

ImageButton, ImageView



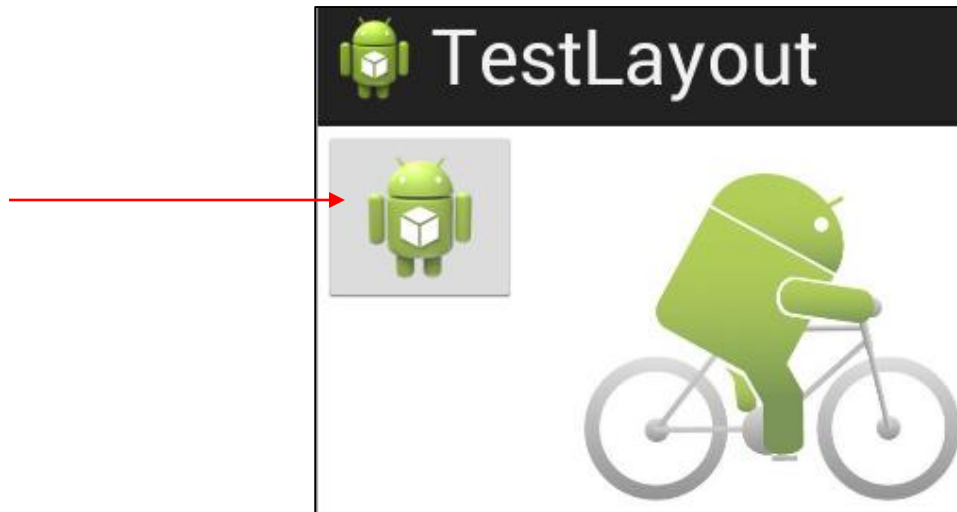


Nội dung bài học

- ❖ ImageButton
- ❖ ImageView

ImageButton

- ❖ ImageButton là 1 Button có thể chứa hình ảnh, có thể đặt prefix **btn**, **imgBtn**



ImageView

- ❖ ImageView là control/view dùng để hiển thị hình ảnh, cho phép tương tác. Có thể thay thế luôn cho ImageButton bằng cách design hình ảnh dạng 3D cho đẹp



ImageView

❖ **ImageView** là loại View dùng để hiển thị tài nguyên hình ảnh như các ảnh Bitmap, các ảnh Drawable. Nó cũng cung cấp các chức năng tùy biến khác nhau như đổ màu nhuộm (tint) vào ảnh, co kéo/cắt ảnh khi hiển thị trên View.

```
<ImageView  
    android:id="@+id/image"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@mipmap/ic_launcher"  
/>
```

Thuộc tính cơ bản của ImageView thường dùng:

Các thuộc tính/sự kiện	Mô tả
android:id	Id của ImageView
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:src	Là thuộc tính chứa hình ảnh cần hiển thị.
android:background	Thuộc tính này xác định màu nền cho ImageView .
android:padding	Thuộc tính này xác định khoảng cách từ đường viền của ImageView với nội dung nó chứa: left, right, top or bottom. - paddingRight: thiết khoảng cách bên phải của ImageView - paddingLeft: thiết khoảng cách bên trái của ImageView.

Thay đổi tỷ lệ căn chỉnh ảnh trong ImageView

Thuộc tính **android:scaleType** dùng để thiết lập thu phóng ảnh, nhận các giá trị như: *fitXY*, *center*, *fitXY* ...,

