

# 과제 진행 순서

## 1. 프로젝트 1 설명

- 1) 프로그램 실행 시간을 측정하는 Java 프로그램:
- 2) 약수 개수를 계산하는 Java 프로그램:
- 3) 주의: 리포트 내용, 제출 양식에 대한 요구사항이 있는 pdf 필독.

## 2. 에디터/명령창에서 구현

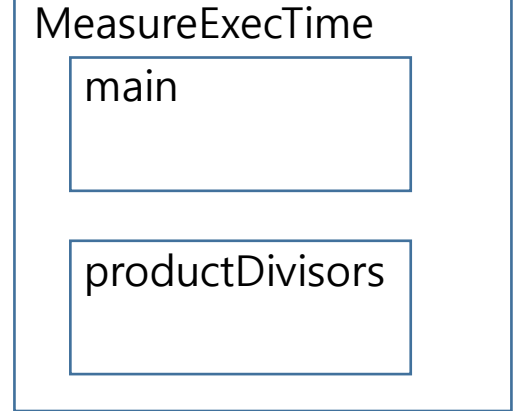
- 1) C언어와의 차이 설명에 유의하면서 editor에서 코드 작성
- 2) javac MeasureExecTime.java
- 3) Java MeasureExecTime
- 4) 관련 ppt 참조

## 3. Eclipse 사용법 및 구현 (다른 개발 도구 사용 가능)

- 1) 같은 프로그램을 Eclipse 에서 작성/실행
- 2) 관련 ppt 참조

# 코드 설명

- `class MeasureExecTime`
  - 모든 함수(메소드)는 class 안에 정의
- `public static main(String[] args)` 작성
  - 실행 시간 측정하는 프로그램으로 구성 (다음 페이지 코드 참조)
  - 추후, 측정 대상에 `productDivisors()` call 을 넣어서 측정
- `static long productDivisors(int num)` 작성
  - 인자로 주어진 정수의 약수들의 곱을 리턴하는 함수
  - 당분간 static은 그대로 사용
  - 검색을 해서 그대로 쓰지 말고,
  - 자신이 코딩을 해보고, 실행 시간을 측정해보기 바람.
  - 주의) 실행 시간을 개선하는 것이 목적이 아님.
    - 실제 실행 시간을 측정하는 방법을 익히기만 하면 됨.



# main () 에 들어갈 실행시간 측정 코드

```
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));
```

# Eclipse에서 project1 project 만들기

## 1. New > Java project

- 예) project name : project1

## 2. New > Class

- 예) class name : MeasureExecTime
- static main() 포함으로

## 3. 소스 프로그램 추가 1

- main()에 코드 추가 : 앞 슬라이드의 코드 사용

## 4. 소스 프로그램 추가 2

- static int numCommonDivisor() : 과제로 작성

## 5. Run> Run Configurations

## 6. Run> Run