

Bài 04:

GIAO DIỆN NGƯỜI DÙNG VÀ

XỬ LÝ SỰ KIỆN

GVGD: ThS. Đặng Thế Hân

Biên soạn: ThS. Giang Hào Côn

Mục tiêu

Cung cấp cho sinh viên kiến thức cơ bản về các các bố cục (Layouts) màn hình của Android, và các điều khiển cơ bản Checkbox và RadioButton trong ngôn ngữ lập trình ứng dụng trên Android.

- 1) View và ViewGroup
- 2) Các Layout trong Android
- 3) Sử dụng CheckBox
- 4) Sử dụng RadioButton.

4.1/ View & View Group

1) View

- View là một lớp cơ sở của tất cả các UI component trong Android. Ví dụ: lớp EditText được sử dụng để nhập đầu vào từ người dùng trong Android, và là lớp con của View.
- Sau đây là một số lớp con phổ biến của View thường được sử dụng trong các ứng dụng Android.
 - TextView
 - EditText
 - Button
 - CheckBox
 - RadioButton
 - ImageButton
 - Progress Bar
 - Spinner

4.1/ View & ViewGroup

2) ViewGroup

- ViewGroup là một lớp con của View và nó sẽ hoạt động như một lớp cơ sở cho layouts và layouts parameters. ViewGroup sẽ cung cấp một bộ chứa (container) vô hình để chứa các View hoặc ViewGroup khác và để xác định các thuộc tính layout.
 - Linear Layout
 - Relative Layout
 - Table Layout
 - Frame Layout
 - Web View
 - List View
 - Grid View
- Ví dụ: Linear Layout là ViewGroup chứa các UI control như button, textview, v.v. và các layout khác.

4.2/ Các Layouts trong Android

- Layout là thành phần định nghĩa cấu trúc giao diện người dùng hay nói cách khác là thành phần quyết định đến giao diện của một màn hình trong ứng dụng Android. Layout hỗ trợ việc căn chỉnh các widget (Ví dụ: TextView, Button, hay EditText...) như chúng ta thấy trong các ứng dụng Android.
- Tùy và ý muốn và cách bố trí các đối tượng hiển thị mà người code có thể sử dụng và kết hợp các Layout khác nhau. Thông thường thì Linear Layout và Relative Layout là được sử dụng nhiều nhất.

4.2/ Các Layouts trong Android

1). FrameLayout

- FrameLayout là một ViewGroup được sử dụng rất nhiều trong android. Bởi vì nó là ViewGroup đơn giản nhất, và thời gian tính toán của nó để layout ra các view con trong nó là thấp nhất nên performance của ViewGroup này là cao nhất.
- FrameLayout được định nghĩa bắt đầu bởi thẻ <FrameLayout> và thẻ đóng </FrameLayout>. Ở giữa thẻ đóng và thẻ mở chính là các view con của nó.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

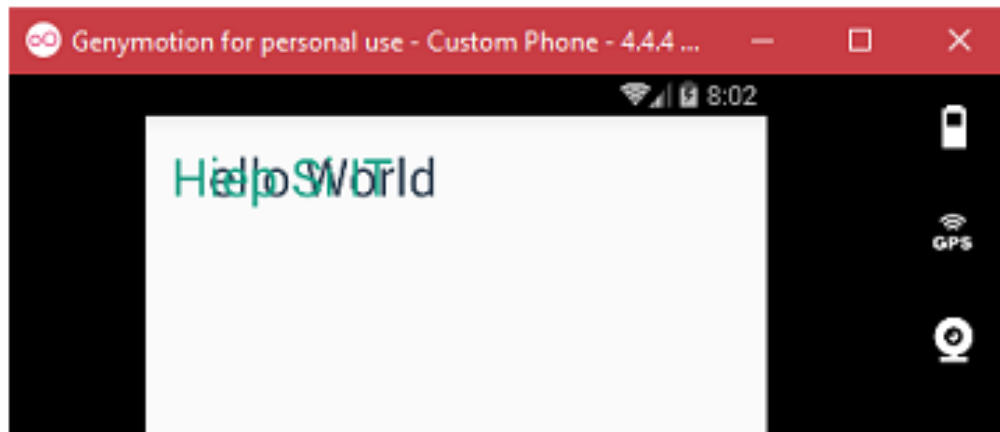
    <!--View Child-->

</FrameLayout>
```

4.2/ Các Layouts trong Android

1). FrameLayout

Quy tắc **FrameLayout** các view con trong **FrameLayout** là các view sẽ nằm chồng lên nhau, view thêm vào sau sẽ nằm đè lên view nằm phía dưới. Ví dụ như sau:



