
Kotlin을 이용한 Android 프로그래밍

- 비만도 계산기 앱 만들기 -

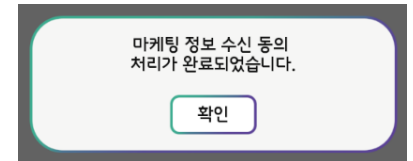
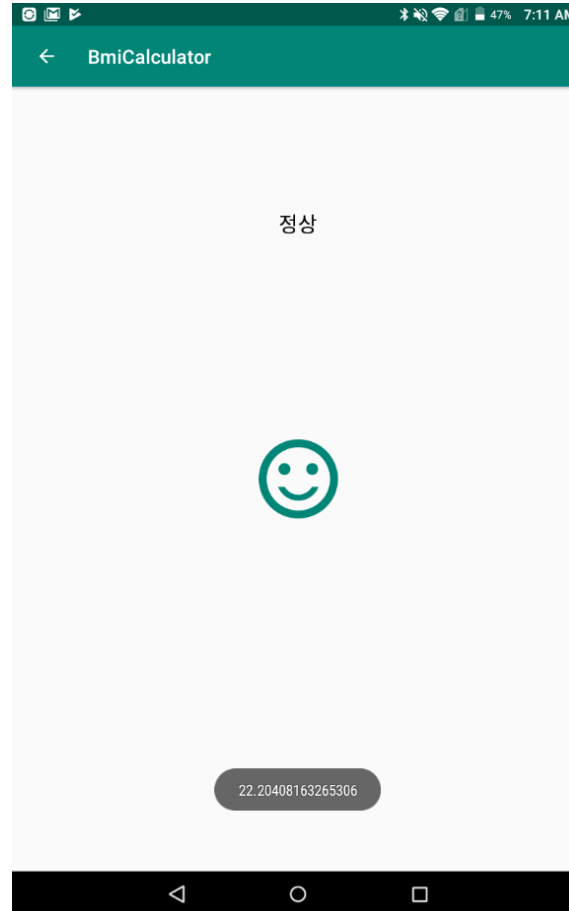
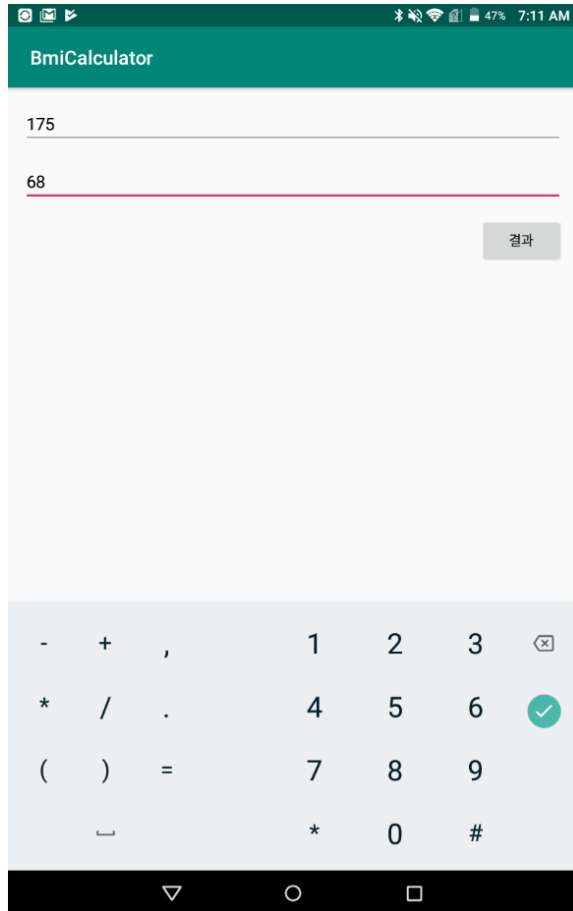
Contents

I. 비만도 계산기 앱 만들기

II. 연습문제

I. 계산기 만들기

비만도 계산기 앱 만들기



비만도 계산기 앱 만들기

▶ 프로젝트 명 : BmiCalculator

▶ 기능

▶ 키와 몸무게를 입력하고 결과 버튼을 누르면 다른 화면으로 전환되어 비만도 결과를 문자와 그림으로 보여줌

▶ 최근에 입력한 키와 몸무게는 자동으로 저장됨

▶ 구성요소

▶ ConstraintLayout : 제약에 따라 뷰를 배치하는 레이아웃

▶ Intent : 화면을 전환에 사용하고 액티비티 간에 간단한 데이터를 전달

▶ SharedPreferences : 간단한 데이터를 저장

▶ 라이브러리 설정

▶ 벡터 드로어블 하위 호환 설정 : 안드로이드 5.0 버전 미만에서 벡터 드로어블을 지원

▶ Anko 라이브러리 : 인텐트, 다이얼로그, 로그 등을 효율적으로 구현하도록 제공

비만도 계산기 앱 만들기

▶ 프로젝트 설계

▶ 두 개의 액티비티로 구성

▶ 키와 몸무게를 입력하는 화면과 비만도 결과를 보여주는 화면

▶ 인텐트에 데이터를 담아서 다른 액티비티로 전달하는 기능을 구현

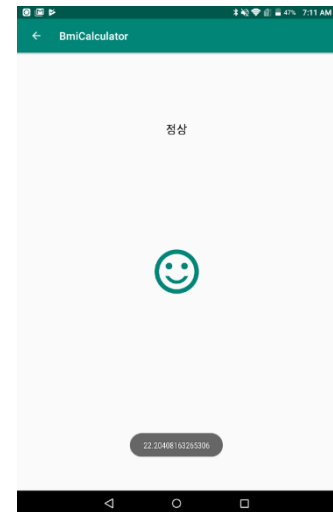
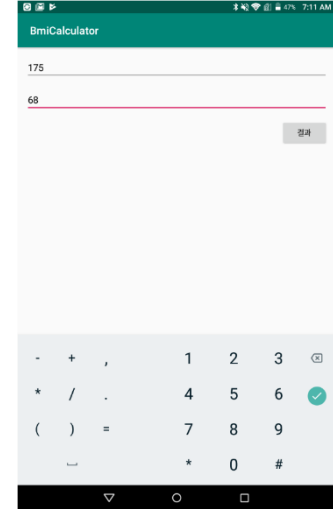
▶ 뷰와 레이아웃을 자유롭게 다룰 수 있어야 함

▶ ConstraintLayout, EditText, ImageView

▶ 벡터 이미지를 사용하므로 벡터 드로어블 하위 호환 설정

▶ Toast 사용

▶ 코드를 기존보다 간단하게 작성할 수 있도록 Anko 라이브러리를 사용



비만도 계산기 앱 만들기

▶ 프로젝트 생성

▶ 프로젝트 명 : BmiCalculator

▶ minSdkVersion : 19



▶ 기본 액티비티 : Empty Activity

A screenshot of the 'New Activity' dialog in Android Studio. The dialog has a preview on the left showing a green header bar with a white back arrow and a white body. On the right, there are several input fields and a dropdown menu. The 'Name' field contains 'BmiCalculator'. The 'Package name' field contains 'kr.ac.kpu.bmicalculator'. The 'Save location' field contains 'C:\Users\Wchoe\AndroidStudioProjects\BmiCalculator'. The 'Language' dropdown menu is set to 'Kotlin'. The 'Minimum API level' dropdown menu is set to 'API 19: Android 4.4 (KitKat)'. The 'Empty Activity' button at the bottom left is highlighted with a red box.

비만도 계산기 앱 만들기

▶ Anko 라이브러리

▶ 코틀린과 안드로이드 스튜디오를 개발한 JetBrains에서 제공하는 라이브러리

▶ Anko 라이브러리 구성

▷ Anko Commons

- 인텐트(Intent), 다이얼로그(Dialog), 로그(Log) 등을 편리하게 사용하는 라이브러리

▷ Anko Layouts

- 안드로이드 레이아웃을 코드로 쉽게 작성하는 라이브러리
 - 레이아웃을 XML이 아닌 코틀린 파일로 작성할 때 사용

▷ Anko SQLite

- SQLite를 쉽게 사용하는 라이브러리

▷ Anko Coroutines

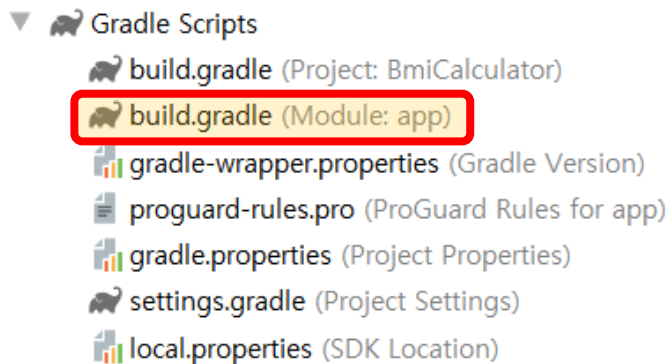
- 코루틴을 쉽게 사용하는 라이브러리

▶ Anko 라이브러리 설정 방법 참고 : <https://github.com/Kotlin/anko>

비만도 계산기 앱 만들기

▶ Anko 라이브러리 사용 준비

- ▶ 안드로이드 개발 과정에서는 코드 편집과 빌드를 분리해서 수행
- ▶ 안드로이드 스튜디오는 소스코드 편집을, 그레이들(gradle)은 빌드를 담당
 - ▷ 그레이들은 SDK와 앱 버전 및 라이브러리를 관리
- ▶ 프로젝트 수준의 그레이들과 모듈 수준의 그레이들로 구분
 - ▷ build.gradle의 괄호안에 project가 적혀있는 것이 프로젝트 수준, module이라 적혀 있는 것이 모듈 수준의 그레이들
- ▶ 프로젝트 창에서 모듈 수준의 build.gradle 파일을 더블 클릭하여 열기



비만도 계산기 앱 만들기

▶ Anko 라이브러리 사용 준비

▶ dependencies 항목에 Anko 라이브러리를 추가

▷ 이러한 과정을 라이브러리 **의존성 추가(주입)**이라고 함

▷ implementation "org.jetbrains.anko:anko-commons:\$anko_version"

