Bài 3: Android Layout Buổi 2

Giáo viên: Lê Quốc Anh

Nội dung

1. Layout trong lập trình Android
2. TableLayout
3. LinearLayout
4. RelativeLayout
5. ConstraintLayout
6. Ví dụ

1. Layout trong lập trình Android

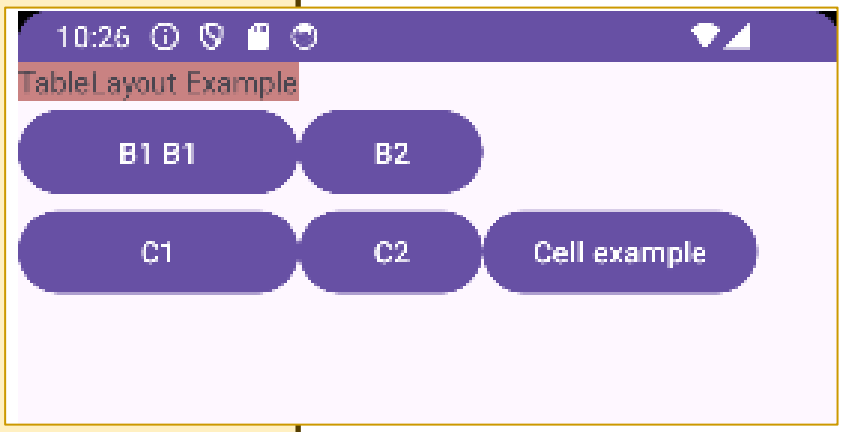
- **Layout** Các layout chính là các View (cụ thể nó kế thừa thừa ViewGroup) được thiết kế với mục đích chứa các View con và điều khiển, sắp xếp vị trí các View con đó trên màn hình, mỗi layout có cơ chế điều khiển vị trí View con riêng của mình.
  - **TableLayout**
  - **LinearLayout**
  - **RelativeLayout**
  - **ConstraintLayout**
  - **GridLayout**
  - **FrameLayout**
  - **CoordinatorLayout**

2. TableLayout

- **TableLayout** nó sẽ sắp xếp các View con bên trong thành dạng bảng
- Mỗi hàng là một đối tượng view **TableRow** bên trong **TableRow** chứa các View con
- mỗi View con này nằm ở vị trí một ô bảng (cell)
- Cột / hàng trong bảng bắt đầu từ số 0

2. TableLayout

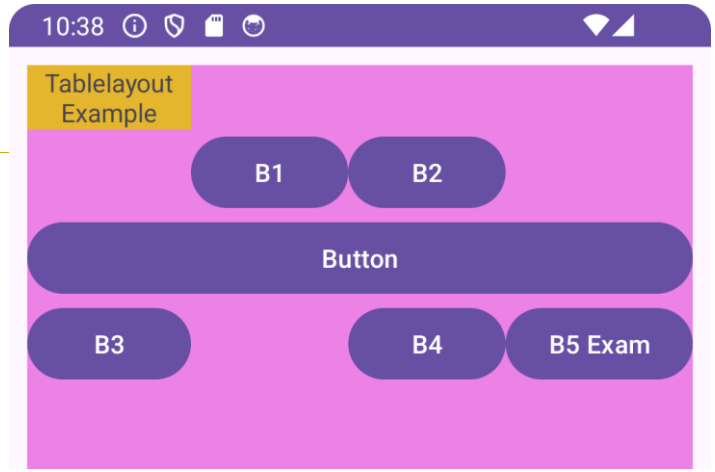
```
<TableLayout android:layout_width="match_parent"
   android:layout_height="match_parent"
xmlns:android="http://schemas.android.com/apk/res/android">
   <TableRow>
      <TextView
         android:text="TableLayout Example"
         android:background="#c98282"
         android:gravity="center"/>
   </TableRow>
   <TableRow>
      <Button android:text="B1 B1" />
      <Button android:text="B2"/>
   </TableRow>
   <TableRow>
      <Button android:text="C1" />
      <Button android:text="C2" />
      <Button android:text="Cell example" />
   </TableRow>
</TableLayout>
```



