진동, 효과음 울리기
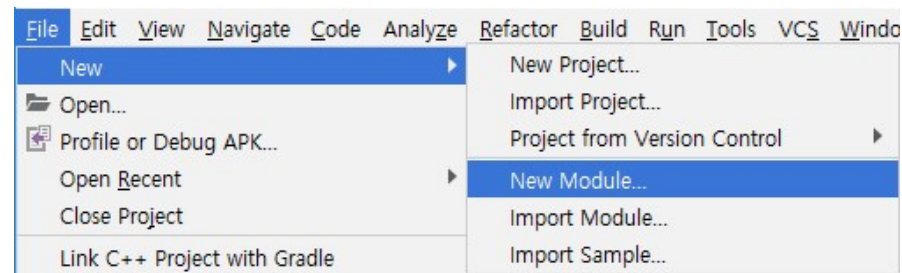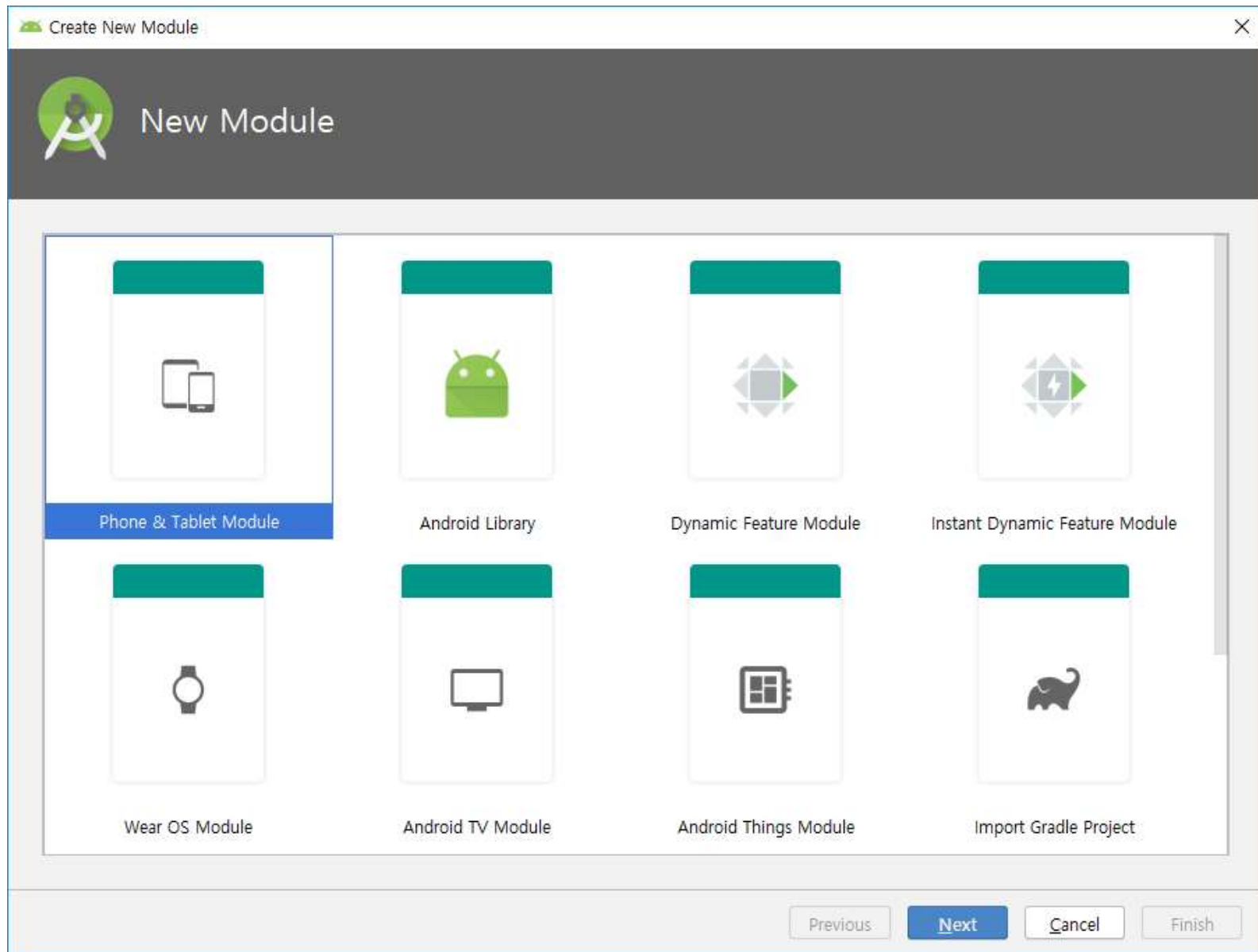
Step 1 _ 모듈 생성