■ 위 코드를 복사하여 아래의 위치에 붙여넣기

```
25 dependencies {  
26     implementation "org.jetbrains.anko:anko-commons:$anko_version"  
27  
28     implementation fileTree(dir: 'libs', include: ['*.jar'])  
29     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
30     implementation 'com.android.support:appcompat-v7:28.0.0'  
31     implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
32     testImplementation 'junit:junit:4.12'  
33     androidTestImplementation 'com.android.support.test:runner:1.0.2'  
34     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

▶ 이와 같이 dependencies 항목에 각 라이브러리를 추가하면 그레이들 빌드 시스템에서 **자동으로 라이브러리를 다운로드** 받아서 프로젝트에서 사용 가능

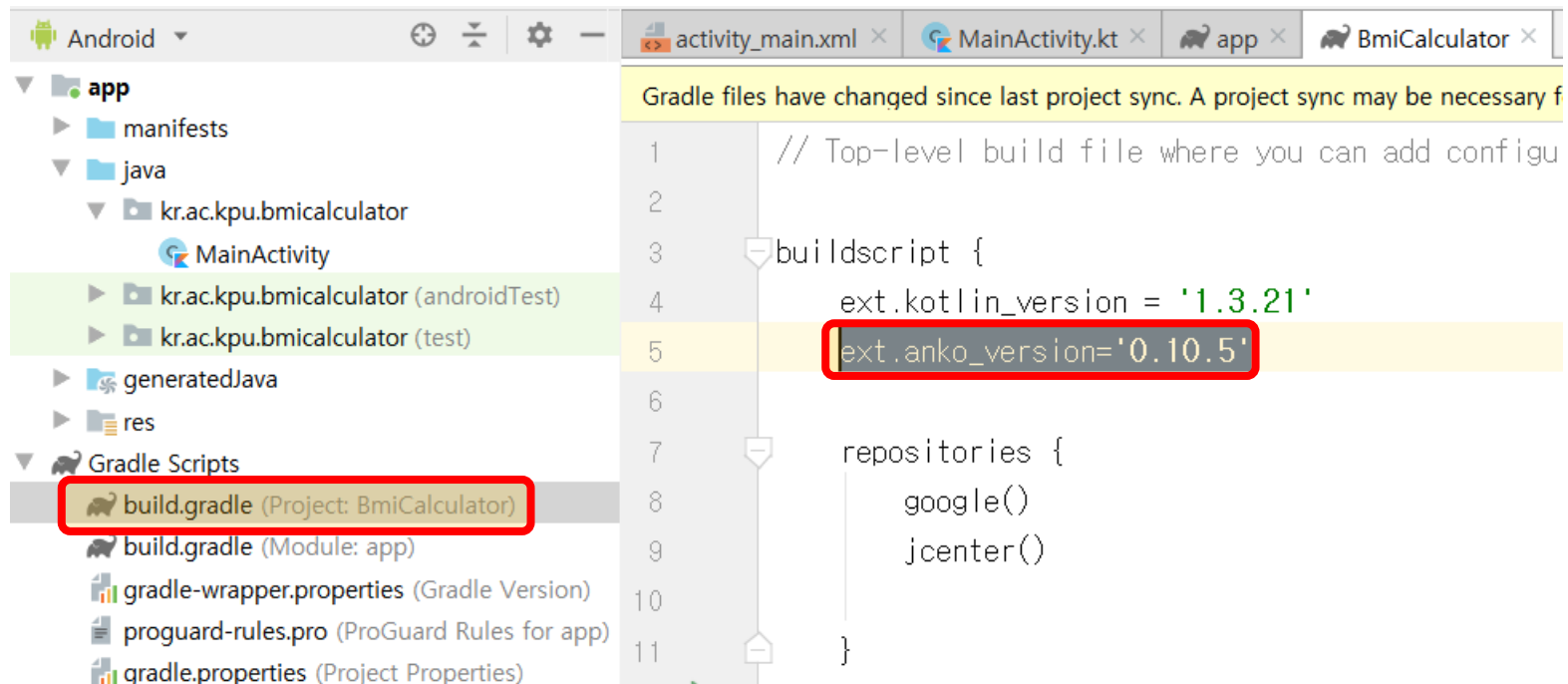
비만도 계산기 앱 만들기

▶ Anko 라이브러리 사용 준비

▶ 프로젝트 수준의 build.gradle 파일을 더블 클릭하여 열기

▶ Anko 라이브러리 버전을 변수에 지정

▶ `ext.anko_version='0.10.5'`



비만도 계산기 앱 만들기

▶ Anko 라이브러리 사용 준비

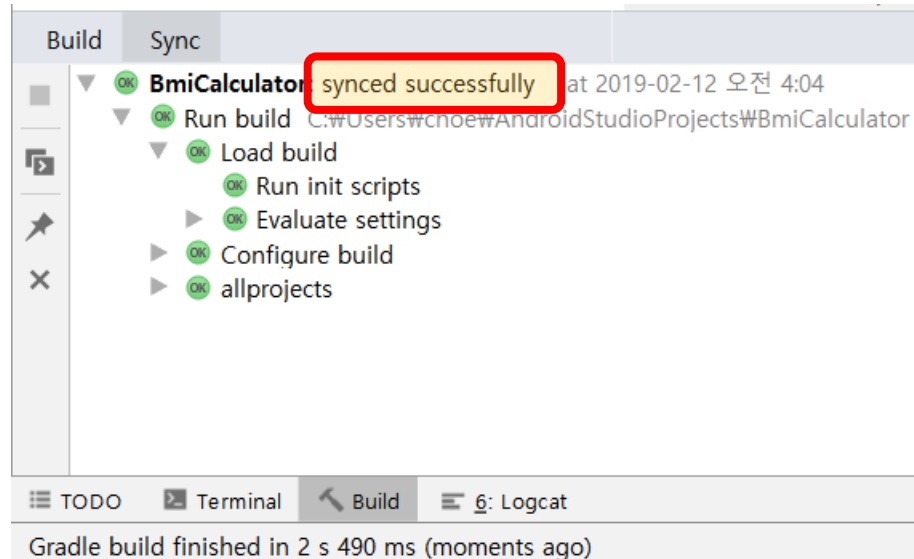
▶ 그레이들 파일을 수정하고 이를 적용하려면 싱크(동기화)가 필요

▶ Sync Now 링크를 클릭하여 진행

Gradle files have changed since last project sync. A project sync may be necessary for [Sync Now](#)

▶ 싱크를 하면 프로젝트가 다시 빌드되며 과정 중에 오류가 발생하지 않는다면 성공적으로 Anko 라이브러리 추가

▶ 앱을 개발할 경우에 외부 라이브러리를 자주 사용하므로 해당 과정을 숙지

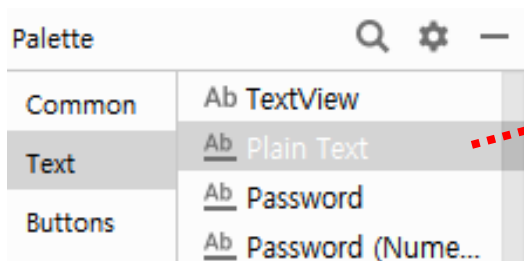


비만도 계산기 앱 만들기

▶ 키와 몸무게를 입력하는 화면 작성

