

BÀI 3: XỬ LÝ SỰ KIỆN

I. Tóm tắt nội dung thực hành

1. Yêu cầu lý thuyết

Xử lý sự kiện (Event Handling) là một khái niệm quan trọng trong lập trình Android Java. Sự kiện là những hành động của người dùng hoặc hệ thống như nhấn nút, chạm màn hình, nhận dữ liệu từ mạng,... Để ứng dụng của phản hồi với những sự kiện này, cần phải xử lý chúng. Dưới đây là một số kiến thức cơ bản về xử lý sự kiện trong Android Java:

1. Listener: Một Listener (Bộ nghe sự kiện) là một đối tượng lớp cài đặt giao diện (interface) đặc biệt để lắng nghe và phản hồi với một loại sự kiện cụ thể. Ví dụ: `OnClickListener` để lắng nghe sự kiện nhấn nút, `TouchListener` để lắng nghe sự kiện chạm màn hình.

2. Đăng ký Listener: Để xử lý sự kiện, phải đăng ký một đối tượng Listener với View (ví dụ: Button, TextView, ImageView) mà muốn theo dõi sự kiện. Điều này thường được thực hiện trong phương thức `onCreate()` của Activity hoặc Fragment.

```
Button myButton = findViewById(R.id.myButton);
myButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Xử lý sự kiện nhấn nút tại đây
    }
});
```

3. Callback Method: Khi sự kiện xảy ra, hệ thống Android sẽ gọi phương thức callback tương ứng trong Listener đã đăng ký. Ví dụ: `onClick()` cho sự kiện nhấn nút, `onTouch()` cho sự kiện chạm màn hình.

4. Xử lý sự kiện trong Callback Method: viết mã để xử lý sự kiện trong phương thức callback của Listener. Điều này có thể bao gồm cập nhật giao diện người dùng, thực hiện một tác vụ, di chuyển đến một Activity khác, ...

5. Hủy đăng ký Listener: Trong một số trường hợp, cần hủy đăng ký Listener để tránh rò rỉ bộ nhớ hoặc khi View không còn cần lắng nghe sự kiện nữa. Điều này thường được thực hiện trong phương thức `onDestroy()` của Activity hoặc Fragment.

Ngoài ra, Android cung cấp các sự kiện hệ thống khác như sự kiện vòng đời Activity, sự kiện nhận dữ liệu từ mạng, sự kiện thay đổi cấu hình thiết bị, v.v. Và có thể xử lý những sự kiện này bằng cách ghi đè các phương thức vòng đời tương ứng trong Activity hoặc Fragment.

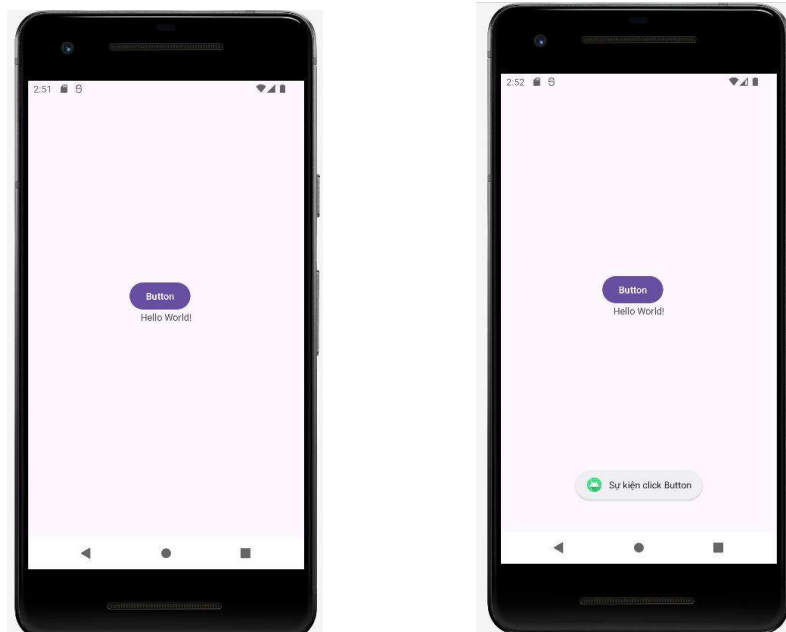
Xử lý sự kiện là một phần quan trọng trong lập trình Android Java để xây dựng ứng dụng phản hồi và tương tác với người dùng. Cần hiểu rõ về các loại sự kiện, cách đăng ký và hủy đăng ký Listener, cũng như cách viết mã xử lý sự kiện trong phương thức callback.