```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="hiepsiit.com.MainActivity" >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World"
    android:textColor="#2c3e50"
    android:textSize="32sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hiệp Sĩ IT"
    android:textColor="#16a085"
    android:textSize="32sp" />
```

```
</FrameLayout>
```


4.2/ Các Layouts trong Android

2). LinearLayout

LinearLayout LinearLayout là ViewGroup được sử dụng khá phổ biến cùng FrameLayout và RelativeLayout, LinearLayout được định nghĩa trong xml bởi cặp thẻ đóng mở `<LinearLayout>` và thẻ đóng `</LinearLayout>`.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!--View child-->

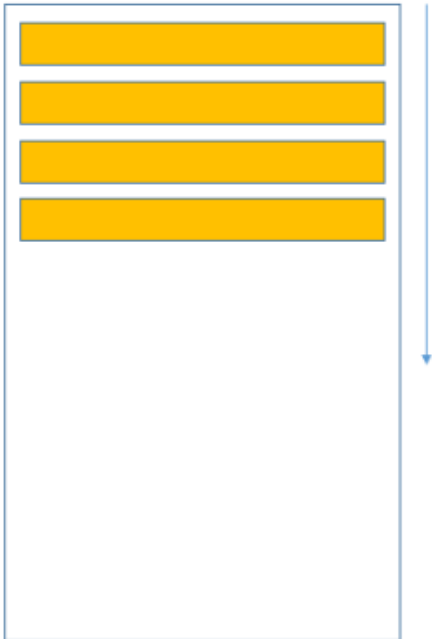
</LinearLayout>
```

4.2/ Các Layouts trong Android

2). LinearLayout

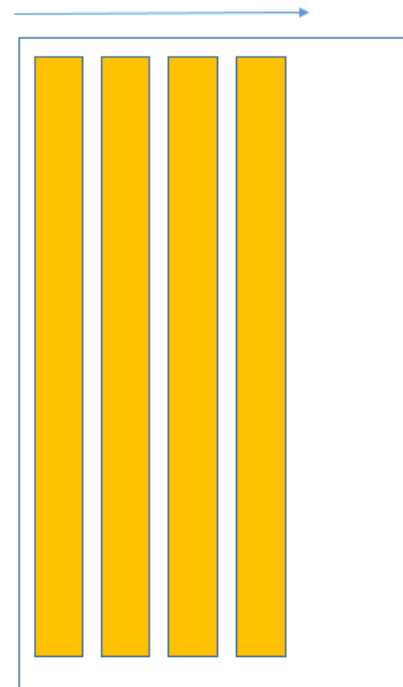
Quy tắc LinearLayout sắp xếp các view con theo hai hướng:

- **Vertical**: Sắp xếp view con theo chiều dọc.



```
android:orientation="vertical"
```

- **Horizontal**: Sắp xếp các view con theo chiều ngang:



```
android:orientation="horizontal"
```

4.2/ Các Layouts trong Android

2). LinearLayout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <View
        android:background="#af5656"
        android:layout_width="match_parent"
        android:layout_height="50dp" />

