Race codition

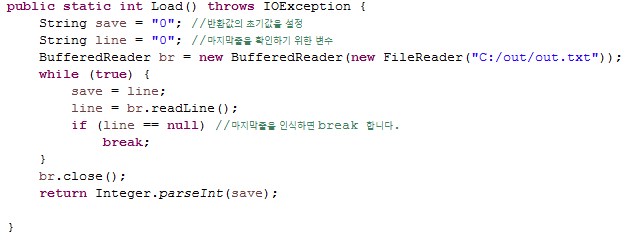
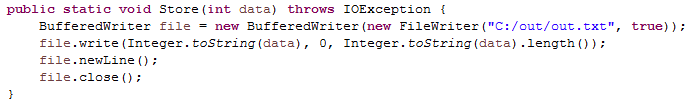
# 컴퓨터공학부 20123420 차경준

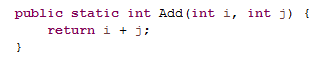
1. 다음 기능을 하는 프로그램(add1)을 만드시오

1)데이타 화일의 마지막 숫자를 읽어서 1을 더해서 그 화일의 끝에 추가해서 기록(데이타 화일은 초기에 0만 써있다.)  
2)위 함수를 1000번 호출  
3)이 프로그램을 실행하면 0 부터 1000까지 1001개의 숫자가 기록될 것이다.