Create New Module   ✕

New Module

Phone & Tablet Module | Android Library | Dynamic Feature Module | Instant Dynamic Feature Module

Wear OS Module | Android TV Module | Android Things Module | Import Gradle Project

Previous   **Next**   Cancel   Finish

Create New Module ✕

Phone & Tablet Module

Configure the new module

Application/Library name

Lab03

Module name

lab03

Package name

com.example.user.lab03   Done

Minimum SDK

API 15: Android 4.0.3 (IceCreamSandwich) ▼

Previous   Next   Cancel   Finish

Create New Module

Add an Activity to Mobile

Add No Activity

Basic Activity

Bottom Navigation Activity

Empty Activity

Previous   Next   Cancel   Finish
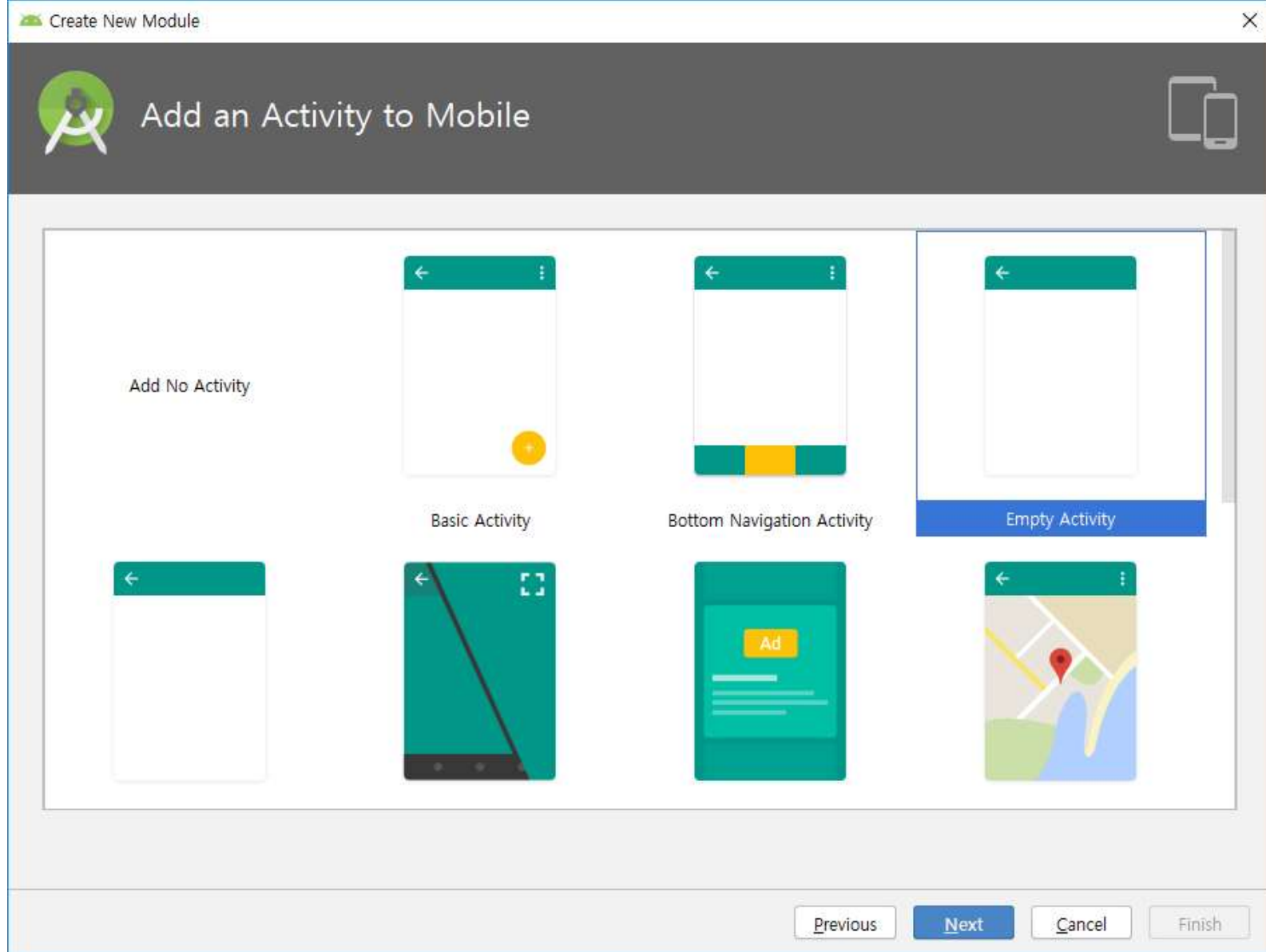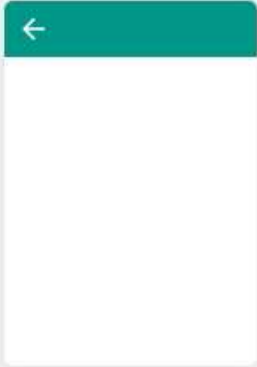
Create New Module   ✕

