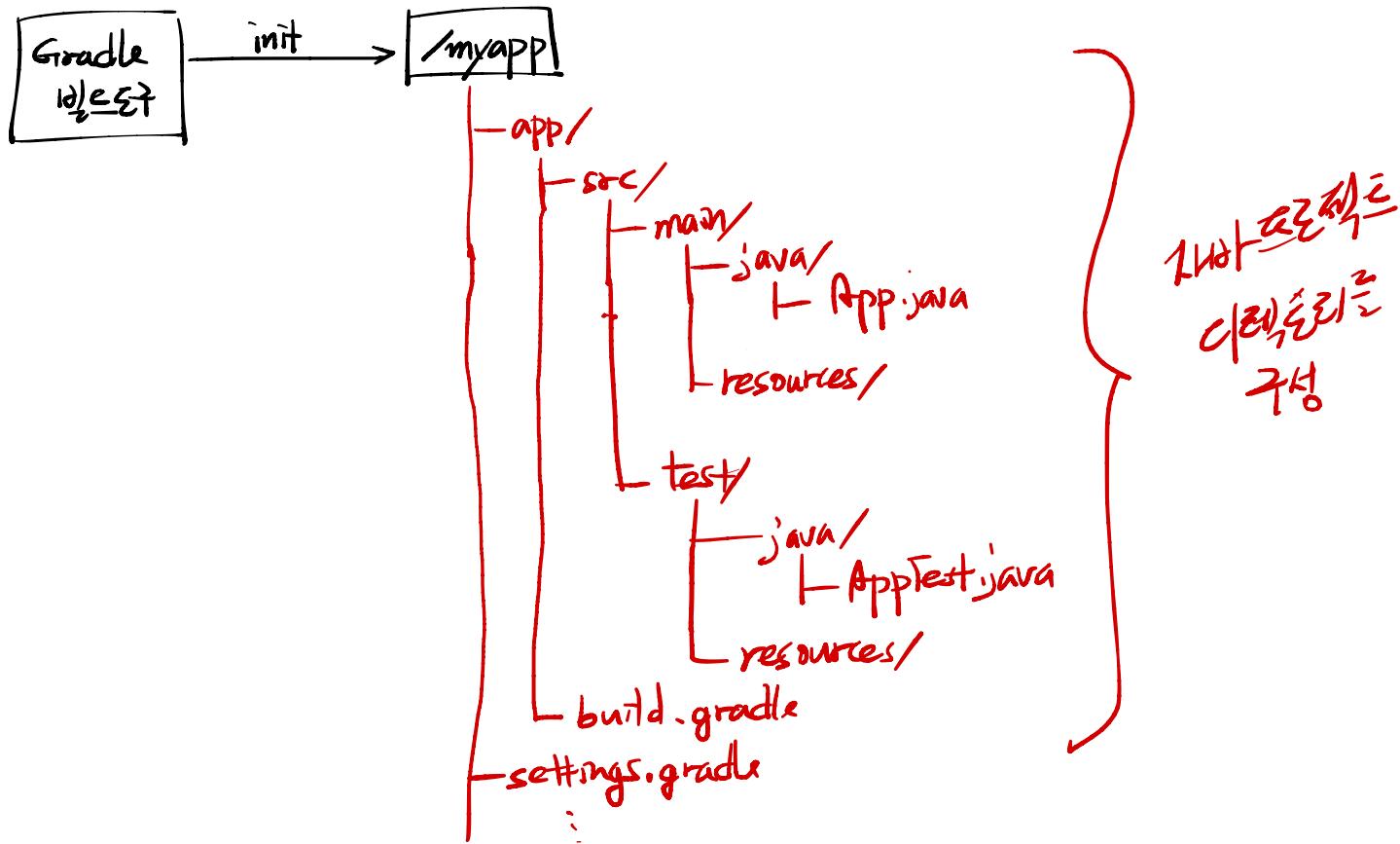
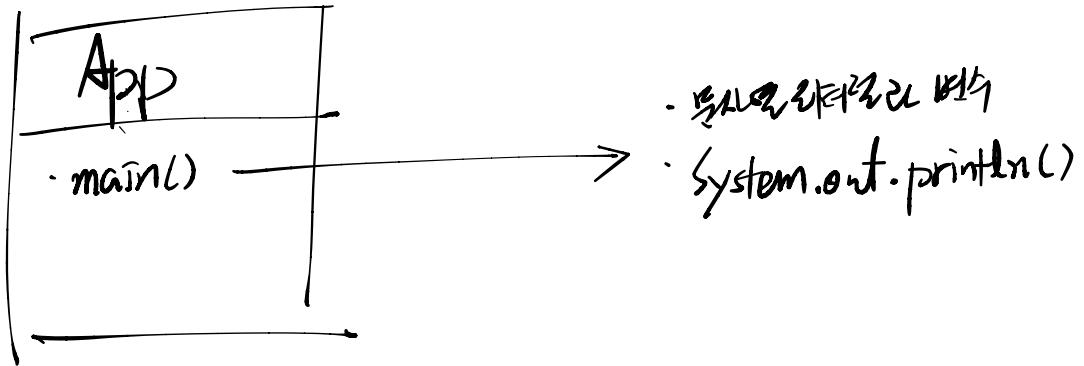