2. Nội dung

Thực hiện ứng dụng Android với xử lý sự kiện cơ bản theo từng bước của hướng dẫn.

II. Bài tập yêu cầu

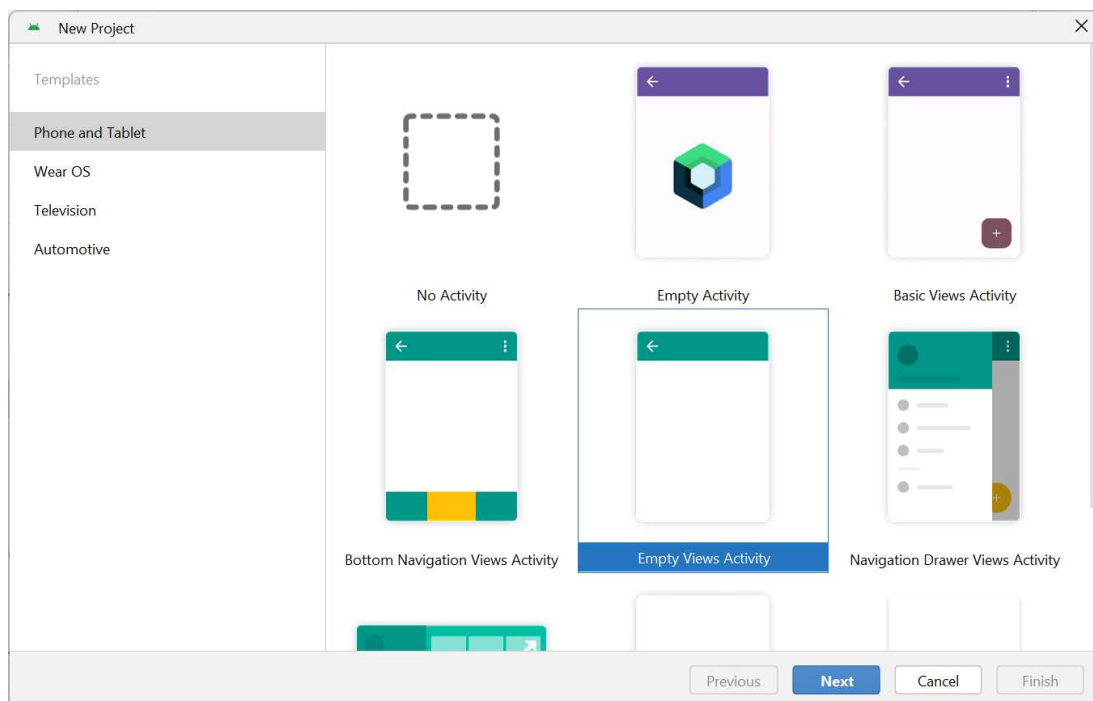
Bài 1: Xây dựng ứng dụng với xử lý sự kiện click Button như hình.



Hình 33 Giao diện bài tập 3.1

Hướng dẫn:

Tạo project mới:



Hình 34 Tạo project mới

Tạo giao diện gồm có Button



Cách 1:

Đăng ký bộ lắng nghe sự kiện khi click Button. Khi click button sẽ hiện ra một thông báo.

```

public class MainActivity extends AppCompatActivity {
    1 usage
    private View.OnClickListener corkyListener = new View.OnClickListener() {
        public void onClick(View v) {
            // do something when the button is clicked
            Toast.makeText(context: MainActivity.this, text: "Sự kiện click Button",
                Toast.LENGTH_SHORT).show();
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Capture our button from layout
        Button button = (Button)findViewById(R.id.btn);
        /* Register the onClick listener with the implementation above */
        button.setOnClickListener(corkyListener);
    }
}

```

Cách 2:

Ta cũng có thể triển khai OnClickListener trong Activity và override lại hàm onClick của interface này như sau:

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    public void onClick(View v) {
        // do something when the button is clicked
        Toast.makeText(context: MainActivity.this, text: "Sự kiện click Button",
            Toast.LENGTH_SHORT).show();
    }