Configure Activity

Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

Source Language: Java   ▼
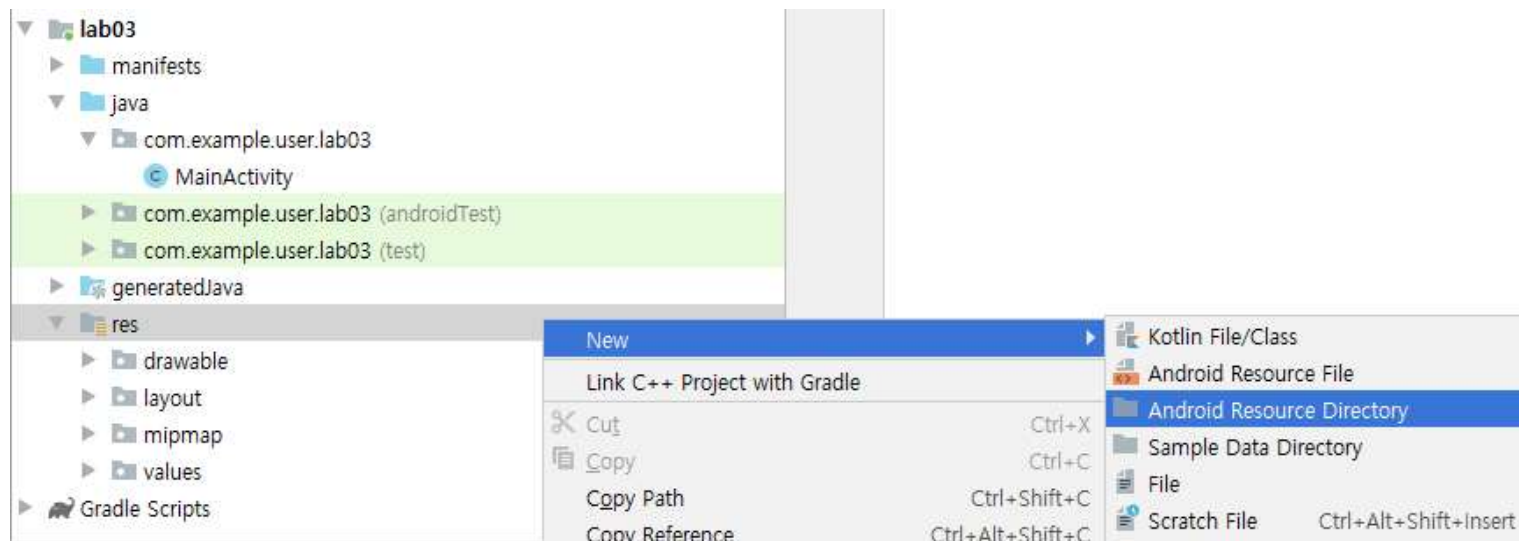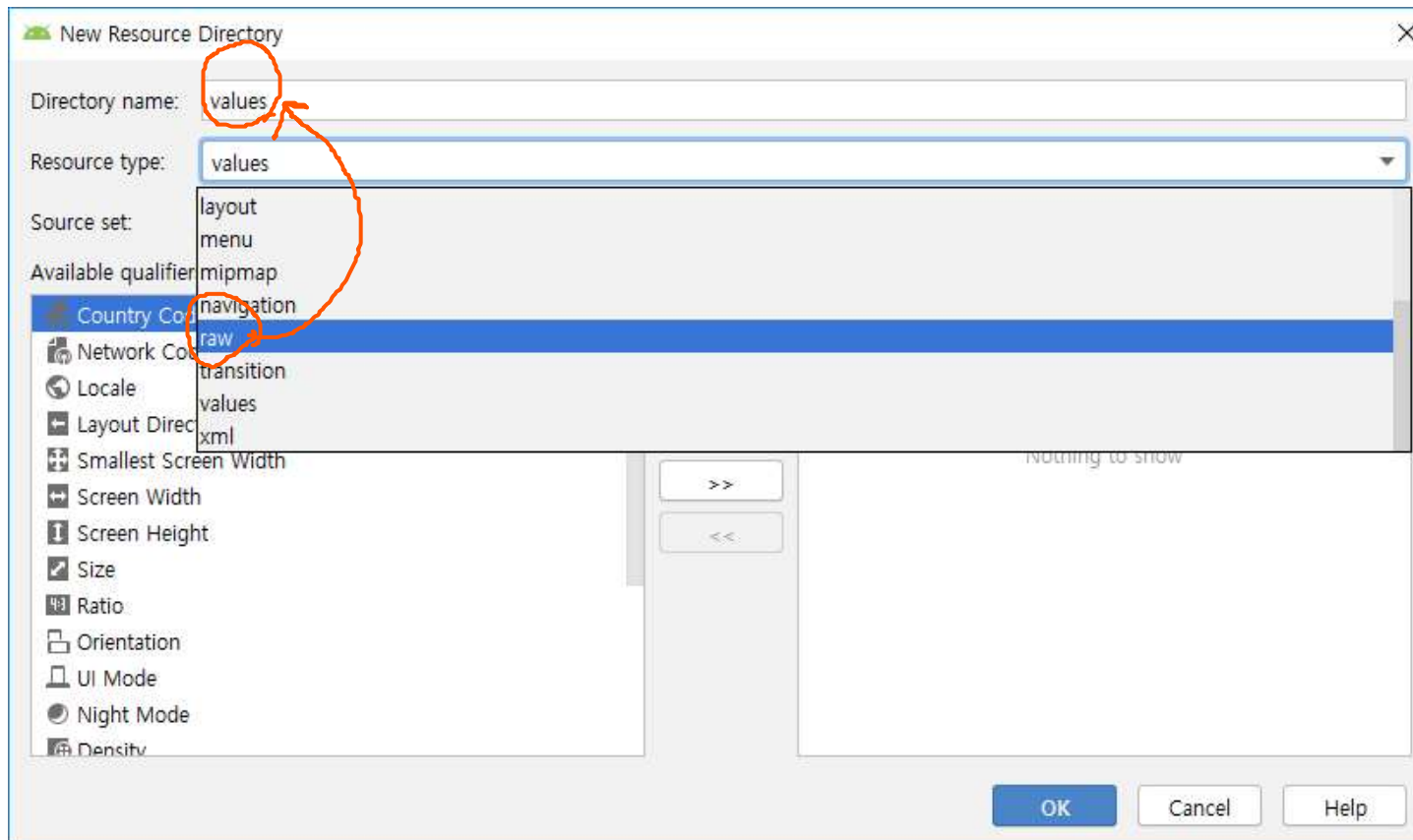
The name of the activity class to create

Previous   Next   Cancel   **Finish**

Step 2 _ raw 폴더 생성

New Resource Directory   ✕

Directory name: values

Resource type: values   ▼

Source set: layout
                    menu
Available qualifier mipmap
                    navigation
   Country Co   raw
   Network Co   transition
   Locale   values
   Layout Direc   xml
   Smallest Screen Width
   Screen Width   >>   Nothing to show
   Screen Height   <<
   Size
   Ratio
   Orientation
   UI Mode
   Night Mode
   Density

