









ANDROID

## 2. Các kiểu lập trình sự kiện

-  2.1 sử dụng thuộc tính Onclick trong file .XML
-  2.2 Lớp ngàm định trong phương thức của nút lệnh (Inline anonymous listener)
-  2.3 Cụ thể hóa phương thức onclick trong Activity
-  2.4 Biến xử lý sự kiện -Listener in variable
-  2.5 Lớp xử lý sự kiện Explicit Listener Class
-  2.6 Lớp View subclassing?



ANDROID

## 2.1 sự kiện Onclick trong file giao diện XML

Sử dụng sự kiện onClick (android:onClick) trong file xml

```
<Button
    android:id="@+id/btnOK"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/editUserName"
    android:onClick="btnOk_click"
    android:text="@string/idstrok" />
```

```
public void btnOk_click(View v)
{
    Toast.makeText(this,
        "You click ok",
        Toast.LENGTH_LONG)
        .show();
}
```

Viết hàm  
xử lý trong  
file .java



## 2.2 Sử dụng sự kiện ngầm định- Inline anonymous listener

- Định nghĩa sự kiện và đưa vào phương thức `setOnClickListener` của nút lệnh trong file `.java`

Bước 1: lấy nút lệnh

Bước 2: đặt sự kiện ngầm định cho nút lệnh.

Bước 3: viết nội dung cho hàm xử lý trong `onClick(...)`

```
Button btnCancel=(Button) findViewById(R.id.btnCancel);  
btnCancel.setOnClickListener(new OnClickListener() {  
    public void onClick(View arg0) {  
        //Close this activity  
        finish();  
    }  
});
```



## 2.3 Cụ thể hóa phương thức onClick trong giao tiếp OnClickListener

Thêm phương thức cụ thể kế thừa giao diện ở lớp mô tả

```
public class LoginExampleImplements extends Activity implements
OnClickListener {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...

        // Set Click Listener
        btnLogin.setOnClickListener(this);
        btnCancel.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if(v==btnLogin) {
            // Check Login

        } else if(v==btnCancel) {
            // Close the application
            finish();
        }
    }
}
```



## 2.4 Sử dụng biến nhận sự kiện -Listener in variable

```
public class LoginExampleVariableImplementation extends Activity {  
    ...
```

```
    OnClickListener myClickListener = new OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            if(v==btnLogin) {  
                // Check Login  
            } else if(v==btnCancel) {  
                // Close the application  
                finish();  
            }  
        }  
    };
```

- Chia sẻ chung biến sự kiện cho các điều khiển khác nhau

```
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        ...  
        // Set Click Listener  
        btnLogin.setOnClickListener(myClickListener);  
        btnCancel.setOnClickListener(myClickListener);  
    }
```



## 2.5 Sử dụng lớp nhận sự kiện- Explicit Listener Class

```
public class FsdFActivity extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        Button btn1= (Button) findViewById(R.id.button1);  
        btn1.setOnClickListener(new button1());  
    }  
  
    class button1 implements View.OnClickListener  
    {  
        public void onClick(View v) {  
            // your code here  
        }  
    }  
}
```

Tách riêng một lớp đóng vai trò là lớp sự kiện riêng.



## Phân loại các phương pháp xử lý sự kiện

1. Onclick trong file .xml
2. - Xử lý sự kiện trong lớp ngầm định (Inline anonymous class) của nút lệnh  
Hoặc cụ thể hóa phương thức onclick trong interface OnClickListener.
3. Biến nhận sự kiện (Variable as listener) hoặc  
-Lớp xử lý sự kiện (Explicit Listener class)

**So sánh ưu, nhược điểm của 3 phương pháp trên?**

# So sánh



ANDROID

## Xử lý inline (trong .xml)

- Ưu điểm: đơn giản, nhỏ gọn, dễ triển khai
- Nhược điểm: không linh động, không sử dụng lại được và bảo trì không tiện lợi

## Cụ thể hóa phương thức onclick trong lớp OnClickListener, sự kiện ngầm của nút lệnh

- Ưu điểm:
  - + các phương thức xử lý có thể được sử dụng lại trong các tiện ích khác nhau.
  - + Mã nguồn ở trong cùng 1 file .java
  - + Mã nguồn có thể dùng để tạo ra phương thức xử lý sự kiện cho hành động tương tự
- Nhược điểm:
  - + mã nguồn không sáng sủa vì các phương thức ngầm định được xử lý lồng nhau.
  - + Nếu các hành động xử lý nhiều tình huống khác nhau phải dùng đến nhiều if/else/if thì người đọc mã nguồn gặp khó khăn
  - + Chỉ nên triển khai 1 sự kiện cho 1 lớp

## Biến nhận sự kiện, lớp xử lý sự kiện

- Ưu điểm:
  - + có thể được sử dụng lại
  - + có thể xử lý sự kiện cho các điều khiển cùng loại trong cùng lớp
  - + Mã nguồn được xử lý trong cùng 1 lớp/biến vậy nên mã nguồn sáng sủa, dễ đọc –readable)
- Nhược điểm:
  - Nếu có quá nhiều lớp, biến, hàm cùng xử lý sự kiện thì mã nguồn sẽ rất phức tạp.





## Demo

1. Onclick xml: Bảng tính toán đơn giản
2. Lớp ngầm định (Inline anonymous class) Chuyển đổi năm dương lịch
  - Cụ thể hóa phương thức onclick trong interface OnClickListener – chỉ số khối lượng cơ thể BMI
3. Biến nhận sự kiện (Variable as listener) – đổi độ C sang độ F
  - Lớp xử lý sự kiện (Explicit Listener class)- giải phương trình bậc 2