2. TableLayout

- android:stretchColumns="col1, col2 ..." (ký hiệu * tất cả các cột): chiếm tất cả khoảng trống còn thừa
- android:shrinkColumns="0,1,.." (ký hiệu * tất cả các cột) : co các cột
- <Button android:text="Singlerow" />
- <Button android:text="Button 0" android:layout_column="1"/>
- android:layout_span="3"

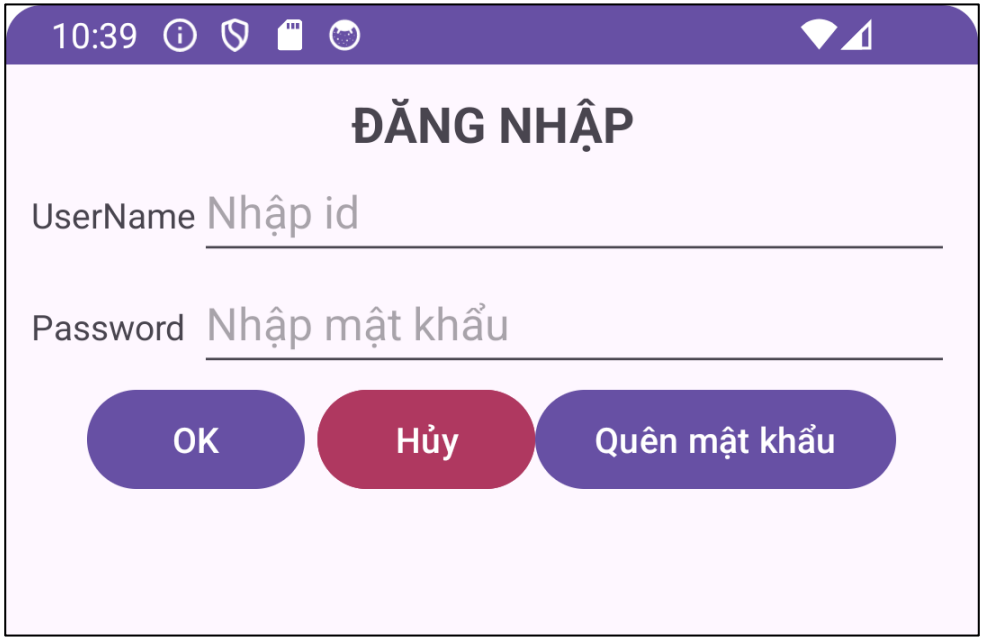
2. TableLayout




```
<TableLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#EC82E6"
    android:layout_margin="10dp"
    android:shrinkColumns="0"
    android:stretchColumns="2"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TableRow>
        <TextView android:text="Tablelayout Example"
            android:background="#E3B62E"
            android:gravity="center"/>
    </TableRow>
    <TableRow>
        <Button  android:text="B1" android:layout_column="1"/>
        <Button  android:text="B2" />
    </TableRow>
    <Button android:text="Button"/>
    <TableRow>
        <Button  android:text="B3" />
        <Button  android:text="B4" android:layout_column="2"/>
        <Button  android:text="B5 Exam" />
    </TableRow>
</TableLayout>
```

2. TableLayout

- Sử dụng TableLayout hoàn thành layout sau



3. LinearLayout

- **LinearLayout** là một layout rất tiện lợi trong thiết kế giao diện, nó sắp xếp các View con một cách liên tục theo tùy chọn xếp theo chiều ngang (một hàng các view) hay chiều đứng (một cột các view)
- **android:orientation** để thiết lập cách sắp xếp phần tử. *android:orientation="horizontal"* xếp theo chiều ngang (ví dụ trên) và *android:orientation="vertical"* xếp theo chiều đứng.

