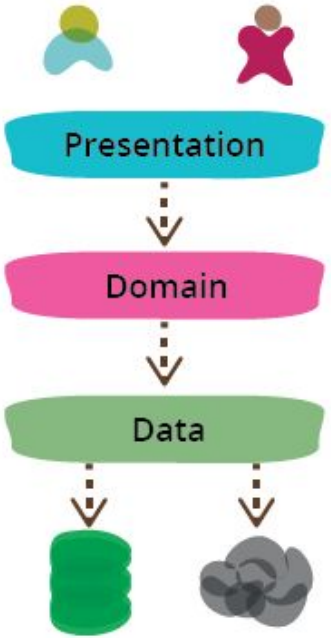
3 w.

App Architecture - MVP

Ivy

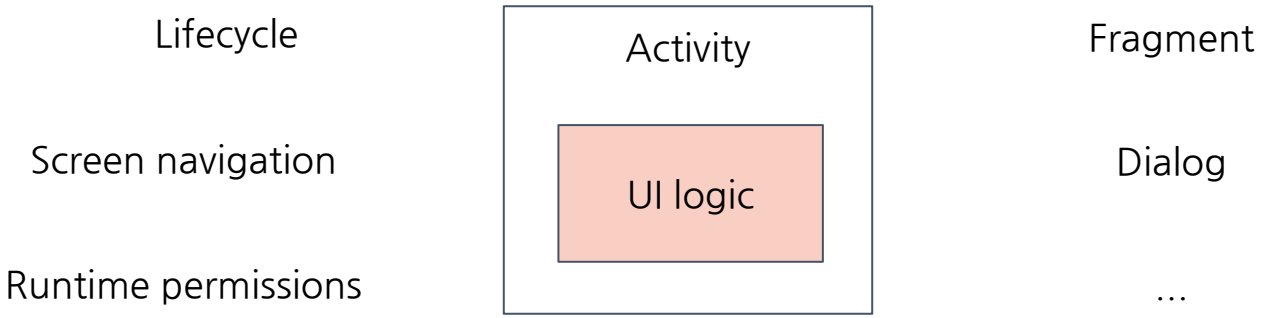
Separation of Concerns

Layered Architecture



- 초기에는 Layer의 경계가 곧 물리적인 경계를 의미했습니다.
- Layer는 관심사에 의해서도 분리됩니다.
- 각 Layer는 역할과 책임이 다릅니다.
- 다른 Layer가 하는 일에는 관심이 없습니다.

UI 구현을 위해 Activity / Fragment 가 처리하던 일들



UI logic을 Activity / Fragment 외부로 분리시킬 방법이 있을까?

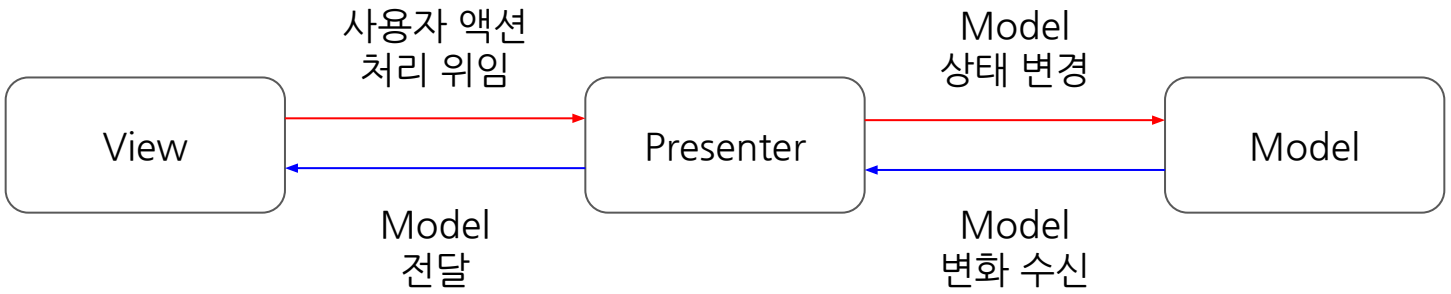
생각해보기

왜 UI(presentation)를 분리하려고 할까?