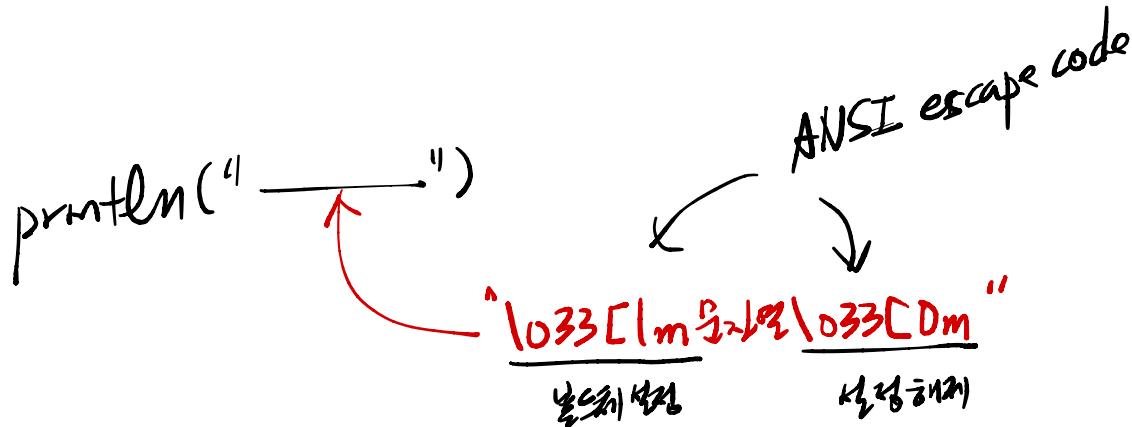
## 01. 자바 프로젝트 준비



02. 자바를 사용하는 방법으로 출력문을 출력하기



03. ANSI escape 코드를 사용하여 콘솔 출력을 조작하기



\* gradle run --quiet

↳ gradle 실행 과정을 제거  
gradle 실행 과정을 설명하는 문구

## 04. 키보드 입력 다루기

main() →

- Scanner keyboard = new Scanner();
- keyboard.nextInt()  
↳ next(), nextLine(), ...
- keyboard.close();

System.io

## 05. 배열을 활용하여 메뉴 출력하기

```
String menu1 = "1. 퇴원";
```

```
String menu2 = "2. 청진";
```

```
:
```

```
:
```

```
:
```