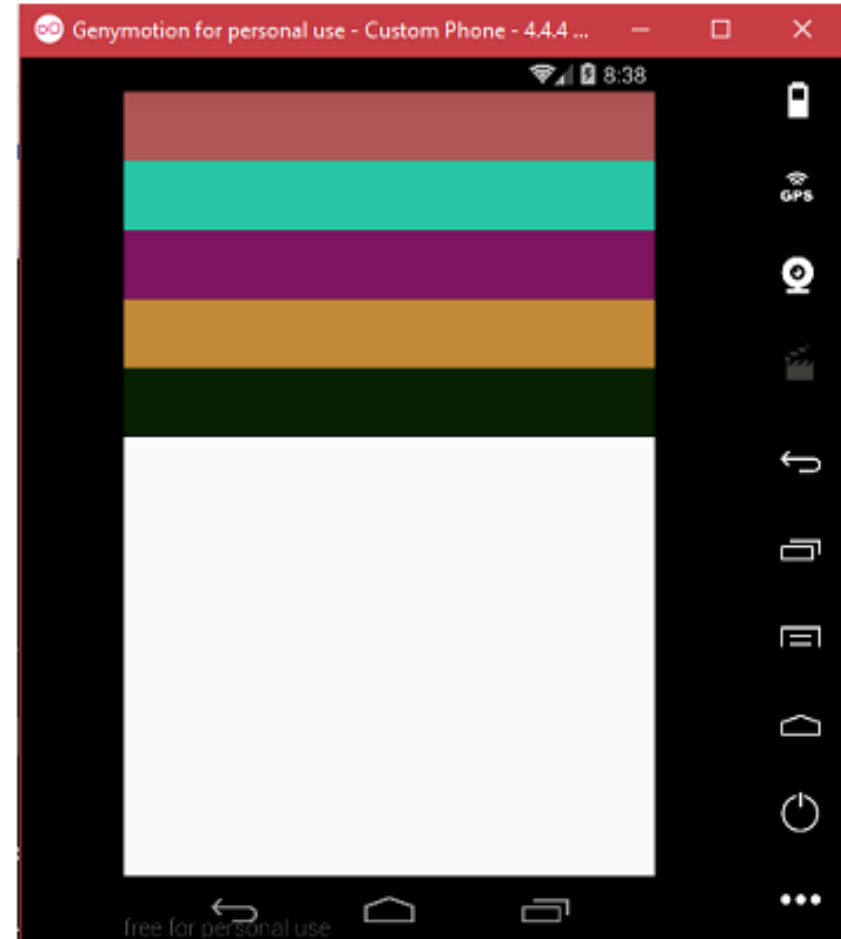
    <View
        android:background="#28c6a6"
        android:layout_width="match_parent"
        android:layout_height="50dp" />

    <View
        android:background="#7c145f"
        android:layout_width="match_parent"
        android:layout_height="50dp" />

    <View
        android:background="#c18936"
        android:layout_width="match_parent"
        android:layout_height="50dp" />

    <View
        android:background="#062001"
        android:layout_width="match_parent"
        android:layout_height="50dp" />

</LinearLayout>
```



4.2/ Các Layouts trong Android

2). LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

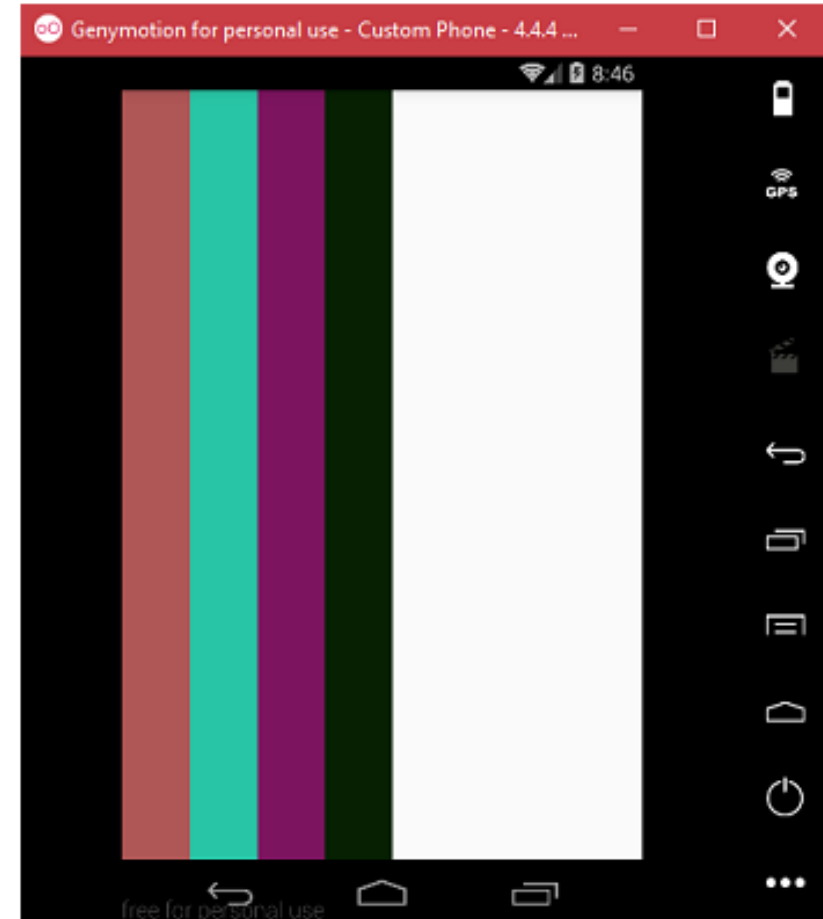
    <View
        android:background="#af5656"
        android:layout_width="50dp"
        android:layout_height="match_parent" />

    <View
        android:background="#28c6a6"
        android:layout_width="50dp"
        android:layout_height="match_parent" />

    <View
        android:background="#7c145f"
        android:layout_width="50dp"
        android:layout_height="match_parent" />

    <View
        android:background="#062001"
        android:layout_width="50dp"
        android:layout_height="match_parent" />

</LinearLayout>
```