3. LinearLayout


```
<LinearLayout
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#a6dfdfdf"
    android:layout_margin="10dp"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:backgroundTint="#dc9e1a"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:backgroundTint="#00c853"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:backgroundTint="#141ba9"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:backgroundTint="#dc1a8b"
        android:text="B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

3. LinearLayout

- Thuộc tính **android:gravity** để căn chỉnh các View nằm ở vị trí nào trong LinearLayout, nó nhận các giá trị (có thể tổ hợp lại với ký hiệu |)

Giá trị	Ý nghĩa
center	Căn ở giữa
top	Ở phần trên
bottom	Phần dưới
center_horizontal	Ở giữa theo chiều ngang
center_vertical	Ở giữa theo chiều đứng
left	Theo cạnh trái
right	Theo cạnh phải
bottom	Cạnh dưới

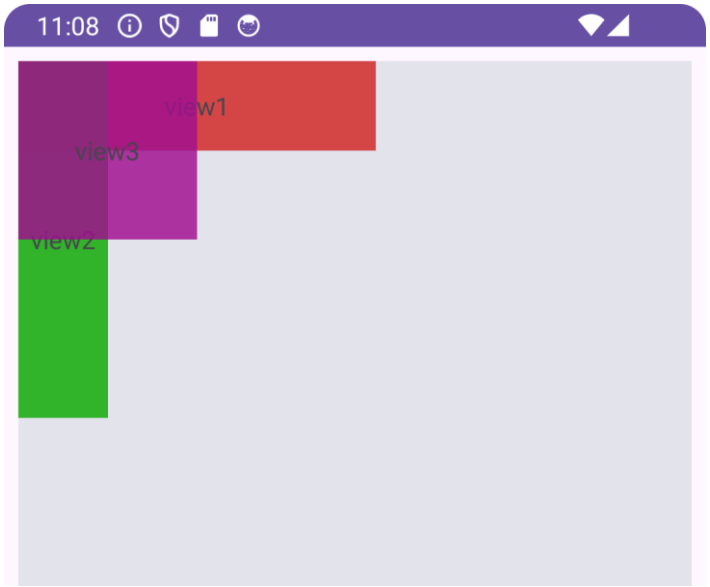
4. RelativeLayout

- RelativeLayout là layout mà các View con được xác định vị trí bởi các mối liên hệ với View cha hoặc với View con như View con nằm dưới một View con khác, View con căn thẳng lề phải với View cha ...
- **RelativeLayout** là layout mà nó hiện thị các view con nó chứa ở các vị trí trong mối liên hệ của chúng với nhau *(như View con này nằm dưới một View con khác, View con này nằm bên trái một View con khác ...)*,
- kể cả mối liên hệ của chúng với chính phần tử cha RelativeLayout (như căn thẳng theo cạnh đấy của phần tử cha, nằm giữa phần tử cha, nằm bên trái phần tử cha ...).

4. RelativeLayout

- **RelativeLayout** là một layout hết sức mạnh mẽ về độ tiện lợi và hiệu quả, nếu giao diện không ở mức quá phức tạp việc chọn RelativeLayout mạng lại hiệu suất còn tốt hơn ConstraintLayout.
- RelativeLayout dùng khi đơn giản, ConstraintLayout khi giao diện phức tạp.

4. RelativeLayout




```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"
    android:background="#e2e3eb">

    <TextView
        android:id="@+id/view1"
        android:text="view1"
        android:gravity="center"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:background="#e8d33636" />

    <TextView
        android:id="@+id/view2"
        android:text="view2"
        android:gravity="center"
        android:layout_width="50dp"
        android:layout_height="200dp"
        android:background="#e71faf15" />
```