- 관심사에 따라 클래스를 분리하는 연습을 하면서 이유 생각해보기

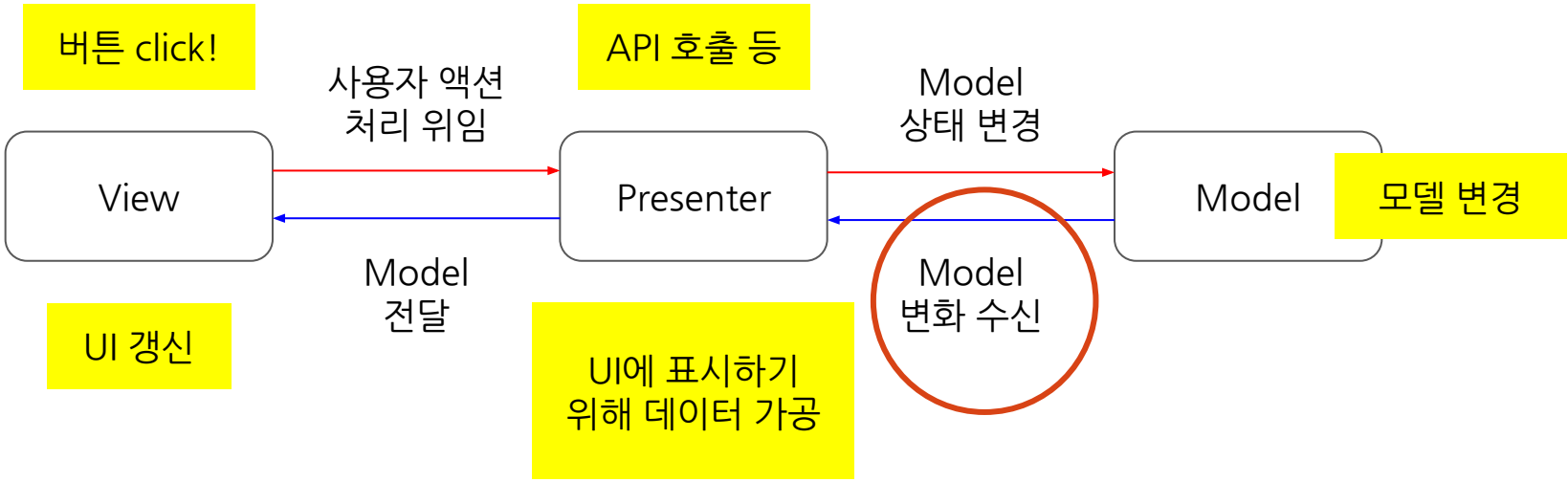
MVP pattern

Model - View - Presenter



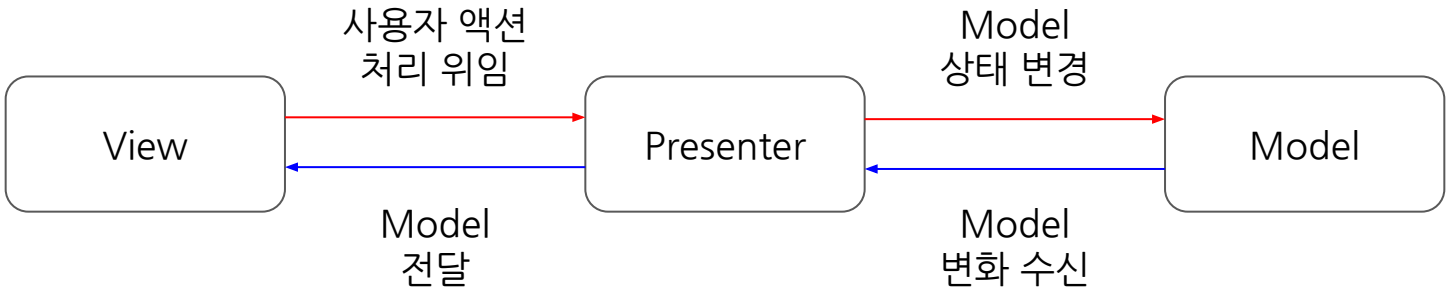
- Model : data layer. 데이터 원형과 그 상태를 관리
- View : UI 표시, 사용자 액션 감지 (Activity, Fragment)
- Presenter : Model, View를 호출하여 UI를 표시하기 위해 필요한 작업 실행