String[] menus = new String[] {

"퇴원", "청진", "재입원", ...

```
}
```

String  
퇴원 선택됨

String  
재입원 선택됨

for ( ; ; ) {

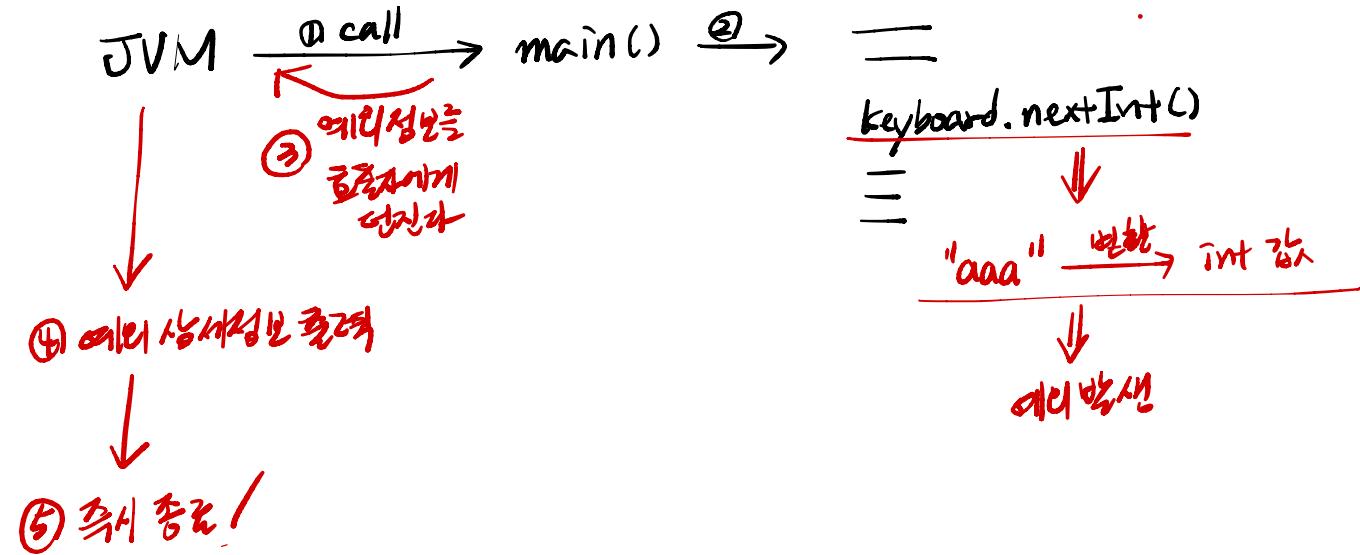
==

```
}
```

for ( ; ; ) {  
 선택된 메뉴 출력하기

## 06. 예외 처리

### 예외 처리 전



## 06. 예외처리

예외처리 후

JVM  $\xrightarrow{\text{① call}}$  main()  $\xrightarrow{\text{②}}$

try {

==

keyboard.nextInt()

==

"aaa"  $\xrightarrow{\text{변환}}$  int 값

↓

예외발생

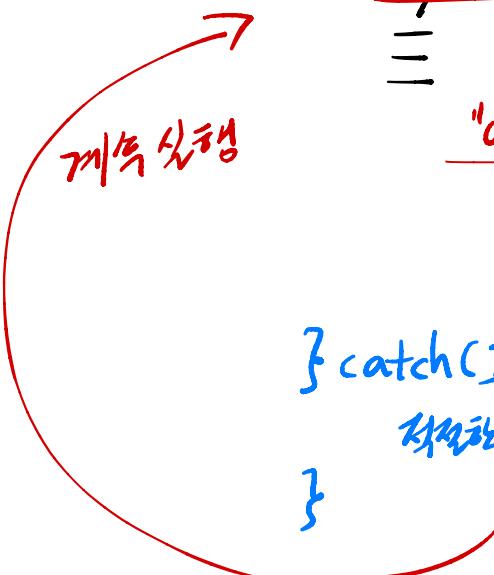
→ 전달

} catch (InputMismatchException ex) {

자세한 조치 취함

}

예외처리 방법을 통해  
예외 상황을 통제하여  
JVM에게 알리지 않고도 예  
외상을 계속 유지.



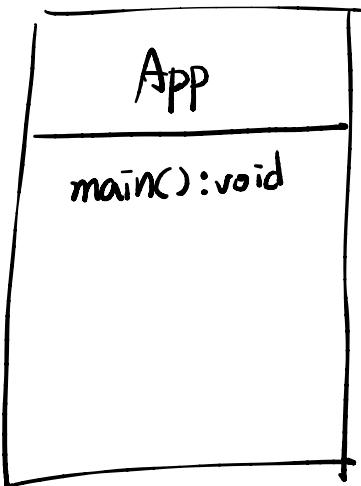
07. 문자열 바꾸기 쓰기위해 어떤 걸까요?

int menuNo = keyboard.nextInt() }  
} ⇒

String command =  
    keyboard.nextLine();  
  
if (command.equals("menu")) {  
    <sup>문자열 비교</sup>      ↑ 문자열 비교  
}  
  
int menuNo =  
    Integer.parseInt(command);  
  
String "2" → 2  
                ↓  
                INT

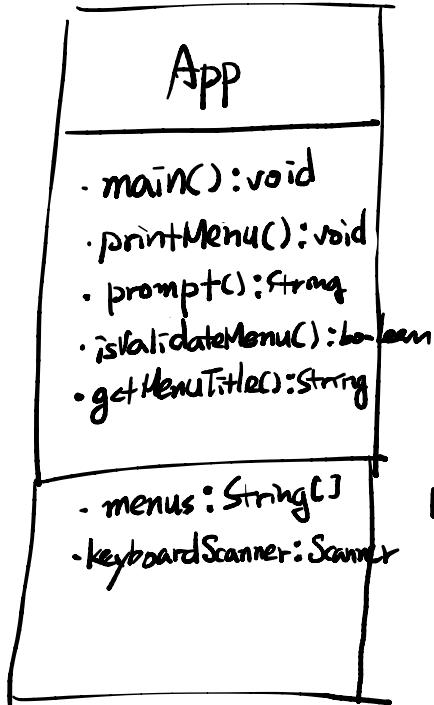
## 08. 1개의 메소드를 대량으로 쓰기 : 클래스 추상화

07.



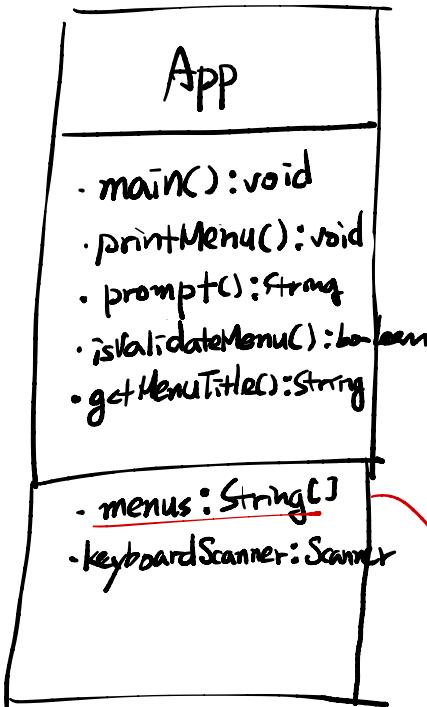
*refactoring*  
↓  
"extract method"

08.