```
    <TextView
        android:id="@+id/view3"
        android:text="view3"
        android:gravity="center"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:background="#d4a00f8f" />
</RelativeLayout>
```

4. RelativeLayout

- Các View con khi đã định vị xong trong RelativeLayout, giả sử coi như tất cả các View con nằm vừa trong một đường biên chữ nhật, thì cả khối các View con này có thể dịch chuyển tới những vị trí nhất định trong RelativeLayout bằng thuộc tính: **android:gravity**

Giá trị	Ý nghĩa
center	Căn ở giữa
top	Ở phần trên
bottom	Phần dưới
center_horizontal	Ở giữa theo chiều ngang
center_vertical	Ở giữa theo chiều đứng
left	Theo cạnh trái
right	Theo cạnh phải
bottom	Cạnh dưới

Định vị View con bằng liên hệ với View cha RelativeLayout

- Vị trí của View con trong RelativeLayout có thể thiết lập bằng cách chỉ ra mối liên hệ vị trí với view cha như căn thẳng cạnh trái View cha với View con, căn thẳng cạnh phải View cha với View con ... Các thuộc tính thực hiện chức năng này như sau

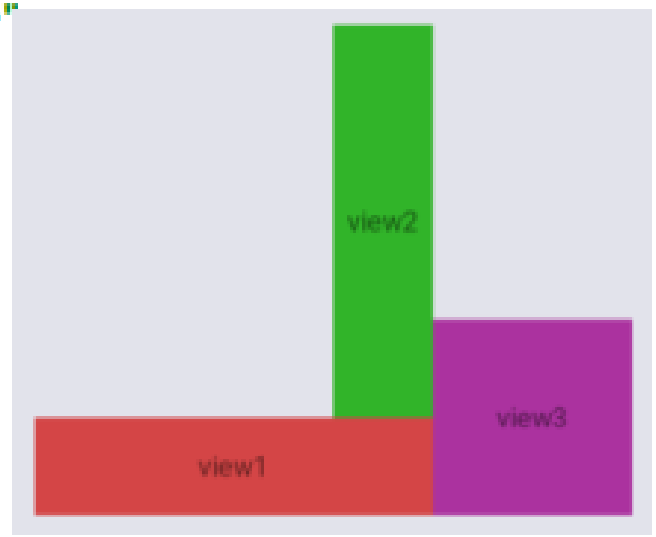
Thuộc tính	Ý nghĩa
android:layout_alignParentBottom	true căn thẳng cạnh dưới view con với cạnh dưới View cha
android:layout_alignParentLeft	true căn thẳng cạnh trái view con với cạnh trái View cha
android:layout_alignParentRight	true căn thẳng cạnh phải view con với cạnh phải View cha
android:layout_alignParentTop	true căn thẳng cạnh trên view con với cạnh trên View cha
android:layout_centerInParent	true căn view con vào giữa View cha
android:layout_centerHorizontal	true căn view con vào giữa View cha theo chiều ngang
android:layout_centerVertical	true căn view con vào giữa View cha theo chiều đứng

Định vị View con bằng liên hệ giữa chúng với nhau

- View con trong RelativeLayout ngoài mối liên hệ với View cha như trên, chúng có có thể thiết lập liên hệ với nhau ví dụ như View con này nằm phía trên một View con khác, nằm phía dưới một view con khác ...

Thuộc tính	Ý nghĩa
android:layout_below	Nằm phía dưới View có ID được chỉ ra
android:layout_above	Nằm phía trên View có ID được chỉ ra
android:layout_toLeftOf	Nằm phía trái View có ID được chỉ ra
android:layout_toRightOf	Nằm phía phải View có ID được chỉ ra
android:layout_alignBottom	Căn thẳng cạnh dưới với cạnh dưới của View có ID được chỉ ra
android:layout_alignLeft	Căn thẳng cạnh trái với cạnh trái của View có ID được chỉ ra
android:layout_alignRight	Căn thẳng cạnh phải với cạnh phải của View có ID được chỉ ra
android:layout_alignTop	Căn thẳng cạnh trên với cạnh trên của View có ID được chỉ ra