ví dụ như **android:scaleType="center"**

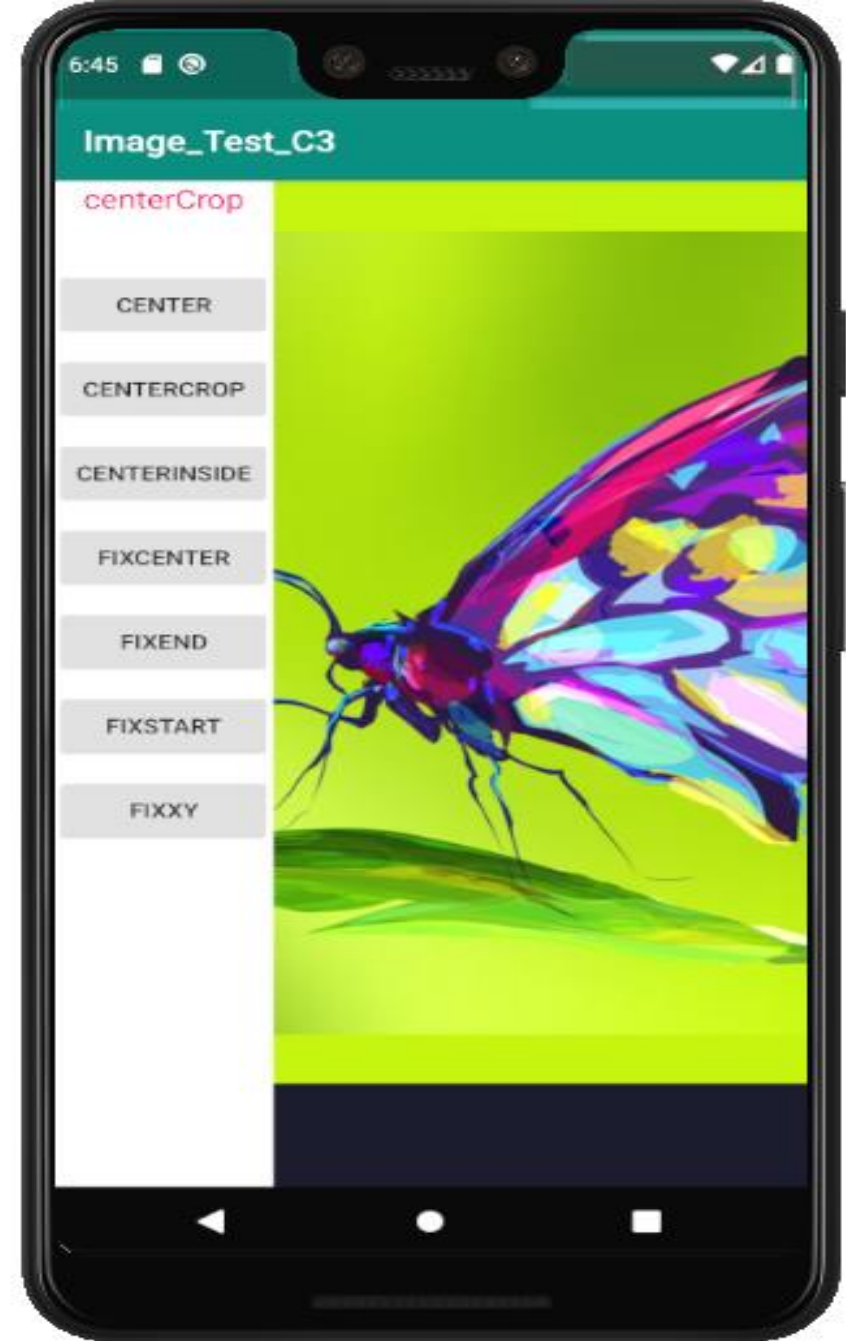
hoặc code Java:

```
imageView.setScaleType(ImageView.ScaleType.CENTER);
```

Các hằng thuộc tính scaleType:

Giá trị	Sử dụng
center	ImageView.ScaleType.CENTER : Đặt ảnh vào giữa ImageView, không có thay đổi tỷ lệ ảnh.
centerCrop	ImageView.ScaleType.CENTER_CROP: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho ảnh phủ kín hết cả ImageView (phần thừa bị cắt)
centerInside	ImageView.ScaleType.CENTER_INSIDE: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiện thị trên ImageView.
fitCenter	ImageView.ScaleType.FIT_CENTER: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiện thị trên ImageView.
fitEnd fitStart	ImageView.ScaleType.FIT_END, ImageView.ScaleType.FIT_START: Co ảnh vừa View, vị trí ảnh ở cuối (ở đầu) ImageView
fitXY	ImageView.ScaleType.FIT_XY: Co ảnh vừa khít cả chiều rộng và cao.