↑  
기능  
↓  
O/FN  
↓  
기능 시각화  
↓  
기능 추상화  
↓  
기능 선택

## 09. 자바 기본문법 활용 연습



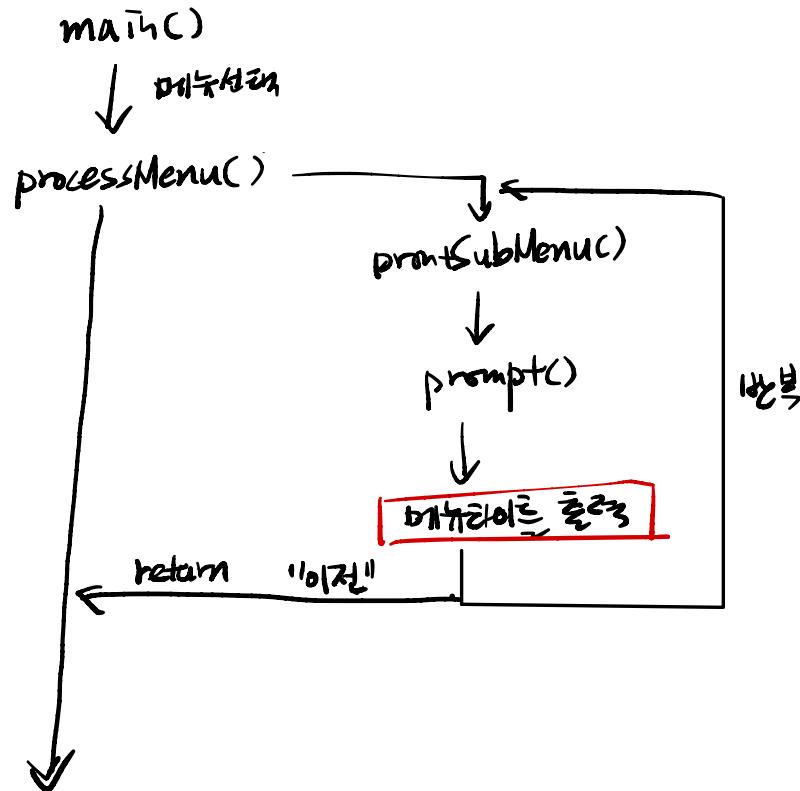
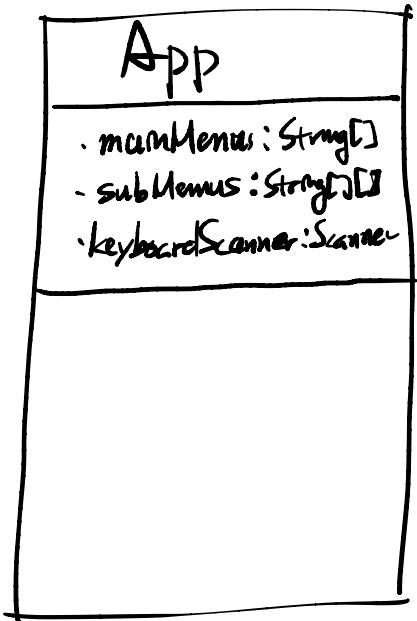
→ Handwritten code generated from the App class:

```
+ printSubMenu(){} -}
+ prompt(String title){} -}
+ validateMenu(int menuNo, String[] menus){} -}
+ getMenuTitle(int menuNo, String[] menus){} -}
+ processMenu(String menuTitle, String[] menus){} -}

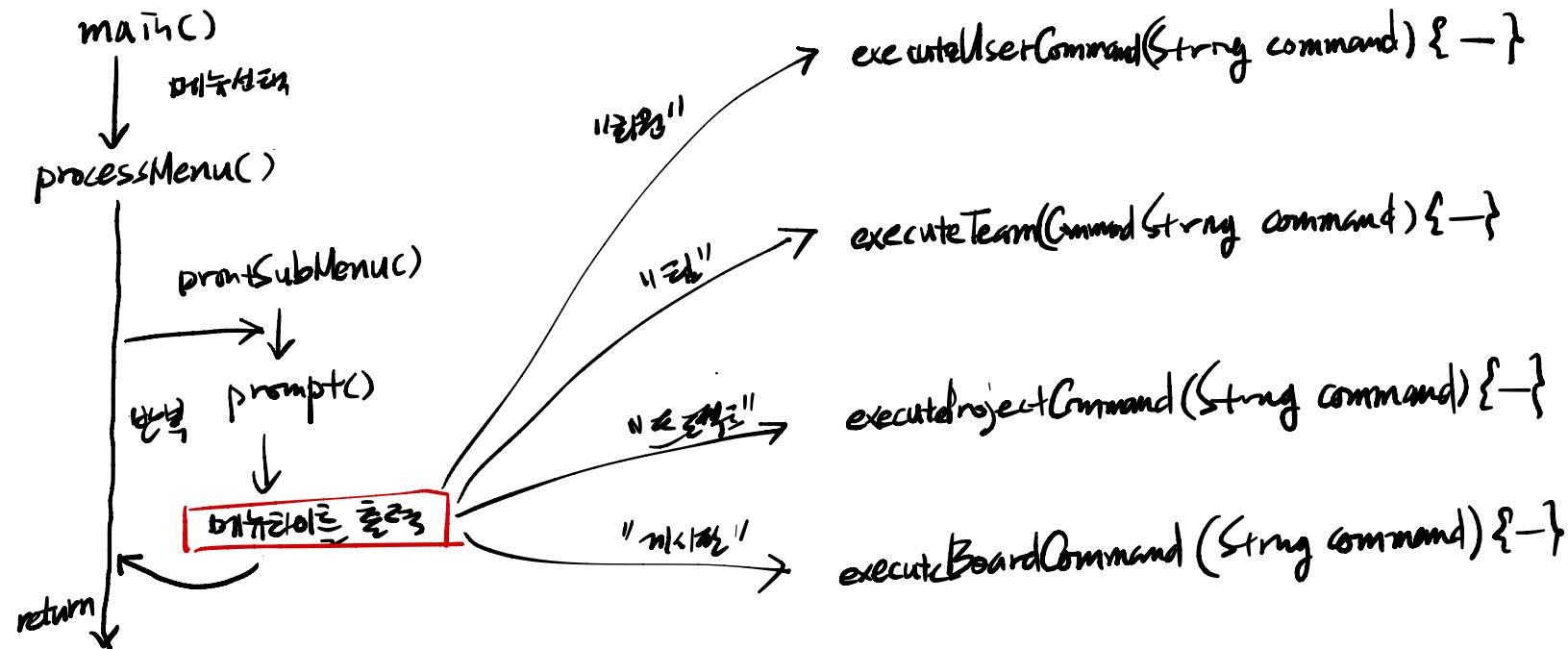
+ mainMenus: String[]
+ subMenus: String[][]
```