▶ 키와 몸무게를 입력하는 에디트텍스트(EditText) 배치

▷ EditText == PlainText



▶ 에디트텍스트 크기와 여백 지정

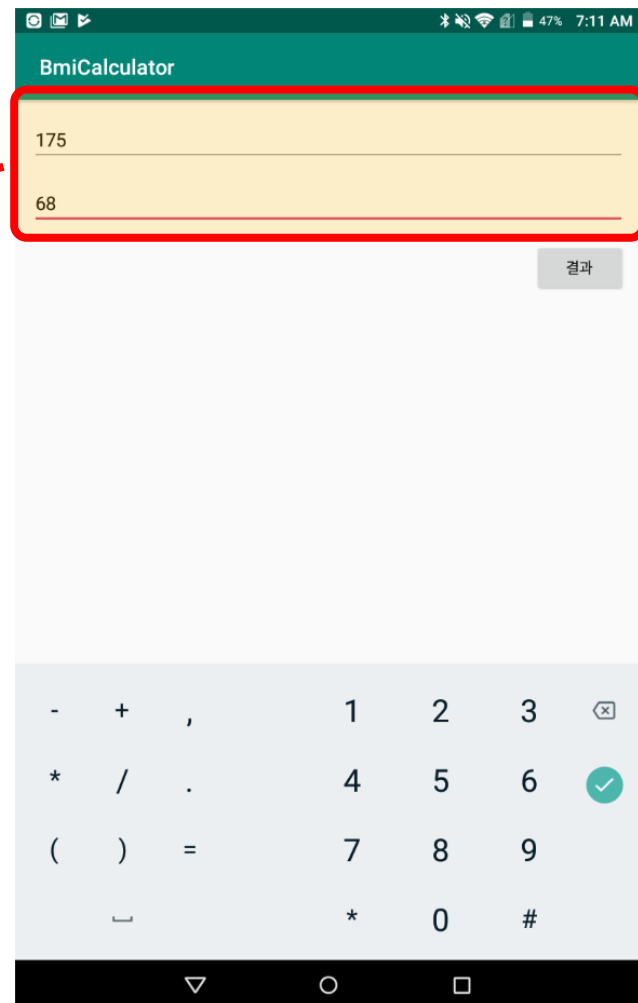
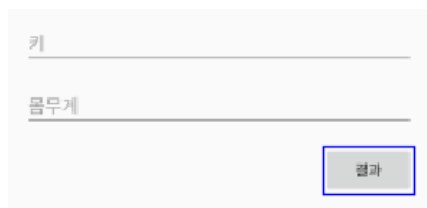
▶ 에디트텍스트에 정수만 입력 받음

▷ 간단한 앱을 작성하기 위하여 실수 입력은 제한

▶ 에디트텍스트에 기본으로 표시되는 문자열 지정

▷ hint 지정

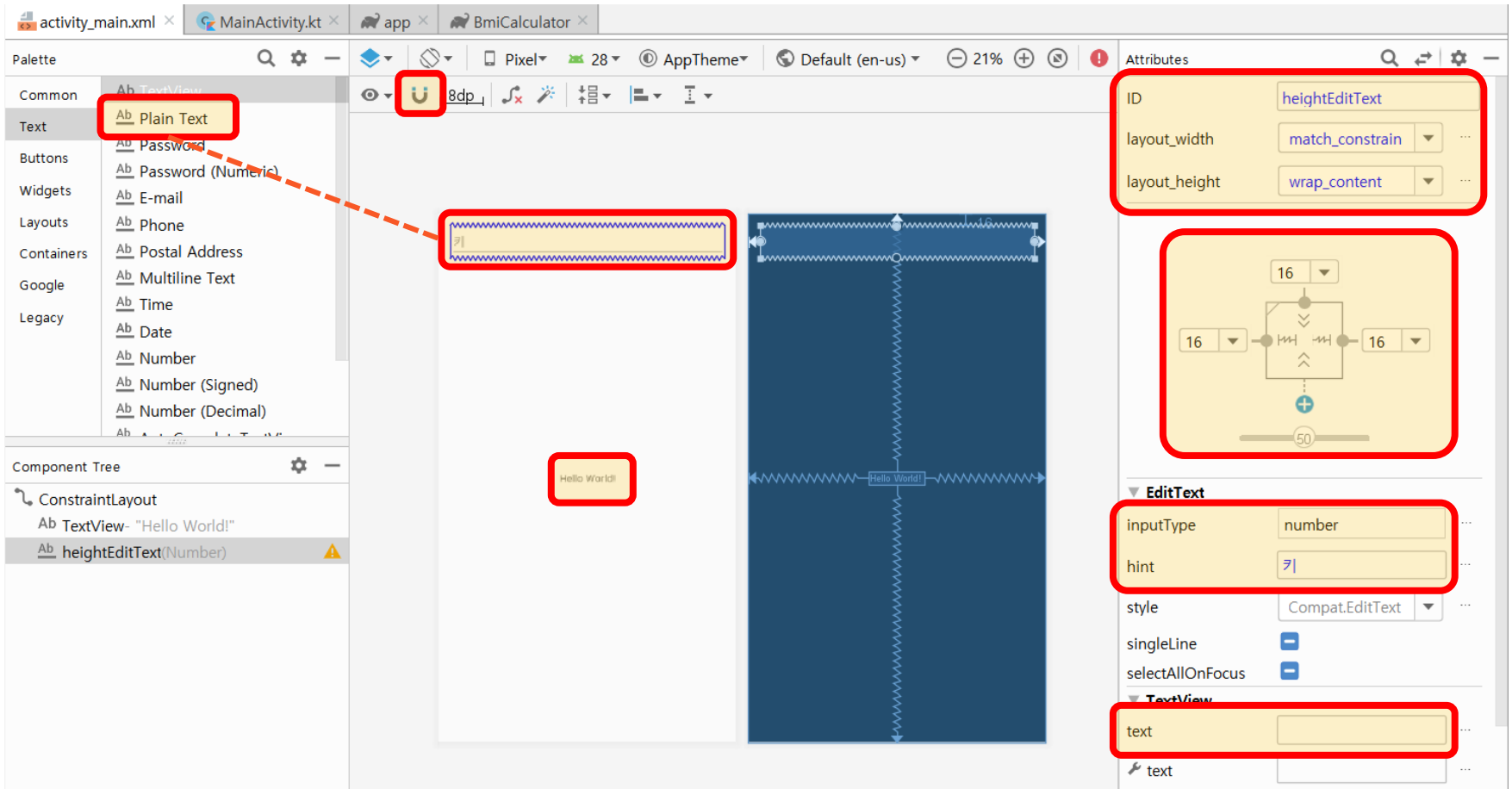
▶ 결과 버튼 배치



비만도 계산기 앱 만들기

▶ 레이아웃 작성

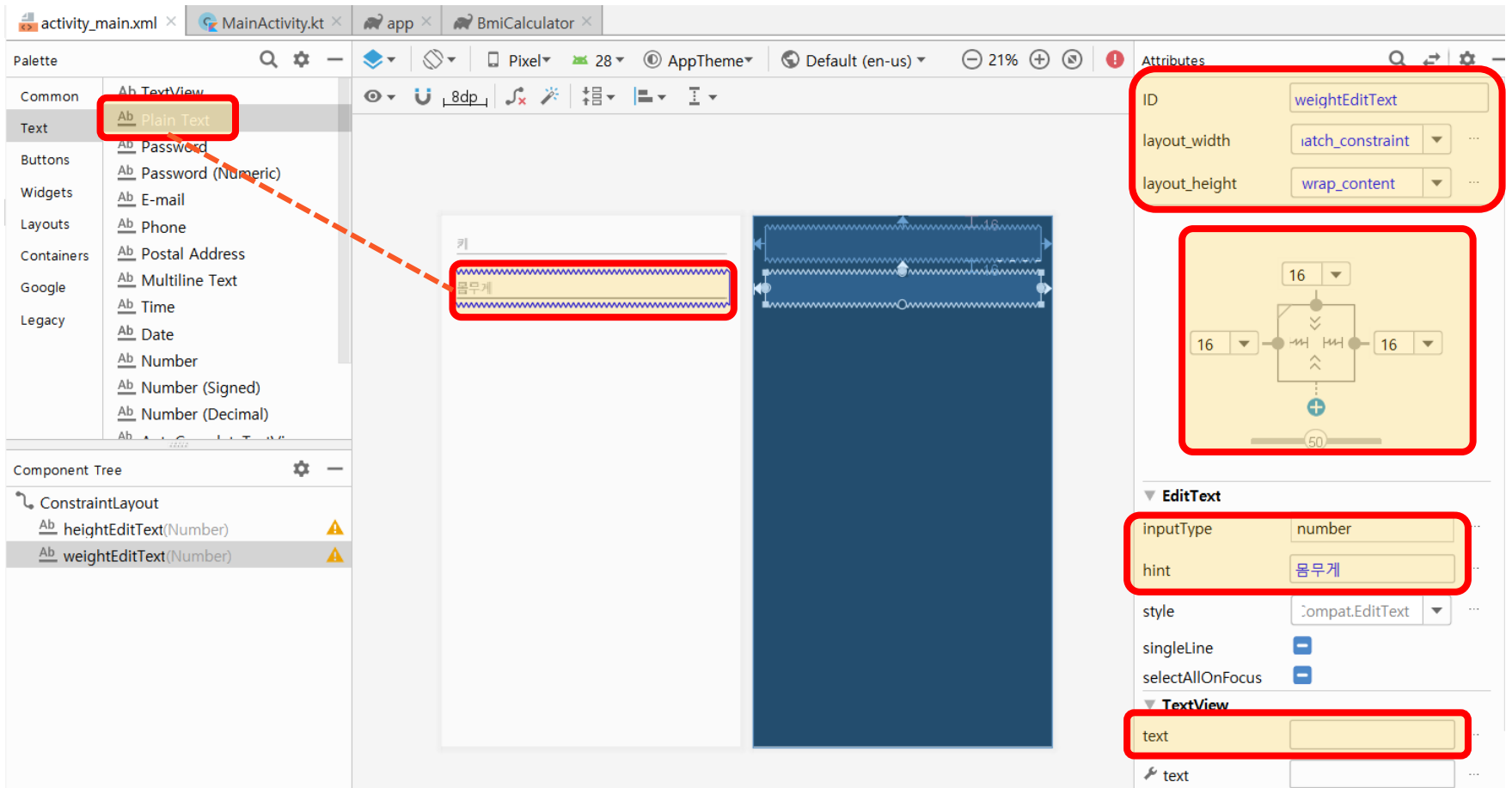
▶ 화면에 Plain Text 추가



비만도 계산기 앱 만들기

▶ 레이아웃 작성

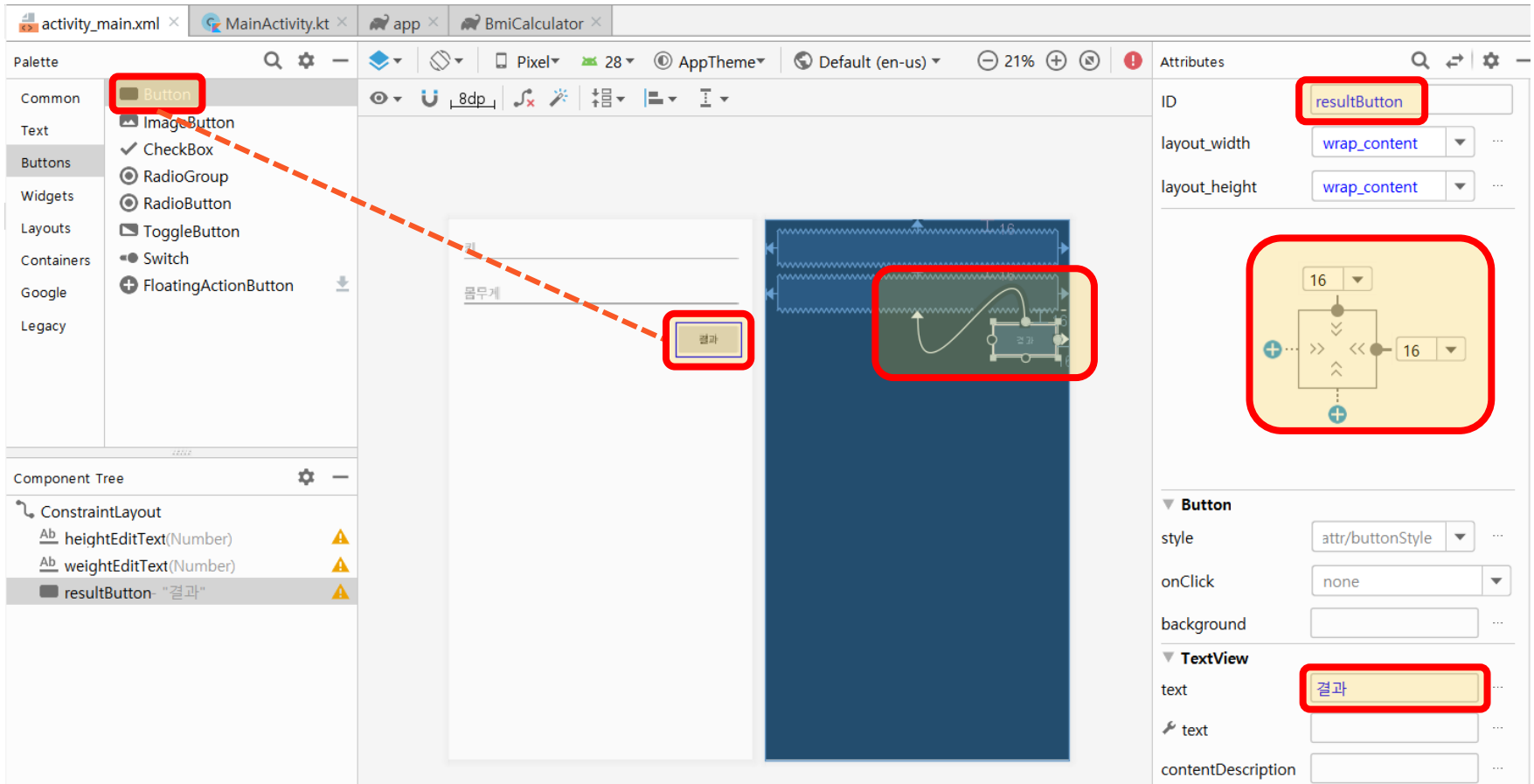
▶ 몸무게를 입력할 수 있도록 화면에 Plain Text를 하나 더 추가



비만도 계산기 앱 만들기

▶ 레이아웃 작성

▶ 결과 버튼 추가



비만도 계산기 앱 만들기

▶ 결과 화면 작성

- ▶ 두 번째 화면에는 키와 몸무게에 대한 비만도가 결정되면 결과를 표시하는 TextView와 세가지 표정 중에서 한 이미지를 표시할 ImageView를 배치

▶ 디자인 순서

▶ 새로운 액티비티 추가

- ResultActivity

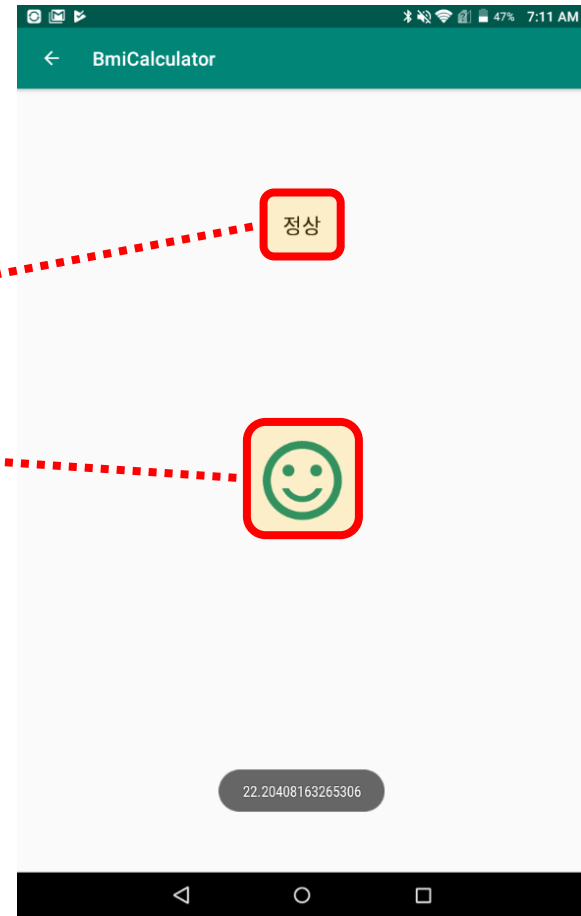
▶ 결과를 표시할 텍스트 뷰 배치

▶ 결과에 대한 이미지를 표시할 이미지 뷰 배치

- 이미지를 res에 미리 추가

▶ 벡터 드로어블 사용 환경 구성

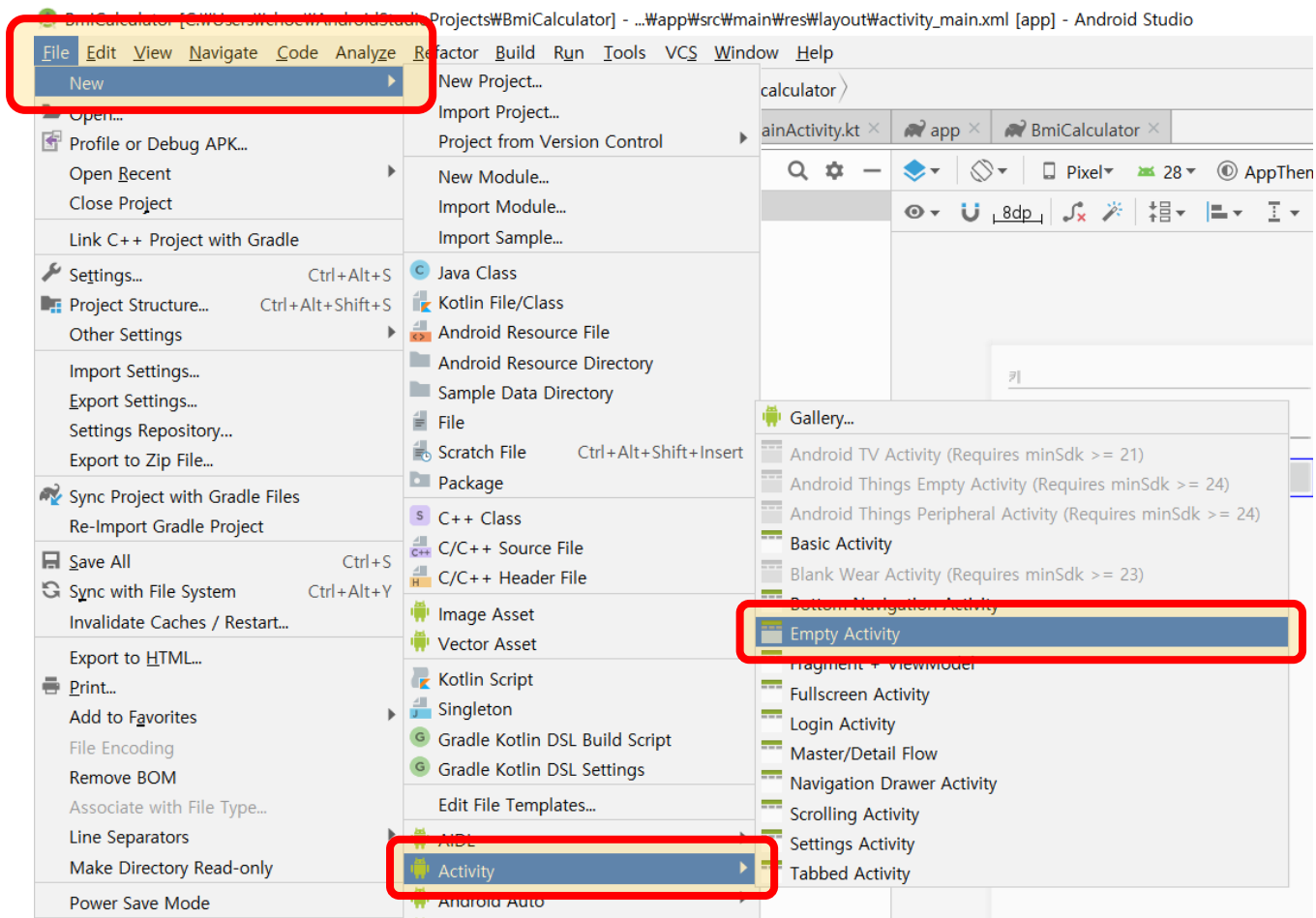
- 안드로이드 5.0미만에서는 벡터 이미지를 사용할 수 없음



비만도 계산기 앱 만들기

▶ 액티비티 추가

▶ [File] - [New] - [Activity] - [Empty Activity]

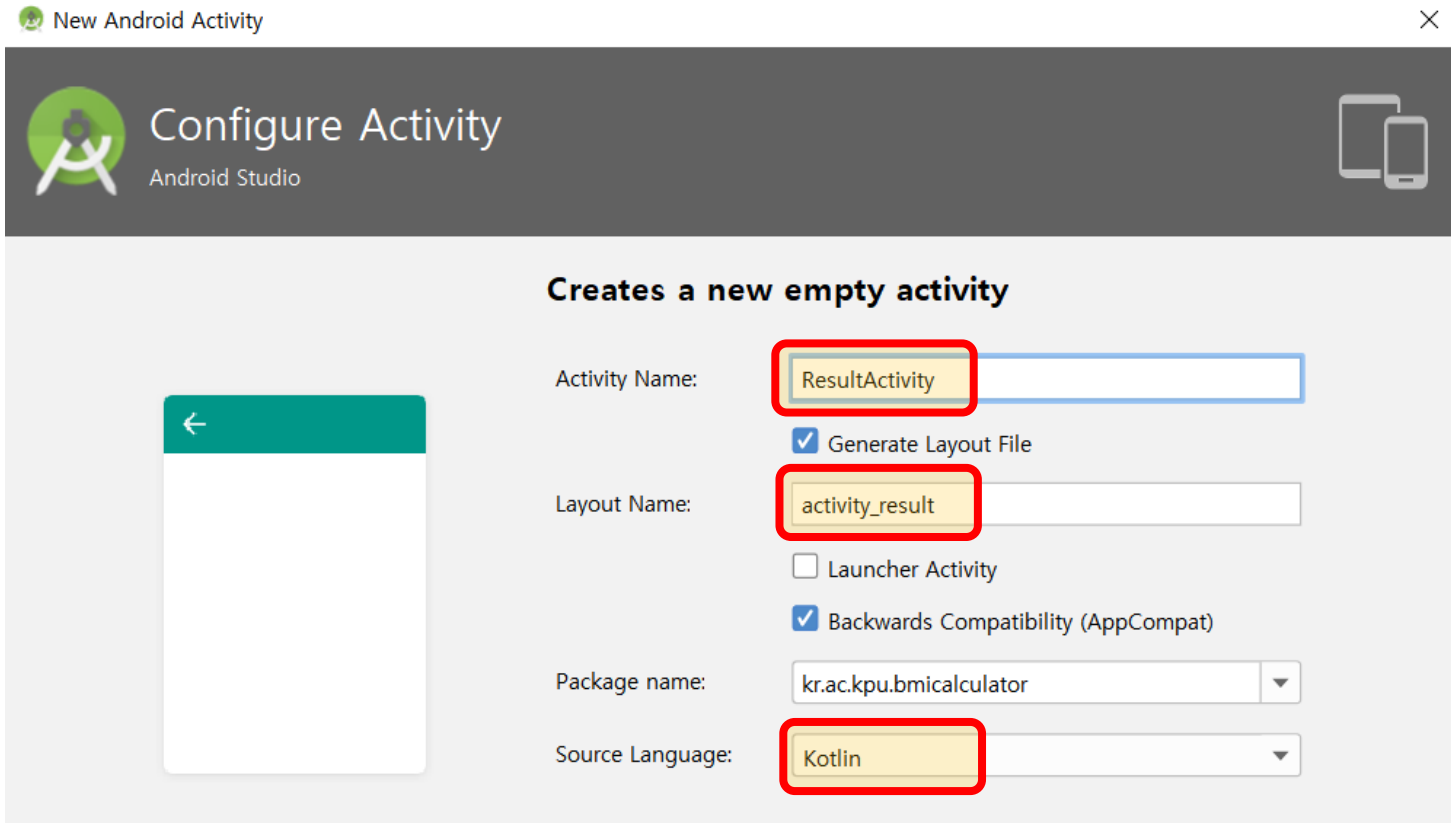


비만도 계산기 앱 만들기

▶ 액티비티 추가

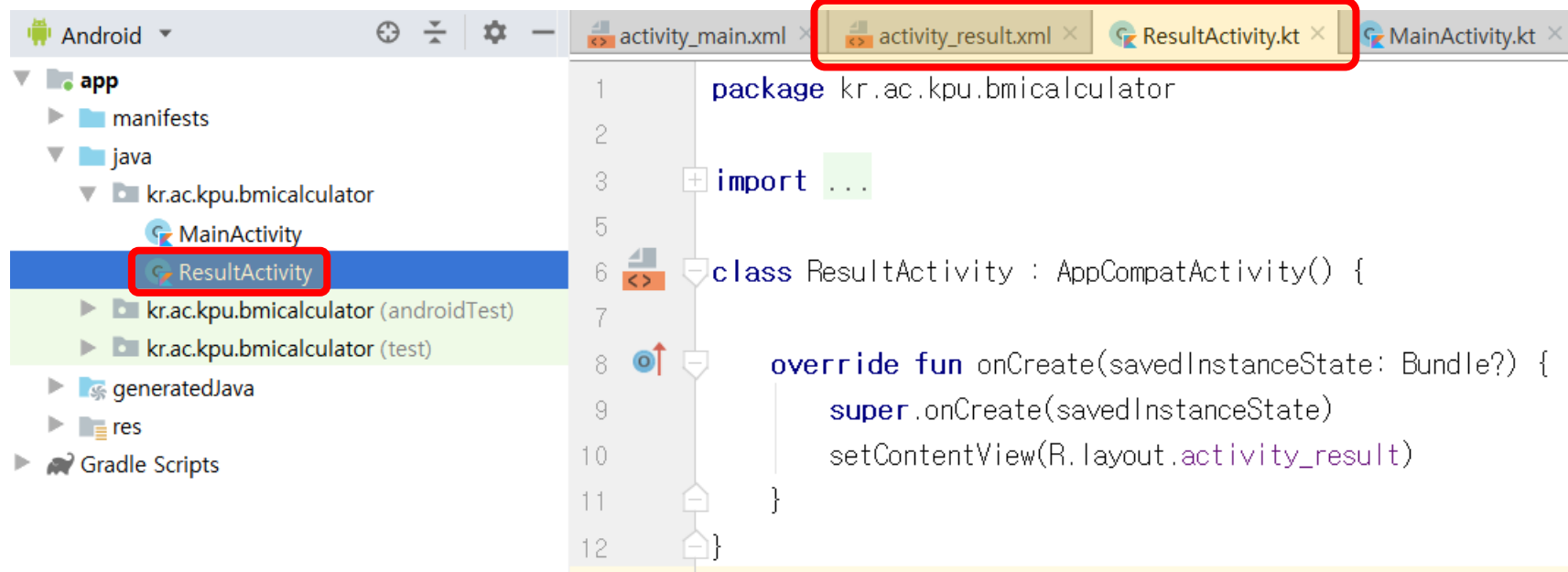
▶ ResultActivity / activity_result

▶ 액티비티 이름을 넣으면 레이아웃은 자동으로 채워짐



비만도 계산기 앱 만들기

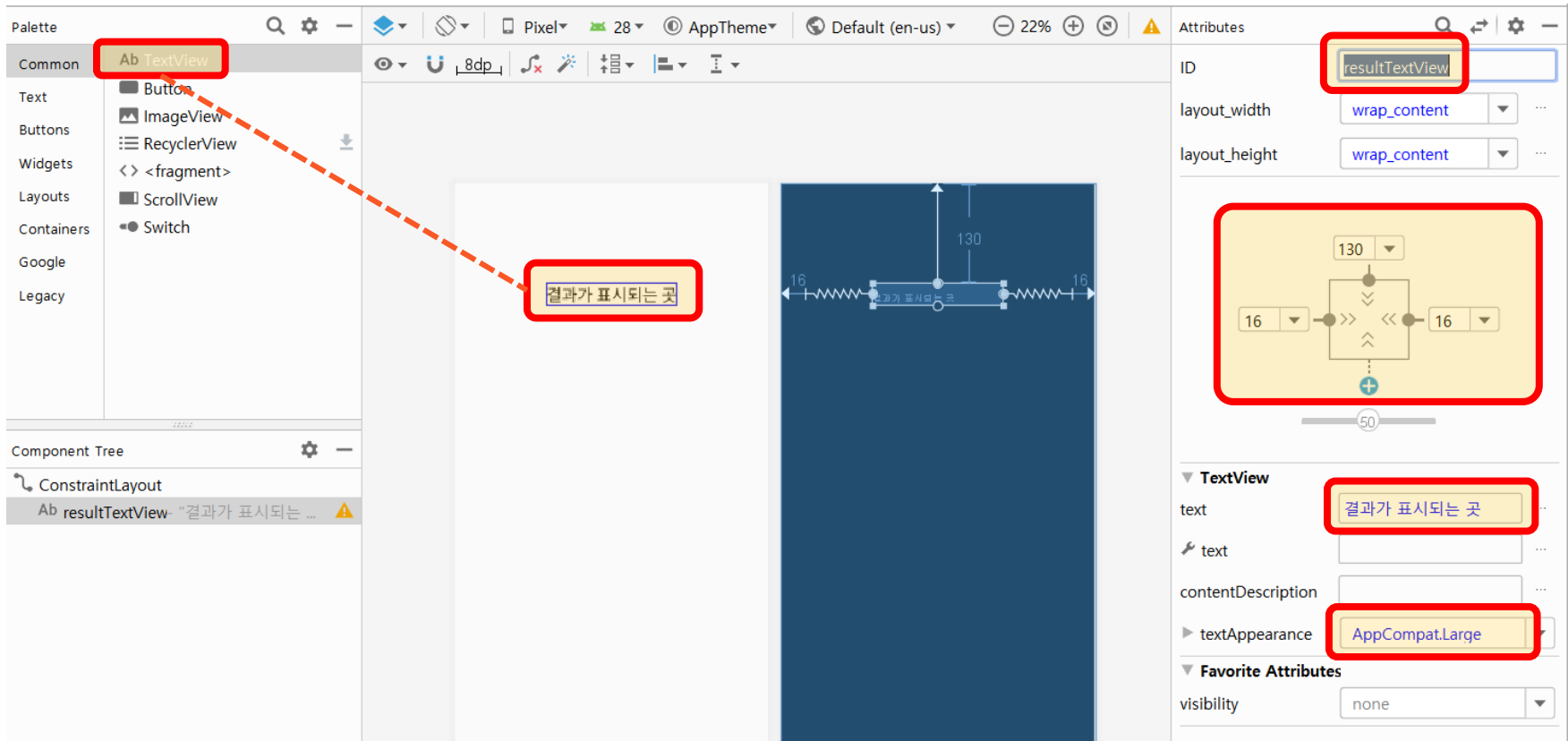
▶ 추가된 액티비티 확인