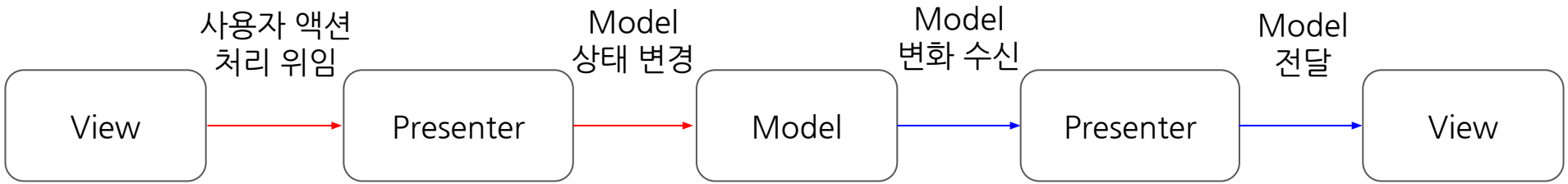
Model - View - Presenter



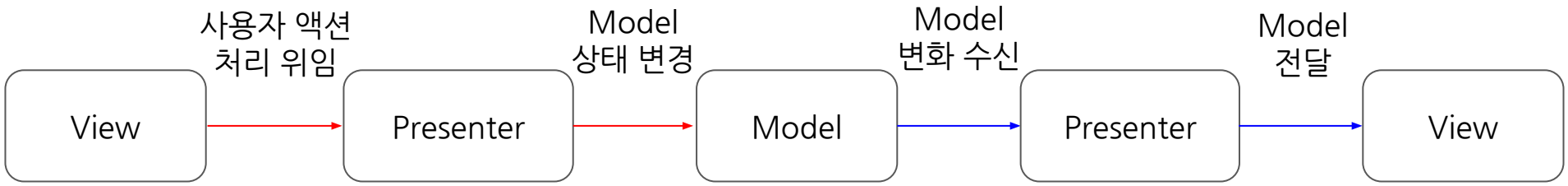
Presenter : Model, View를 호출하여 UI를 표시하기 위해 필요한 작업 실행

Model - View - Presenter





Model - View - Presenter

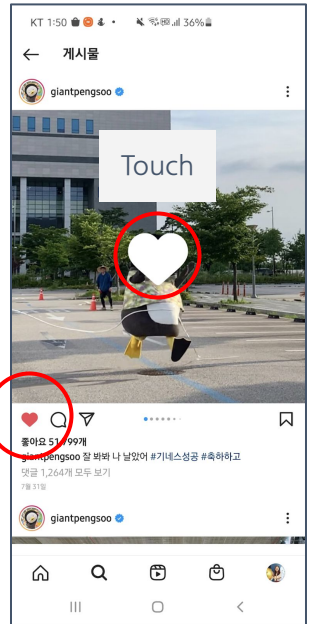


펼쳐서 생각하는 것이 더 도움된다.