## 10. CRUD 퀴즈하기

설명문

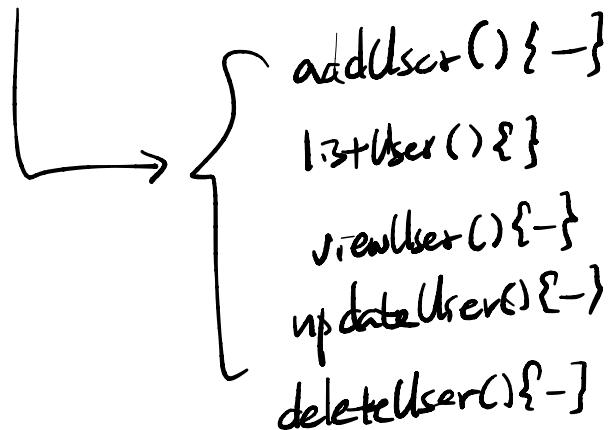


## 10. CRUD 툴 만들기 (2/3)

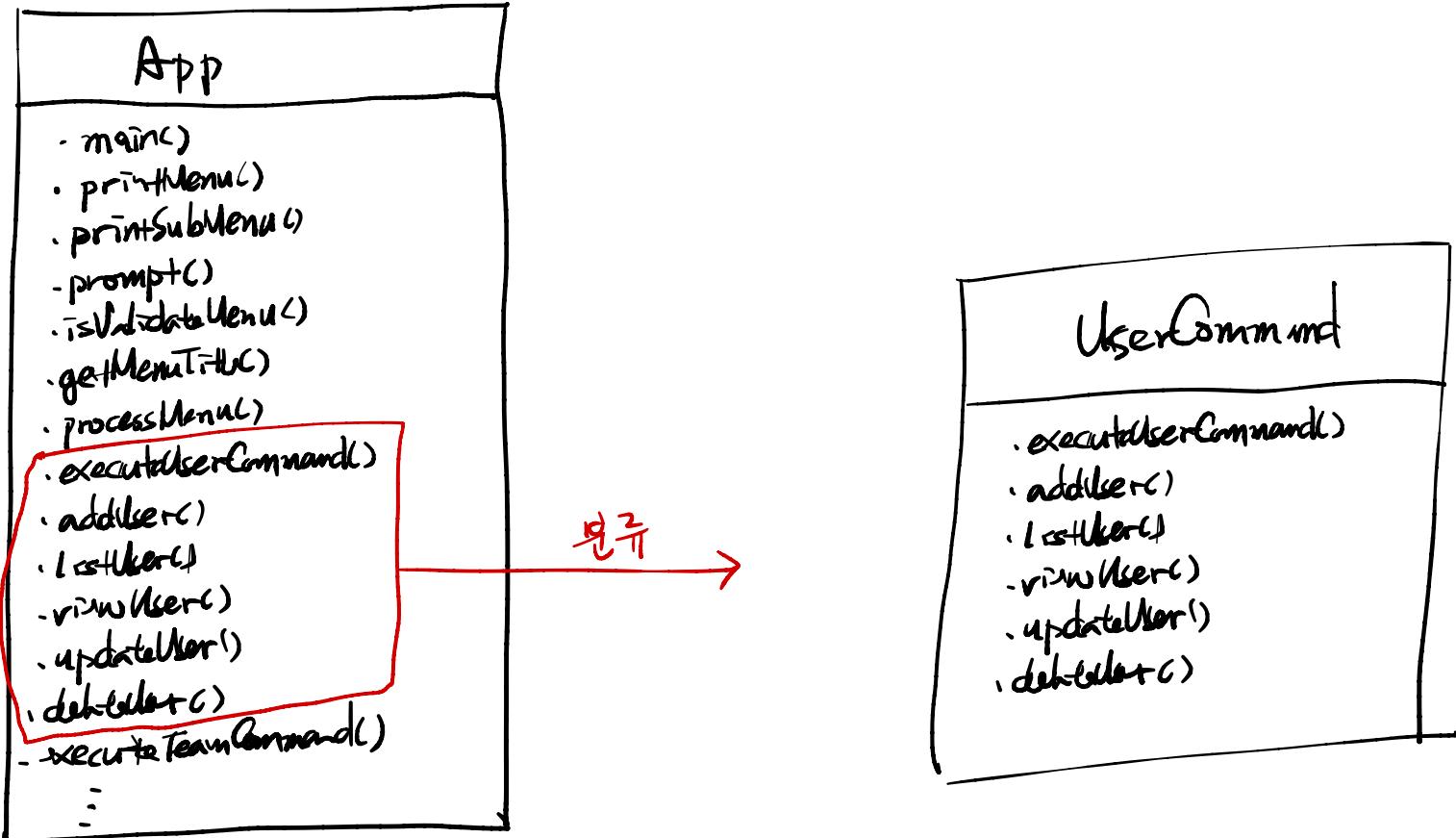


