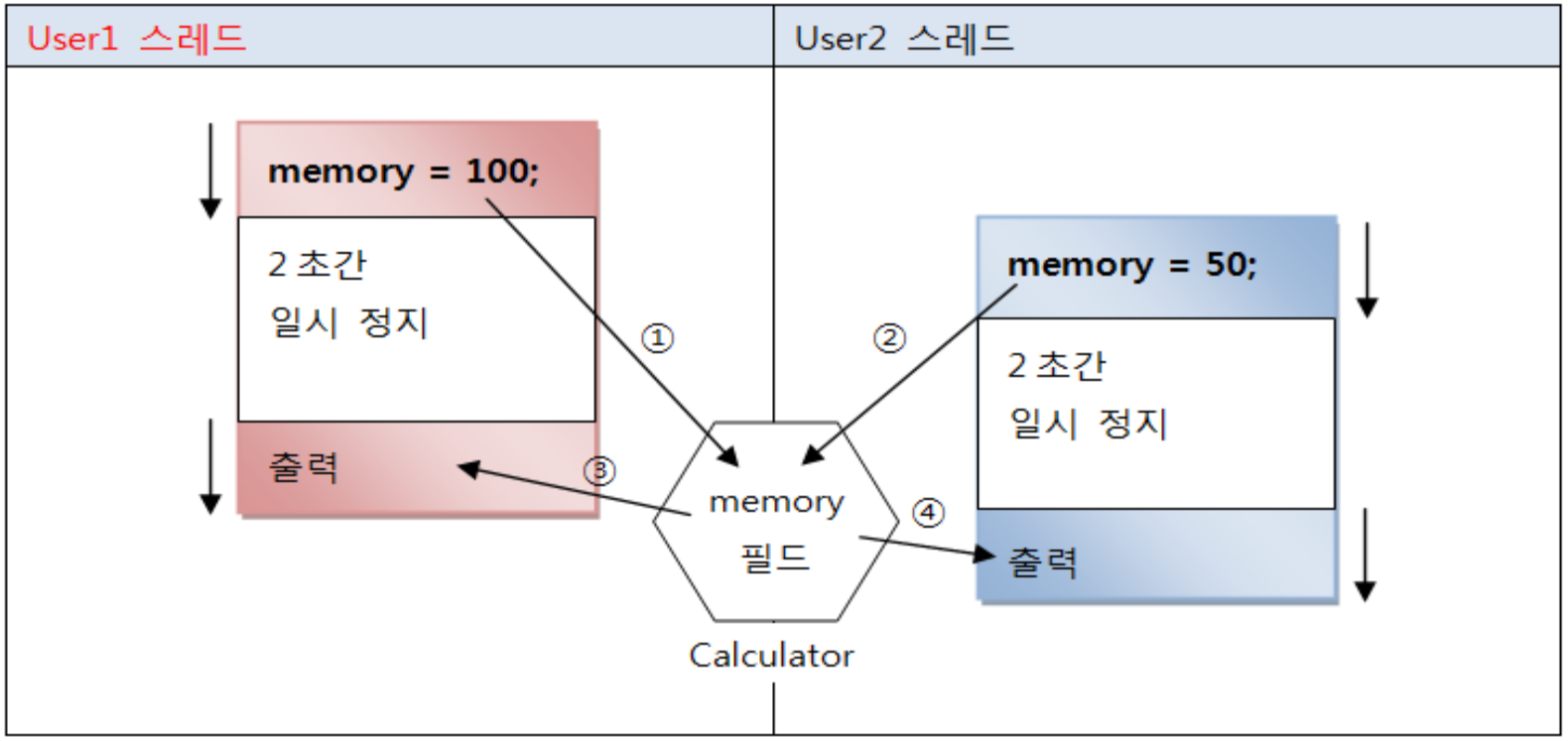
동기화 메소드와 동기화 블록

동기화 메소드와 동기화 블록

❖ 공유 객체를 사용할 때의 주의할 점

- ㅇ 멀티 스레드가 하나의 객체를 공유해서 생기는 오류



동기화 메소드와 동기화 블록

❖ 공유객체 : Calculator.java

```
public class Calculator {
    private int memory;

    public int getMemory() {
        return memory;
    }

    public void setMemory(int memory) {
        this.memory = memory;
        try {
            Thread.sleep(2000);
        } catch(InterruptedException e) {}
        System.out.println(Thread.currentThread().getName() +
                                    ": " +  this.memory);
    }
}
```