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 결과를 표시할 텍스트 뷰 배치



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

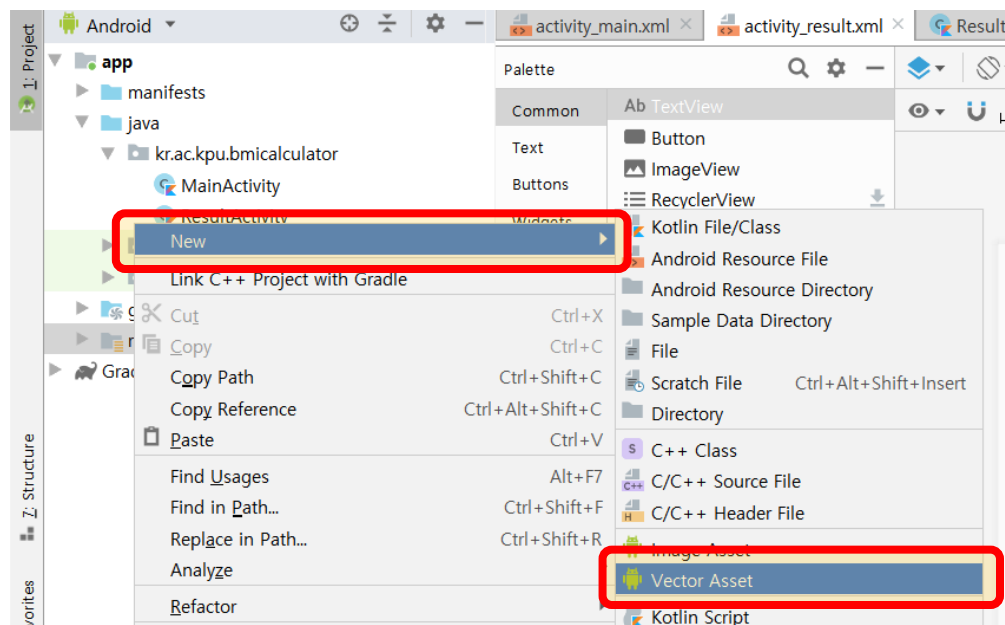
▶ 이미지를 표시할 이미지 뷰 배치

▷ 이미지 뷰를 추가하기 위해서는 적당한 이미지가 필요함

▷ VectorDrawable이라는 벡터 이미지 리소스를 작성

- 앱 개발에서는 주로 PNG, JPG 등의 비트맵 이미지 파일과 SVG, EPS 등의 벡터 이미지를 사용

▷ 프로젝트 창의 res 폴더에서 마우스 우 클릭 - [New] - [Vector Asset]을 클릭



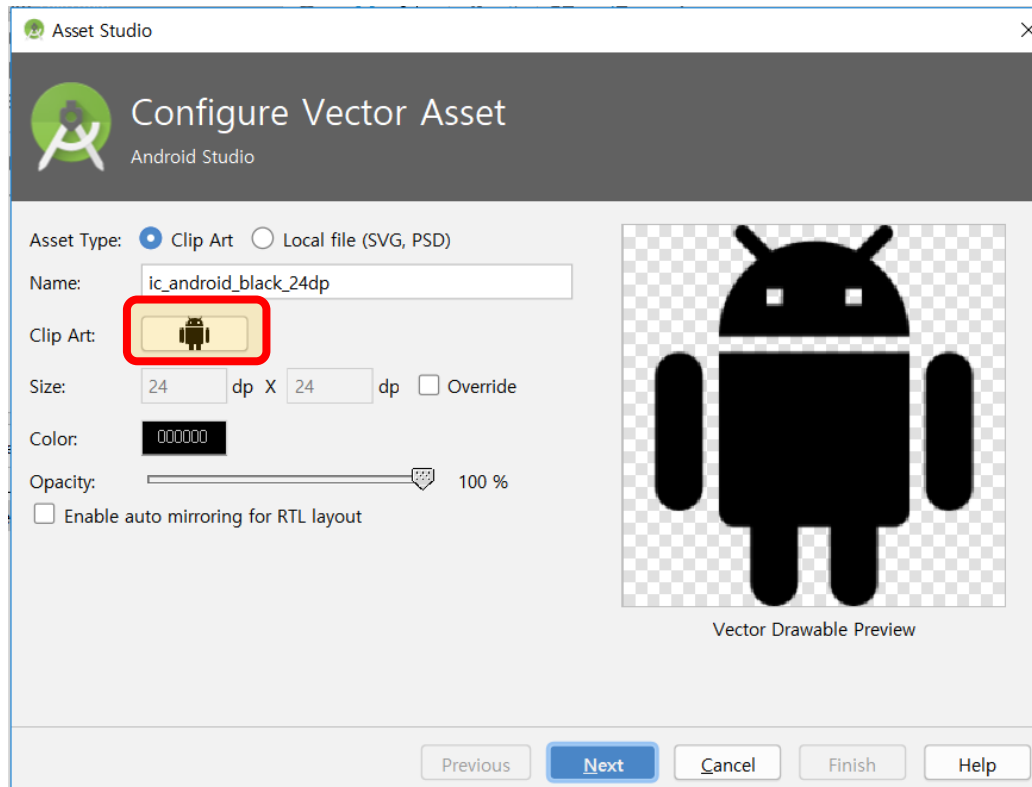
비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ Asset Studio 화면 표시

▶ 여기서 벡터 이미지를 생성 가능

▶ Clip Art 아이콘을 클릭하여 다양한 이미지를 확인



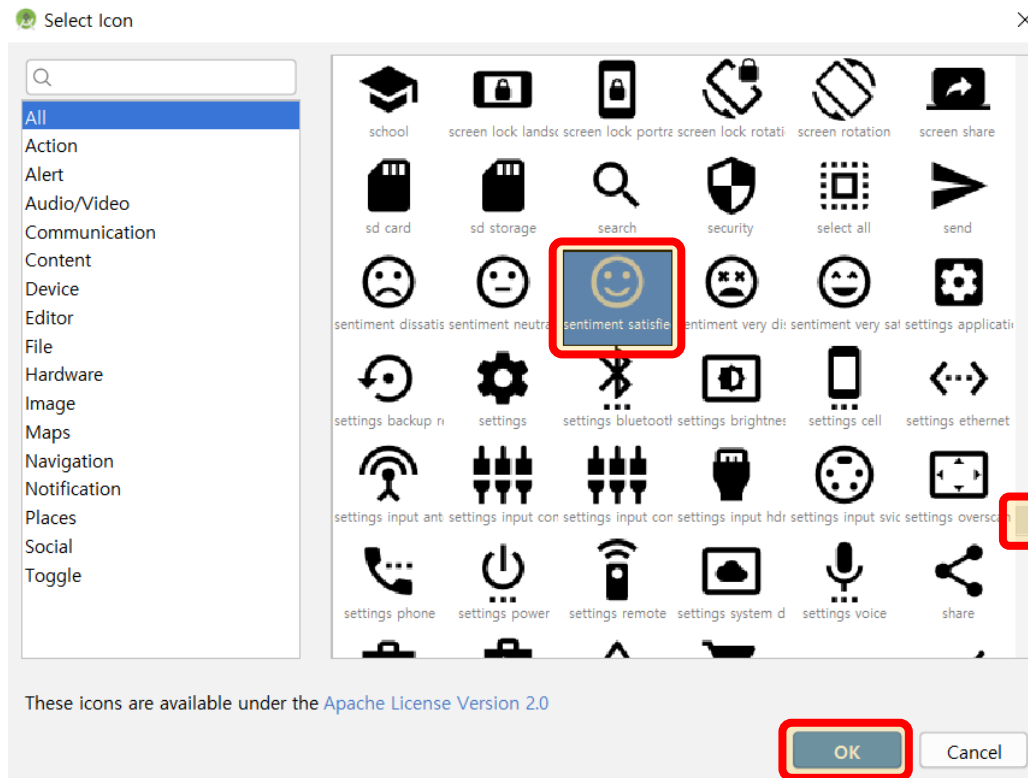
비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 비만도가 정상일 경우 사용할 이미지를 선택 : 웃는 얼굴

▶ sentiment satisfied

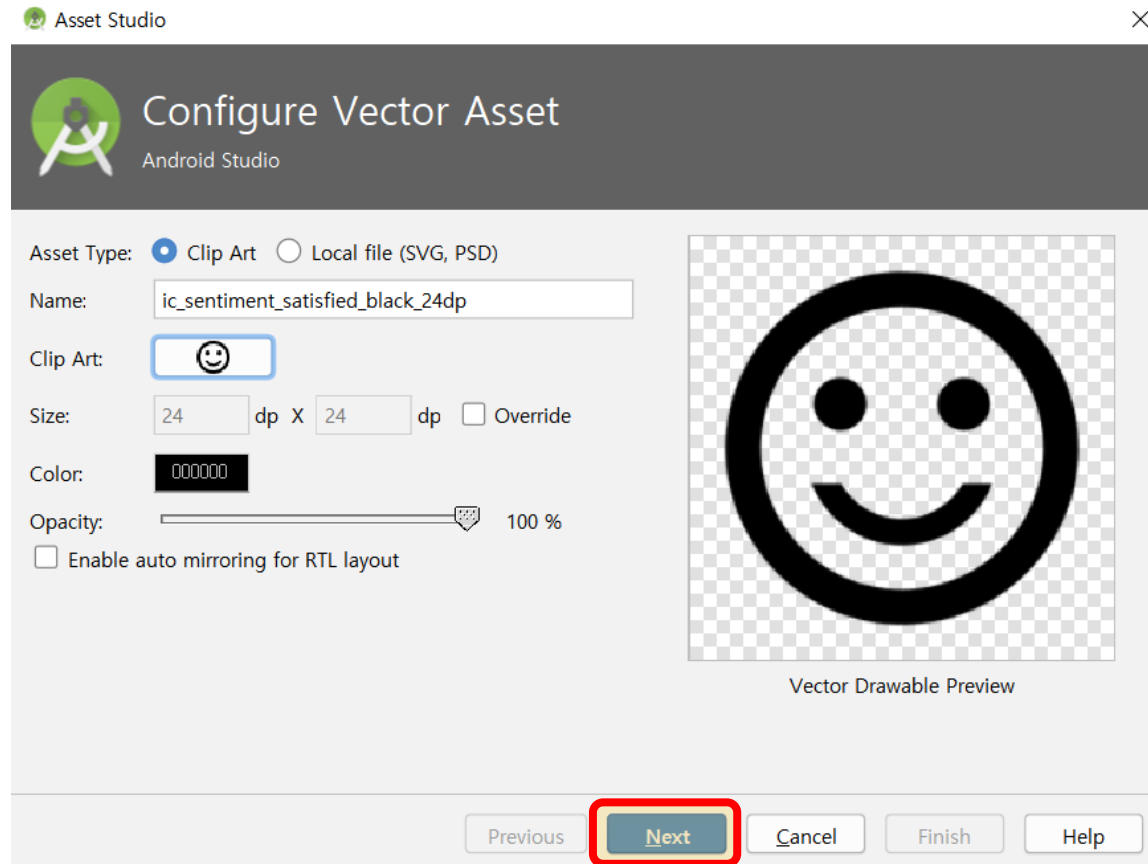
▶ 스크롤바를 그림과 같이 아래로 이동하여 검색



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 이미지 확인 후 Next

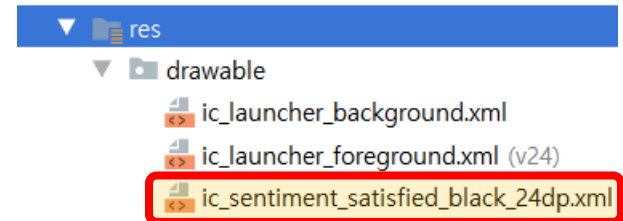
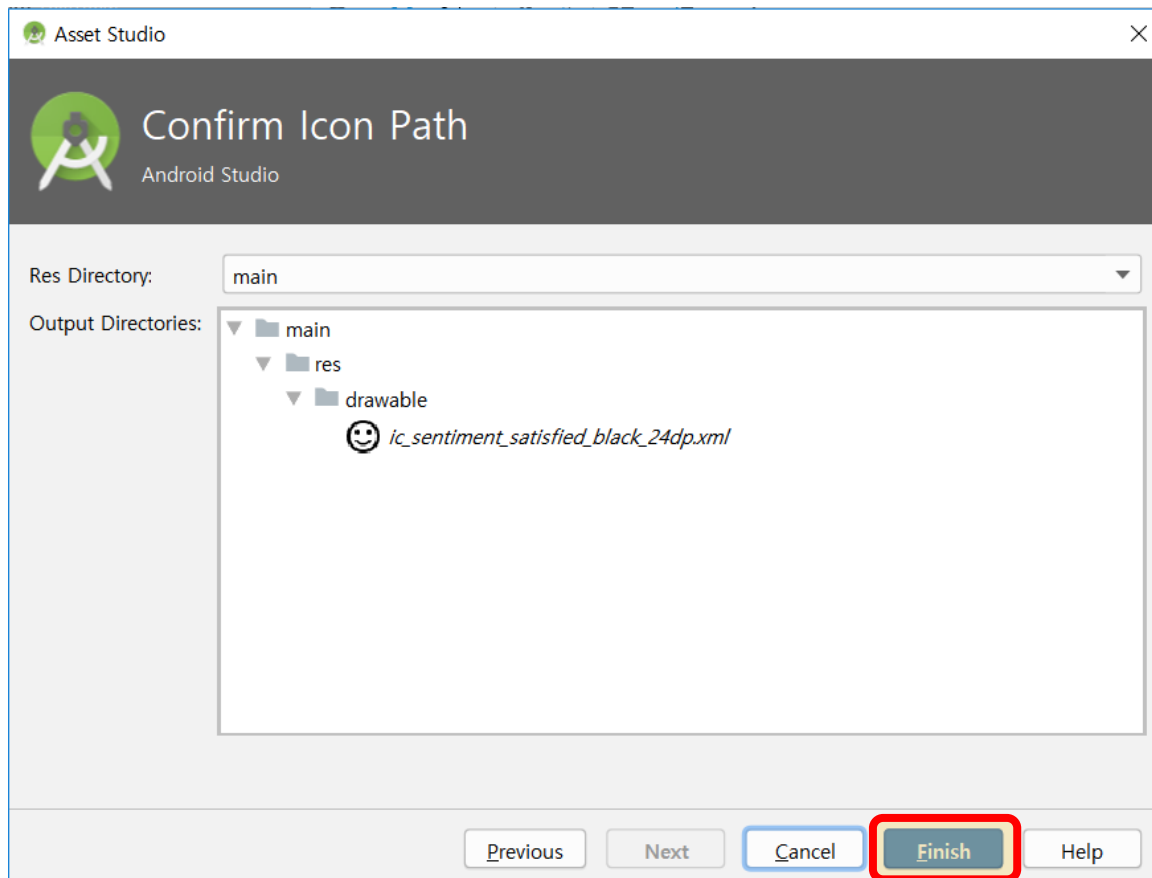


비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 아이콘이 저장될 위치 확인

▶ 앱 개발에 사용되는 이미지 리소스는 drawable 폴더에서 관리

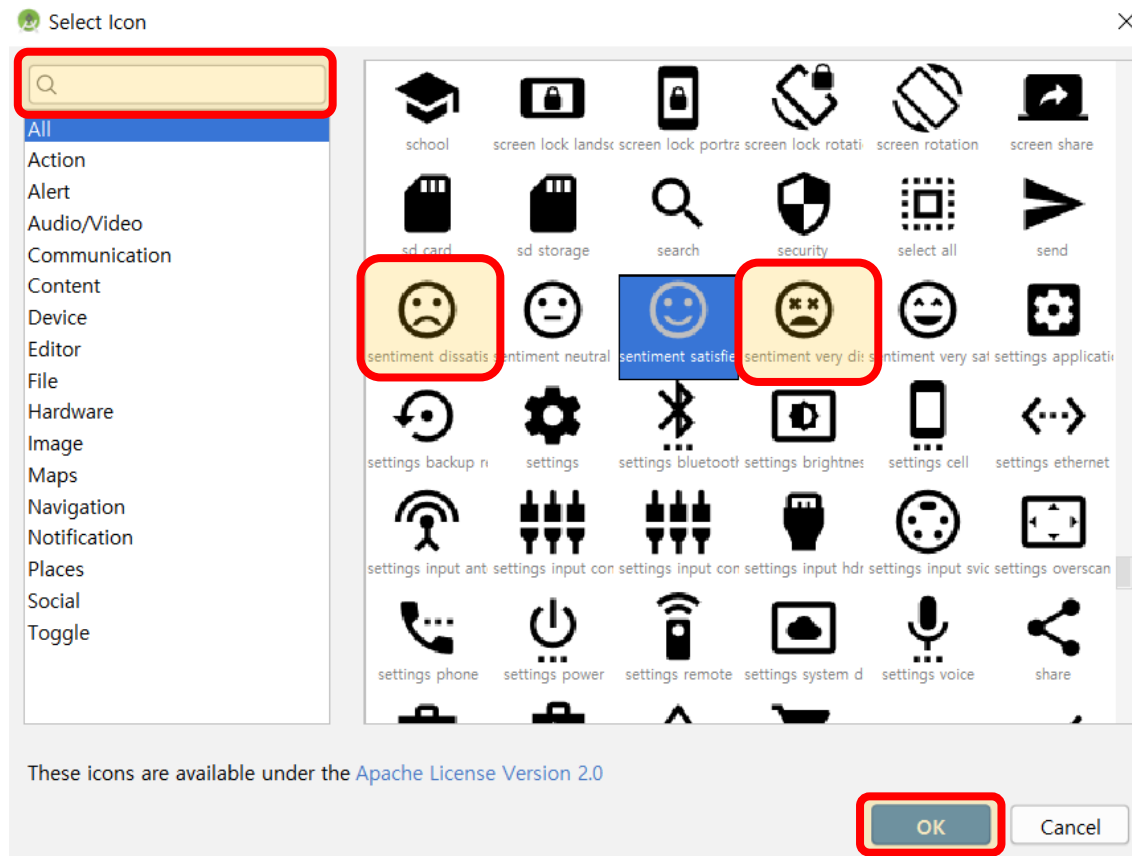


비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 동일한 방법으로 비만과 저 체중일 경우 표시할 이미지 추가

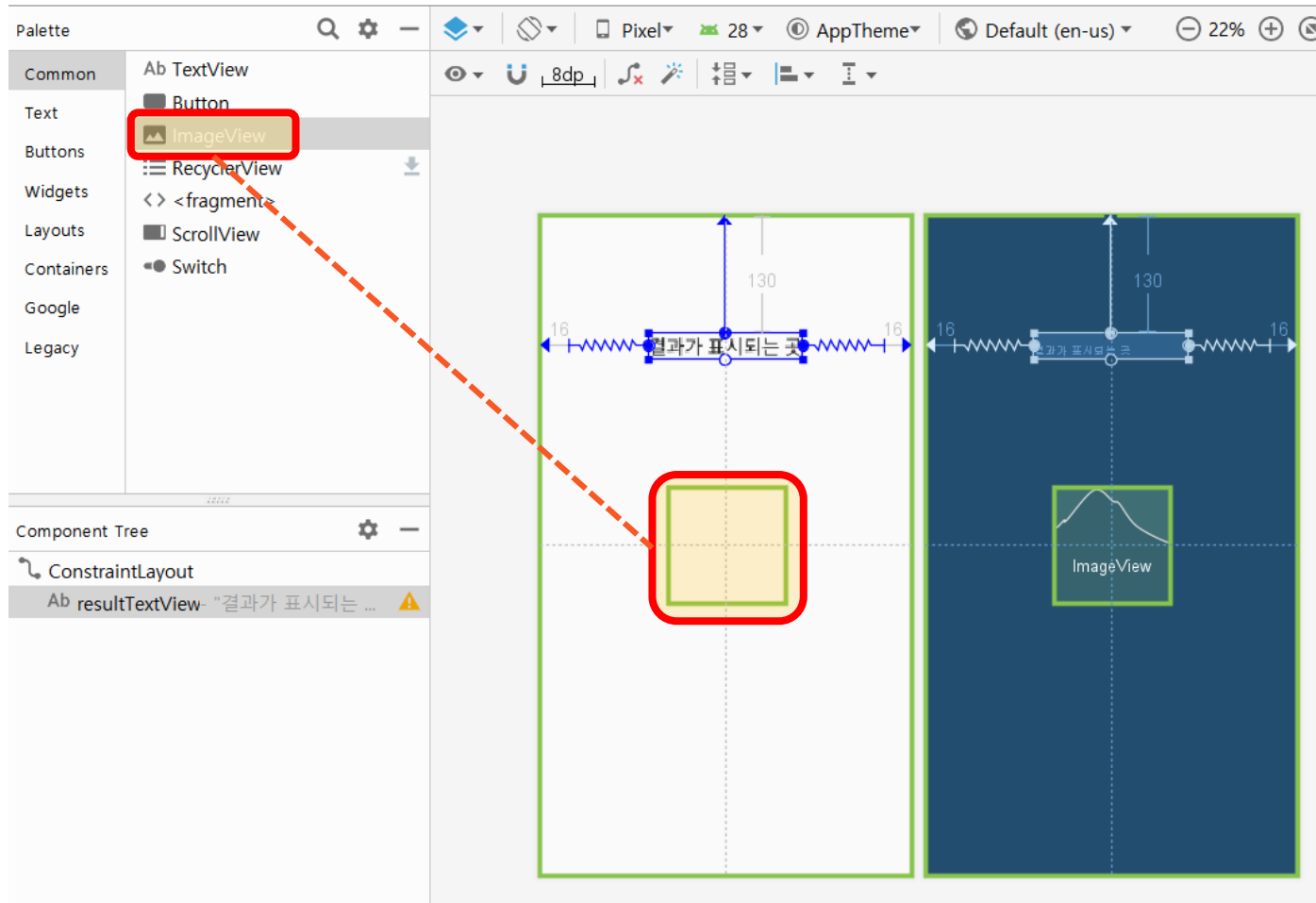
▶ 검색창에서 “sentiment”로 검색



