

# *REPORT*



과 목 명 : 디자인패턴

담당교수 : 박제호 교수님

소 속 : 소프트웨어학과

학 번 : 32151671

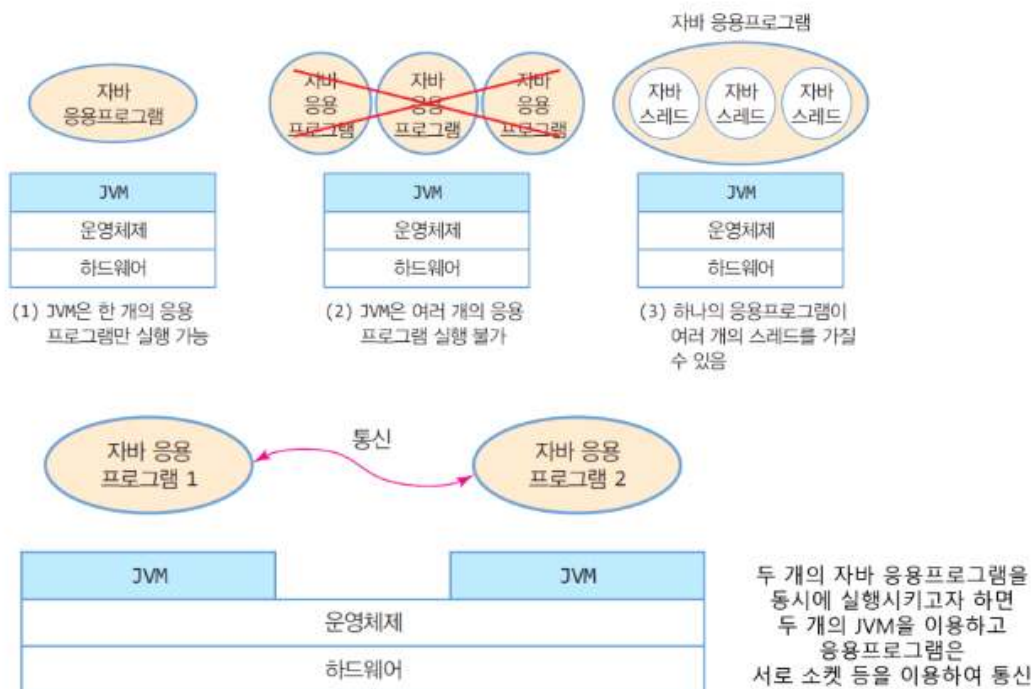
이 름 : 박민혁



**단국대학교**  
Dankook University

### 13. 스레드와 멀티태스킹

- 멀티 태스킹 : 하나의 응용프로그램이 여러 개의 태스크를 동시에 처리
- 스레드 : 사용자가 작성한 코드로서, JVM에 의해 스케줄링 되어 실행되는 단위
- 멀티스레딩의 효과
  - ✓ 한 스레드가 대기하는 동안 다른 스레드 실행
  - ✓ 프로그램 전체적으로 시간 지연을 줄임
- 자바 스레드 : 자바 가상 기계에 의해 스케줄 되는 실행 단위의 코드 블록



- 스레드 만드는 2가지 방법
  - ✓ java.lang.Thread 클래스 사용
  - ✓ java.lang.Runnable 인터페이스 이용

```
class TimerThread extends Thread {
    .....
    @Override
    public void run() { // run() 오버라이딩
        .....
    }
}
```

- 스레드 만들 때 주의사항
  - ✓ run() 메소드가 종료하면 스레드는 종료한다.
  - ✓ 한번 종료한 스레드는 다시 실행 시킬 수 없다.
  - ✓ 한 스레드에서 다른 스레드를 강제 종료할 수 있다.