동기화 메소드와 동기화 블록

❖ User1 스레드 : User1.java

```
public class User1 extends Thread {
    private Calculator calculator;

    public void setCalculator(Calculator calculator) {
        this.setName("User1");
        this.calculator = calculator;
    }

    public void run() {
        calculator.setMemory(100);
    }
}
```

❖ User2 스레드 : User2.java

```
public class User2 extends Thread {
    private Calculator calculator;

    public void setCalculator(Calculator calculator) {
        this.setName("User2");
        this.calculator = calculator;
    }

    public void run() {
        calculator.setMemory(50);
    }
}
```

❖ 메인 스레드가 실행하는 코드 : MainThreadExample.java

```
public class MainThreadExample {
    public static void main(String[] args) {
        Calculator calculator = new Calculator();

        User1 user1 = new User1();
        user1.setCalculator(calculator);
        user1.start();

        User2 user2 = new User2();
        user2.setCalculator(calculator);
        user2.start();
    }
}
```

동기화 메소드와 동기화 블록

❖ 동기화 메소드 및 동기화 블록 – synchronized

- 단 하나의 스레드만 실행할 수 있는 메소드 또는 블록
- 다른 스레드는 메소드나 블록이 실행이 끝날 때까지 대기해야
- 동기화 메소드

```
public synchronized void method() {
    임계 영역;  //단 하나의 스레드만 실행
}
```

동기화 메소드와 동기화 블록

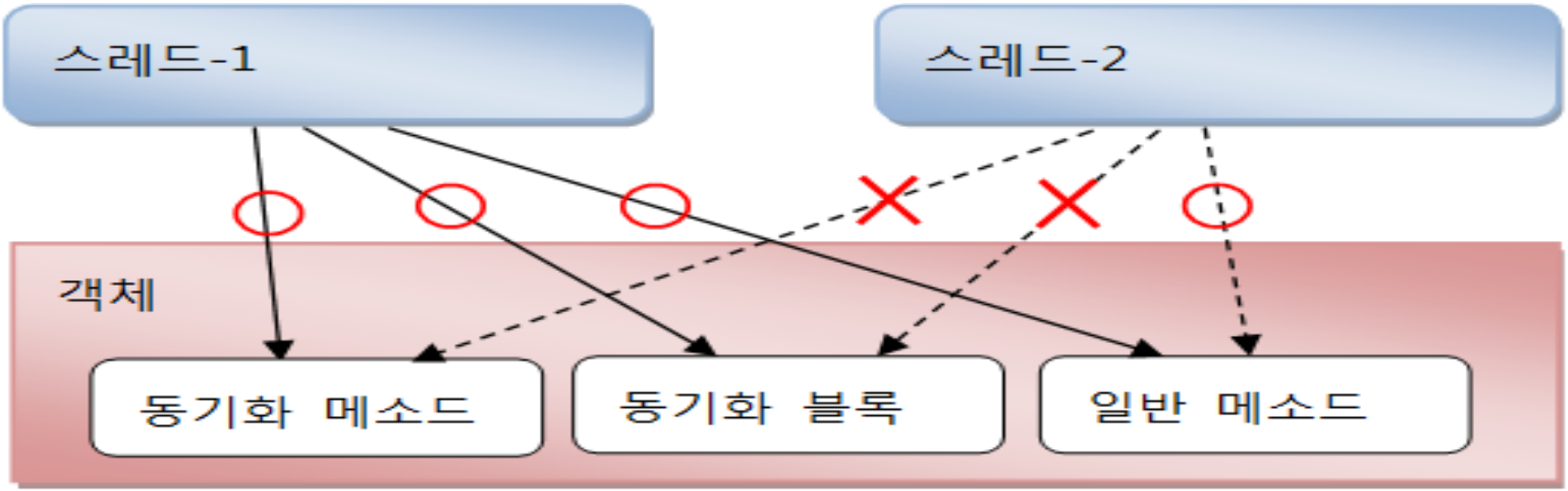
❖ 동기화 메소드 및 동기화 블록 – synchronized

- ○ 동기화 블록

```
public void method () {
   //여러 스레드가 실행 가능 영역
   ...
   synchronized(공유객체) {
      임계 영역 //단 하나의 스레드만 실행
   }
   //여러 스레드가 실행 가능 영역
   ...
}
```