4.2/ Các Layouts trong Android

3). RelativeLayout

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).
- RelativeLayout là ViewGroup cũng được sử dụng khá nhiều trong android, RelativeLayout được định nghĩa trong xml bởi cặp thẻ đóng mở **<RelativeLayout>** và thẻ đóng **</RelativeLayout>**.

4.2/ Các Layouts trong Android

3). RelativeLayout

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).
- RelativeLayout là ViewGroup cũng được sử dụng khá nhiều trong android, RelativeLayout được định nghĩa trong xml bởi cặp thẻ đóng mở **<RelativeLayout>** và thẻ đóng **</RelativeLayout>**.

4.2/ Các Layouts trong Android

3). RelativeLayout

Quy tắc của **RelativeLayout** khá giống như **FrameLayout**, nhưng có một số điểm đặc biệt là các view có thể xác định bằng vị trí tương đối (relative) với các view khác thông qua id. Các vị trí tương đối này như sau:

- **android:layout_above="id_name"**: view hiện tại sẽ nằm phía trên view có thuộc tính id là id_name.
- **android:layout_below="id_name"**: view hiện tại sẽ nằm phía dưới view có thuộc tính id là id_name.
- **android:layout_toLeftOf="id_name"**: view hiện tại sẽ nằm bên trái dưới view có thuộc tính id là id_name.
- **android:layout_toRightOf="id_name"**: view hiện tại sẽ nằm phía bên phải view có thuộc tính id là id_name.

4.2/ Các Layouts trong Android

3). RelativeLayout

Quy tắc của **RelativeLayout** khá giống như **FrameLayout**, nhưng có một số điểm đặc biệt là các view có thể xác định bằng vị trí tương đối (relative) với các view khác thông qua id. Các vị trí tương đối này như sau:

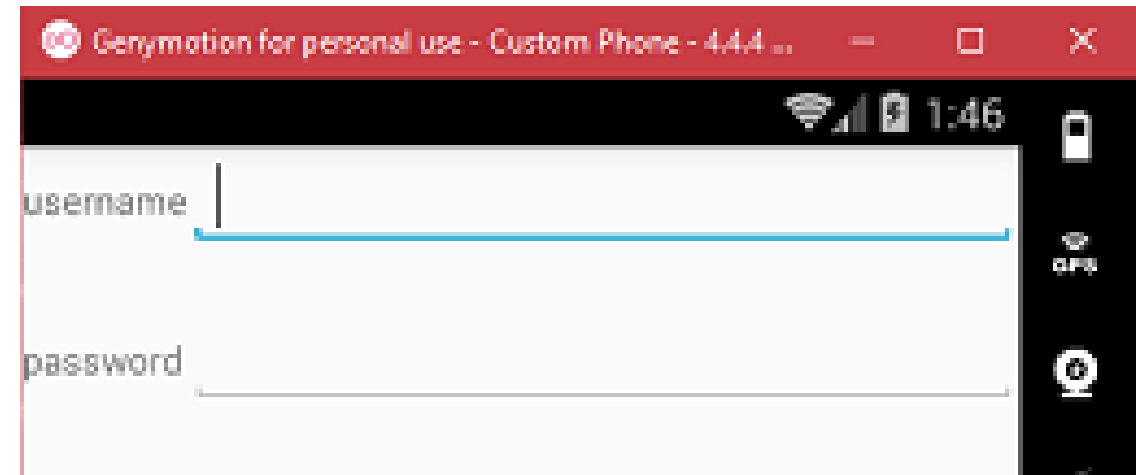
- **android:layout_alignParentBottom="boolean"**: Căn dưới phần tử hiện theo phần tử cha nếu set là true.
- **android:layout_alignParentTop="boolean"**: Căn trên phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout_alignParentRight="boolean"**: Căn phải phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout_alignParentLeft="boolean"**: Căn trái phần tử hiện tại theo phần tử cha nếu set là true..