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

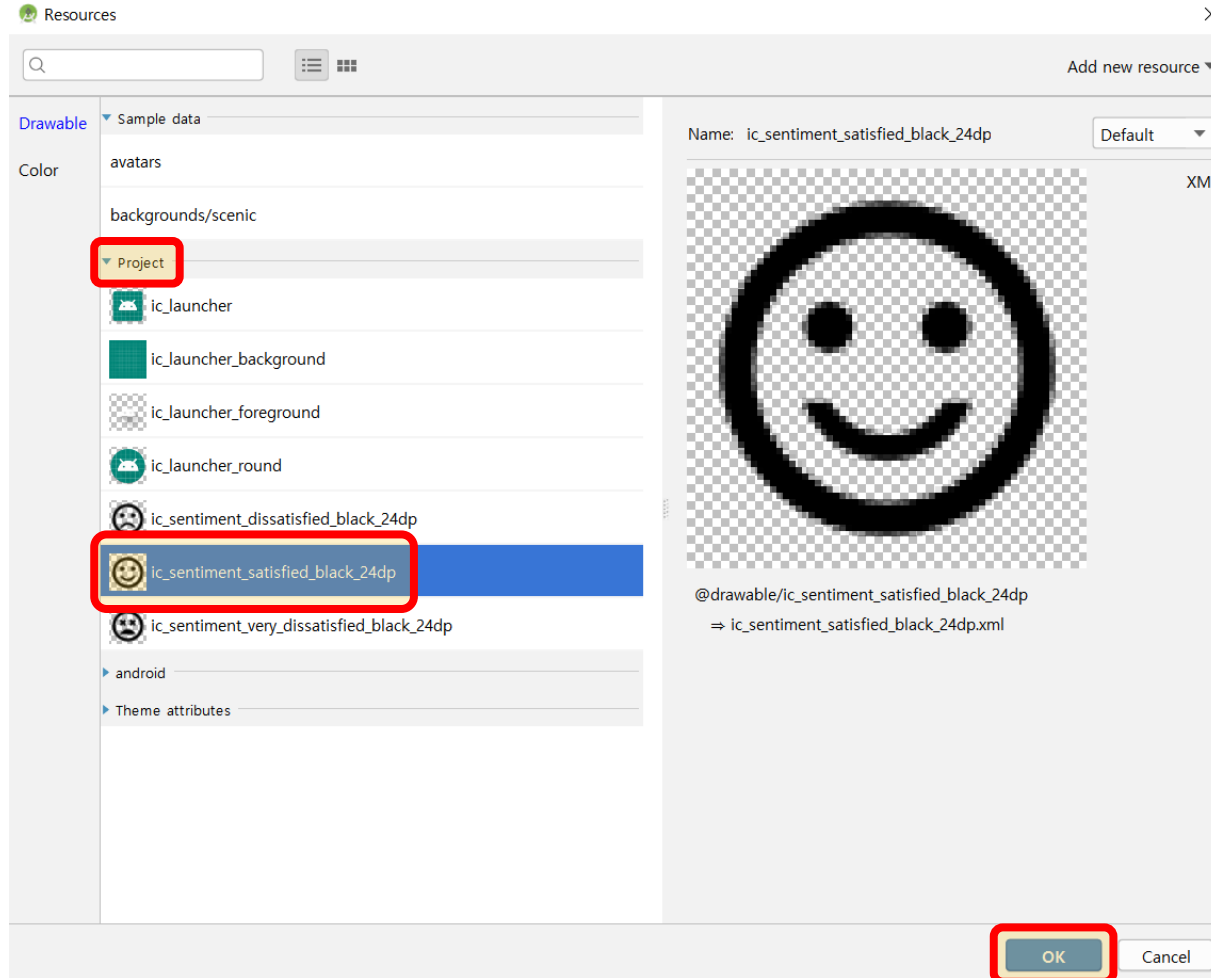
▶ ImageView를 화면의 정중앙에 배치



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 리소스를 선택하는 창에서 웃는 얼굴 이미지를 선택

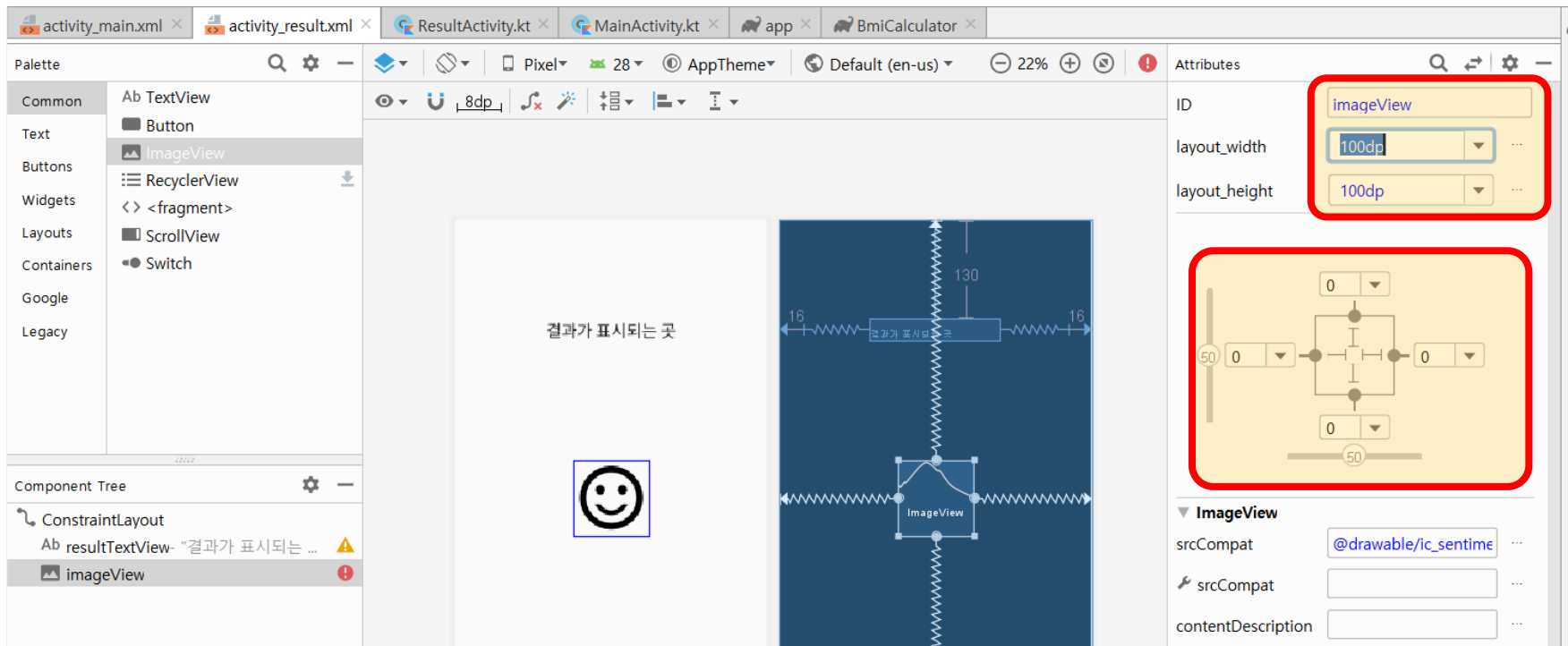


비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 이미지의 크기가 작으므로 크게 보이도록 조절

▶ 예제에서는 100dp로 설정



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

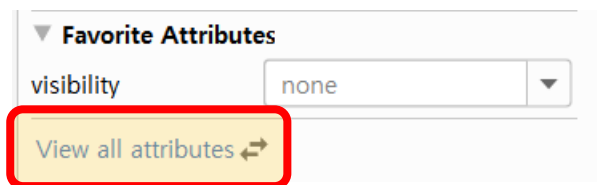
▶ tint 속성을 이용하여 이미지의 색상을 타이틀바와 어울리도록 변경



▷ 색상 조합을 위하여 기본 테마의 대표 색상을 이미지 뷰의 벡터 이미지에 적용

▷ Tint 속성이 보이지 않는다면 속성 창 하단에 [view all attributes]클릭

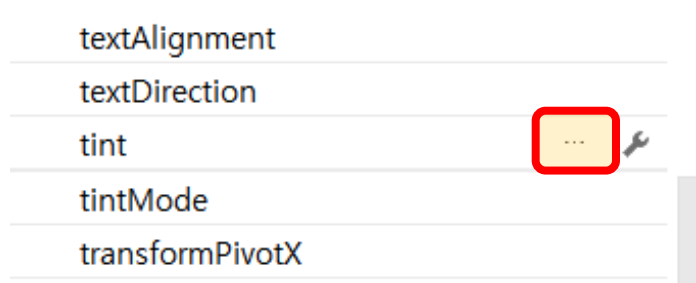
■ 뷰의 전체 속성이 표시됨



▷ 또는 상단에 Attributes 탭과 아이콘을 클릭



▷ tint 속성에서 ...을 클릭



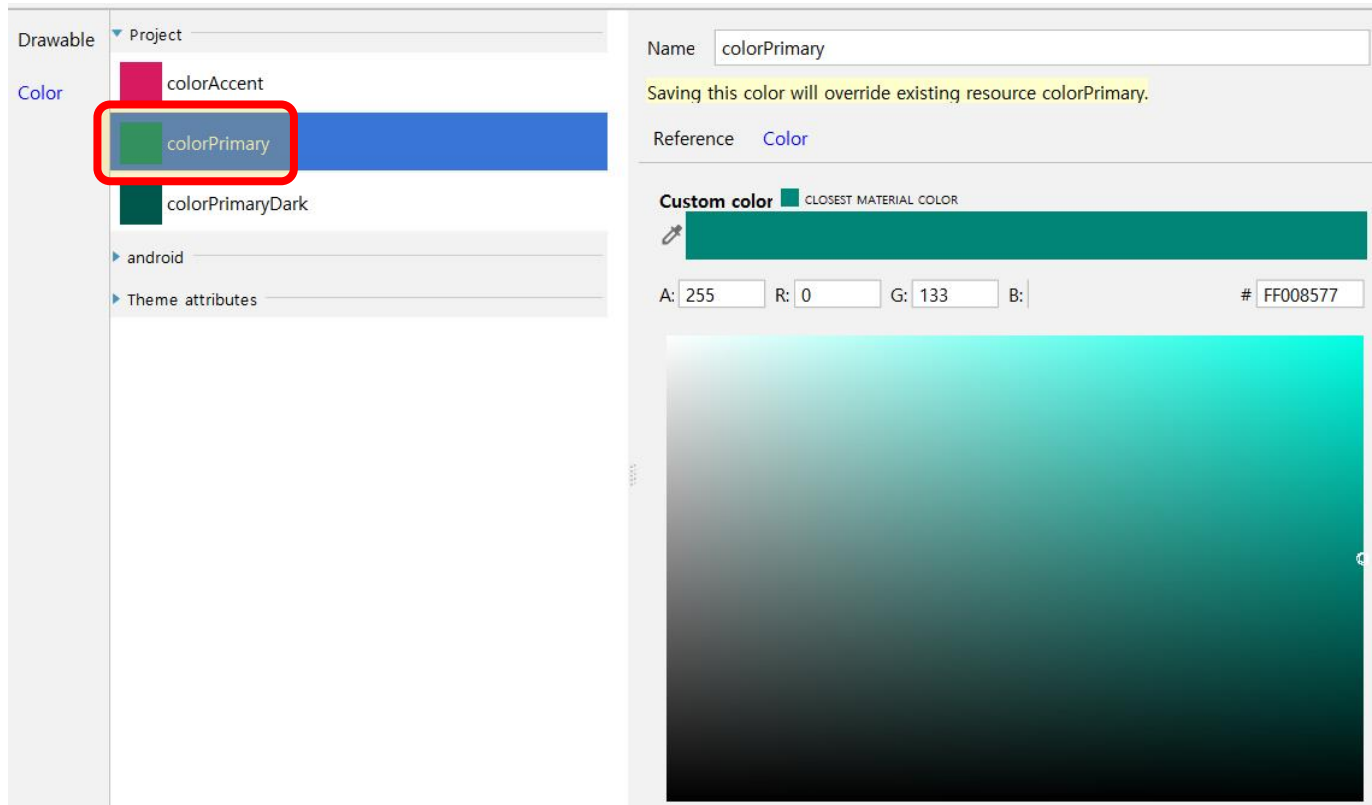
비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 세 가지의 대푯값 중에 원하는 것을 선택

▶ 예제에서는 colorPrimary를 선택

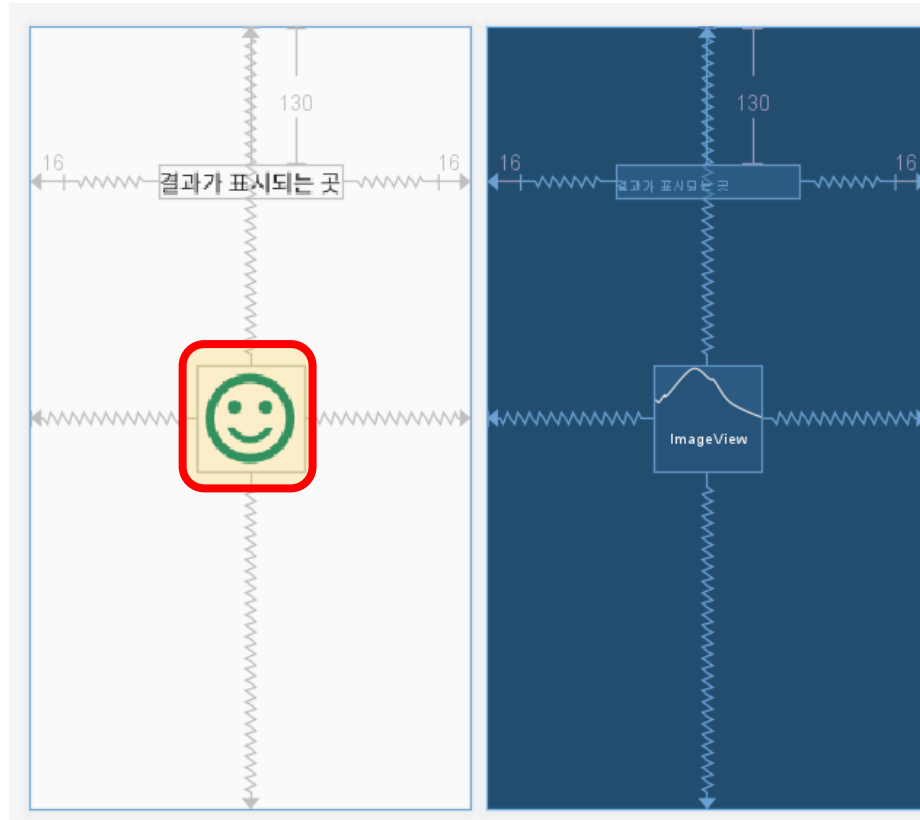
- 안드로이드 스튜디오 버전에 따라 색상은 약간씩 다를 수 있으니 참고



비만도 계산기 앱 만들기

▶ 결과를 표시할 화면 디자인

▶ 이미지가 선택한 색상으로 변경



비만도 계산기 앱 만들기

▶ 벡터 드로어블(VectorDrawable) 사용 환경 설정

- ▶ 결과 화면에서 사용한 벡터 이미지는 벡터 드로어블 리소스로 분류
- ▶ 벡터 드로어블 리소스는 안드로이드 5.0부터 사용 가능하지만 현재 프로젝트 버전은 4.4
- ▶ 안드로이드 5.0 미만의 버전에서도 해당 리소스가 표시되도록 모듈 수준의 build.gradle 파일의 defaultConfig 내부에 아래 코드를 추가 후 싱크
 - ▶ `vectorDrawables.useSupportLibrary = true`

