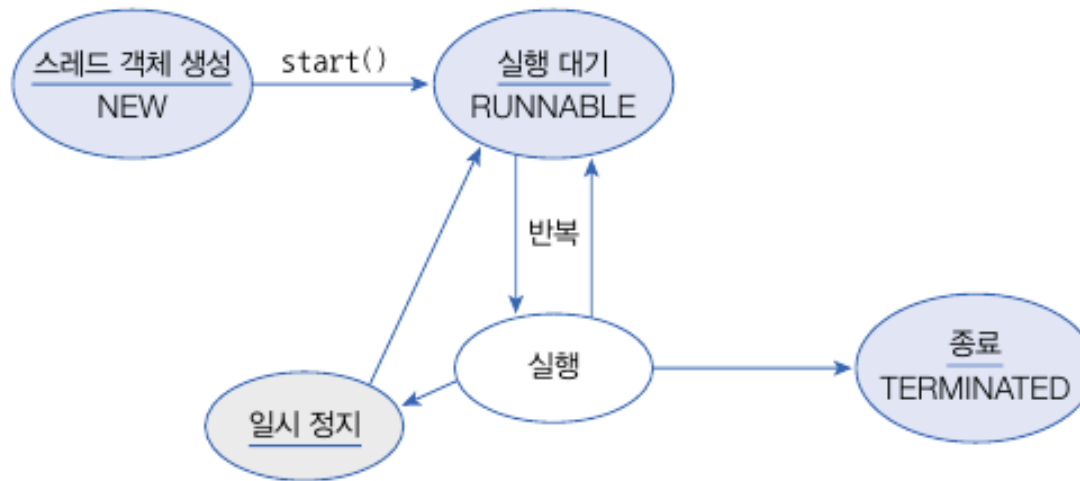


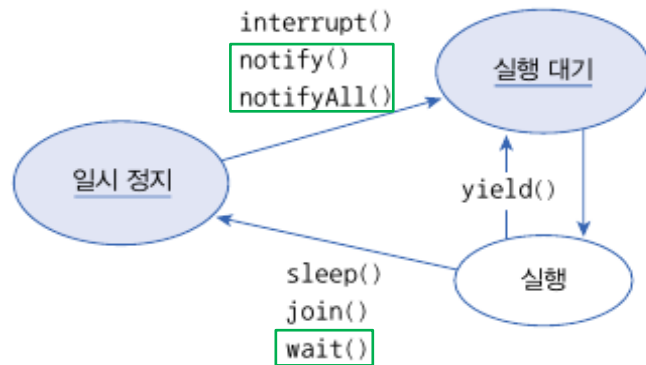
스레드 상태

▣ 스레드 상태

- 스레드 객체를 생성(NEW)하고, start() 메소드를 호출하면 곧바로 스레드가 실행되는 것이 아니라 **실행 대기 상태(RUNNABLE)**가 된다. 실행 대기 상태란 실행을 기다리고 있는 상태를 말한다.
- 실행 대기하는 스레드는 **CPU 스케줄링**에 따라 **CPU를 점유**하고 **run()** 메소드를 실행한다. 이때를 **실행(RUNNING)** 상태라고 한다. 실행 스레드는 run() 메소드를 모두 실행하기 전에 스케줄링에 의해 다시 실행 대기 상태로 돌아갈 수 있다. 그리고 다른 스레드가 실행 상태가 된다. 이렇게 스레드는 **실행 대기 상태**와 **실행 상태**를 **번갈아 가면서** 자신의 **run()** 메소드를 **조금씩 실행**한다.
- 실행 상태에서 run() 메소드가 종료되면 더 이상 실행할 코드가 없기 때문에 스레드의 실행은 멈추게 된다. 이 상태를 **종료 상태(TERMINATED)**라고 한다.
- 실행 상태에서 일시 정지 상태로 가기도 하는데, **일시 정지 상태**는 스레드가 실행할 수 없는 상태를 말한다. 스레드가 다시 실행 상태로 가기 위해서는 일시 정지 상태에서 실행 대기 상태로 가야만 한다. 이때 사용하는 메소드에는 **interrupt()**, **notify()**, **notifyAll()** 이 있다.



- wait(), notify(), notifyAll()은 **Object** 클래스의 메소드이고, 그 외는 **Thread** 클래스의 메소드이다.



구분	메소드	설명
일시 정지로 보냄	sleep(long millis)	주어진 시간 동안 스레드를 일시 정지 상태로 만든다. 주어진 시간이 지나면 자동적으로 실행 대기 상태가 된다.
	join()	join() 메소드를 호출한 스레드는 일시 정지 상태가 된다. 실행 대기 상태가 되려면, join() 메소드를 가진 스레드가 종료 되어야 한다.
	wait()	동기화 블록 내에서 스레드를 일시 정지 상태로 만든다.
일시 정지에서 벗어남	interrupt()	일시 정지 상태일 경우, InterruptedException을 발생시켜 실행 대기 상태 또는 종료 상태로 만든다.
	notify() notifyAll()	wait() 메소드로 인해 일시 정지 상태인 스레드를 실행 대기 상태로 만든다.
실행 대기로 보냄	yield()	실행 상태에서 다른 스레드에게 실행을 양보하고 실행 대기 상태가 된다.

◎ 주어진 시간 동안 일시 정지 (sleep)

- 실행 중인 스레드를 일정 시간 멈추게 하고 싶다면 `Thread` 클래스의 정적 메소드인 `sleep()`을 이용하면 된다.
- 매개값에는 얼마 동안 일시 정지 상태로 있을 것인지 밀리세컨드(1/1000) 단위로 시간을 주면 된다.

```
try {  
    Thread.sleep(1000); // 1초 동안 일시 정지상태  
} catch (InterruptedException e) {  
    // interrupt() 메소드가 호출되면 실행  
}
```

◎ 다른 스레드의 종료를 기다림 (join)

- 스레드는 다른 스레드와 독립적으로 실행하지만 다른 스레드가 종료될 때까지 기다렸다가 실행을 해야 하는 경우도 있다. 이를 위해 스레드는 join() 메소드를 제공한다.
- ThreadA가 ThreadB의 join() 메소드를 호출하면, ThreadA는 ThreadB가 종료할 때까지 일시 정지 상태가 된다.
- ThreadB의 run() 메소드가 종료되고 나서야 비로소 ThreadA는 일시 정지에서 풀려 다음 코드를 실행한다.



◎ 다른 스레드에게 실행 양보 (yield)

- yield()를 호출한 스레드는 실행 대기 상태로 돌아가고, 다른 스레드가 실행 상태가 된다.

