

* 계산기 만들기 — 1 단계 : 프로젝트 생성

↳ 작은 프로젝트를 통해 배운 방법을 정리해 보자!

```
bitcamp-study$ mkdir project-app1
```

```
    $ cd project-app1
```

```
project-app1$ gradle init
```

```
    :
```

```
패키지명 com.eomcs.app1
```

* 계산기 만들기 — 단계: eclipse IDE로 프로젝트 생성

- ① build.gradle 파일 편집 → ② eclipse IDE 설정 파일 생성 → ③ eclipse IDE로 프로젝트 인ポート
- eclipse 플러그인 추가
 - JDK 설정 추가
 - gradle eclipse
 - 프로젝트명 설정 추가

* 계산기 만들기 - 3단계 : 명령창에서 계산기 실행하기

\$ java -cp bin/main com.eomcs.appl.App add 100 200

↓ 실행파일
↑ 자바 프로그램
↑ 명령
↑ 인자

100 + 200 = 300

* 패키지

- 프로그램 아규먼트 다루기 (com.eomcs.lang.ex07.Exam05.x)
- 소인수 (com.eomcs.lang.ex06.x)
- 유클리드 (com.eomcs.lang.ex09.x)
- 피보나치 (com.eomcs.lang.ex05.x)
- 배열 (com.eomcs.lang.ex04.x)

* 계산기 만들기 - 4단계 : 도출할 기능 추가

\$ java -cp bin/main com.eomcs.appl.App } 실행결과

사용법:

App [명령] [값1] [값2]

명령:

add [값1] [값2] 더하기 계산은 수행한다 예) App add 100 200

minus [값1] [값2] 빼기 계산은 수행한다 예) App minus 100 200

* 계산기 만들기 — 5단계 : 키보드로부터 입력 받기

\$ java -cp bin/main com.eomcs.appl.App

> help

add [값1] [값2] 덧셈하기 계산은 수행한다

minus [값1] [값2] 뺄셈하기 계산을 수행한다

help 도움말을 출력한다

> add 100 200

100 + 200 = 300

> add 100

명령어 입력 형식이 틀리 있습니다.

> multiple 100 200

지원하지 않는 연산자 입니다.

* 필기사항

- 키보드 입력은 다루는 법
(com.eomcs.lang.expr.*)

* 계산기 만들기 - 6단계 : 리팩토링 (메시드 활용)

① prompt 기능을 메서드로 분리 \longrightarrow prompt() {-}

② 각 명령어 처리 코드를 메서드로 분리.

add 명령어 \longrightarrow doAdd() {-}

minus 명령어 \longrightarrow doMinus() {-}

help 명령어 \longrightarrow doHelp() {-}

* 메서드 분법은 이렇게
코드를 기능단위로 분리.



✓ 유지보수가 쉬움.

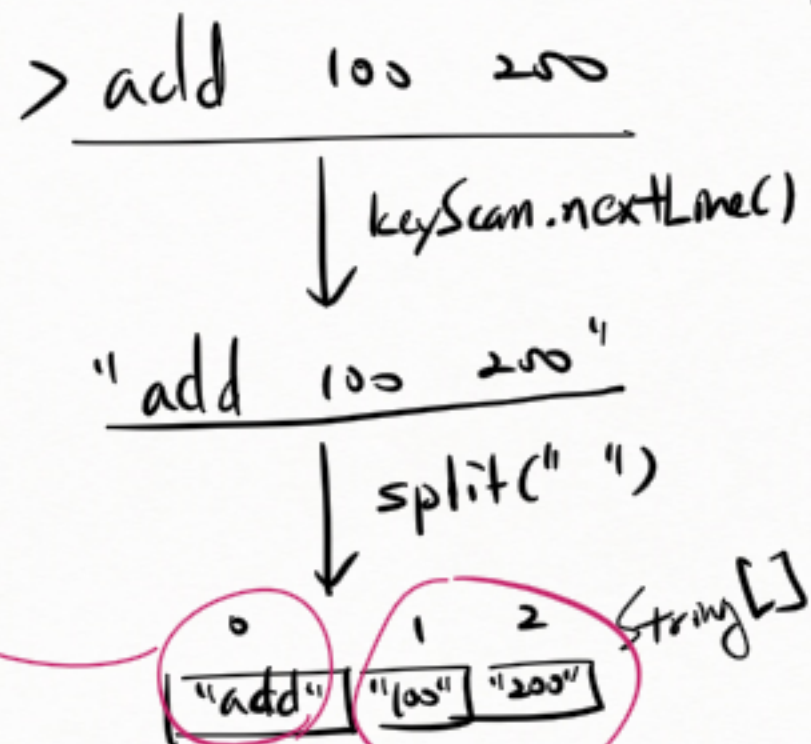
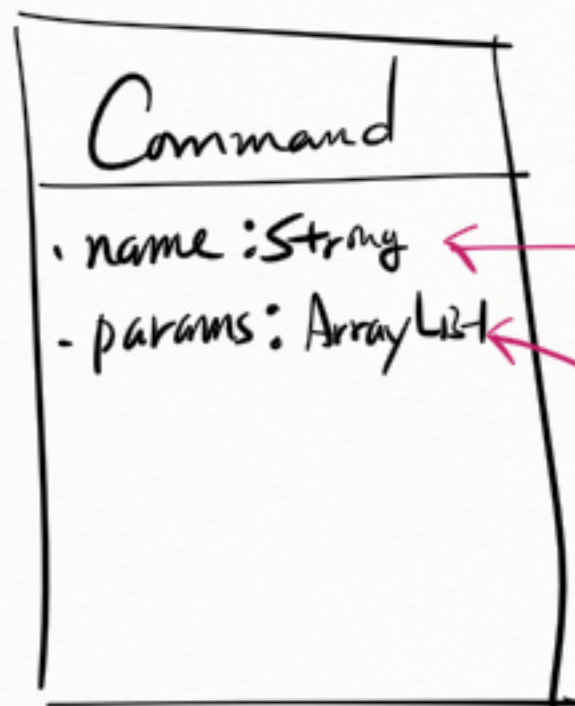
✓ 재사용성이 높음.

✓ 코드 중복을 줄임.

* 계산기 만들기 — 7단계 : 리팩토링 (클래스 분할)

① 사용자가 입력한 명령은 객체로 바꾸기 → Command 클래스 정의

↳ 명령어와 값을 사용하기 위해
필요한 분리해서 저장한다.



저장

저장

* 계산기 만들기 — 응용예 : 리팩토링 (클래스원형)