동기화 메소드와 동기화 블록

❖ 동기화 메소드 및 동기화 블록



동기화 메소드와 동기화 블록

❖ 동기화 메서드로 수정된 공유 객체 : Calculator.java

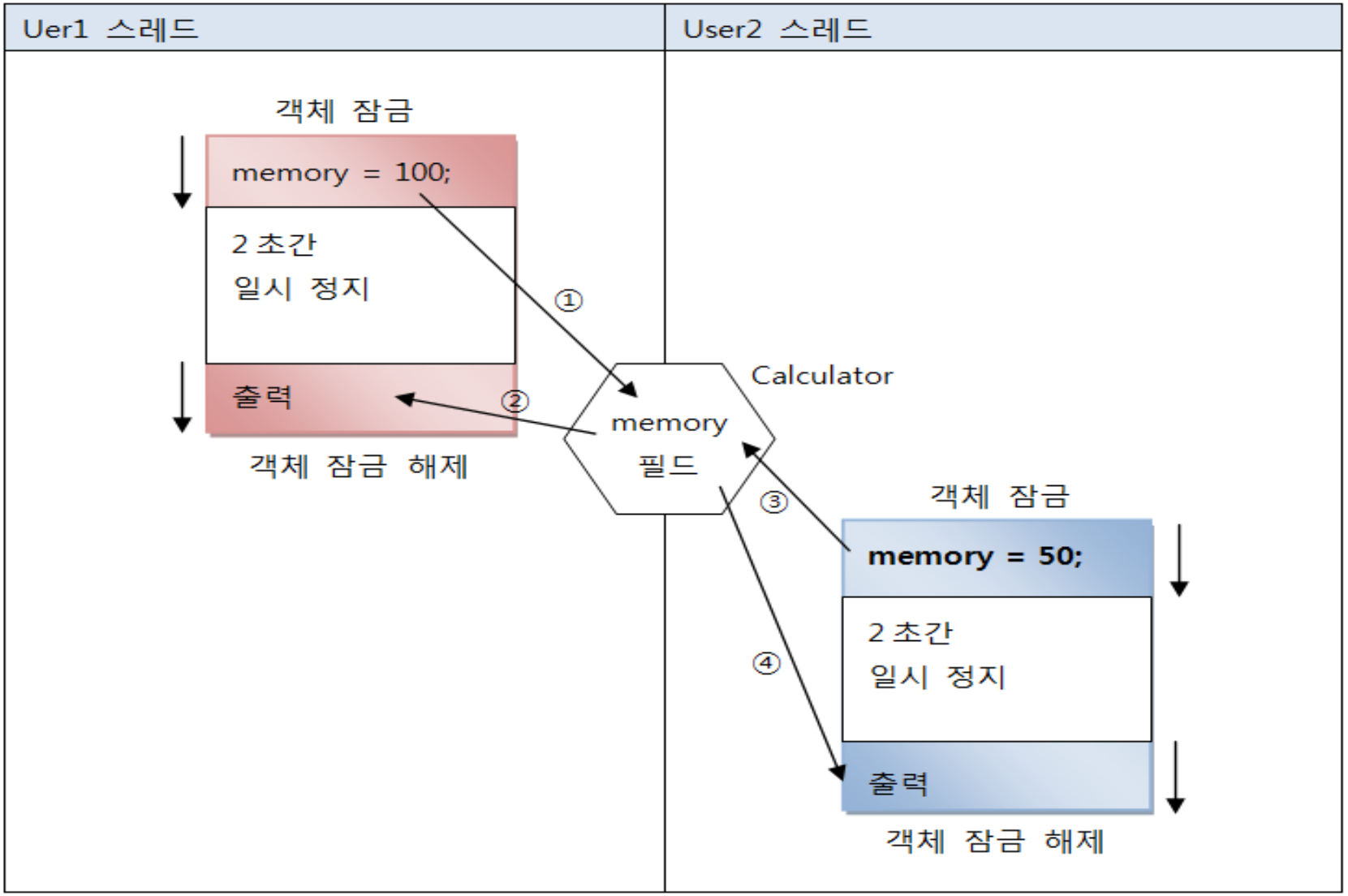
```
public class Calculator {
    private int memory;

    public int getMemory() {
        return memory;
    }

    public synchronized void setMemory(int memory) {
        this.memory = memory;
        try {
            Thread.sleep(2000);
        } catch(InterruptedException e) {}
        System.out.println(Thread.currentThread().getName() +
                           ": " +  this.memory);
    }
}
```

동기화 메소드와 동기화 블록

❖ 동기화 메소드 및 동기화 블록



동기화 메소드와 동기화 블록

❖ 동기화 메소드 및 동기화 블록

```
public void setMemory(int memory) {
    synchronized (this) {
        this.memory = memory;
        try {
            Thread.sleep(2000);
        } catch(InterruptedException e) {}
        System.out.println(Thread.currentThread().getName() + ": " +  this.memory + " 저장");
    }
}
```

공유 객체인 Calculator의 참조(잠금 대상)