4.2/ Các Layouts trong Android

3). RelativeLayout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tv_username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        .        android:text="username" />
    .
    .
    .    <EditText
    .        android:id="@+id/edt_username"
    .        android:layout_width="match_parent"
    .        android:layout_height="wrap_content"
    .        android:layout_below="@+id/tv_username" />
    .
    .    <TextView
    .        android:id="@+id/tv_password"
    .        android:layout_width="wrap_content"
    .        android:layout_height="wrap_content"
    .        android:layout_below="@+id/edt_username"
    .        android:text="password" />
    .
    .    <EditText
    .        android:id="@+id/edt_password"
    .        android:layout_width="match_parent"
    .        android:layout_height="wrap_content"
    .        android:layout_below="@+id/tv_password"
    .        android:inputType="textPassword" />
    .
    </RelativeLayout>
```



4.2/ Các Layouts trong Android

4). TableLayout

- Kết hợp với thẻ TableRow để tạo ra các hàng các cột cho layout. Cách sử dụng cũng khá đơn giản. Khi dùng TableLayout, mặc định sẽ tạo ra các đường viền xung quanh các ô. Các ô có thể chứa nội dung là bất kỳ thành phần view nào khác (có thể chứa LinearLayout, FrameLayout hoặc TableLayout cũng được).
- Mỗi thẻ TableRow nằm trong thẻ TableLayout tạo ra một hàng. Thẻ này sẽ thực sự sẽ chứa đựng các thành phần view khác.
- TableLayout sẽ xem dòng nào có số lượng control nhiều nhất để xác định rằng nó có bao nhiêu cột (lấy dòng có số lượng control nhiều nhất làm số cột chuẩn).

4.2/ Các Layouts trong Android

4). TableLayout

Ví dụ 01

0	1		
0	1	2	
0	1	2	3

TableLayout này có 4 cột, 3 dòng.

Ví dụ 02

Label (URL)	EditText	EditText-span	EditText-span
Column 0	Column 1	Column 2 Button Cancel	Column 3 Button OK

android:layout_span="3"

android:layout_columns="2"

4.2/ Các Layouts trong Android

4). TableLayout

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/TableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="#ffffff"
    android:stretchColumns="*"
    tools:context="hiepsiiit.com.MainActivity" >
```