```
<TextView
    android:id="@+id/view1"
    android:text="view1" android:gravity="center"
    android:layout_toLeftOf="@id/view3"
    android:layout_alignBottom="@id/view3"
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:background="#e8d33636" />
<TextView
    android:id="@+id/view2" android:text="view2"
    android:layout_alignRight="@id/view1"
    android:layout_above="@id/view1"
    android:gravity="center"
    android:layout_width="50dp" android:layout_height="200dp"
    android:background="#e71faf15" />
<TextView
    android:id="@+id/view3"
    android:layout_centerInParent="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="30dp"
    android:text="view3" android:gravity="center"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#d4a00f8f" />
```



5. ConstraintLayout

- Dùng ConstraintLayout để xây dựng các layout trong lập trình Android, với tính năng ràng buộc các phần tử, thiết lập các xích đa dạng
- **ConstraintLayout** là một layout mạnh, khuyến khích bạn dùng nếu có thể vì nó giúp tạo ra các giao diện phức tạp, mềm dẻo (hạn chế tối đa sử dụng các layout lồng nhau).
- Nó giúp định vị, sắp xếp các View con dựa trên sự ràng buộc liên hệ của các View con với View cha và sự liên hệ ràng buộc giữa các View con với nhau, với cơ chế tạo xích các View, gán trọng số hay sử dụng trợ giúp giao diện với Guideline.

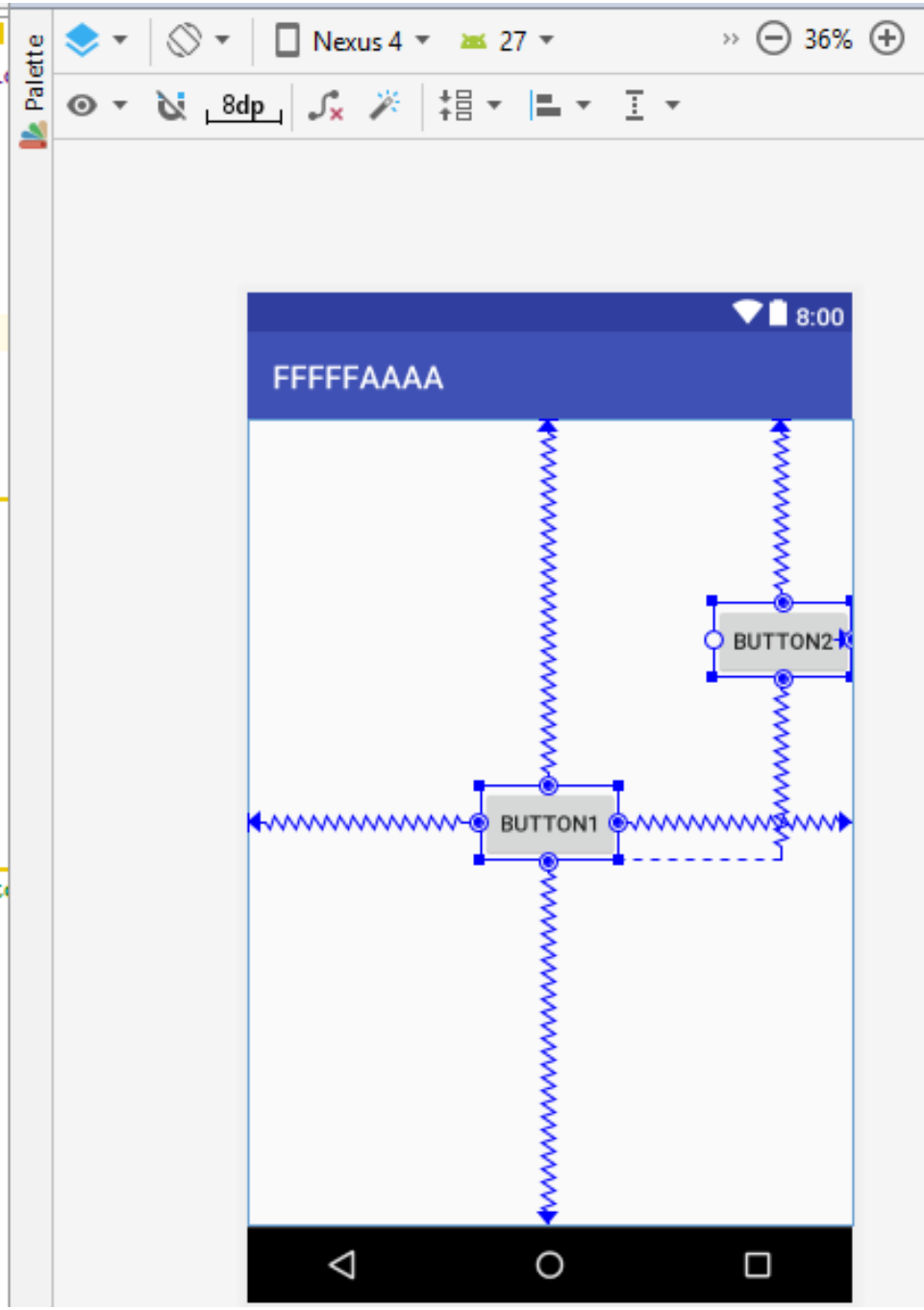
5. ConstraintLayout

- **Sự ràng buộc**
  - Mỗi view trong ConstraintLayout để định vị được chính xác cần tối thiểu 2 ràng buộc, một theo phương ngang (X) và một theo phương đứng (Y)
- Khái niệm ràng buộc giữa các phần tử ở đây ám chỉ sự liên kết với nhau của các phần tử với nhau (kể cả với phần tử cha ConstraintLayout)
- **app:layout_constraintBottom_toBottomOf="parent"**

5. ConstraintLayout

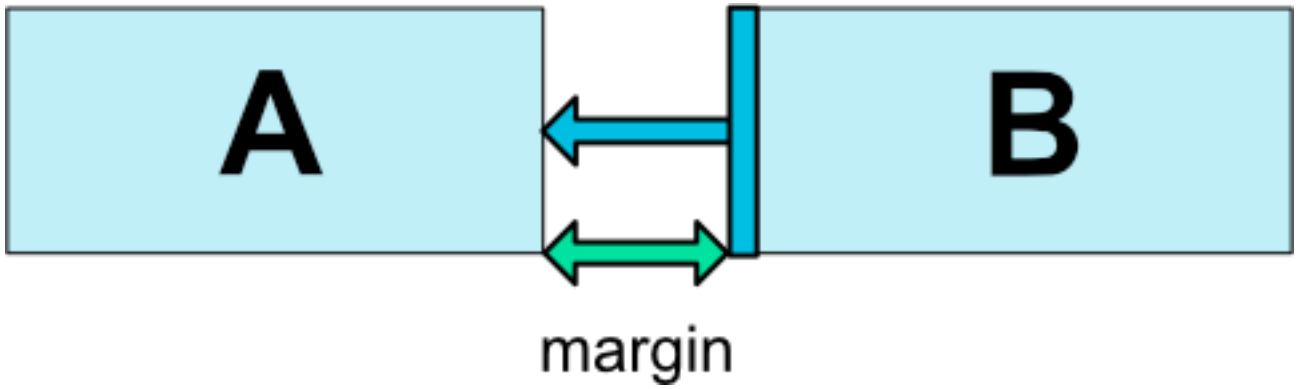
Ràng buộc	Ý nghĩa ràng buộc
layout_constraintLeft_toLeftOf	Rang buộc cạnh trái của phần tử tới phần tử chỉ ra trong giá trị (gán ID)
layout_constraintLeft_toRightOf	Bên trái với bên phải của phần tử chỉ ra
layout_constraintRight_toLeftOf	Bên phải với bên trái
layout_constraintRight_toRightOf	Phải với phải
layout_constraintTop_toTopOf	Cạnh trên với cạnh trên
layout_constraintTop_toBottomOf	Cạnh trên nối với cạnh dưới
layout_constraintBottom_toTopOf	Dưới với trên
layout_constraintBottom_toBottomOf	Dưới với dưới
layout_constraintBaseline_toBaselineOf	Trùng Baseline
layout_constraintStart_toEndOf	Bắt đầu - Kết húc
