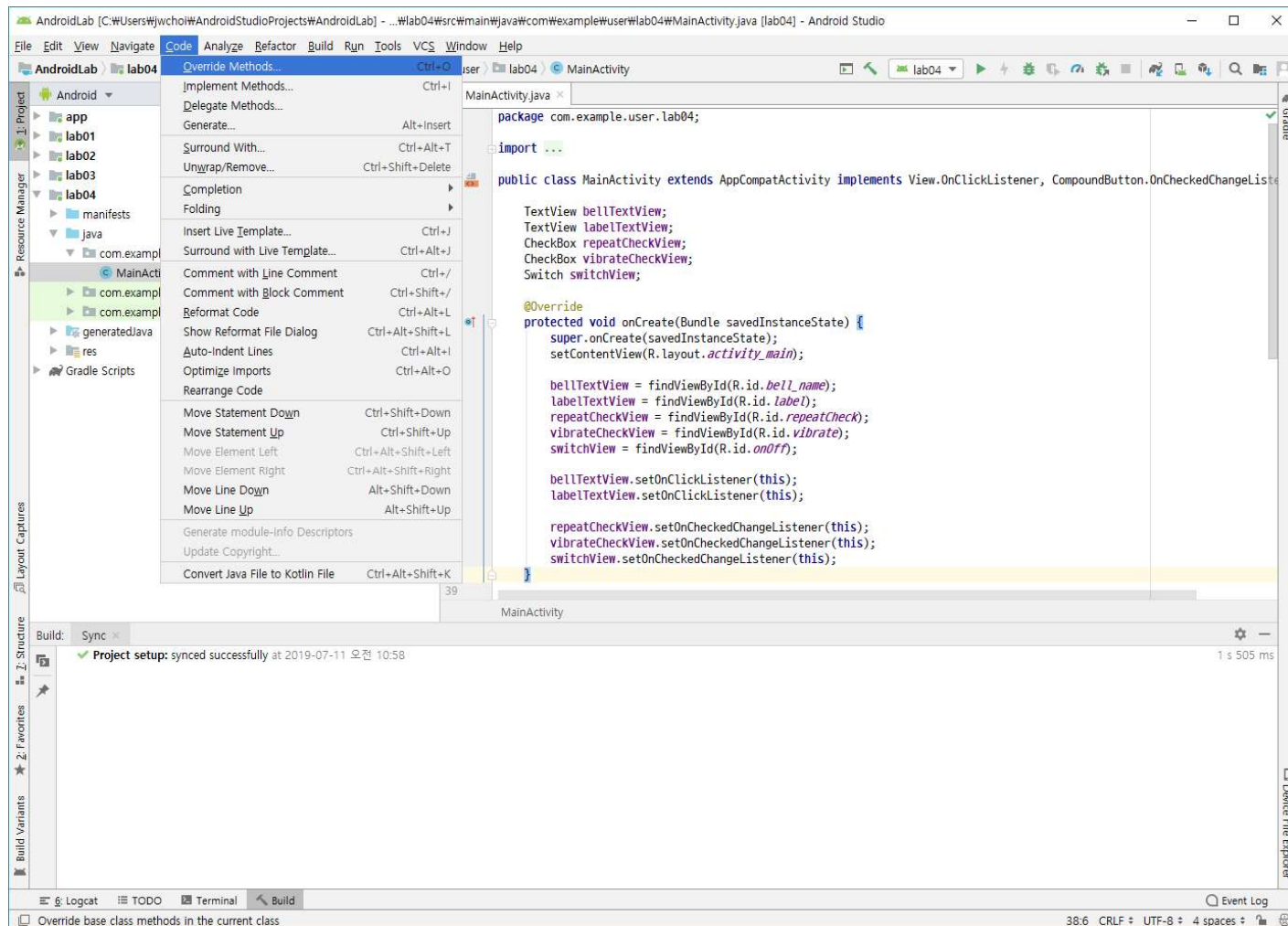
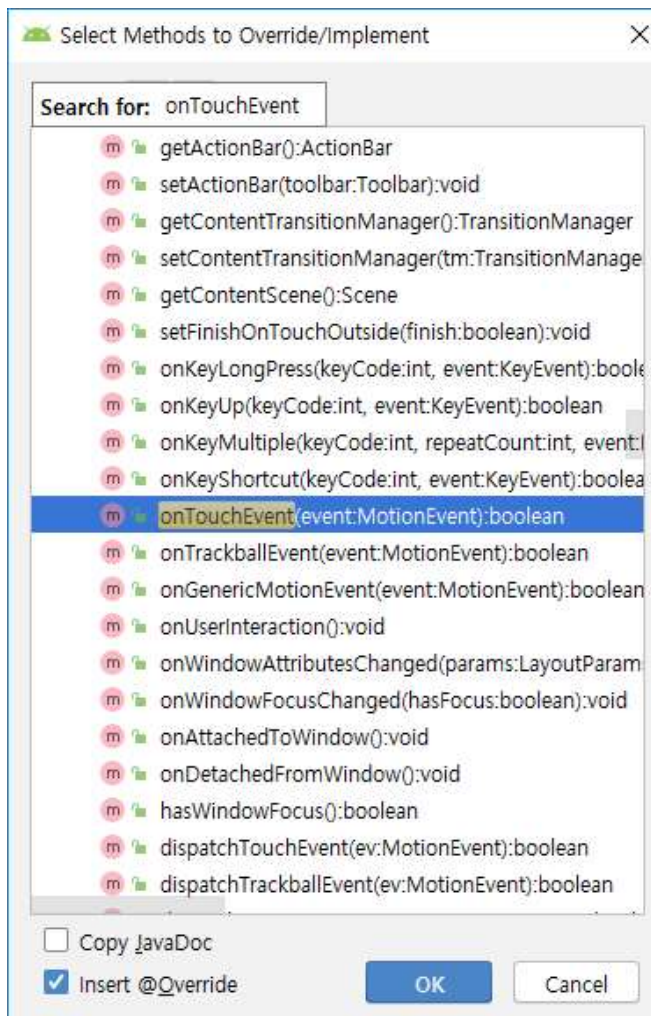