4.2/ Các Layouts trong Android

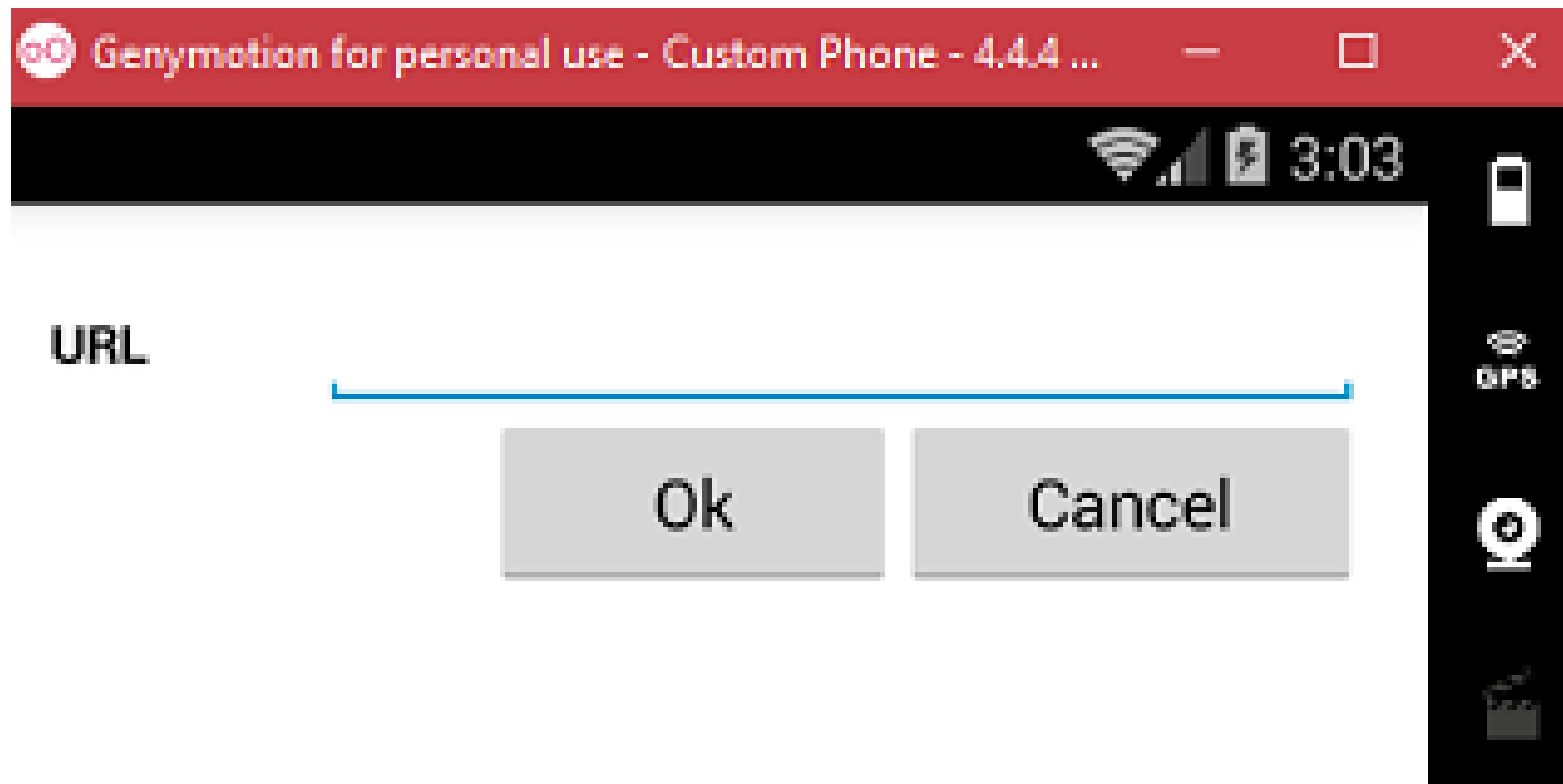
4). TableLayout

```
<TableRow
    android:id="@+id/tableRow1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
    <TextView
        android:id="@+id/tvURL"
        android:text="URL"
        android:textStyle="bold"
        android:textColor="#000000" />
    <EditText
        android:id="@+id/edtURL"
        android:layout_span="3"
        android:textColor="#000000" />
```

```
<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
    <Button
        android:id="@+id/btnOK"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="2"
        android:text="Ok" />
    <Button
        android:id="@+id/btnCancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel" />
</TableRow>
```

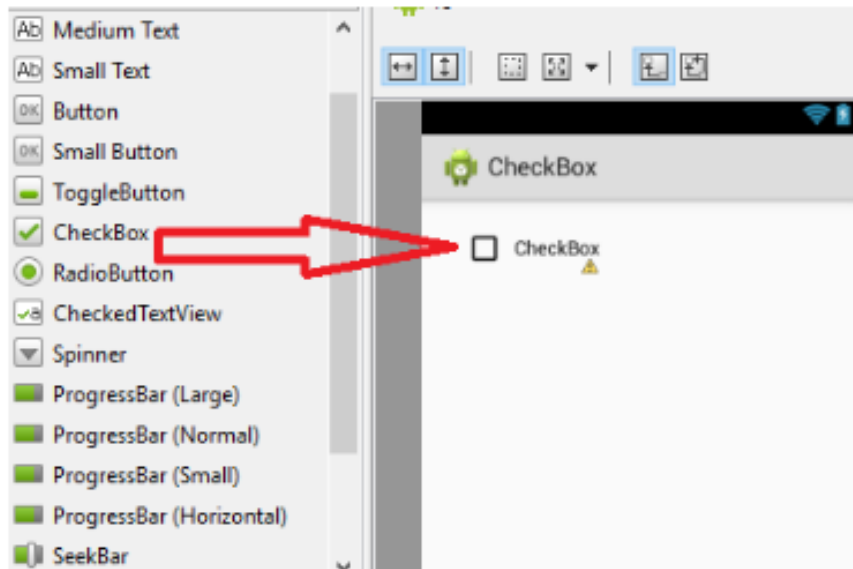
4.2/ Các Layouts trong Android

4). TableLayout



4.3/ Sử dụng CheckBox

- CheckBox là thành phần thể hiện trạng thái chọn (checked) hoặc không chọn (unchecked) CheckBox thường dùng khi người dùng có nhiều lựa chọn và được phép chọn một hoặc nhiều lựa chọn cùng lúc