Ví dụ 1: Thiết kế ứng dụng sau:



```
private void addControls() {  
    btncenter = this.<Button>findViewById(R.id.btncenter);  
    btncenterCrop = this.<Button>findViewById(R.id.btncenterCrop);  
    btncenterInside = this.<Button>findViewById(R.id.btncenterInside);  
    btnfixCenter = this.<Button>findViewById(R.id.btnfixCenter);  
    btnfixEnd = this.<Button>findViewById(R.id.btnfixEnd);  
    btnfixStart = this.<Button>findViewById(R.id.btnfixStart);  
    btnfixXY = this.<Button>findViewById(R.id.btnfixXY);  
    txtKQ = this.<TextView>findViewById(R.id.txtKQ);  
    imgHinh = this.<ImageView>findViewById(R.id.imgHinh);  
}
```

Sử dụng Activity As Listener

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    TextView txtKQ;
    Button
btncenter,btncenterCrop,btncenterInside,btnfixCenter,btnfixEnd,btnfixStart,btnfixXY;
    ImageView imgHinh;

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

    private void addEvents() {
        btnfixXY.setOnClickListener(this);
        btnfixStart.setOnClickListener(this);
        btnfixEnd.setOnClickListener(this);
        btncenterInside.setOnClickListener(this);
        btncenterCrop.setOnClickListener(this);
        btnfixCenter.setOnClickListener(this);
        btncenter.setOnClickListener(this);
    }
}
```

```
@Override
public void onClick(View v) {
    ImageView.ScaleType scaletype = ImageView.ScaleType.CENTER;
    switch (v.getId())
    {
        case R.id.btncenter:
            scaletype = ImageView.ScaleType.CENTER;
            break;
        case R.id.btncenterCrop:
            scaletype = ImageView.ScaleType.CENTER_CROP;
            break;
        case R.id.btncenterInside:
            scaletype = ImageView.ScaleType.CENTER_INSIDE;
            break;
        case R.id.btnfixCenter:
            scaletype = ImageView.ScaleType.FIT_CENTER;
            break;
        case R.id.btnfixEnd:
            scaletype = ImageView.ScaleType.FIT_END;
            break;
        case R.id.btnfixStart:
            scaletype = ImageView.ScaleType.FIT_START;
            break;
        case R.id.btnfixXY:
            scaletype = ImageView.ScaleType.FIT_XY;
            break;
    }