터치, 키 이벤트

Step 1 _ onTouchEvent() 함수 추가



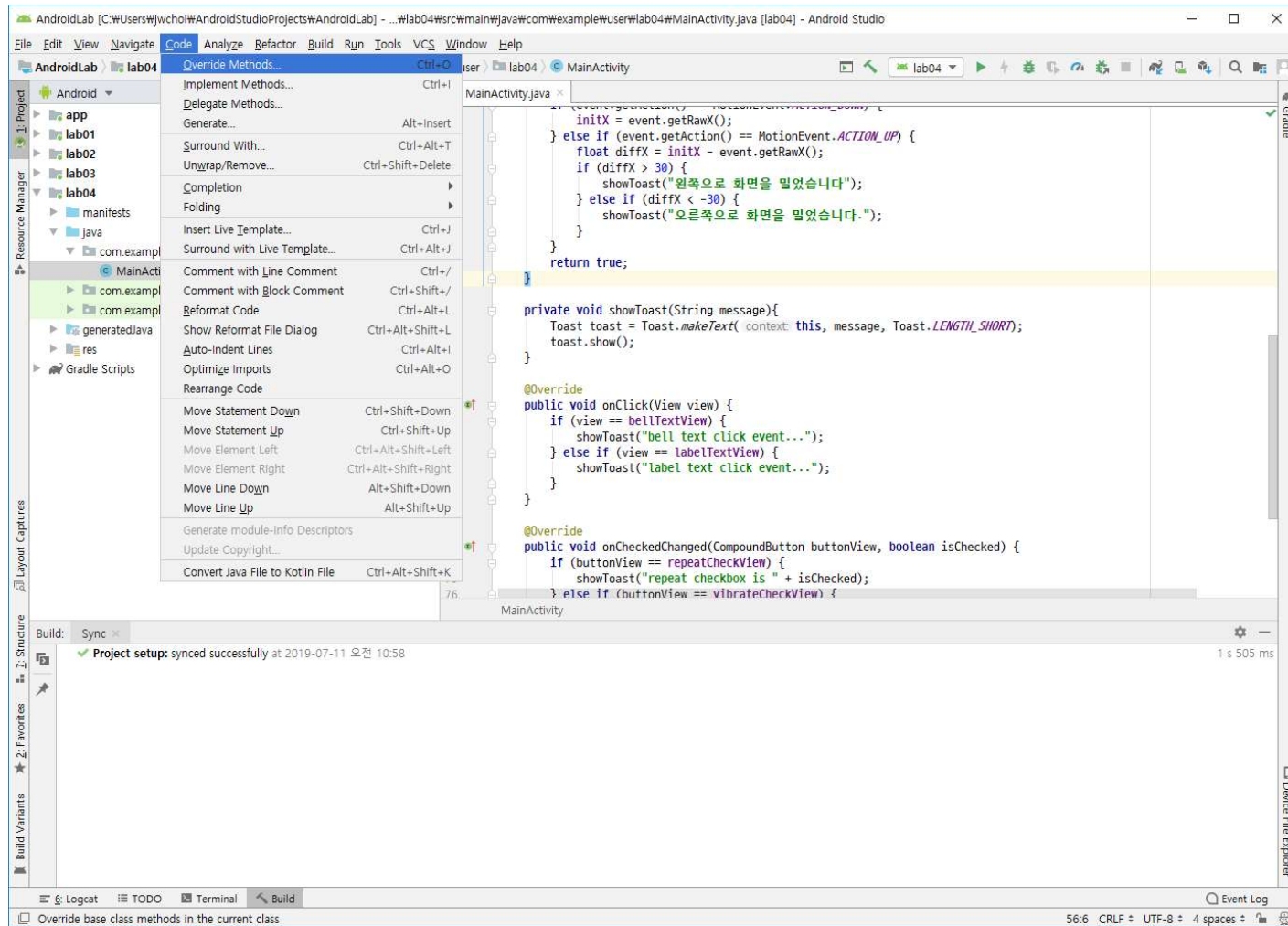


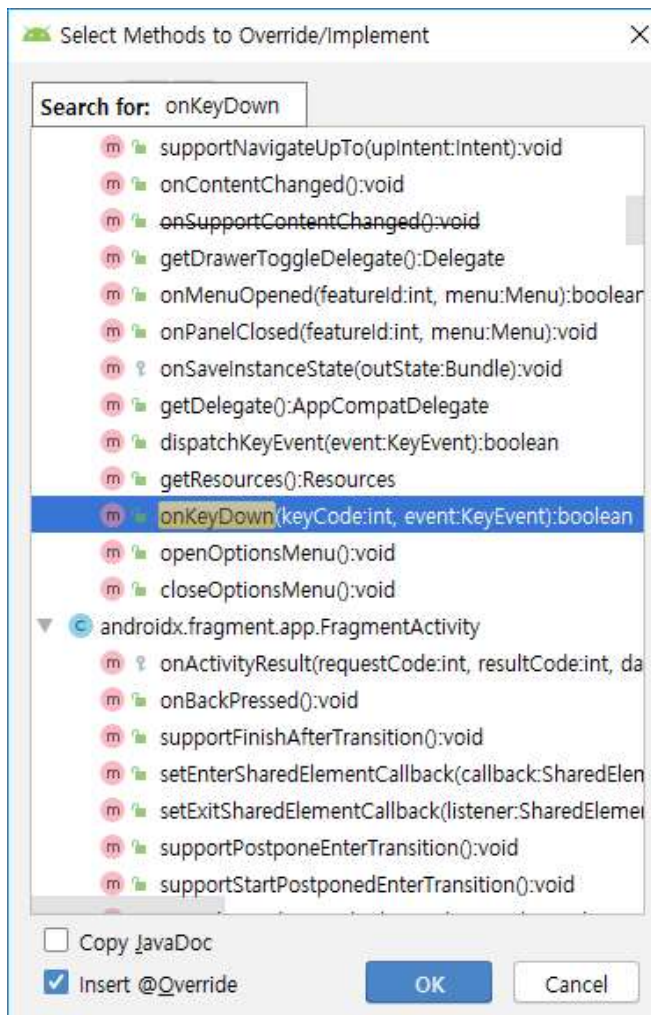
```
MainActivity.java x
1 package com.example.user.lab04;
2
3 import ...
13
14 public class MainActivity extends AppCompatActivity implements View.OnClickListener, CompoundButton.OnCheckedChangeListener {
15
16     TextView bellTextView;
17     TextView labelTextView;
18     CheckBox repeatCheckView;
19     CheckBox vibrateCheckView;
20     Switch switchView;
21
22     float initX;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28
29         bellTextView = findViewById(R.id.bell_name);
30         labelTextView = findViewById(R.id.label);
31         repeatCheckView = findViewById(R.id.repeatCheck);
32         vibrateCheckView = findViewById(R.id.vibrate);
33         switchView = findViewById(R.id.onOff);
34
35         bellTextView.setOnClickListener(this);
36         labelTextView.setOnClickListener(this);
37
38         repeatCheckView.setOnCheckedChangeListener(this);
39         vibrateCheckView.setOnCheckedChangeListener(this);
40         switchView.setOnCheckedChangeListener(this);
41     }
42
43     @Override
44     public boolean onTouchEvent(MotionEvent event) {
45         return super.onTouchEvent(event);
46     }
}
```