CheckBox code trong XML

```
<CheckBox  
    android:id="@+id/chksimpleCheckBox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Simple CheckBox"/>
```


4.3/ Sử dụng CheckBox

CheckBox code trong Java

```
//initiate a check box  
CheckBox chksimpleCheckBox = (CheckBox) findViewById(R.id.chksimpleCheckBox);  
  
//check current state of a check box (true or false)  
Boolean checkBoxState = chksimpleCheckBox.isChecked();
```

Thuộc tính thường dùng của CheckBox

```
<CheckBox  
    android:id="@+id/chksimpleCheckBox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Simple CheckBox"  
    android:checked="true"/> <!--set the current state of the check box-->
```


4.3/ Sử dụng CheckBox

Các phương thức của CheckBox

Phương thức	Ý nghĩa
<code>public boolean isChecked()</code>	True nếu CheckBox là checked, ngược lại false
<code>public void setChecked(boolean status)</code>	Thay đổi trạng thái của CheckBox

4.3/ Sử dụng CheckBox

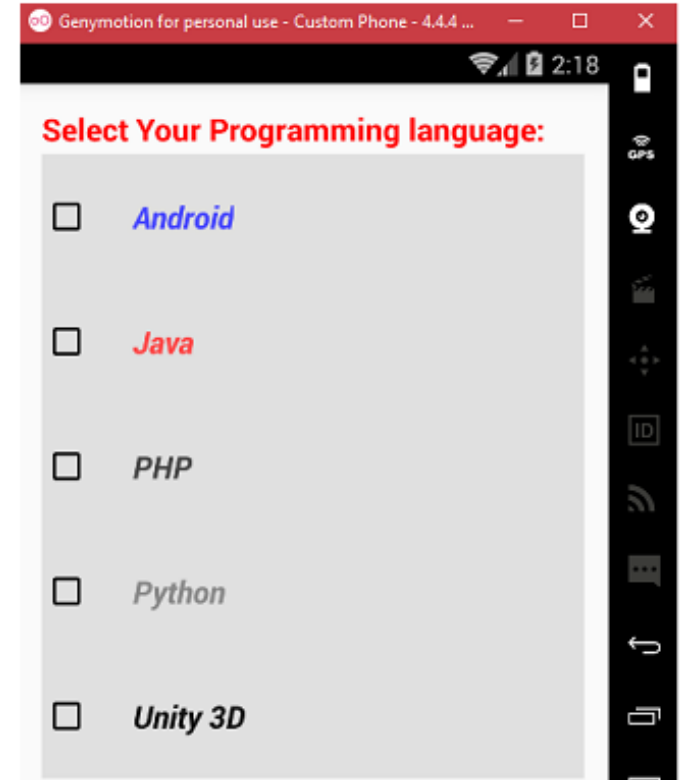
Ví dụ:

```
<CheckBox
    android:id="@+id/chkAndroid"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="27dp"
    android:text="@string/android"
    android:textColor="#44f"
    android:textSize="20sp"
    android:textStyle="bold|italic" />
```

```
<CheckBox
    android:id="@+id/chkJava"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="27dp"
    android:text="@string/java"
    android:textColor="#f44"
    android:textSize="20sp"
    android:textStyle="bold|italic" />
```

```
<CheckBox
    android:id="@+id/chkPHP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="27dp"
    android:text="@string/php"
    android:textColor="#444"
    android:textSize="20sp"
    android:textStyle="bold|italic" />
```

```
<CheckBox
    android:id="@+id/chkPython"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="27dp"
    android:text="@string/python"
    android:textColor="#888"
    android:textSize="20sp"
    android:textStyle="bold|italic" />
```