    imgHinh.setScaleType(scaletype);
    txtKQ.setText( ((Button)v).getText() );
}
```

AlertDialog



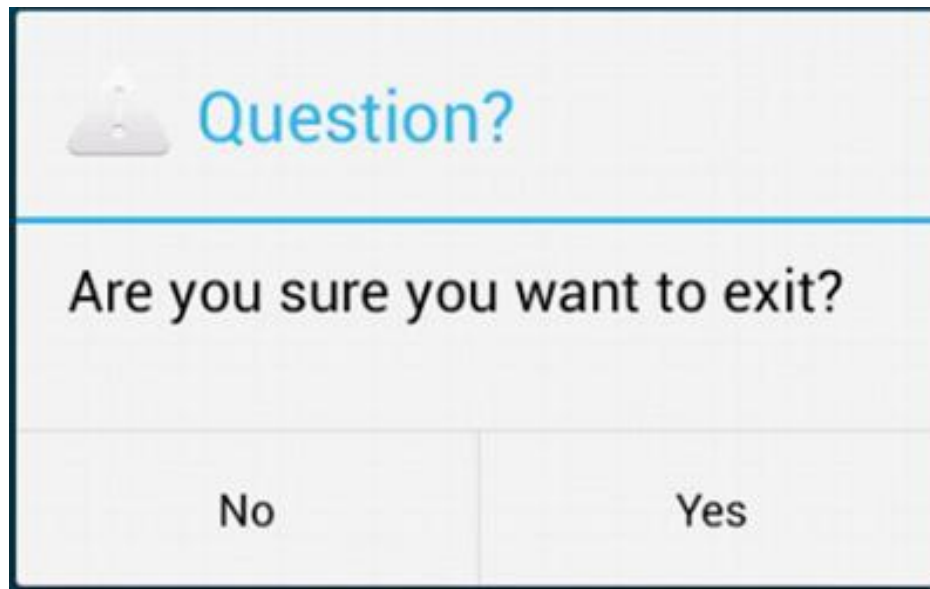


Nội dung bài học

- ❖ Giới thiệu AlertDialog
- ❖ Các phương thức thường dùng
- ❖ Xử lý sự kiện trên AlertDialog

Giới thiệu AlertDialog

- Hiển thị cửa sổ và cho phép người dùng tương tác



AlertDialog.Builder b=new AlertDialog.Builder(YourActivity.this);

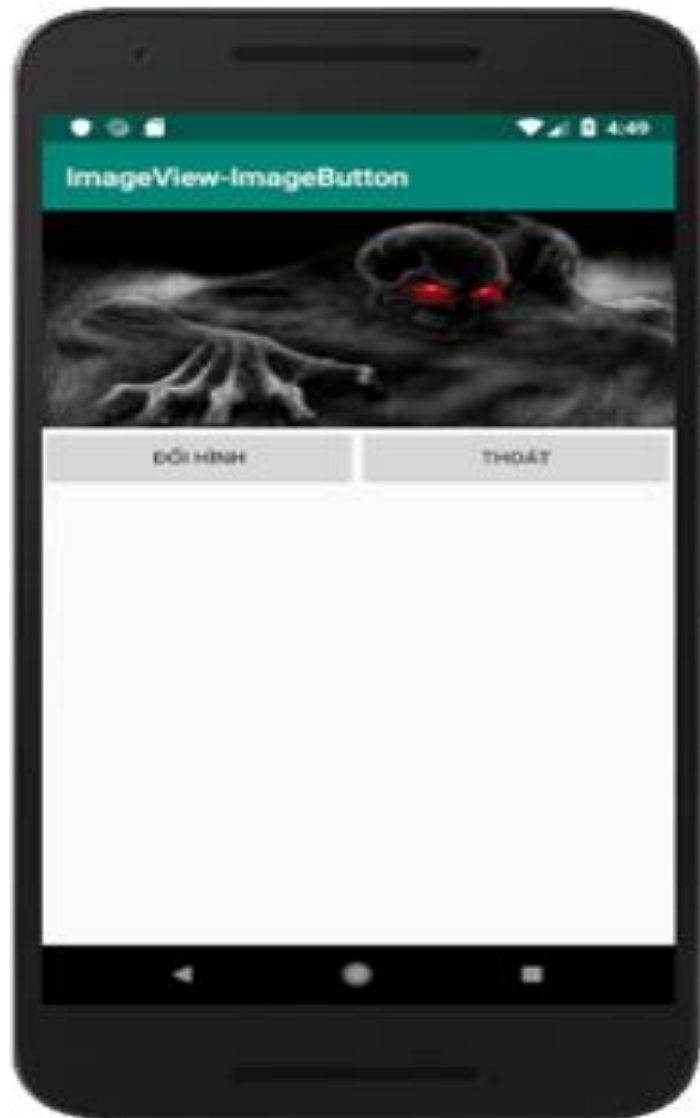


Các phương thức thường dùng

- **setTitle** : thiết lập tiêu đề cho Dialog
- **setMessage**: Thiết lập nội dung cho Dialog
- **setIcon** : để thiết lập Icon
- **setPositiveButton, setNegativeButton** thiết lập hiển thị Nút chọn cho Dialog
- **create()** để tạo Dialog
- **show()** để hiển thị Dialog.

Các phương thức thường dùng