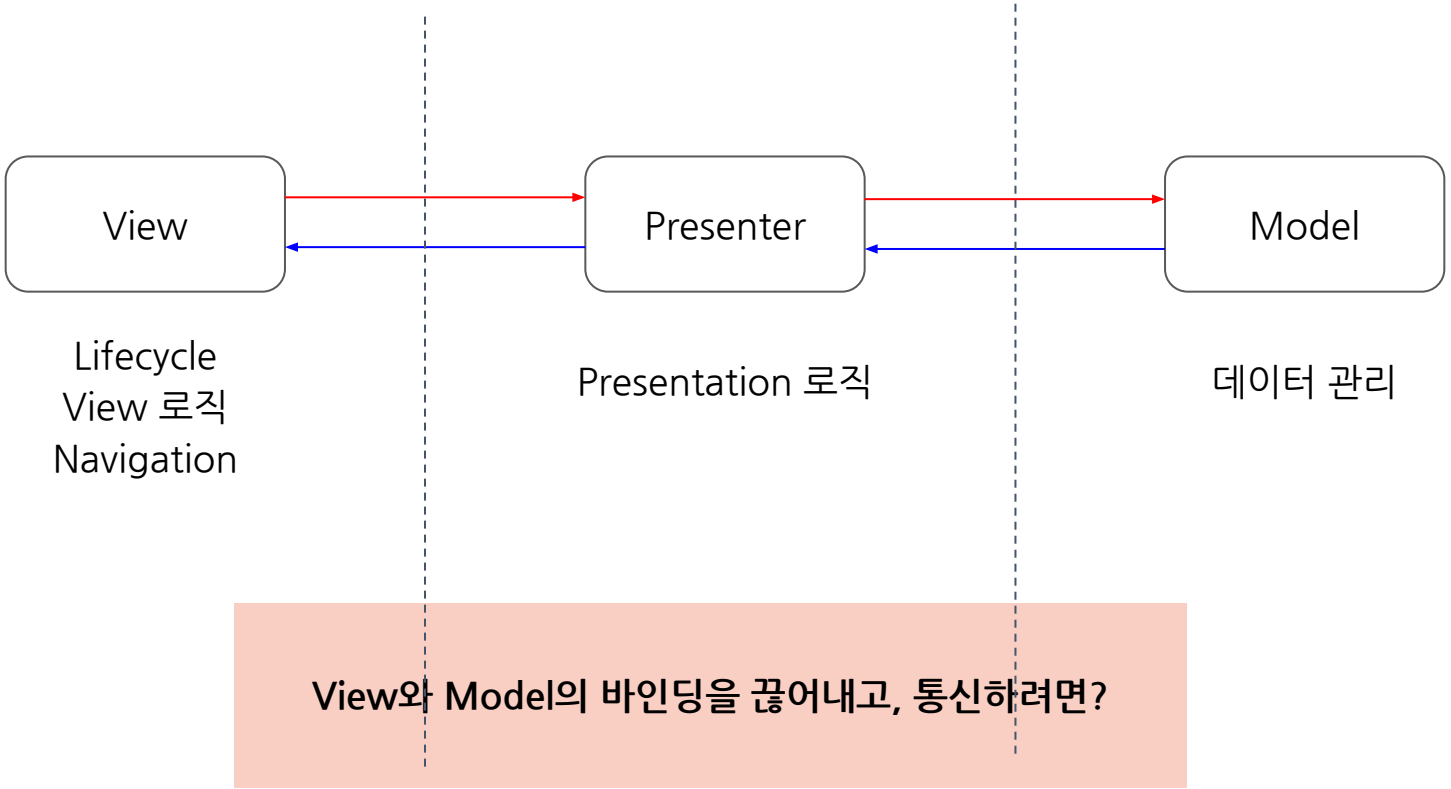
많이 하는 실수

1. 사용자 액션에 대한 처리를 할 때는 Model의 상태만 변경하는 것이다. 바로 마지막의 View를 바꾸는 것이 아니다!
2. 처음의 View와 마지막 View는 다를 수 있다!