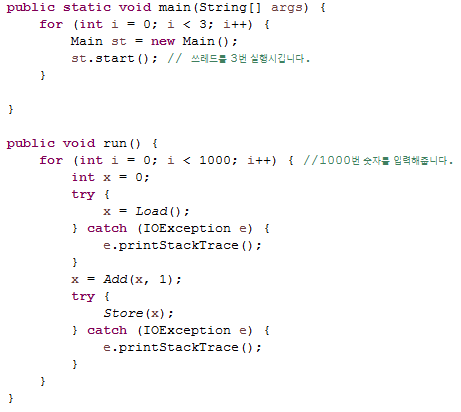
**사용프로그램(언어) : Eclipse(JAVA)**

**결과값 :C:/out/out.txt 출력**

1. Load()
2. Store
3. Add



1. Call

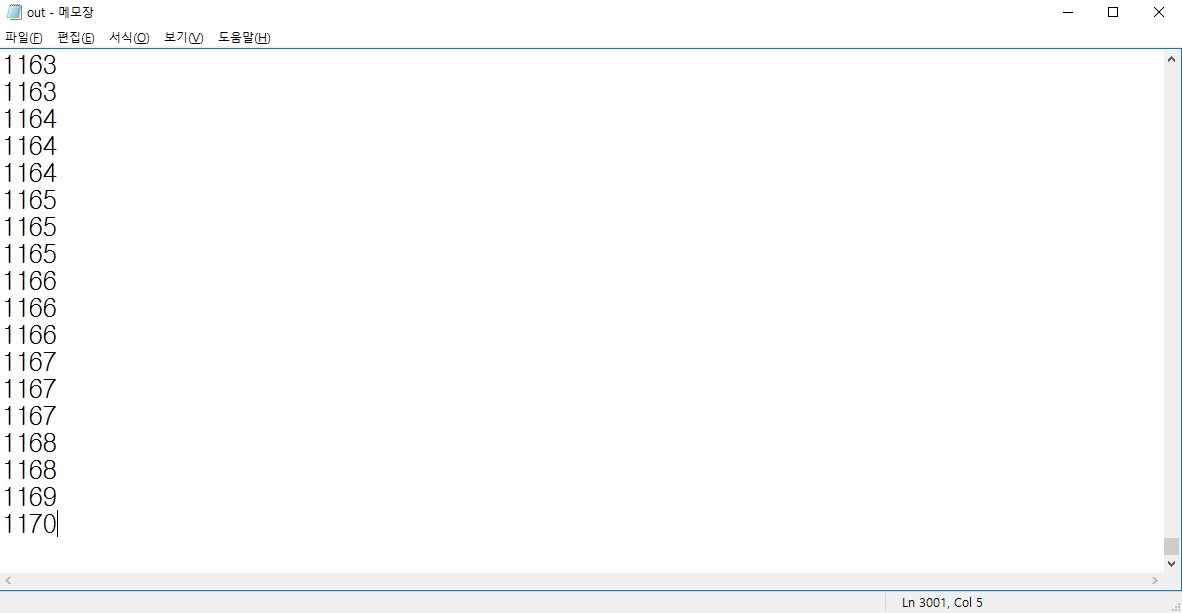


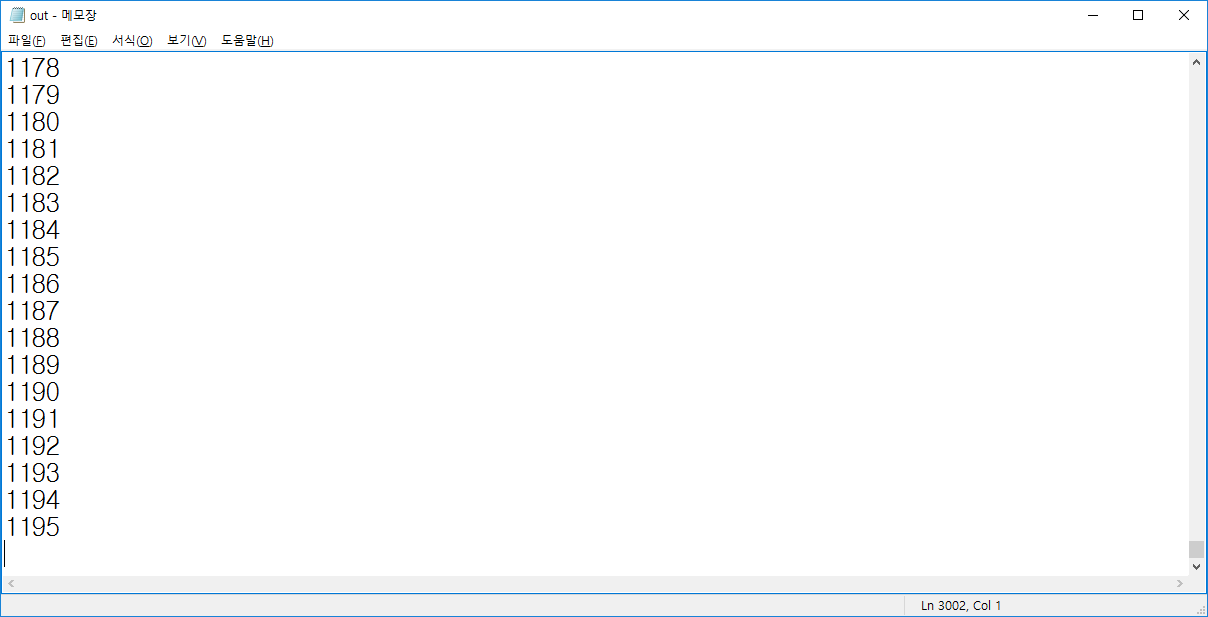
2. 이 프로그램을 두개 이상 동시에 실행하여 결과를 살펴보시오

1) n개를 동시에 실행했을 때 화일의 마지막 숫자가 1000\*n인가?

Thread를 이용하여 프로그램 내부에서 3번 Thread가 동시에 실행하는 환경을 만들고 실행한다.

**결과**





결과값은 1000 \* 3 = 3000이 아니라 1170이다

출력물의 3001line을 보면 Thread는 Call1000번씩 3번씩 실행했지만 최종 값은 1195이다.

2) 결과가 1000\*n이 아니라면 그 원인은 무엇인지 설명하시오

[out.text]는 최종 값을 출력하는 결과물이기도 하지만 동시에 프로그램이 Load()를 사용하여 마지막 줄의 숫자(자원)을 가져오는 Memory인데 여기서 3개의 thread 가 동시에 실행되면서 같은 자원을 동시에 사용하여 숫자를 중복되게 사용한다. 그리하여 총 숫자는 3000번 입력되었지만 최종 값은 1195, 1170 같은 적은 숫자가 출력된다.

사용자가 출력 값이 3000이기를 의도한다면 thread가 자원을 사용할 때 다른 thread가 같은 자원을 사용할 때 중복 사용되는 것을 거부하는 memory의 잠금 기능이나, 다른 thread의 접근을 인식하는 기능을 추가하면 의도한 값으로 출력될 것이다.