멀티쓰레드 프로그래밍

멀티쓰레드란

- 실행 흐름을 의미하는 쓰레드를 여러 개 동시에 병렬적으로 사용하여 프로그램의 성능을 향상시키는 것

- java에서 멀티쓰레드를 구현하는 데에 사용되는 2가지 방법
- 1. thread 클래스
- 2. runnable 인터페이스

THREAD 클래스

- thread 클래스를 상속받아 새로운 쓰레드를 정의함
- thread 클래스의 run() 메소드를 오버라이딩하여 사용
- 이후 run이 아니라 start() 메소드를 통해 사용

RUNNABLE 인터페이스

- 쓰레드 클래스를 상속받는 대신 **runnable** 인터페이스를 **implements** 하여 사용
- 마찬가지로 **run()** 메소드를 구현하여 사용
- **new thread(new myRunnable)** 과 같이 **thread**의 생성자를 이용한 후 **start** 메소드로 실행

쓰레드 상태

- 1. new (생성된 상태)
- 2. runnable (실행 대기)
- 3. waiting, timed waiting, blocked (일시 정지)
- 4. terminated (종료)

- getState() 메소드로 상태 파악 가능

쓰레드 우선순위

- 1부터 10까지의 값으로 조정 가능
- 수가 클수록 우선순위가 높음
- setPriority(원하는 우선순위) 메소드를 이용하여 우선순위 설정

MAIN 쓰레드

- 자바 프로그램이 시작될 때 실행되는 주요 쓰레드
- main 쓰레드로 시작하고 main 쓰레드가 종료되면 프로그램이 종료됨
- c에서의 메인함수로 이해

쓰레드 동기화

- 하나의 데이터에 여러 쓰레드들이 동시에 접근하는 것을 대비하여 제한을 걸어주는 것

- **synchronized** 키워드를 이용하여 동기화 -> 하나의 쓰레드가 접근할 시 **lock**이 걸림
- 메서드를 정의할 때 키워드를 붙여 메소드를 동기화 가능
- 단일 코드에 키워드를 붙여 객체를 동기화 가능

쓰레드 데드락

- 두 개 이상의 서로 다른 쓰레드에서 교차적으로 lock이 발동 된 상태
- 교착상태 라고도 함
- 4가지 발생조건이 있으며 모두 충족할 때 데드락이 발동 됨
- 상호 배제 / 점유 대기 / 비선점 / 순환 대기
- 하나만 막더라도 데드락을 막을 수 있음

QNA