threads in Java

A Thread class manages a single sequential thread of control. Threads may be created and deleted dynamically.

The Thread class executes instructions from its method run(). The actual code executed depends on the implementation provided for run() in a derived class.

```java
class MyThread extends Thread {
    public void run() {
        //......
    }
}
```

Creating a thread object:
```
Thread a = new MyThread();
```
threads in Java

Since Java does not permit multiple inheritance, we often implement the `run()` method in a class not derived from `Thread` but from the interface `Runnable`.

```java
public interface Runnable {
    public abstract void run();
}

class MyRun implements Runnable{
    public void run() {
        //....
    }
}

Creating a thread object:

    Thread b = new Thread(new MyRun());
```
thread life-cycle in Java

An overview of the life-cycle of a thread as state transitions:

The predicate `isAlive()` can be used to test if a thread has been started but not terminated. Once terminated, it cannot be restarted (cf. mortals).
thread alive states in Java

Once started, an alive thread has a number of substates:

Also, `wait()` makes a Thread Non-Runnable, and `notify()` makes it Runnable (used in later chapters).
Java thread lifecycle - an FSP specification

\[
\begin{align*}
\text{THREAD} &= \text{CREATED,} \\
\text{CREATED} &= (\text{start} \rightarrow \text{RUNNABLE} \\
& \quad | \text{stop} \rightarrow \text{TERMINATED}), \\
\text{RUNNING} &= (\{\text{suspend, sleep}\} \rightarrow \text{NON\_RUNNABLE} \\
& \quad | \text{yield} \rightarrow \text{RUNNABLE} \\
& \quad | \{\text{stop, end}\} \rightarrow \text{TERMINATED} \\
& \quad | \text{run} \rightarrow \text{RUNNING}), \\
\text{RUNNABLE} &= (\text{suspend} \rightarrow \text{NON\_RUNNABLE} \\
& \quad | \text{dispatch} \rightarrow \text{RUNNING} \\
& \quad | \text{stop} \rightarrow \text{TERMINATED}), \\
\text{NON\_RUNNABLE} &= (\text{resume} \rightarrow \text{RUNNABLE} \\
& \quad | \text{stop} \rightarrow \text{TERMINATED}), \\
\text{TERMINATED} &= \text{STOP}.
\end{align*}
\]
Java thread lifecycle - an FSP specification

end, run, dispatch are not methods of class Thread.

States 0 to 4 correspond to CREATED, TERMINATED, Runnable, RUNNING, and NON-RUNNABLE respectively.