



소프트웨어프로젝트 02분반

project1 report

소프트웨어학부

20181269 박현우

I. 소스 프로그램

```
public class ExecTime {  
    public static void main(String[] args) {  
        int a=3;  
        int result;  
        long startTime, endTime, execTime1,execTime2;  
  
        startTime=System.nanoTime();  
  
        result=a*a;  
  
        endTime=System.nanoTime();  
        execTime1=endTime-startTime;  
  
        startTime=System.nanoTime();  
  
        result=(int)Math.pow(a,2);  
  
        endTime=System.nanoTime();  
        execTime2=endTime-startTime;  
  
        System.out.println("Execution Time of a*a = " +execTime1);  
        System.out.println("Execution Time of a^2 = " +execTime2);  
    }  
}
```

II. 프로그램 설명

일단 class 이름을 ExecTime으로 지정해주고 a를 3으로 초기화시킨다. 그리고 계산 결과값을 담을 변수 result를 선언하고 각각 계산 전 시간, 계산 후 시간, 두 시간의 차이를 나타내는 변수 startTime, endTime, execTime1, execTime2를 선언한다.

먼저 $a \times a$ 로 계산하는 방법의 수행시간을 알아보기 위해 계산을 수행하기 전의 시간을 startTime에 저장한다. 이 때, 현재 시간을 나타내주는 메소드인 System.nanoTime()을 이용한다. 그 후에 $a \times a$ 의 계산을 진행한다. 계산을 수행하고 나서의 시간을 endTime에 저장한다. 두 시간의 차이가 계산 수행 시간이므로 이 값을 execTime1에 저장한다.

이번에는 지수승을 이용하여 계산하는 방법의 수행시간을 알아본다. 앞에서와 마찬가지로 계산을 수행하기 전의 시간을 startTime에, 계산을 수행한 후의 시간을 endTime에 저장한다. 그리고 이 두 시간의 차이를 execTime2에 저장한다. a의 제곱을 계산할 때에는 Math.pow()라는 메소드를 이용한다. 이 메소드의 입출력은 모두 double형이기 때문에 int형으로 형 변환을 시켜준 후 result에 값을 저장해야 한다.

마지막으로, execTime1과 execTime2에 저장된 값을 System.out.println을 이용해 출력하면 두 값을 비교해볼 수 있다.

III. 실행 화면

command 창에서의 실행 화면

```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\Java\jdk1.8.0_201\bin

C:\Program Files\Java\jdk1.8.0_201\bin>javac ExecTime.java

C:\Program Files\Java\jdk1.8.0_201\bin>java ExecTime
Execution Time of a*a = 427
Execution Time of a^2 = 20906

C:\Program Files\Java\jdk1.8.0_201\bin>java ExecTime
Execution Time of a*a = 0
Execution Time of a^2 = 14507

C:\Program Files\Java\jdk1.8.0_201\bin>java ExecTime
Execution Time of a*a = 427
Execution Time of a^2 = 31574
```

Eclipse에서의 실행 화면

```
ExecTime.java
1 package exectime;
2
3 public class ExecTime {
4     public static void main(String[] args) {
5         int a=3;
6         int result;
7         long startTime, endTime, execTime1,execTime2;
8
9         startTime=System.nanoTime();
10
11         result=a*a;
12
13         endTime=System.nanoTime();
14         execTime1=endTime-startTime;
15
16         startTime=System.nanoTime();
17
18         result=(int)Math.pow(a,2);
19
20         endTime=System.nanoTime();
21         execTime2=endTime-startTime;
22
23         System.out.println("Execution Time of a*a = " +execTime1);
24         System.out.println("Execution Time of a^2 = " +execTime2);
25     }
26 }
```

```
<terminated> ExecTime (1) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (2019. 3. 18. 오전 1:22:35)
Execution Time of a*a = 426
Execution Time of a^2 = 24747
```

IV. 추가 질문에 대한 의견

① 수행시간이 짧아서 nano-second보다 작다면 어떻게 측정할까?

System.nanoTime 외에 현재 시간을 나타내주는 다른 메소드가 있는지 알아보아야 할 것 같다. 그 메소드는 nano-second보다 작은 범위까지 나타내는지를 확인해보았다. 다른 메소드 중에 System.currentTimeMillis라는 메소드가 있는데, 이것은 milli-second 단위로 표시를 해주어 nano-second보다 훨씬 좁은 범위만을 나타내준다. 따라서 nano-second보다 수행시간이 짧은 경우에 어떻게 측정해야 하는지에 대한 해답은 내리지 못했다.

② 측정값의 편차가 크다면 어떻게 보정할 수 있을까?

측정값의 편차가 크다면 측정을 여러 번 실행한 후 그것의 평균값을 구하는 것을 통해 보정을 할 수 있을 것 같다.

V. 자체평가표

| 평가 항목 | 학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견) | 평가 (빈칸) | 점수 (빈칸) |
|---------------------------------------------------------------------------------------------------|----------------------------------------------------|------------|------------|
| 에디터와 명령어를 이용한 실행 - 각 단계별 결과물 과 그 의미는? - 요구사항을 만족한 구현? - 충분한 실험? (수행시간이 작은 경우 포함?) | 각 단계별로 결과물을 도출하여 글을 작성하지 않고 전체적인 프로그램 설명만을 한 것 같다. | | |
| Eclipse를 이용한 실행 | 따로 기술할 부분은 없다. | | |
| 추가 질문에 대한 의견 | 추가 질문에 대한 해답을 제대로 내리지 못한 것 같다. | | |
| 리포트 작성 - 평가 항목에 맞게 리포트 작성? - 모든 파일을 하나의 문서로. | 모든 파일을 하나의 문서로 묶었다. | | |
| 기타 추가 설명 (필요한 경우) | | | |
| 총평/계 | 리포트를 체계적으로 작성하지 못한 것 같아 아쉬움이 있다. | | |