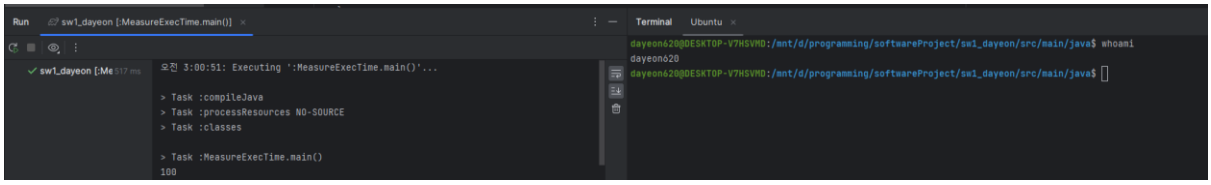
2025 소프트웨어 프로젝트 과제1

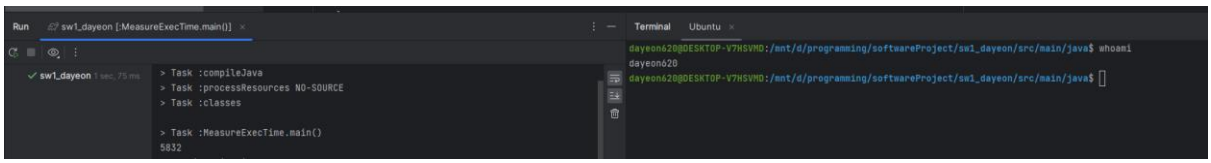
산업보안학과 20213509 정다연

MeasureExecTime.java 기능 테스트

우선적으로, 임의의 수를 한번 반복하여 원하는 결과가 제대로 나오는지 테스트를 진행하였다.



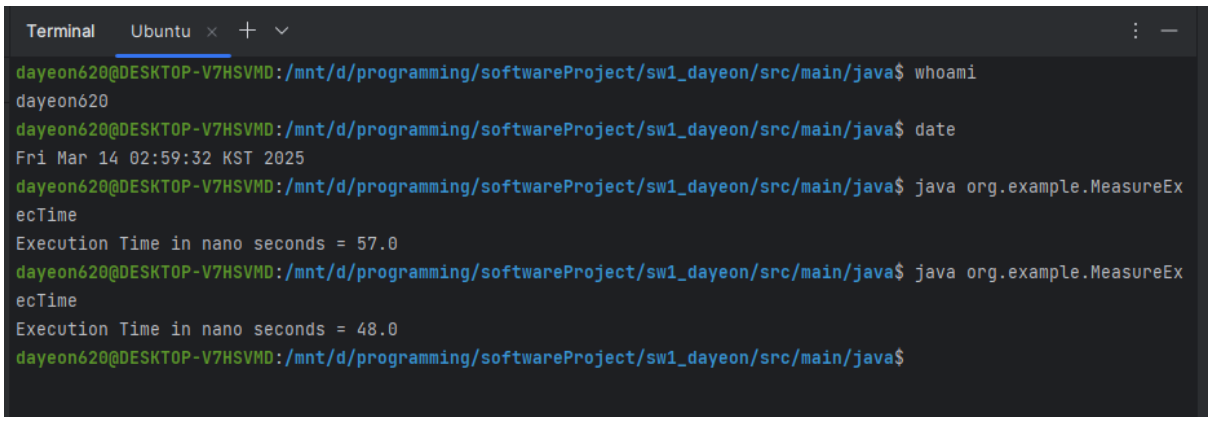
입력값을 10으로 두었을 때 정상적으로 100이 출력되는 것을 확인할 수 있다.



18일 때에도 정상적으로 5832가 출력되는 것을 확인하였다. 이를 기반으로 N만큼 반복하여 실행 시간을 측정해보도록 한다.

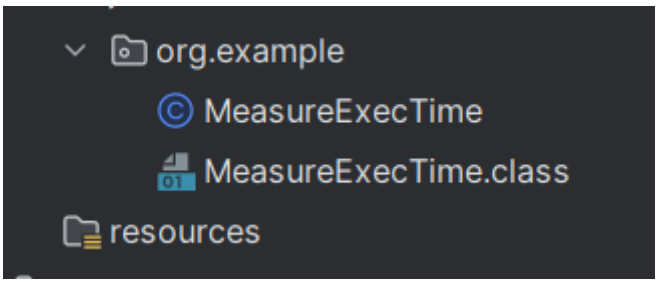
Ubuntu 환경의 터미널 창을 이용한 실행

우분투 환경에서 컴파일 후 실행한 결과 78.0의 결과가 나오는 것을 확인할 수 있다.



