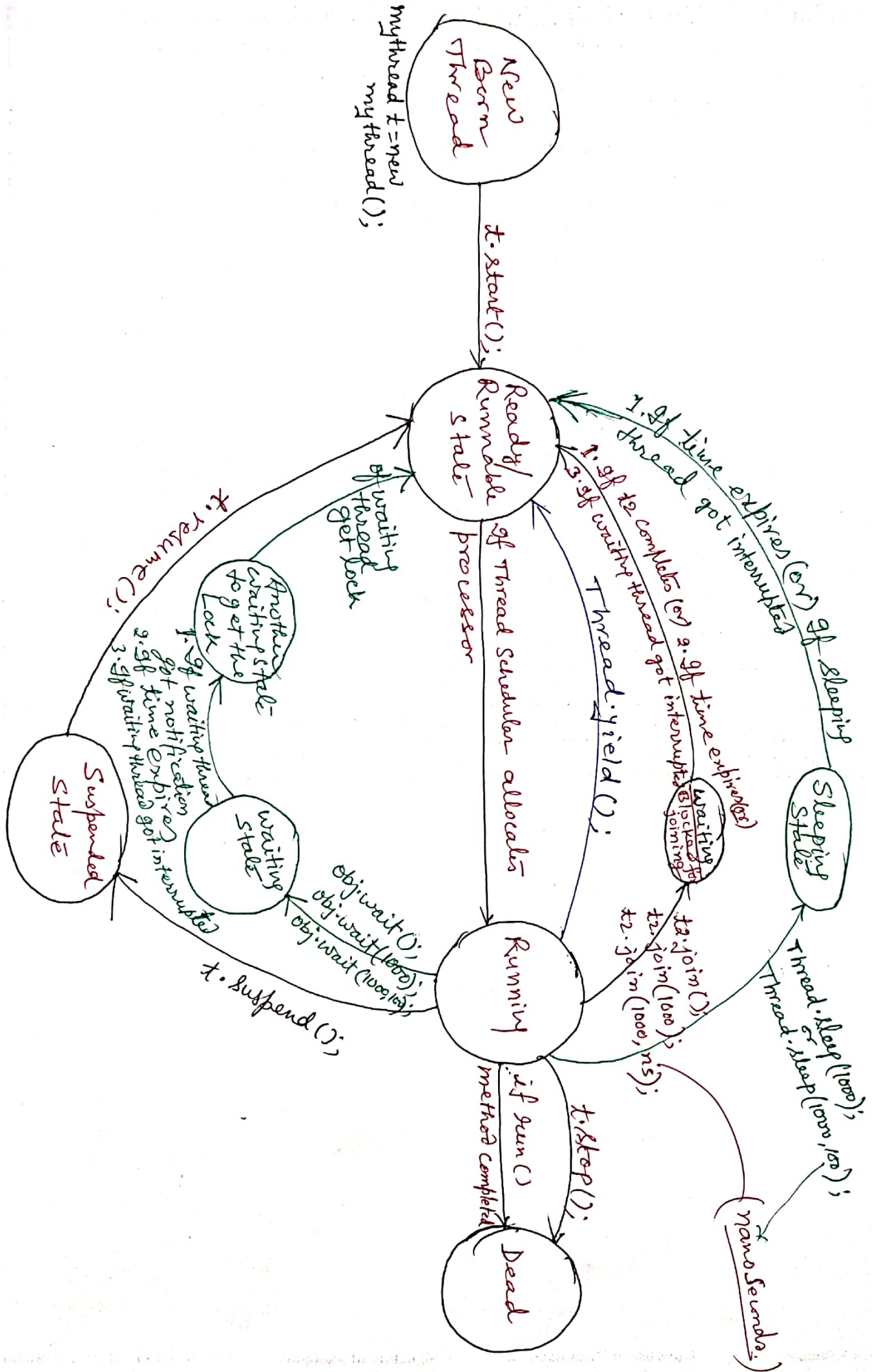
Comparable

1. Comparable interface is meant for Default-natural Sorting order.
2. It is present in java.lang package.
3. It has only one method — compareTo()
4. All wrapper classes and String class is implementing Comparable interface.

Comparator

1. Comparator interface is meant for Customized Sorting order.
2. Comparator is present in java.util package
3. It has two methods —
 - a) compare()
 - b) equals()
4. The only implementing classes are: —
 - a) Collator
 - b) RuleBasedCollator

Thread Life Cycle Diagram



Explanation of Life Cycle of a Thread .

- ★ When we initialize our class having multithreading capabilities like:-

```
mythread t = new mythread();
```

A new born thread ~~is~~ comes into existence.
- ★ When we execute start method the thread will come into Ready/Ready to Run/Runnable state.
- ★ If thread scheduler allocates processor then the thread enters into Running state.
- ★ If run method completes its work, then the thread entered into dead state or when stop() method is called then also ~~it~~ thread entered into dead state.
- ★ The thread enters into suspended state when suspend() method is called.
- ★ The thread enters into Ready/Runnable state again when thread invokes resume() method.
- ★ The thread comes into waiting state when ^{another} ~~it~~ thread calls join(); or join(1000); or join(1000, 100); methods.
 The thread is blocked for joining and again comes into Ready/Runnable state when another thread completes or time expires or if waiting thread got interrupted.

- * The ^{running} thread comes into sleeping state when user (Program) calls `sleep(1000);` or `sleep(1000, 100);`
- * If time expires or if sleeping thread got interrupted then sleeping thread comes into Ready/Runnable state again.
- * `Thread.yield()` method is when called, running thread again enters into Ready/Runnable state.
- * The running thread enters into waiting state when `wait();` or `wait(1000);` or `wait(1000, 100);` method is called.
- * The thread comes into another waiting state to get the lock if waiting thread got notification or if time expires or if waiting thread got interrupted.

This is how a thread spans its life cycle during its life cycle.