## 10. CRUD 주제(1) (2/3)

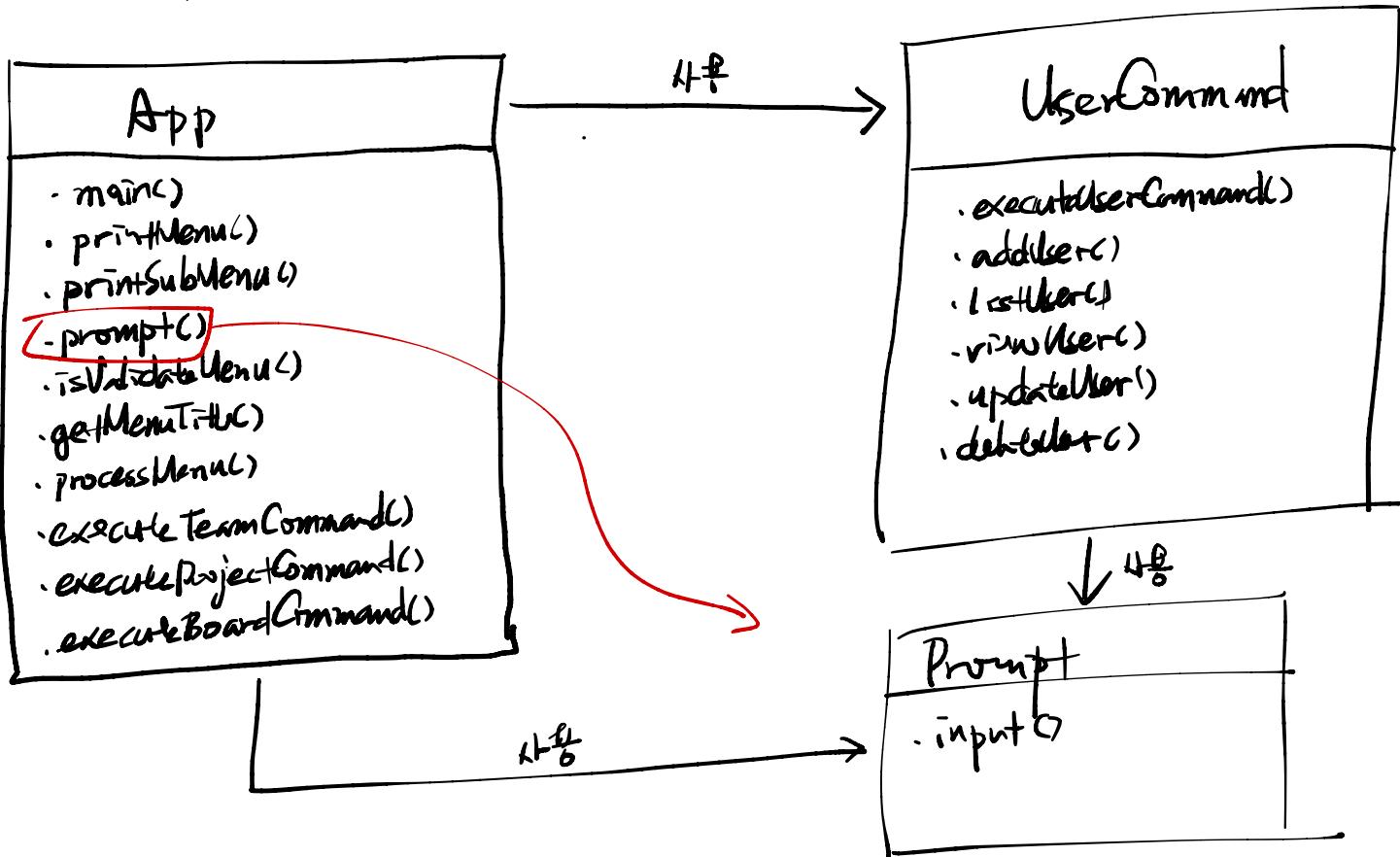
executeUserCommand(String command) { - }