1. 사용자 액션 : 사진 터치
2. Model 상태 변경
   - 터치되었음
3. Model 변경 알림
4. Model 변경 수신
5. View 업데이트



Model - View - Presenter

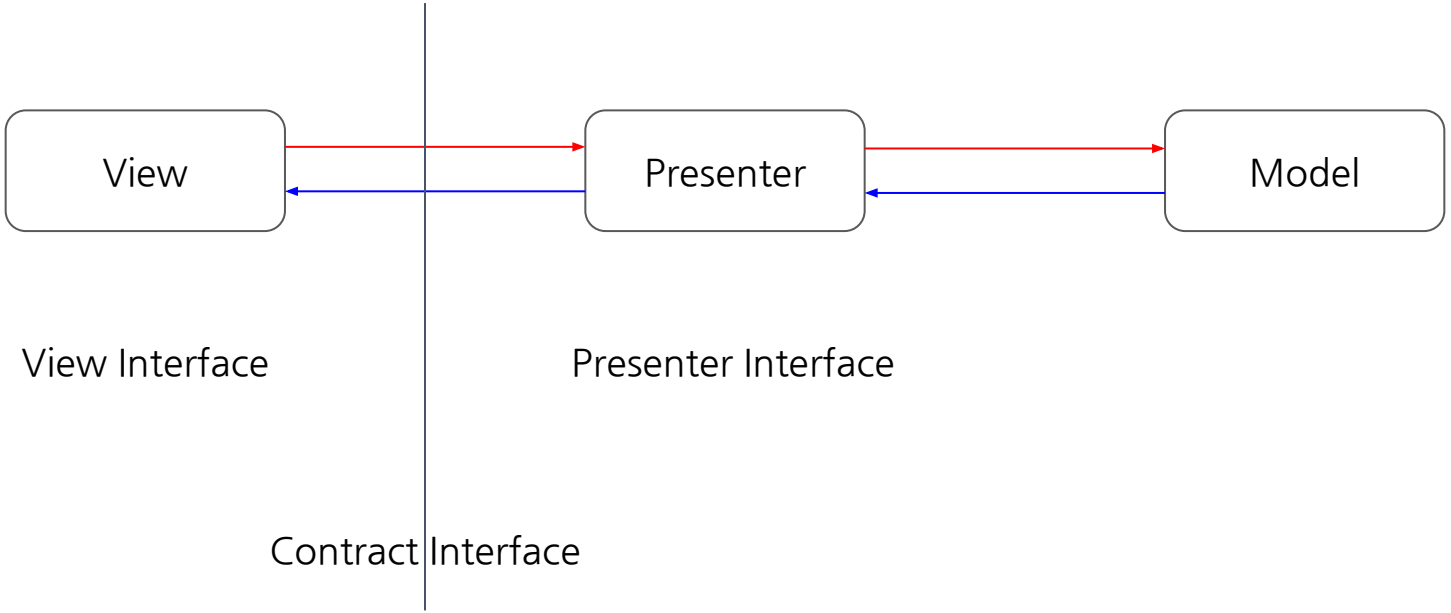


연습 1. 검색 버튼 클릭 후, 결과를 목록에 그리기

실습 : 검색 결과 요청

Step 1. Contract

- 왜 가장 먼저 작성할까?
- 어떤 내용이 포함될 수 있을까?



Step 1. Contract

```
interface SearchContract {

    interface SearchView {

    }

    interface SearchPresenter {

    }
}
```

사용자가 검색어를 입력하는 이벤트를 처리하기 위해 수행해야 하는 작업들

- 검색어 입력
- 검색 요청
- 검색 결과
- UI 갱신
- …

Step 1. Contract

- 각 객체의 역할은?
- 역할을 수행하기 위해 필요한 객체가 있나요?

Step 2. Presenter

- Presenter는 Model을 알아도 되는 걸까요?
- Model의 업데이트된 상태는 어떻게 알 수 있을까요?

```
// remoteDataSource.
fun getSearchResults(query: String): LiveData<SearchResult> ...
```

Step 3. Model

- Model은 Presenter를 알아도 되나요?
- Model의 역할은?
- Model의 업데이트된 상태는 어떻게 알 수 있을까요?

Reference

- Google / todo-mvp-kotlin