layout_constraintStart_toStartOf	Bắt đầu - Bắt đầu
layout_constraintEnd_toStartOf	Cuối với bắt đầu
layout_constraintEnd_toEndOf	Cuối với cuối

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:androi
    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">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BUTTON2"
        app:layout_constraintBottom_toBottomOf="@+id/butt
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

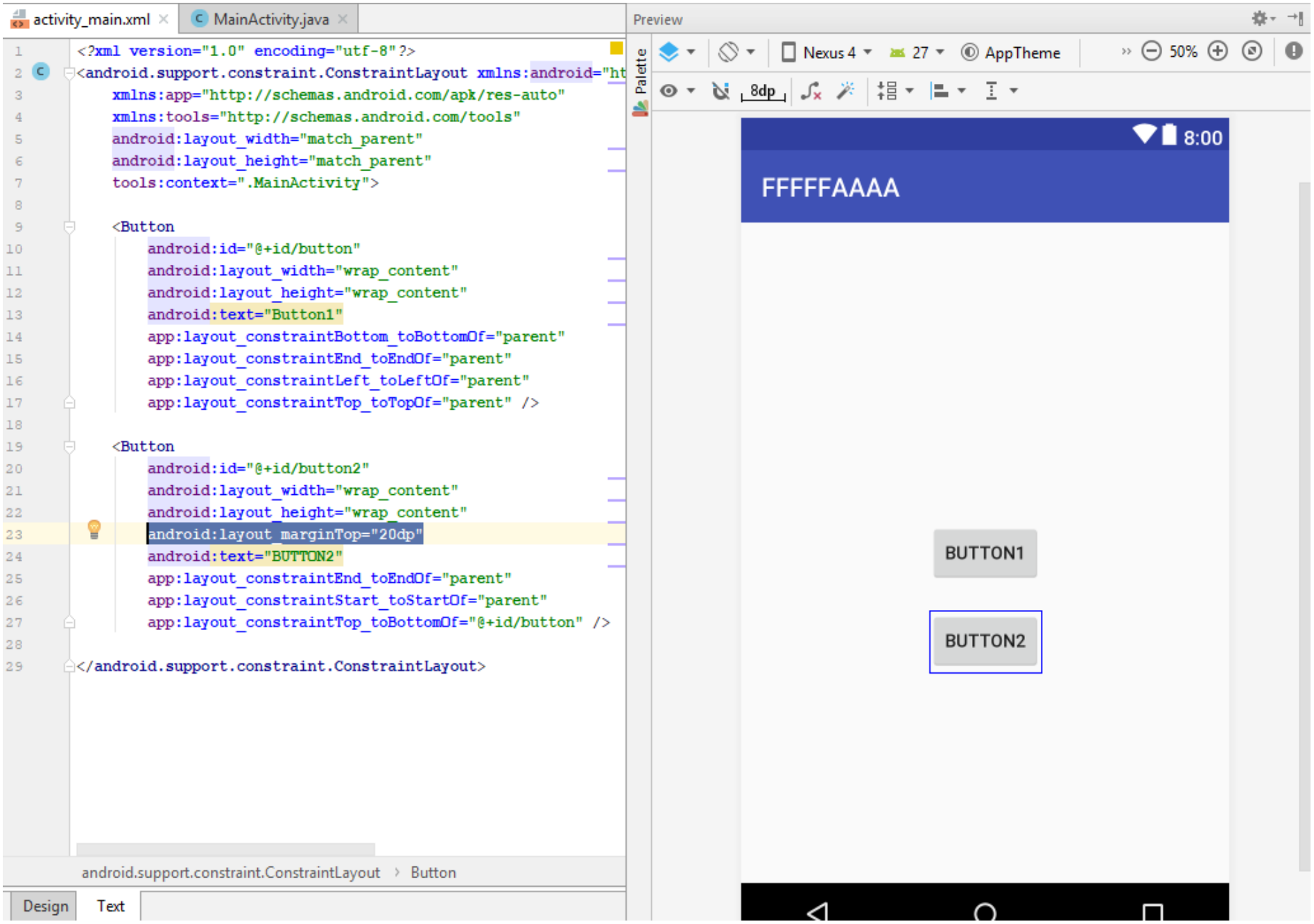


5. ConstraintLayout

- Cạnh nào của View con có ràng buộc thì có thể thiết lập thêm thuộc tính Margin để điều chỉnh thêm khoảng cách các cạnh tới điểm nối ràng buộc.



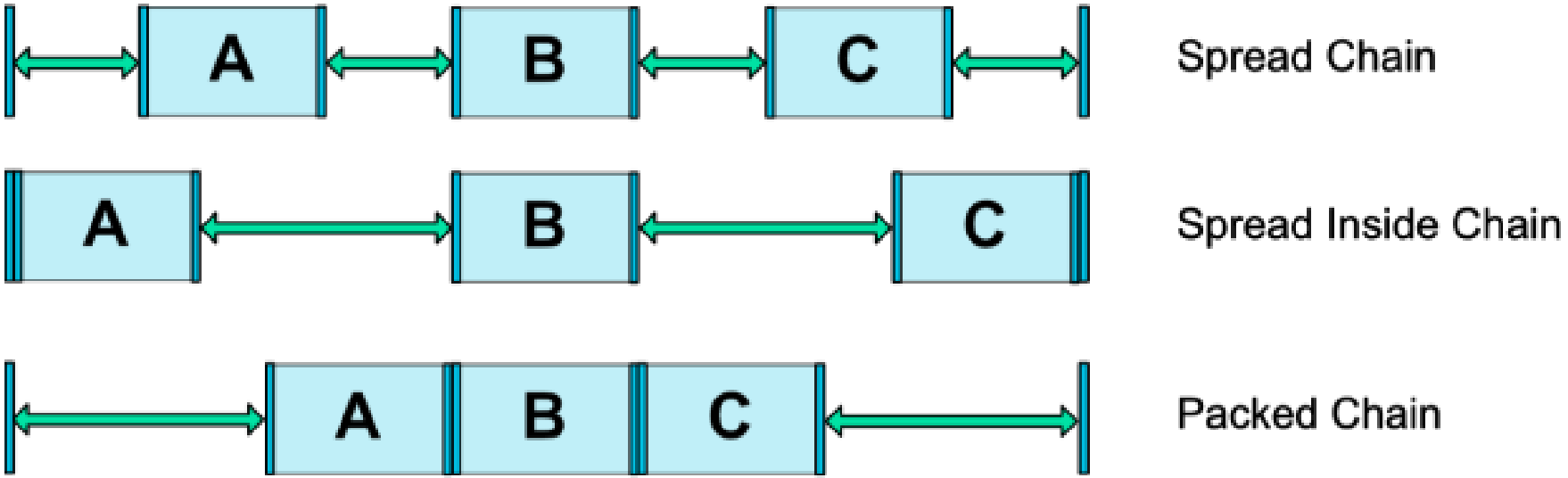
- Các thuộc tính margin theo các cạnh:
android:layout_marginStart, android:layout_marginEnd, android:layout_marginLeft, android:layout_marginTop, android:layout_marginRight, android:layout_marginBottom


activity_main.xml
MainActivity.java
Preview
Palette
Nexus 4
27
AppTheme
50%
8dp
8:00
FFFFFAAAA
BUTTON1
BUTTON2
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="ht
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">
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button1"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:text="BUTTON2"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button" />