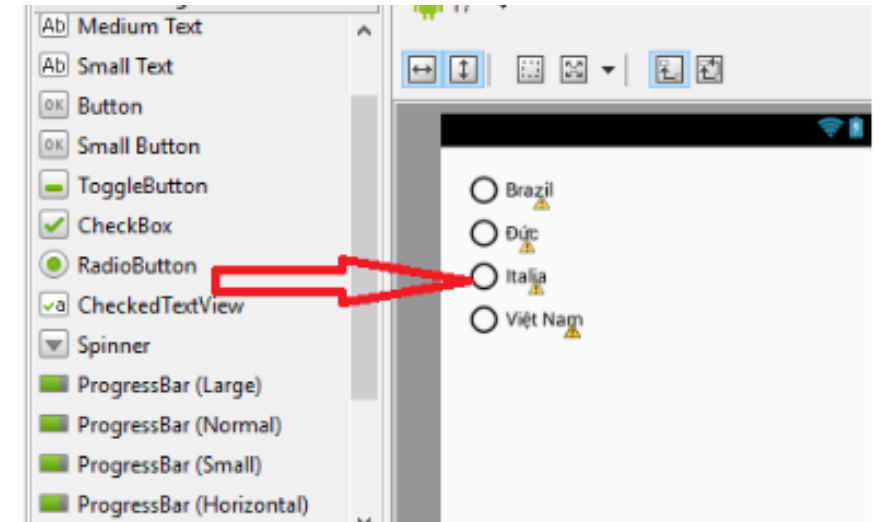
4.4/ Sử dụng RadioButton

- Radiobutton thường được đưa ra 2 hoặc nhiều hơn hai phần tử trong đó người dùng chỉ được chọn một phần tử, để làm được như vậy chúng ta cần nhóm chúng vào một nhóm đó chính là `GroupRadiobutton` để khi người dùng chọn thì chỉ chọn được duy nhất.
- `RadioButton` là một mục chọn gồm có 2 trạng thái `checked` và `unchecked`, nếu một button đang ở trạng thái `unchecked` thì người sử dụng có thể check vào, còn button đang ở trạng thái `checked` thì người sử dụng không thể `unchecked`.

4.4/ Sử dụng RadioButton

Các phương thức thường sử dụng:

Phương thức	Ý nghĩa
public boolean isChecked()	True nếu CheckBox là checked, ngược lại false
public void setChecked(boolean status)	Thay đổi trạng thái của CheckBox



```
<RadioButton
    android:id="@+id/rdisimpleRadioButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nữ"/>
```

4.4/ Sử dụng RadioButton

Có 2 cách xử lý RadioButton nào được checked như sau:

Cách 1: Dựa vào RadioGroup để biết chính xác Id của RadioButton nào được checked. Dựa vào Id này ta sẽ xử lý đúng nghiệp vụ:

```
rdigroupCountry = (RadioGroup) findViewById(R.id.rdigroupCountry);
int idChecked = rdigroupCountry.getCheckedRadioButtonId();
switch(idChecked){
    case R.id.rdiBrazil:
        Toast.makeText(getApplicationContext(), "Bạn đã chọn nước Brazil" , Toast.LENGTH_LONG).show();
        break;
    case R.id.rdiGermany:
        Toast.makeText(getApplicationContext(), "Bạn đã chọn nước Đức" , Toast.LENGTH_LONG).show();
        break;
    case R.id.rdiItalia:
        Toast.makeText(getApplicationContext(), "Bạn đã chọn nước Italia" , Toast.LENGTH_LONG).show();
        break;
    case R.id.rdiVietnam:
        Toast.makeText(getApplicationContext(), "Bạn đã chọn nước Việt Nam" , Toast.LENGTH_LONG).show();
        break;
}
```

4.4/ Sử dụng RadioButton

Có 2 cách xử lý RadioButton nào được checked như sau:

Cách 2: Kiểm tra trực tiếp từng RadioButton đó có được checked hay không?

```
RadioButton rdiBrazil, rdiGermany, rdiVietnam;  
rdiBrazil = (RadioButton) findViewById(R.id.rdiBrazil);  
rdiGermany = (RadioButton) findViewById(R.id.rdiGermany);  
rdiVietnam = (RadioButton) findViewById(R.id.rdiVietnam);  
if(rdiBrazil.isChecked())  
    Toast.makeText(getApplicationContext(), "Bạn đã chọn nước "+rdiBrazil.getText() , Toast.LENGTH_LONG).show();  
else if(rdiGermany.isChecked())  
    Toast.makeText(getApplicationContext(), "Bạn đã chọn nước "+rdiGermany.getText() , Toast.LENGTH_LONG).show();  
else if(rdiVietnam.isChecked())  
    Toast.makeText(getApplicationContext(), "Bạn đã chọn nước "+rdiVietnam.getText() , Toast.LENGTH_LONG).show();
```

Câu hỏi Thảo luận

- 1) Trình bày ứng dụng của các dạng layouts đã học.
- 2) Phân biệt View và ViewGroup
- 3) Trình bày điểm khác biệt của CheckBox và RadioButton.