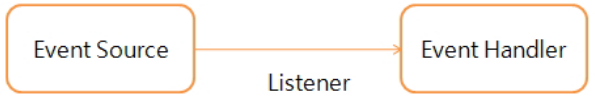
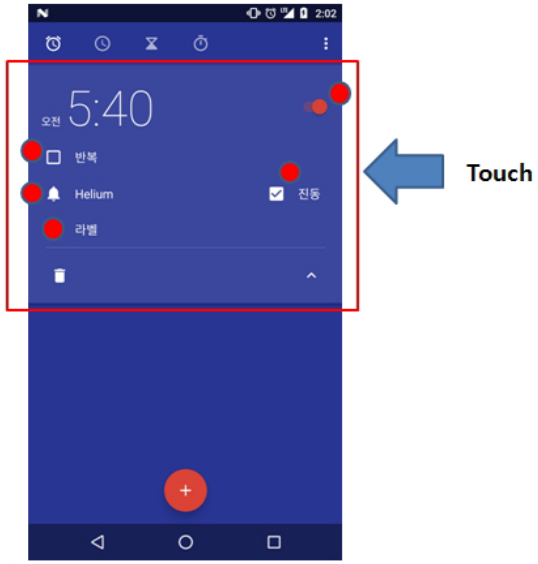
6장. 사용자 이벤트 처리

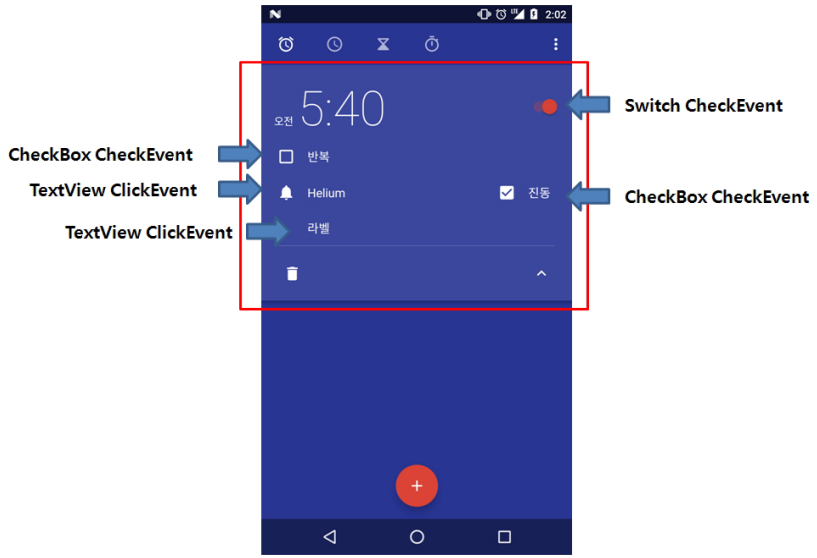
6.1 Delegation Event Model

6.1.1. 이벤트 프로그램 구조



- 이벤트 소스(Event Source): 이벤트가 발생한 뷰 객체
- 이벤트 핸들러(Event Handler): 이벤트 처리 내용을 가지는 객체
- 리스너(Listener): 이벤트 소스와 이벤트 핸들러를 연결하는 작업





6.1 Delegation Event Model


```
vibrateCheckView.setOnCheckedChangeListener(new MyEventHandler());
```

```
class MyEventHandler implements CompoundButton.OnCheckedChangeListener{
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        // 이벤트 처리 로직 작성
    }
}
```

1. 객체에 Event 발생

```
vibrateCheckView.setOnCheckedChangeListener(new MyEventHandler());
```

2. Listener로 등록된 EventHandler 의 함수 실행

```
class MyEventHandler implements CompoundButton.OnCheckedChangeListener{
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

    }
}
```