@Override

```
public boolean onTouchEvent(MotionEvent event) {  
    if (event.getAction() == MotionEvent.ACTION_DOWN) {  
        initX = event.getRawX();  
    } else if (event.getAction() == MotionEvent.ACTION_UP) {  
        float diffX = initX - event.getRawX();  
        if (diffX > 30) {  
            showToast("왼쪽으로 화면을 밀었습니다");  
        } else if (diffX < -30) {  
            showToast("오른쪽으로 화면을 밀었습니다.");  
        }  
    }  
    return true;  
}
```

Step 2 _ onKeyDown() 함수 추가






```

MainActivity.java x
1 package com.example.user.lab04;
2
3 import ...
4
14
15 public class MainActivity extends AppCompatActivity implements View.OnClickListener, CompoundButton.OnCheckedChangeListener {
16
17     TextView bellTextView;
18     TextView labelTextView;
19     CheckBox repeatCheckView;
20     CheckBox vibrateCheckView;
21     Switch switchView;
22
23     float initX;
24
25     long initTime;
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState) {
29         super.onCreate(savedInstanceState);
30         setContentView(R.layout.activity_main);
31
32         bellTextView = findViewById(R.id.bell_name);
33         labelTextView = findViewById(R.id.label);
34         repeatCheckView = findViewById(R.id.repeatCheck);
35         vibrateCheckView = findViewById(R.id.vibrate);
36         switchView = findViewById(R.id.onOff);
37
38         bellTextView.setOnClickListener(this);
39         labelTextView.setOnClickListener(this);
40
41         repeatCheckView.setOnCheckedChangeListener(this);
42         vibrateCheckView.setOnCheckedChangeListener(this);
43         switchView.setOnCheckedChangeListener(this);
44     }
45
46     @Override
47     public boolean onTouchEvent(MotionEvent event) {
48         if (event.getAction() == MotionEvent.ACTION_DOWN) {
49             initX = event.getRawX();
50         } else if (event.getAction() == MotionEvent.ACTION_UP) {
51             float diffX = initX - event.getRawX();
52             if (diffX > 30) {
53                 showToast("왼쪽으로 화면을 밀었습니다");
54             } else if (diffX < -30) {
55                 showToast("오른쪽으로 화면을 밀었습니다.");
56             }
57         }
58         return true;
59     }
60
61     @Override
62     public boolean onKeyDown(int keyCode, KeyEvent event) {
63         return super.onKeyDown(keyCode, event);
64     }

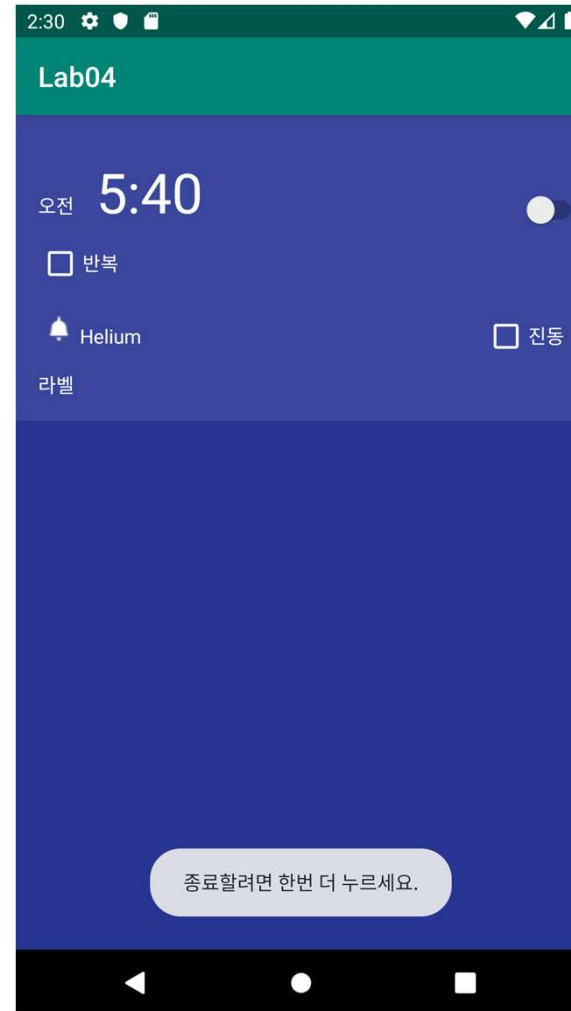
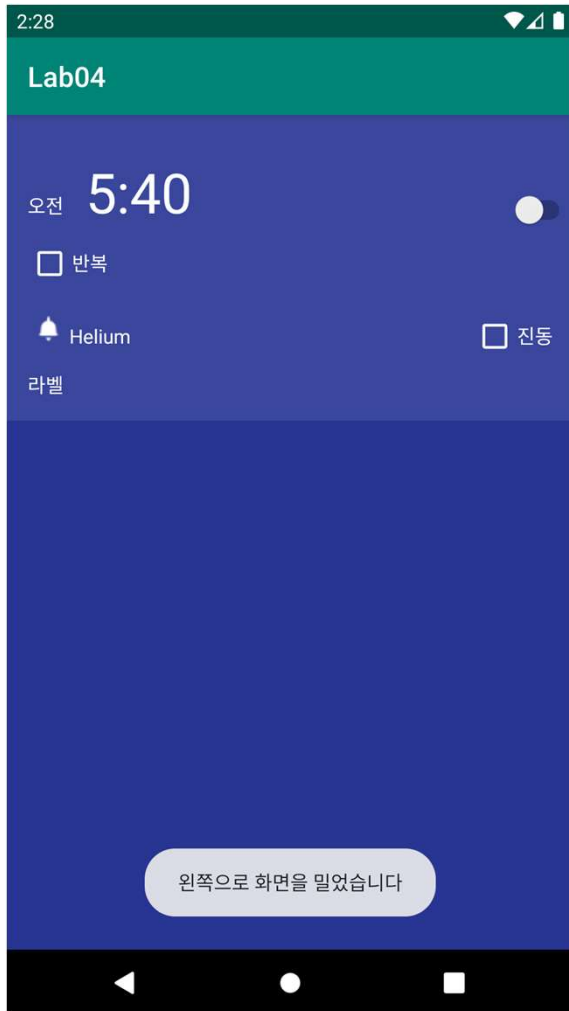
```