</android.support.constraint.ConstraintLayout>
android.support.constraint.ConstraintLayout › Button
Design
Text

Chains trong Constraint layout

- Các Chain (chuỗi) cung cấp hành vi giống như nhóm theo một trục đơn (theo chiều ngang hoặc chiều dọc)
- Để tạo một chain, ta kết nối các View với nhau theo cả 2 hướng (bi-directional connection)
- Hai hướng ở đây có nghĩa là đuôi của view 1 neo vào đầu của view 2 và đầu của view 2 cũng cần được neo vào đuôi của view 1. (nếu không sẽ không tính là chain)

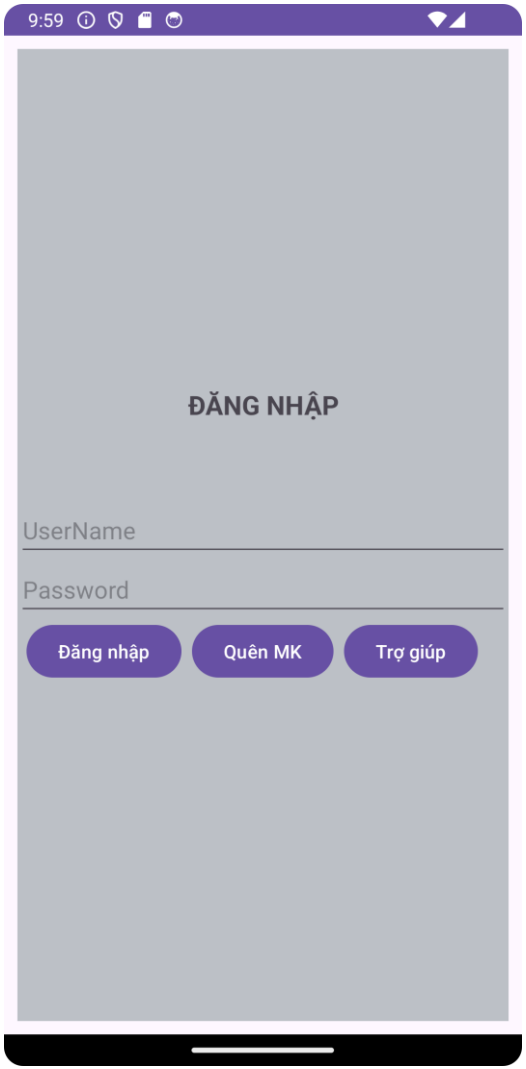
chainStyle của Chain



```
<Button
    android:id="@+id/label_a"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="A"
    android:textSize="22sp"
    app:layout_constraintEnd_toStartOf="@+id/label_b"
    app:layout_constraintHorizontal_chainStyle="packed"
    app:layout_constraintStart_toStartOf="parent"/>
<Button
    android:id="@+id/label_b"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="B"
    android:textSize="22sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/label_a"/>
```

Ví dụ

- Sử dụng các loại layout đã học để hoàn thành bài sau



THANK YOU for YOUR ATTENTION