```
7  android {  
8      compileSdkVersion 28  
9      defaultConfig {  
10         applicationId "kr.ac.kpu.bmicalculator"  
11         minSdkVersion 19  
12         targetSdkVersion 28  
13         versionCode 1  
14         versionName "1.0"  
15         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
16     }  
17     vectorDrawables.useSupportLibrary = true  
18 }
```

Gradle files have changed since last project sync. A project sync may be necessary fo. [Sync Now](#)

비만도 계산기 앱 만들기

▶ 인텐트(Intent)로 화면 전환

▶ 인텐트를 이용하여 첫 번째 화면의 버튼을 누르면 두 번째 화면이 표시되도록 구현

▶ 인텐트는 다른 액티비티를 실행시키거나 간단한 데이터를 전달하는 기능을 함

▶ 개발 순서

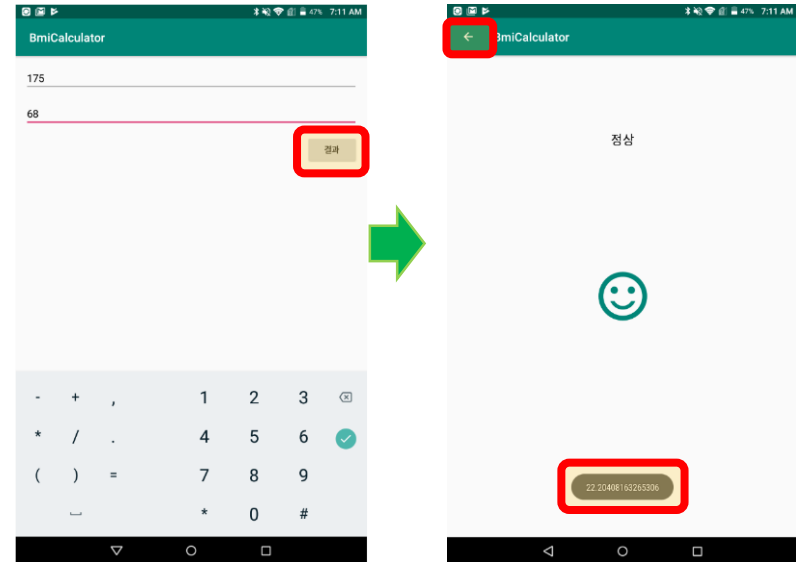
▶ 버튼을 클릭하면 결과 화면으로 전환하는 코드 작성

▶ 이전 화면으로 돌아가기 위한 앱 네비게이션 구현

▶ 인텐트에 데이터를 넣고 꺼내기

▶ 비만도를 계산하여 분기 처리

▶ Toast를 사용하여 간단한 메시지 표시



비만도 계산기 앱 만들기

▶ 버튼을 클릭하면 결과 화면으로 전환되는 코드 작성

▶ MainActivity.kt 파일에 작성

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    resultButton.setOnClickListener { it: View!  
        val intent = Intent(packageContext: this, ResultActivity::class.java)  
        startActivity(intent)  
    }  
}
```