## 10. CRUD 구현하기 (2/3)

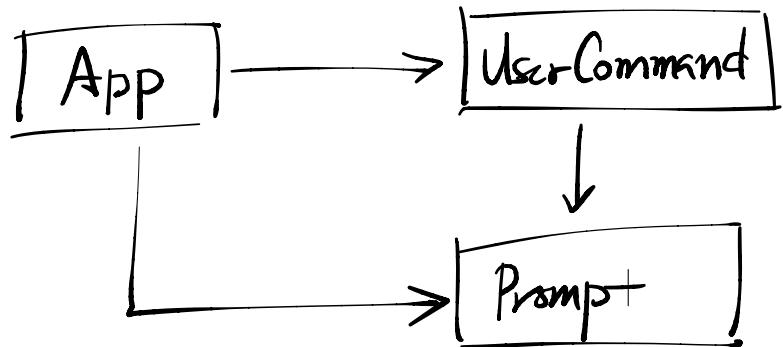


## 10. CRUD 구현하기 (2/3)



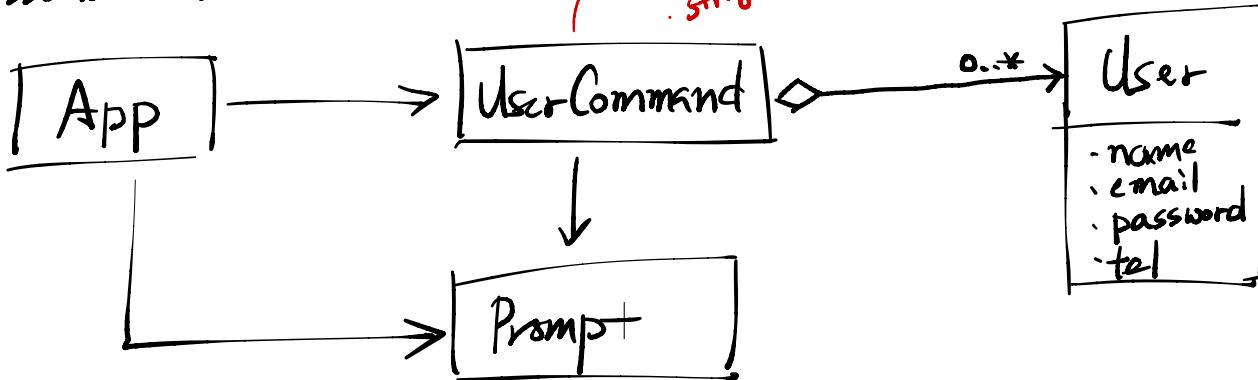
## 10. CRUD 구현하기 (2/3)

\* Association (연관)



