1. 간단한 인스타그램 클래스를 작성해보자!

|  |  |
| --- | --- |
| Class | instagram |
| Field | (+):**int** like;  (+):**String** talk; |
| 생성자 |  |
| Method | **(+): void** talk\_like()  **(+): void** talk\_hate()  **(+): void** talk\_upload(String)  **(+): void** print() |

-talk\_like 메소드는 호출시 필드에 있는 like가 증가합니다.

-talk\_hate 메소드는 호출시 필드에 있는 like가 감소합니다.

-talk\_upload 메소드는 매개변수를 통해 글을 작성 후 필드에있는 talk에 저장해줍니다.

-print 메소드는 현재 필드에 저장된 글과 좋아요 개수를 출력해줍니다.

-모든 값들은 테스트 클래스에서 설정해줍니다.

\*테스트 클래스 생성 후 객체 생성해서 테스트한 화면을 보여주세요.



1. 간단한 은행 클래스를 작성해보자!

|  |  |
| --- | --- |
| Class | Bank |
| Field | (+):**int** pay;  (+):**int** real\_pay;  (+):**double** tex;  (+):**String** name; |
| 생성자 |  |
| Method | **(+): void** setReal\_pay()  **(+): void** setTex() |

-setReal\_pay 메소드는 세금을 제외한 금액을 정수형 변수 real\_pay에 저장하는 메소드입니다.

-setTest 메소드는 원래 월급에 4.7%센트 적용합니다.

-모든 값들은 테스트 클래스에서 설정해줍니다.

\*테스트 클래스 생성 후 객체 생성해서 테스트한 화면을 보여주세요.



1. 2차원 배열을 사용해서 행렬 곱 프로그램을 만들자.

- 무한 반복문 사용할 것

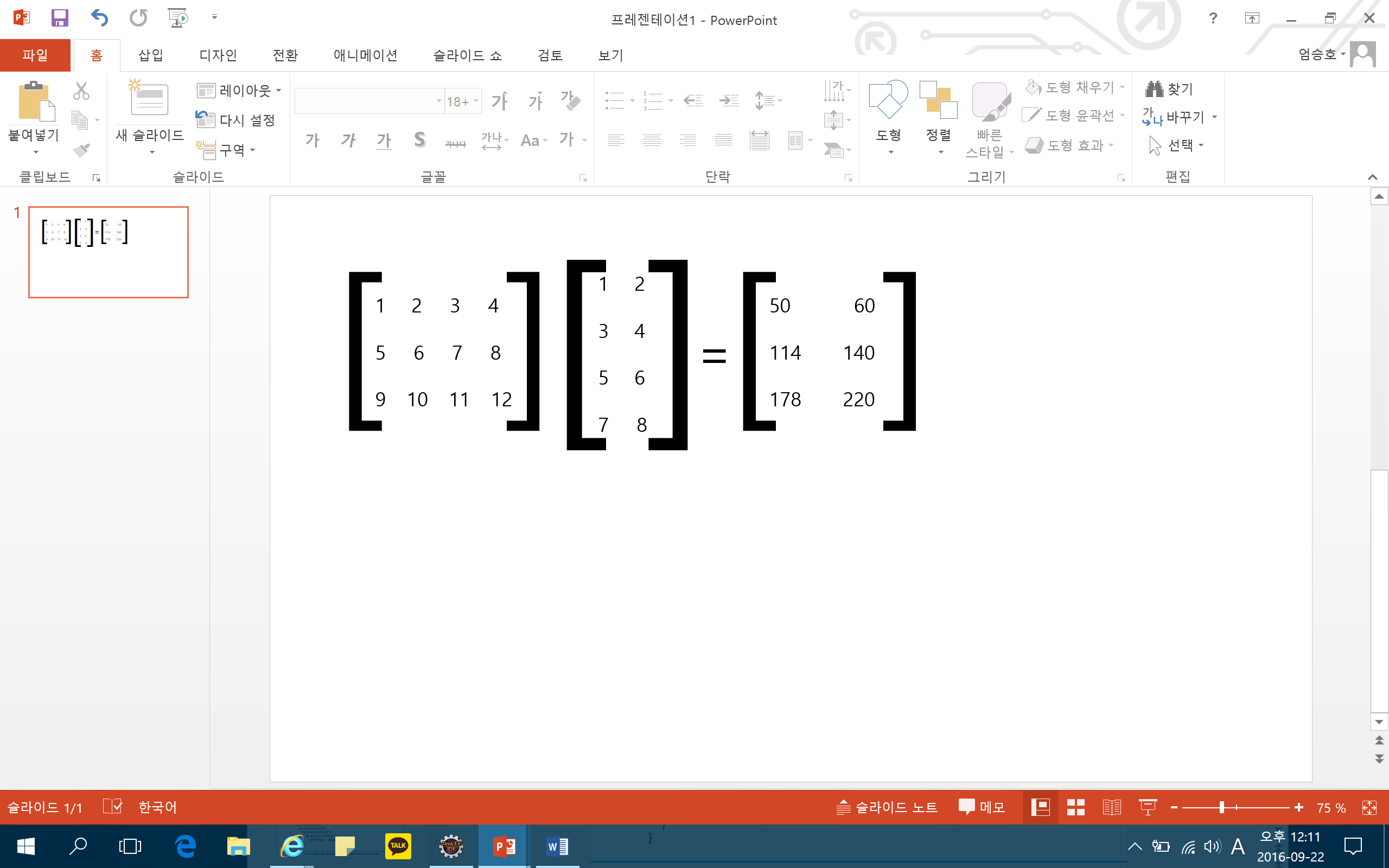
- 행렬1과 행렬2의 배열의 크기를 입력 받아서 사이즈를 정해줄 것

- 행렬1의 열과 행렬2의 행이 일치하지 않으면 종료(탈출조건)

- 행렬3의 행렬 크기는 행렬1의 행과 행렬2의 열이다.

- 행렬1과 행렬2의 원소는 1~10까지의 랜덤 숫자를 넣어줄 것

- 행렬 값들은 for each문으로 출력할 것^^



[행렬1] [행렬2] [행렬3]