- Runnable interface로 스레드 만들기

```

class TimerRunnable implements Runnable {
    .....
    @Override
    public void run() { // run() 메소드 구현
        .....
    }
}

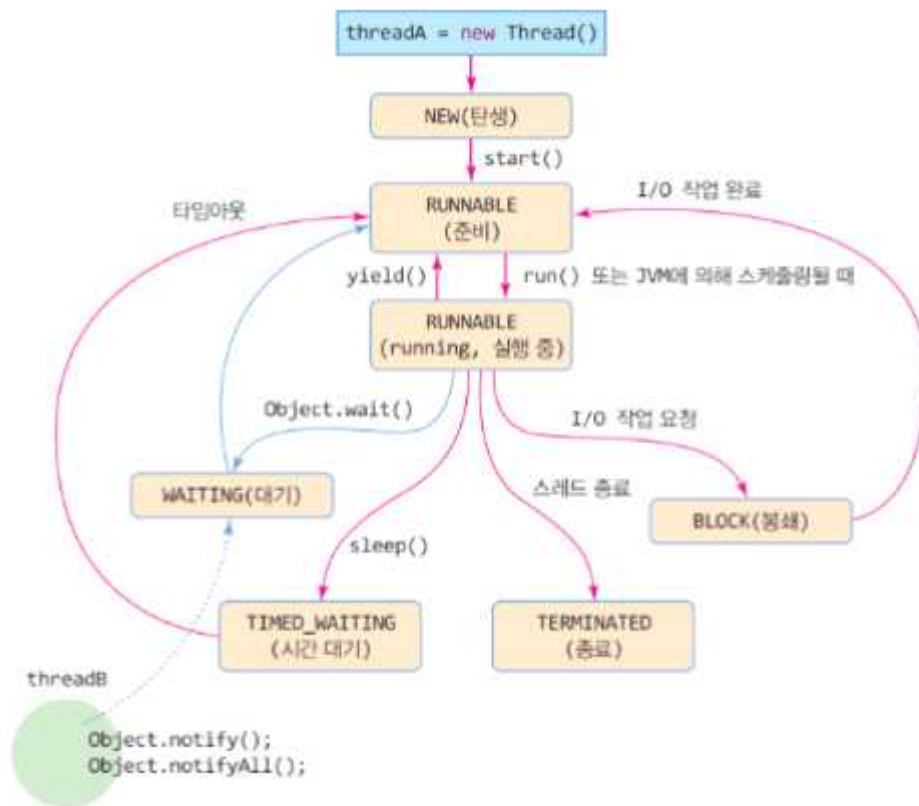
```

- 스레드 정보

필드	타입	내용
스레드 이름	스트링	스레드의 이름으로서 사용자가 지정
스레드 ID	정수	스레드 고유의 식별자 번호
스레드의 PC(Program Count)	정수	현재 실행 중인 스레드 코드의 주소
스레드 상태	정수	NEW, RUNNABLE, WAITING, TIMED_WAITING, BLOCK, TERMINATED 등 6개 상태 중 하나
스레드 우선순위	정수	스레드 스케줄링 시 사용되는 우선순위 값으로서 1~10 사이의 값이며 10이 최상위 우선순위
스레드 그룹	정수	여러 개의 자바 스레드가 하나의 그룹을 형성할 수 있으며 이 경우 스레드가 속한 그룹
스레드 레지스터 스택	메모리 블록	스레드가 실행되는 동안 레지스터들의 값

- 스레드 상태

- ✓ NEW : 스레드가 생성 되었지만 스레드가 아직 실행할 준비가 되지 않음
- ✓ RUNNABLE : 스레드가 현재 실행되고 있거나 실행 준비되어 스케줄링을 기다리는 상태
- ✓ WAITING : wait()를 호출한 상태, 스레드 동기화를 위해 사용
- ✓ TIMED\_WAITING : sleep(n)을 호출하여 n 밀리초 동안 잠을 자고 있는 상태
- ✓ BLOCK : 스레드가 I/O 작업을 요청하면 JVM이 자동으로 BLOCK 상태로 만들
- ✓ TERMINATED : 스레드가 종료한 상태



- 스레드 동기화 : 공유 데이터에 동시에 접근하는 다수의 스레드가 공유 데이터를 배타적으로 접근하기 위해 상호 협력 하는 것

- ✓ synchronized로 동기화 블록 지정
- ✓ wait() - notify() 메소드로 스레드 실행 순서 제어

- producer - consumer 문제와 동기화

- ✓ producer : 공유 메모리에 데이터를 공급하는 스레드
- ✓ consumer : 공유 메모리의 데이터를 소비하는 스레드

- wait() : 다른 스레드가 notify()를 불러줄 때까지 기다림

- notify() : wait()를 호출하여 대기중인 스레드를 깨우고 RUNNABLE 상태로 만든다.

- notifyAll() : 모든 스레드를 깨운다.