▶ 인텐트를 사용하기 위하여 인텐트 클래스를 자동 임포트

- 빨간색으로 오류 표시가 되어 있는 부분으로 이동 후 키보드 alt + enter

```
? android.content.Intent Alt+Enter  
resultButton.setOnClickListener { it: View!  
    val intent = Intent(this, ResultActivity::class.java)  
    startActivity(intent)  
}
```

비만도 계산기 앱 만들기

▶ 버튼을 클릭하면 결과 화면으로 전환되는 코드 작성

▶ Anko 라이브러리를 사용하여 코드 변경

▶ 기존 코드

```
resultButton.setOnClickListener { it: View!  
    val intent = Intent(packageContext: this, ResultActivity::class.java)  
    startActivity(intent)  
}
```

▶ Anko 라이브러리를 사용하여 동일한 기능은 간단하게 작성

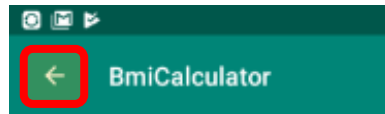
```
resultButton.setOnClickListener { it: View!  
    startActivity<ResultActivity>()  
}
```

▶ 정상 동작 확인

비만도 계산기 앱 만들기

▶ 이전 화면으로 돌아가기

▶ 앱 네비게이션을 사용하여 “뒤로가기” 아이콘 활성화



▷ 뒤로가기 클릭하여 이전화면으로 돌아갈 수 있음

▷ MainActivity에서 ResultActivity를 시작시키므로 ResultActivity에서 MainActivity를 부모 액티비티로 지정

▶ 액티비티간에 상관관계를 연결하도록 메니페스트에 parentActivityName 속성 추가

▷ android:parentActivityName=".MainActivity"

- 위의 코드를 복사하여 아래의 위치에 붙여넣기(“”에 오류가 발생하면 해당 부분만 삭제하고 직접 입력)

```
<activity
    android:name=".ResultActivity"
    android:parentActivityName=".MainActivity">
</activity>
```

▶ 추가 후 실행하여 잘 동작하는지 확인

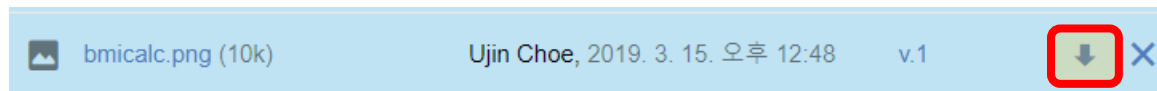
비만도 계산기 앱 만들기

▶ 앱 아이콘 변경

▶ 기본 아이콘에서 사용자가 원하는 디자인의 아이콘으로 변경 가능

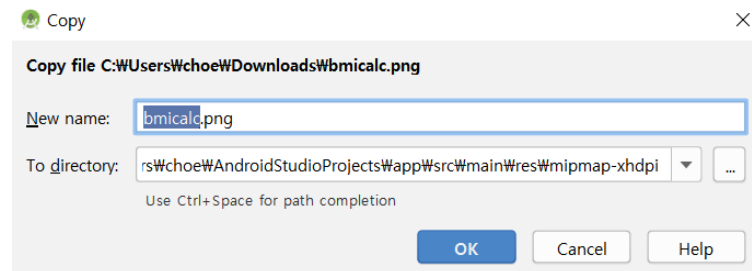
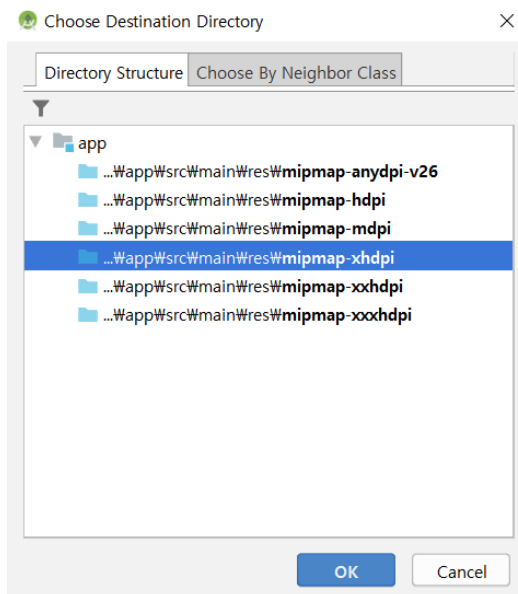
▶ 강의사이트에서 아이콘 다운로드

▶ <https://sites.google.com/site/choeuzin/korearental>



▶ Res - mipmap에 복사

▶ Ctrl + c Ctrl + v



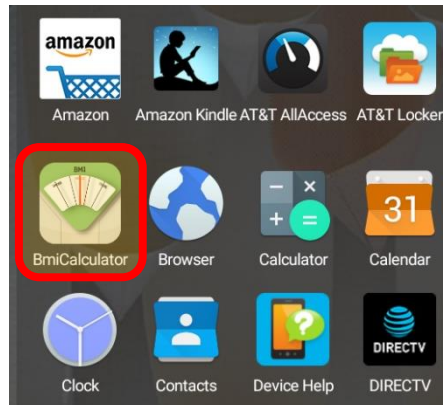
비만도 계산기 앱 만들기

▶ 앱 아이콘 변경

▶ AndroidManifest.xml 변경

- ▶ 아이콘 이미지는 mipmap에 복사됨
- ▶ `android:icon = "@mipmap/bmicalc"`
- ▶ `android:roundIcon = "@mipmap/bmicalc"`
 - 안드로이드 7.1이상에서 적용
- ▶ 잘 적용되었는지 실행하여 확인