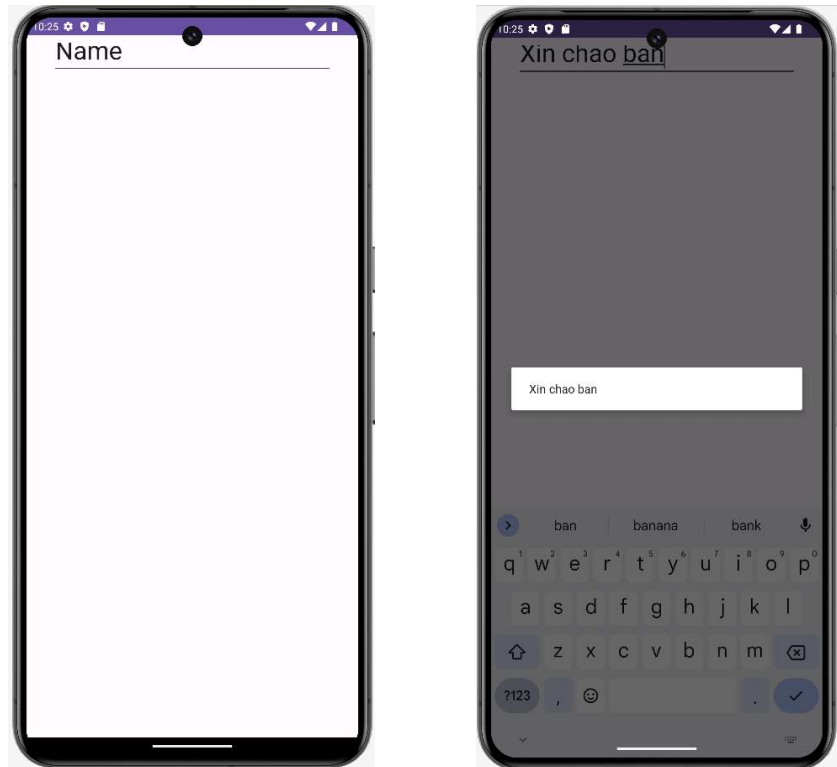
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Capture our button from layout
        Button button = (Button)findViewById(R.id.btn);
        /* Register the onClick listener with the implementation above */
        button.setOnClickListener(this);
    }
}

```

Chạy chương trình và xem kết quả.

Bài 2: Xây dựng ứng dụng xử lý sự kiện như hình. Ứng dụng có 1 EditText cho người dùng gõ vào văn bản. Khi người dùng nhấn phím Enter sẽ xuất ra thông báo là đoạn văn bản vừa gõ.



Hình 35 Giao diện bài tập 3.2

Hướng dẫn:

-Xây dựng giao diện:

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

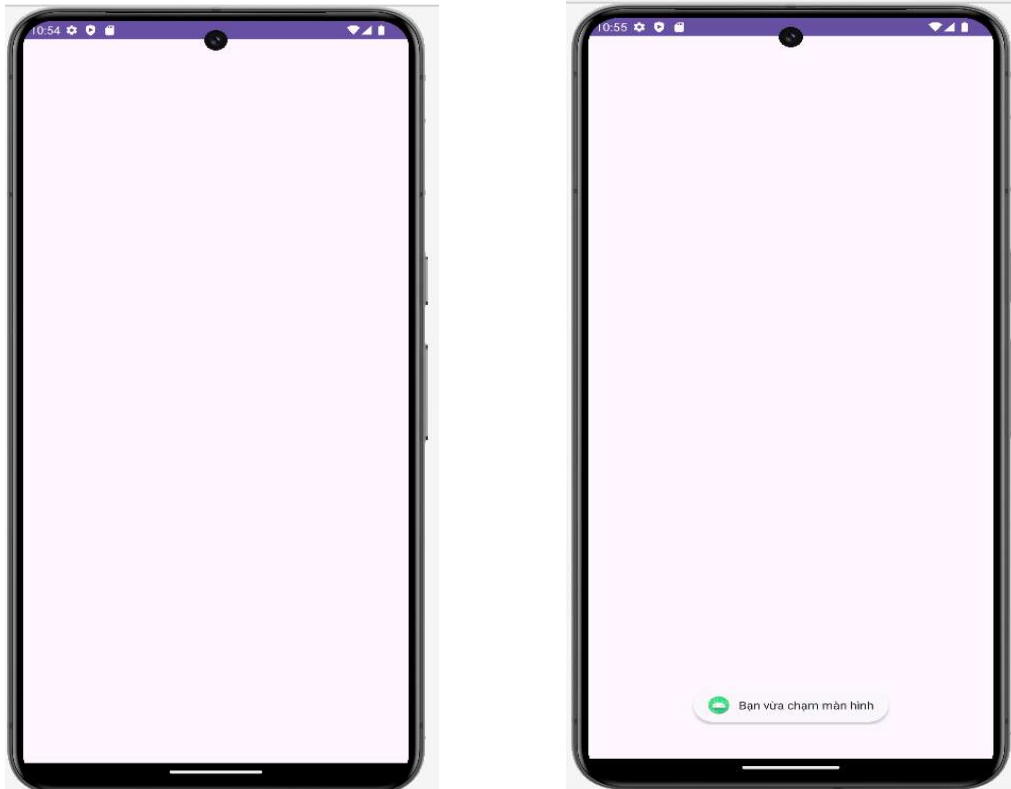
    <EditText
        android:id="@+id/inputText"
        android:layout_width="378dp"
        android:layout_height="53dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:inputType="text"
        android:text="Name"
        android:textSize="34sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="16dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

-Viết xử lý cho nút giữa thông qua sự kiện onKeyDown của EditText

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final EditText editText = (EditText) findViewById(R.id.inputText);  
  
        editText.setOnKeyListener(new View.OnKeyListener() {  
            @Override  
            public boolean onKeyDown(View v, int keyCode, KeyEvent event) {  
                if (event.getAction() == KeyEvent.ACTION_DOWN && keyCode == KeyEvent.KEYCODE_ENTER) {  
                    String message = editText.getText().toString();  
                    Toast.makeText(context: MainActivity.this, message, Toast.LENGTH_SHORT).show();  
                    return true;  
                }  
                return false;  
            }  
        });  
    }  
}
```

