Android 프로그래밍 Pattern: MVVM





- Step 1
 - ▶ Gradle Scripts -> build.gradle.kts(Module :app) 파일의
 - ▶ android 블록 내에
 - ▶ buildFeatures { databinding = true } 입력하고
 - > Sync now 링크를 클릭해서 변경 내용 적용





- Step 2
 - ▶ res > layout 에서
 - ▶ activity_main.xml 파일의 root layout에서
 - ▶ 마우스 오른쪽 버튼 클릭 / 표시되는 메뉴에서
 - "Convert to data binding layout" 선택





- Step 3
 - ▶ 기본 패키지 경로에서
 - ▶ Model 클래스 생성
 - ▶ Model 클래스는 앱에서 관리하는 데이터와 데이터 조작을 위한 메서드 선언/정의





- Step 4
 - ▶ ViewModel 클래스 생성
 - ▶ 클래스 이름: MainViewModel
 - ▶ 클래스 코드:

```
public class MainViewModel extends ViewModel {

private Model model = new Model();
private MutableLiveData<Integer> count = new MutableLiveData<>();
private MutableLiveData<Integer> totalPrice = new MutableLiveData<>();

// 생성자
public MainViewModel() {
    updateValues();
}
```





```
// 두 가지 값에 대한 accessor
public LiveData<Integer> getCount() { return count; }
public LiveData<Integer> getTotalPrice() { return totalPrice; }
public void add() {
  model.add();
  updateValues();
public void subtract() {
  model.subtract();
  updateValues();
private void updateValues() {
  count.setValue(model.getCount());
  totalPrice.setValue(model.getTotalPrice());
```





- Step 5
 - ▶ layout 파일(active_main.xml) 작성 layout design
 - ▶ <data> </data> 블록에서
 - ▶ MainViewModel 사용





- ▶ layout에서
 - ▶ 수량을 나타내는 textView의 text 속성 값 변경 "@{String.valueOf(viewModel.count)}"
 - ▶ 금액을 나타내는 textView의 text 속성 값 변경
 "@{NumberFormat.getNumberInstance(
 Locale.getDefault()).format(viewModel.totalPrice) + '원'}"/>





- Step 6
 - ▶ MainActivity에서
 - ▶ Activity와 Layout을 연결하는 변수를 클래스 레벨에서 선언
 - ▶ 예) private ActivityMainBinding binding; // activity_main.xml 파일과 activity 연결
 - ▶ Activity와 ViewModel을 연결하는 변수를 클래스 레벨에서 선언
 - ▶ 예) private MainViewModel viewModel
 - ▶ OnCreate 메서드에서 위의 두 가지 변수 초기화
 - ▶ 예) binding = ActivityMainBinding.inflate(getLayoutInflater();
 - setContentView(binding.getRoot());
 - ▶ 예) binding = DataBindingUtil.setContentView(this, R.layout.activity_main);
 - ▶ 예) viewModel = new ViewModelProvider(this).get(MainViewModel.class);





- ▶ binding 변수와 viewModel 연결
 - ▶ 예) binding.setViewModel(viewModel);
 - binding.setLifeCycleOwner(this);
- ▶ 레이아웃에 디자인된 버튼에 대한 이벤트 리스너 설정
 - **)** binding.btnDecrease.setOnClickListener(v -> viewModel.subtract());
 - binding.btnIncrease.setOnClickListener(v -> viewModel.add());