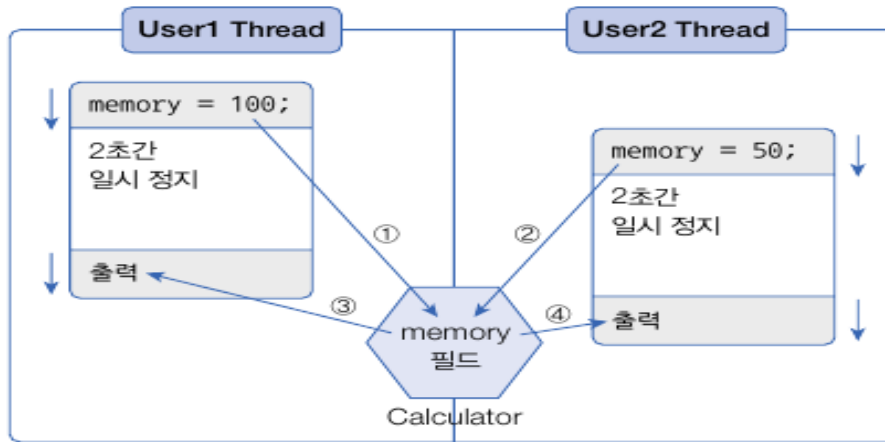


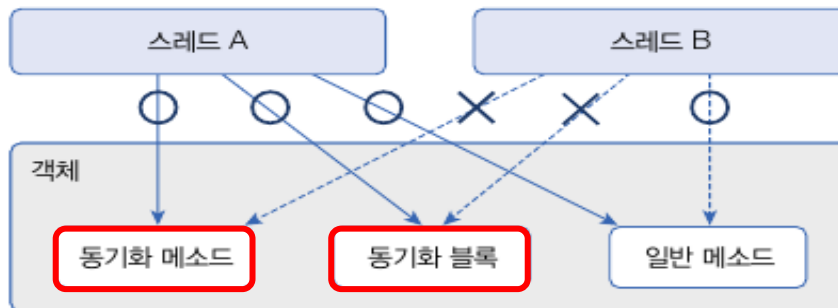
스레드 동기화

■ 스레드 동기화

- 멀티 스레드는 하나의 객체를 공유해서 작업할 수도 있다. 이 경우, 다른 스레드에 의해 객체 내부 데이터가 쉽게 변경될 수 있기 때문에 의도했던 것과는 다른 결과가 나올 수 있다.



- 스레드가 사용 중인 객체를 다른 스레드가 변경할 수 없도록 하려면 스레드 작업이 끝날 때까지 객체에 **잠금**을 걸면 된다. 이를 위해 자바는 **동기화(synchronized) 메소드**와 **동기화(synchronized) 블록**을 제공한다.
- 객체 내부에 동기화 메소드와 동기화 블록이 여러 개 있다면 스레드가 이들 중 하나를 실행할 때 다른 스레드는 해당 메소드는 물론이고 다른 동기화 메소드 및 블록도 실행할 수 없다.
하지만 일반 메소드는 실행이 가능하다.



◎ 동기화 메소드 및 블록 선언

스레드가 동기화 메소드를 실행하는 즉시 객체는 잠금이 일어나고, 메소드 실행이 끝나면 잠금이 풀린다.

<유형1> 메소드 전체 잠금

```
public synchronized void method() {  
    //단 하나의 스레드만 실행하는 영역  
}
```

<유형2> 메소드 내부에서 특정 영역만 잠금

```
public void method () {  
    //여러 스레드가 실행할 수 있는 영역  
  
    synchronized(공유객체) {  
        //단 하나의 스레드만 실행하는 영역  
    }  
  
    //여러 스레드가 실행할 수 있는 영역  
}
```

◎ wait() 과 notify()를 이용한 스레드 제어

- 한 스레드가 작업을 완료하면,
notify() 메소드를 호출해서 일시 정지 상태에 있는 다른 스레드를 실행 대기 상태로 만들고,
자신은 두 번 작업을 하지 않도록 wait() 메소드를 호출하여 일시 정지 상태로 만든다.
- notify()는 wait()에 의해 일시 정지된 스레드 중 한개를 실행 대기 상태로 만들고,
- notifyAll()은 wait()에 의해 일시 정지된 모든 스레드를 실행 대기 상태로 만든다.
- 주의할 점은 이 두 메소드는 동기화 메소드 또는 동기화 블록 내에서만 사용할 수 있다는 것이다.