Bài 3: Xây dựng ứng dụng xử lý sự kiện Touch trên layout như hình. Khi người dùng chạm vào màn hình, sẽ hiện thông báo.



Hình 36 Giao diện bài tập 3.3

Hướng dẫn:

Tạo project và giao diện có 1 Layout tên view1

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!--View (Sub-Class) where onTouchListener is implemented-->
    <LinearLayout
        android:id="@+id/view1"
        android:layout_width="300sp"
        android:layout_height="400sp"
        android:layout_centerInParent="true"
        android:orientation="horizontal">

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Trong code Java, viết hàm xử lý sự kiện onTouch trên View

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        View v1 = findViewById(R.id.view1);  
        v1.setOnTouchListener(new View.OnTouchListener() {  
            @Override  
            public boolean onTouch(View v, MotionEvent event) {  
                if (event.getAction() == MotionEvent.ACTION_DOWN)  
                    Toast.makeText(context: MainActivity.this, text: "Bạn vừa chạm màn hình",  
                        Toast.LENGTH_SHORT).show();  
                return false;  
            }  
        });  
    }  
}
```

Chạy chương trình và kiểm tra kết quả.

III. Câu hỏi ôn tập:

1. Phương thức nào được sử dụng để thiết lập một sự kiện onClick trên một Button?
 - A. setOnClickListener()
 - B. setClickListener()
 - C. registerClickListener()
2. Để xử lý sự kiện onClick trên một Button, chúng ta cần thực hiện những bước nào?
 - A. Thiết lập sự kiện onClick trên Button, và định nghĩa phương thức xử lý sự kiện
 - B. Chỉ cần thiết lập sự kiện onClick trên Button
 - C. Chỉ cần định nghĩa phương thức xử lý sự kiện
3. Phương thức nào được sử dụng để thiết lập một sự kiện onTouch trên một View?
 - A. setOnTouchListener()
 - B. setTouchListener()
 - C. registerTouchListener()

4. Khi sử dụng sự kiện onTouch, các thông tin nào có thể được lấy từ đối tượng MotionEvent?
- A. Vị trí điểm chạm
 - B. Loại sự kiện chạm (touch event)
 - C. Cả A và B đều đúng
5. Phương thức nào được sử dụng để thiết lập một sự kiện onKeyDown trên một View?
- A. setOnKeyListener()
 - B. setKeyListener()
 - C. registerKeyListener()
6. Khi sử dụng sự kiện onKeyDown, các thông tin nào có thể được lấy từ đối tượng KeyEvent?
- A. Mã phím được bấm
 - B. Trạng thái phím (pressed, released, etc.)
 - C. Cả A và B đều đúng
7. Phương thức nào được sử dụng để thiết lập một sự kiện onFocus trên một View?
- A. setOnFocusChangeListener()
 - B. setFocusListener()
 - C. registerFocusListener()