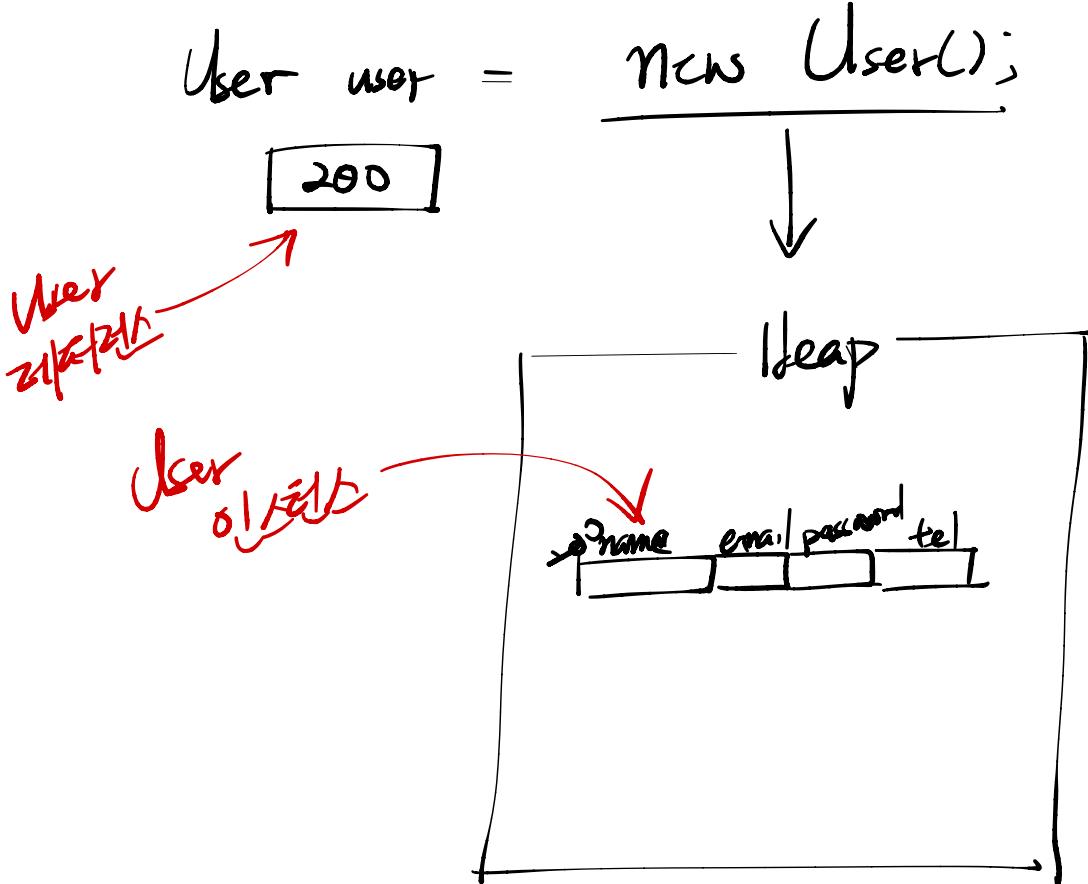
## 10. CRUD 구현하기 (2/3)

### \* Association (연관)



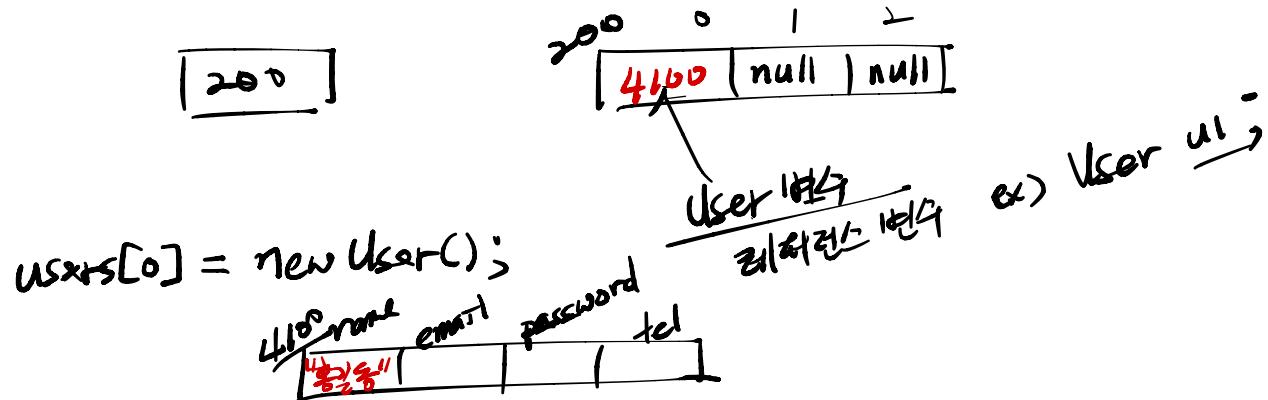
## \* 인스턴스 풀드

```
class User {  
    String name;  
    String email;  
    String password;  
    String tel;  
}
```



\* 주의점

User[] users = new User[3];



user[0].name = "홍길동";

user[1].name = "임꺽정";

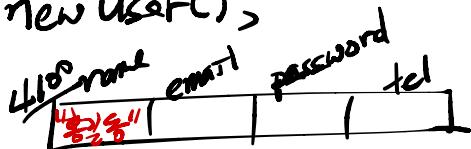
NullPointerException  $\xrightarrow{\text{null}} \text{발생!}$

\* 주소리스트 초기화

User[] users = new User[3];



users[0] = new User();



User user = users[0];



users[0].name = "aaa";  
user.name = "aaa";

## \* 흔한 문법