```
AlertDialog.Builder b=new AlertDialog.Builder(YourActivity.this);  
b.setTitle("Question");  
b.setMessage("Are you sure you want to exit?");  
b.setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which){  
        finish();  
    }  
});  
b.setNegativeButton("No", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which){  
        dialog.cancel(); }  
});  
b.create().show();
```



Hình 1: Ban đầu



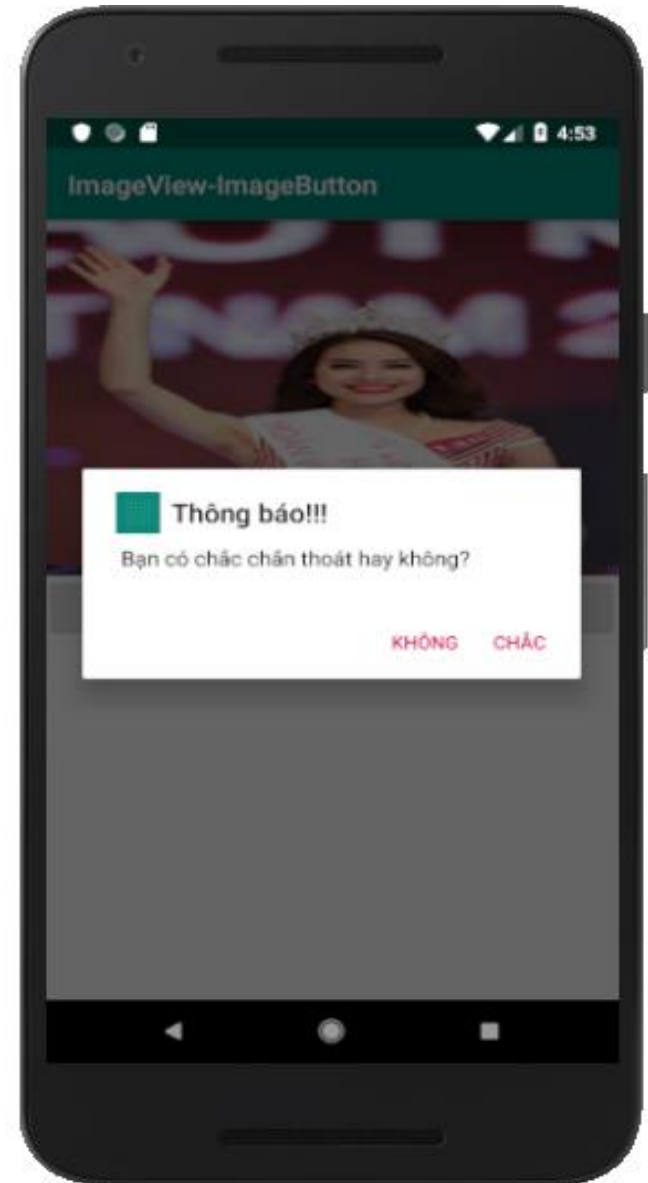
Hình 2: Khi nhấn Đổi Hình

Khi nhấn vào chức năng đổi hình sẽ biến đổi

Khi nhấn Thoát sẽ hiển thị thông báo dưới dạng (AlertDialog)

Khi nhấn vào Button “Không” thì sẽ quay lại chương trình.

Khi nhấn vào Button “Chắc” thì sẽ thoát luôn chương



Khai báo các View trong phương thức addControls()

```
private void addControls() {  
    imgHinh = this.<~>findViewById(R.id.imgHinh);  
  
    btnDoiHinh = this.<~>findViewById(R.id.btnDoiHinh);  
    btnThoat = this.<~>findViewById(R.id.btnThoat);  
    imgHinh.setImageResource(R.drawable.ma);  
}
```

Viết phương thức addEvents() theo Anonymous-Listener

```
private void addEvents() {  
    btnDoiHinh.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            xuLyDoiHinh();  
        }  
    });  
    btnThoat.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            xuLyThoat();  
        }  
    });  
}
```

Viết phương thức xuLyDoiHinh()

```
private void xuLyDoiHinh() {  
    imgHinh.setImageResource(R.drawable.huong);  
}
```

```
private void xuLyThoat() {  
    AlertDialog.Builder b = new AlertDialog.Builder( context: MainActivity.this);  
    // thiết lập tiêu đề  
    b.setTitle("Thông báo!!!");  
    // nội dung thông báo  
    b.setMessage(" Bạn có chắc chắn thoát hay không? ");  
    //biểu tượng thông báo  
    b.setIcon(R.drawable.ic_launcher_background);  
    //thiết lập các nút lệnh cho người dùng tương tác  
    b.setPositiveButton( text: "Chắc", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            finish();  
        }  
    });  
  
    b.setNegativeButton( text: "Không", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            dialogInterface.dismiss();  
        }  
    });  
    // tạo cửa sổ Dialog  
    AlertDialog d = b.create();  
    d.setCanceledOnTouchOutside(false);  
    d.show();  
}
```



Gỡ hiện rra

END