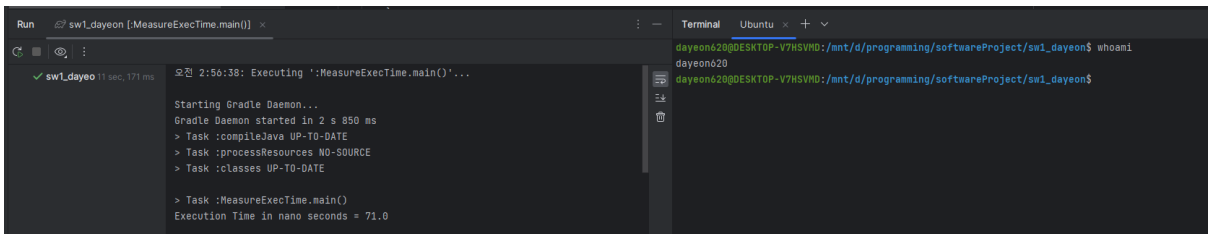
반복적으로 수행해본 결과 주로 40~70 사이의 숫자를 보이는 것을 확인할 수 있었다.

이때, javac 명령어를 이용하여 컴파일 하게 되면 class 파일이 나오게 된다.

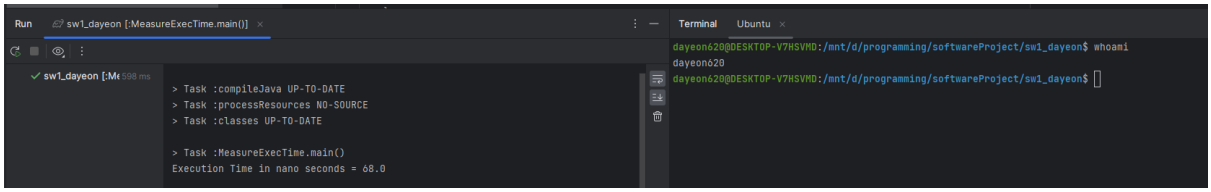


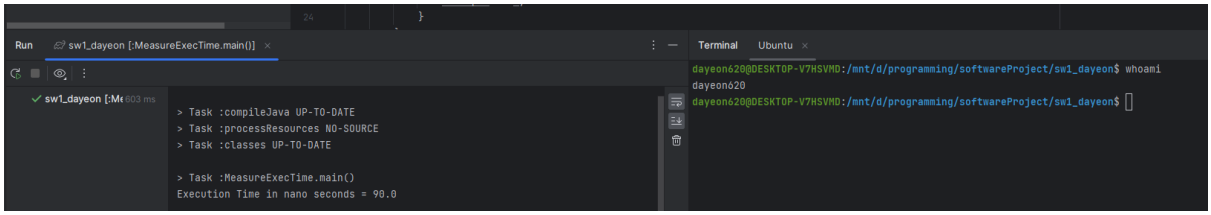
해당 파일을 java 명령어를 통해 실행할 수 있다. 여기서의 .class 파일은 바이트코드 파일로 JVM 이 이해할 수 있는 명령어로 변환된 파일이며 이를 이용하여 java 명령어를 이용하여 실행할 수 있게 된다.

Intellij 개발도구를 활용한 실행



해당 환경에서는 72이라는 결과가 나오는 것을 확인할 수 있다. 이때, 여러 번 실행해 본 결과





68 90 등 50~100 사이의 결과값을 보이는 것을 확인할 수 있다.

소스 프로그램

```
package org.example;

public class MeasureExecTime {
    public static void main(String[] args) {

        long startTime, endTime, execTime;
        int N = 100000; // 반복  수행  횟수
        startTime = System.nanoTime();
        for (int i=0; i<N; i++) {// 반복  수행을  해서  측정을  하면, 보다  정확한  결과
            // statement(s) to be measured
            productDivisors (18);
        }
        endTime = System.nanoTime();
        execTime = endTime - startTime;
        System.out.println("Execution Time in nano seconds = " + (double)(execTime/N));

    }

    static long productDivisors(int num){
        long multiple = 1;
        for (int i=1; i<=num; i++){
            if (num % i == 0){
                multiple *= i;
            }
        }

        return multiple;

    }
}
```