6.1 Delegation Event Model

6.1.2. 다양한 이벤트 처리

표 6-1 주요 이벤트

Event	설명
OnClickListener	뷰 클릭 시 발생하는 이벤트
OnLongClickListener	뷰를 오래 클릭했을 때 발생하는 이벤트
OnCheckedChangeListener	CheckBox의 상태 변경 이벤트
OnItemClickListener	ListView의 항목 선택 이벤트
OnDateSetListener	DatePicker의 날짜 선택 이벤트
OnTimeSetListener	TimePicker의 시간 선택 이벤트

OnClickListener

```
btn.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {

    }
});
```

6.1 Delegation Event Model

- OnLongClickListener

```
btn.setOnLongClickListener(new View.OnLongClickListener(){
    @Override
    public boolean onLongClick(View v) {
        return false;
    }
});
```

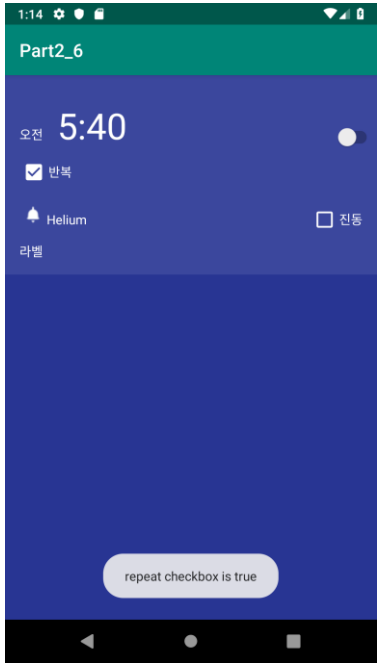
- OnCheckedChangeListener

```
checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

    }
});
```

Step by Step 6-1 – Delegation Event

- 모듈 생성
- 파일 복사
- 이벤트 리스너 인터페이스 선언
- onClick 함수 구현
- onCheckedChanged 함수 구현
- 실행



6.2 Hierarchy Event Model

6.2.1. 터치 이벤트

```
public boolean onTouchEvent(MotionEvent event) {
    return super.onTouchEvent(event);
}
```

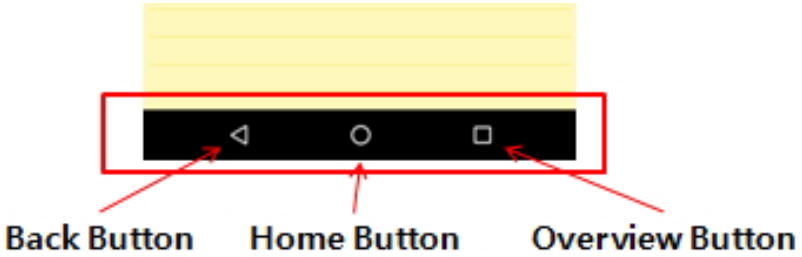
- ACTION_DOWN: 화면에 터치된 순간의 이벤트
- ACTION_UP: 터치를 떼는 순간의 이벤트
- ACTION_MOVE: 터치한 후 이동하는 순간의 이벤트

- getX()
- getY()
- getRawX()
- getRawY()

```
public boolean onTouchEvent(MotionEvent event) {
    if(event.getAction()==MotionEvent.ACTION_DOWN){
        initX=event.getRawX();
    }
    return true;
}
```

6.2 Hierarchy Event Model

6.2.2. 키 이벤트



- onKeyDown: 키가 눌린 순간의 이벤트
- onKeyUp: 키를 떼는 순간의 이벤트
- onKeyLongPress: 키를 오래 누르는 순간의 이벤트

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if(keyCode==KeyEvent.KEYCODE_BACK){

    }
    return super.onKeyDown(keyCode, event);
}
```

```
public void onBackPressed() {
    super.onBackPressed();
}
```

Step by Step 6-2 – 터치, 키 이벤트

- onTouchEvent() 함수 추가
- onKeyDown() 함수 추가
- 결과 실행