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/bmicalc"
    android:label="Bmi Calculator"
    android:roundIcon="@mipmap/bmicalc"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
```



비만도 계산기 앱 만들기

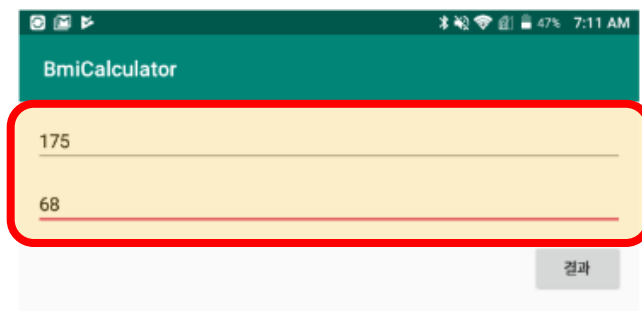
▶ 인텐트에 데이터 저장

▶ 기존의 Intent 사용 코드

```
resultButton.setOnClickListener { it: View!  
    val intent = Intent( packageContext: this, ResultActivity::class.java)  
    intent.putExtra( name: "weight", weightEditText.text.toString())  
    intent.putExtra( name: "height", heightEditText.text.toString())  
    startActivity(intent)  
}
```

▶ Anko 라이브러리로 intent 사용 코드

```
resultButton.setOnClickListener { it: View!  
    startActivity<ResultActivity>( ...params:  
        "weight" to weightEditText.text.toString(),  
        "height" to heightEditText.text.toString()  
    )  
}
```



비만도 계산기 앱 만들기

▶ 인텐트에서 데이터 추출

▶ 인텐트에서 데이터를 추출할 경우 getXXXExtra() 메서드를 사용

▷ XXX는 타입을 의미

■ getStringExtra, getIntExtra, ...

▶ 첫 화면에서 전달한 데이터는 String 타입이기 때문에 getStringExtra() 메서드를 사용

▷ 추출한 문자열은 toInt() 메서드를 사용하여 정수형으로 변환

▶ ResultActivity.kt에 전달받은 데이터를 추출하는 코드 작성

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_result)  
  
    val height = intent.getStringExtra(name: "height").toInt()  
    val weight = intent.getStringExtra(name: "weight").toInt()  
}
```

비만도 계산기 앱 만들기

▶ BMI를 계산하는 코드 추가

▶ 키를 100으로 나눈 값을 제공한 후, 몸무게에서 나누면 BMI 값을 계산

▶ BMI = 몸무게 ÷ (키/100)²

```
val height = intent.getStringExtra(name: "height").toInt()
val weight = intent.getStringExtra(name: "weight").toInt()
```

```
val bmi = weight / Math.pow(height / 100.0, 2.0)
```

▶ when 문을 사용하여 BMI 계산 결과가 특정 구간에 있으면 해당 메시지를 텍스트 뷰에 표시

```
val bmi = weight / Math.pow(height / 100.0, 2.0)
```

```
when {
    bmi >= 35 -> resultTextView.text = "고도 비만"
    bmi >= 30 -> resultTextView.text = "2단계 비만"
    bmi >= 25 -> resultTextView.text = "1단계 비만"
    bmi >= 23 -> resultTextView.text = "과체중"
    bmi >= 18.5 -> resultTextView.text = "정상"
    else -> resultTextView.text = "저체중"
}
```

비만도 계산기 앱 만들기

- ▶ BMI 값에 따라 다른 이미지를 표시하는 코드 추가
 - ▶ when 문을 사용하여 BMI 값의 구간에 따라 다른 이미지로 변경
 - ▶ 텍스트 뷰 코드의 바로 아래에 추가

```
when {  
    bmi >= 23 ->  
        imageView.setImageResource(  
            R.drawable.ic_sentiment_very_dissatisfied_black_24dp)  
    bmi >= 18.5 ->  
        imageView.setImageResource(  
            R.drawable.ic_sentiment_satisfied_black_24dp)  
    else ->  
        imageView.setImageResource(  
            R.drawable.ic_sentiment_dissatisfied_black_24dp)  
}
```

비만도 계산기 앱 만들기

▶ Toast를 사용하여 간단한 메시지 표시

▶ Toast를 이용하여 안드로이드 기기에서 BMI 값을 잠깐 보였다가 사라지도록 구현

▶ ResultAcitivity 클래스의 onCreate() 메서드의 가장 아래에 코드 추가

▶ 기존의 Toast 코드

```
// 토스트 메시지로 BMI 값 표시  
Toast.makeText(context: this, text: "$bmi", Toast.LENGTH_SHORT).show()
```

▶ Anko 라이브러리를 이용한 Toast 코드

```
// 토스트 메시지로 BMI 값 표시  
toast("$bmi")
```

비만도 계산기 앱 만들기

▶ SharedPreferences로 데이터 저장하기

- ▶ 가장 최근에 입력한 키와 몸무게가 다음 번 앱을 실행했을 때 남아 있도록 SharedPreferences에 저장하고 복원하는 기능을 작성
- ▶ 구현 순서
 - ▷ 데이터 저장
 - ▷ 데이터 열기

비만도 계산기 앱 만들기

▶ SharedPreferences로 데이터 저장하기

▶ MainActivity.kt에 키와 몸무게를 저장하는 메서드를 작성

```
private fun saveData(height: Int, weight: Int) { // 프리퍼런스 매니저를 이용하여 프리퍼런스 객체를 가져옴
    val pref = PreferenceManager.getDefaultSharedPreferences(context: this)
    val editor = pref.edit() // 프리퍼런스 객체의 에디터 객체를 가져옴
                                // 이 객체를 사용해 프리퍼런스에 데이터를 담을 수 있음

    editor.putInt("KEY_HEIGHT", height) // putXXX()메서드를 이용하여 키와 값을 쌍으로 저장, xxx는 데이터 타입
        .putInt("KEY_WEIGHT", weight)
        .apply() //설정된 내용을 반영
}
```

▶ 위의 메서드를 버튼을 클릭하고 화면이 전환되는 과정 사이에 추가하여 키와 몸무게 데이터를 저장

```
resultButton.setOnClickListener { it: View!
    saveData(heightEditText.text.toString().toInt(),
        weightEditText.text.toString().toInt())
    startActivity<ResultActivity>( ...params:
        "weight" to weightEditText.text.toString(),
        "height" to heightEditText.text.toString()
    )
}
```

비만도 계산기 앱 만들기

▶ SharedPreferences로 데이터 불러오기

▶ MainActivity.kt에 키와 몸무게를 불러오는 메서드를 작성

```
private fun loadData() { // 프리퍼런스 매니저를 이용하여 프리퍼런스 객체를 가져옴
    val pref = PreferenceManager.getDefaultSharedPreferences(context: this)
    val height = pref.getInt(key: "KEY_HEIGHT", defValue: 0) // getXXX()메소드와 키 값을 이용하여 데이터 로딩
    val weight = pref.getInt(key: "KEY_WEIGHT", defValue: 0) // 두번째 인자는 문제 발생시 디폴트 값

    if (height != 0 && weight != 0) { // 키와 몸무게 데이터를 정상적으로 가져오면
        heightEditText.setText(height.toString()) // 에디터 텍스트에 표시
        weightEditText.setText(weight.toString())
    }
}
```

▶ 액티비티가 실행되면 가장 최근에 입력한 값으로 표시하기

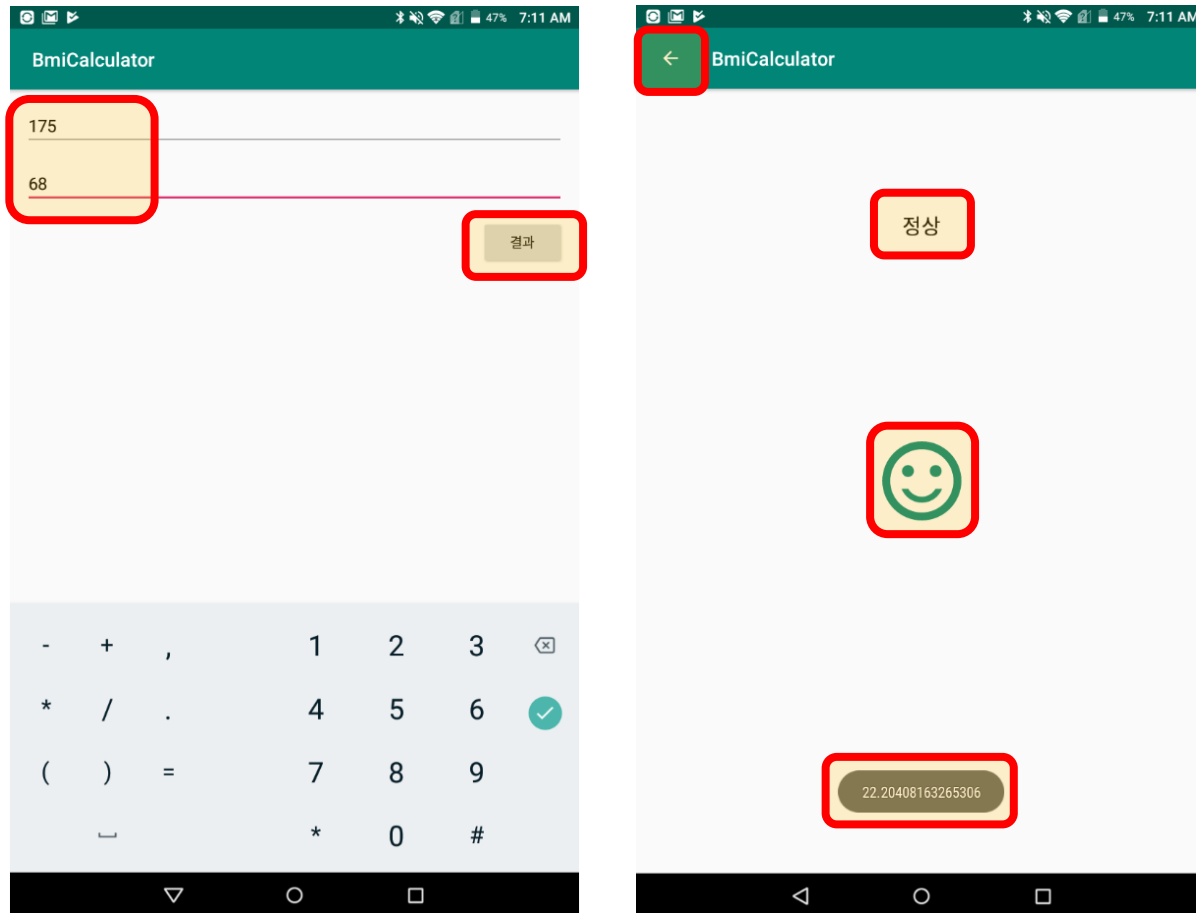
```
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)
```

```
loadData()
```

비만도 계산기 앱 만들기

▶ SharedPreferences로 데이터 불러오기

▶ 실행화면



연습문제

▶ 앞의 예제를 활용하여 사칙연산을 수행하는 앱을 작성하시오.

▷ 프로젝트를 새롭게 생성

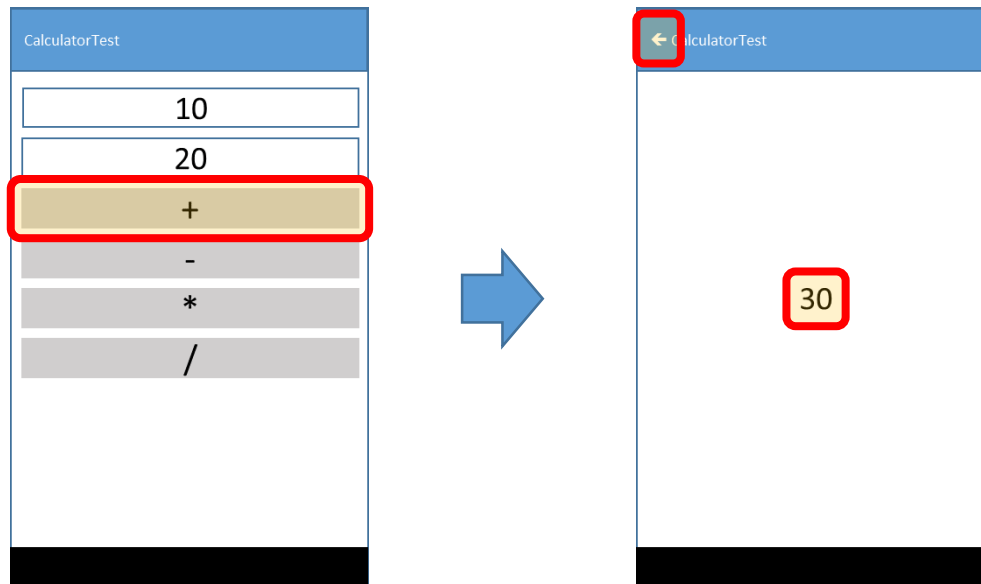
- 프로젝트 명 : KRCalculator

▷ 피연산자를 입력 받을 두 개의 에디트 텍스트와 각각의 사칙연산을 수행하는 버튼 4개로 구성

- 사칙연산 결과를 출력할 액티비티 추가 : PlusActivity, MinusActivity, DivideActivity, MultiplyActivity

- 각 액티비티에 뒤로 가기 버튼 활성화

▷ 최근에 입력한 피연산자를 sharedPreferences를 이용하여 저장/불러오기



Q & A