                                      OK   Cancel   Help

Step 3 _ 음원 파일 복사

Step 4 _ 퍼미션 설정

진동기능 허락받음.

```
lab03
  manifests
    AndroidManifest.xml
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.user.lab03">

    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```
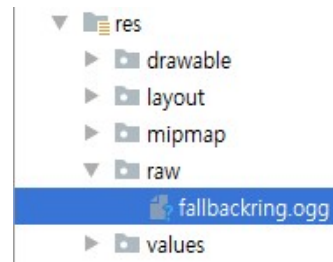
System, user
에게 권한을 받음.

진동권한

.apk 설치시
처음 여는 파일

컴포넌트 종류

Step 5 _ activity_main.xml 파일 작성

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/btn_vibration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="vibration" />

    <Button
        android:id="@+id/btn_system_beep"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="system beep" />

    <Button
        android:id="@+id/btn_custom_sound"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="custom sound" />

</LinearLayout>
```
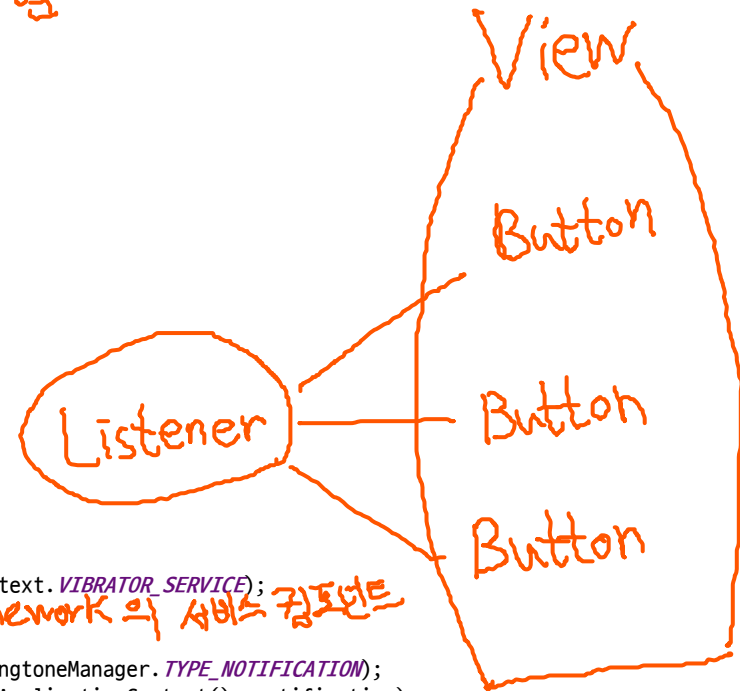
Step 6 _ MainActivity 작성

```java
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button vibrationBtn;
    Button systemBeepBtn;
    Button customBeepBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        vibrationBtn = findViewById(R.id.btn_vibration);
        systemBeepBtn = findViewById(R.id.btn_system_beep);
        customBeepBtn = findViewById(R.id.btn_custom_sound);

        vibrationBtn.setOnClickListener(this);
        systemBeepBtn.setOnClickListener(this);
        customBeepBtn.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        if (view == vibrationBtn) {
            Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
            vibrator.vibrate(1000);
        } else if (view == systemBeepBtn) {
            Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
            Ringtone ringtone = RingtoneManager.getRingtone(getApplicationContext(), notification);
            ringtone.play();
        } else if (view == customBeepBtn) {
            MediaPlayer player = MediaPlayer.create(this, R.raw.fallbackring);
            player.start();
        }
    }
}
```

Handwritten annotations:
- Listener 직접 구현
- 1조
- View
- Button
- Listener
- Button
- Button
- Framework의 서비스 컴포넌트
- URI

Vibrator

- public void vibrate (long milliseconds)
- public void vibrate (long[] pattern, int repeat)
    - pattern
        - {off, on, off, on, ...}
    - repeat
        - -1: 한 번만 패턴대로 진동
        - 0: 진동이 무한반복
            - 코드에서 적절한 시점에 cancel() 함수를 호출하여 진동 울리기를 멈추어 주어야 함

```
Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
vibrator.vibrate(new long[]{500, 1000, 500, 1000}, -1);
```

0.5초 대기 → 1초 진동 → 0.5초 대기 → 1초 진동

RingtoneManager

int	**TYPE_ALARM** Type that refers to sounds that are used for the alarm.
int	**TYPE_NOTIFICATION** Type that refers to sounds that are used for notifications.
int	**TYPE_RINGTONE** Type that refers to sounds that are used for the phone ringer.

Step 7 _ 실행

시스템 제공