@Override

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    if (keyCode == KeyEvent.KEYCODE_BACK) {  
        if (System.currentTimeMillis() - initTime > 3000) {  
            showToast("종료하려면 한번 더 누르세요.");  
            initTime = System.currentTimeMillis();  
        } else {  
            finish();  
        }  
        return true;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.

Step 3 _ 결과 실행



Hierarchy Event Model

- 액티비티에서 터치 이벤트와 키 이벤트를 직접 처리하고 싶다면?
 - 이벤트 발생 시 자동 호출되는 함수만 액티비티 내에 재정의하면 됨.

Hierarchy Event Model – 터치 이벤트

- 이벤트가 발생한 뷰 내에서의 좌퓯값
 - getX()
 - getY()
- 화면에서의 좌퓯값
 - getRawX()
 - getRawY()

Hierarchy Event Model – 키 이벤트

- 안드로이드의 소프트 키보드는 키 이벤트로 처리할 수 없음
 - EditText의 `addTextChangedListener()`로 리스너를 등록하여 처리
- 소프트 키보드가 아닌 하드웨어 키보드가 제공되는 스마트폰이라면?
 - 해당 하드웨어 키보드의 키가 눌렸을 때 키 이벤트로 처리할 수 있음
- 홈과 전원, 오버뷰(Overview) 버튼은 일반 애플리케이션에서 이벤트 처리로 제어할 수 없음
 - 따라서 앱에서 키 이벤트 처리는 뒤로가기 버튼 처리가 대부분
 - 뒤로가기 버튼 이벤트를 처리할 수 있는 **Activity**의 메서드
 - `onKeyDown()`
 - `onBackPressed()` : 뒤로가기 버튼 제어만을 목적으로 만들어졌음

