

스레드 안전 종료

▣ 스레드 안전 종료

- 스레드는 자신의 run() 메소드가 모두 실행되면 자동적으로 종료되지만, 경우에 따라서는 실행 중인 스레드를 즉시 종료할 필요가 있다. <예> 동영상 끝까지 보지 않고 사용자가 멈춤을 요구하는 경우이다.
- 스레드를 강제 종료시키기 위해 Thread는 stop() 메소드를 제공하고 있으나 이 메소드는 deprecated(더 이상 사용하지 않음)되었다.
- 스레드를 안전하게 종료하는 방법은 사용하던 리소스들을 정리하고 run() 메소드를 빨리 종료하는 것이다. 주로 <방법1> 조건 이용 방법 과 <방법2> interrupt() 메소드 이용 방법을 사용한다.

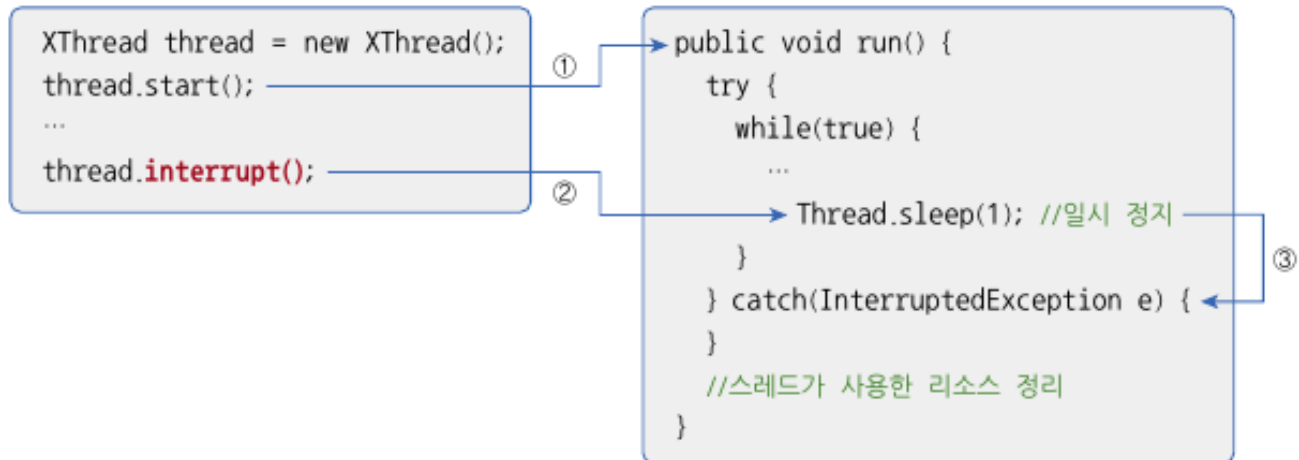
◎ 조건 이용

- 스레드가 while 문으로 반복 실행할 경우, 조건을 이용해서 run() 메소드의 종료를 유도할 수 있다.
- stop 필드 조건에 따라서 run() 메소드의 종료를 유도한다.

```
public class XXXThread extends Thread {  
    private boolean stop; •----- stop이 필드 선언  
  
    public void run() {  
        while( !stop ) { •----- stop이 true가 되면 while 문을 빠져나감  
            //스레드가 반복 실행하는 코드;  
        }  
        //스레드가 사용한 리소스 정리 •----- 리소스 정리  
    } •----- 스레드 종료  
}
```

◎ interrupt() 메소드 이용

<방법1> interrupt() 메소드는 스레드가 일시 정지 상태에 있을 때, **InterruptedException** 예외를 발생시켜 예외 처리 블록으로 이동한다. 이것을 이용하면 예외 처리를 통해 run() 메소드를 정상 종료시킬 수 있다.



- <방법2> 일시 정지를 만들지 않고도 interrupt() 메소드 호출 여부를 알 수 있는 방법이 있다.
- Thread의 interrupted() 와 isInterrupted() 메소드는 interrupt() 메소드 호출 여부를 리턴한다.
 - interrupted()는 정적 메소드이고, isInterrupted()는 인스턴스 메소드이다.

```
boolean status = Thread.interrupted();  
boolean status = objThread.isInterrupted();
```