

BUỔI 1: CƠ BẢN LẬP TRÌNH ANDROID STUDIO

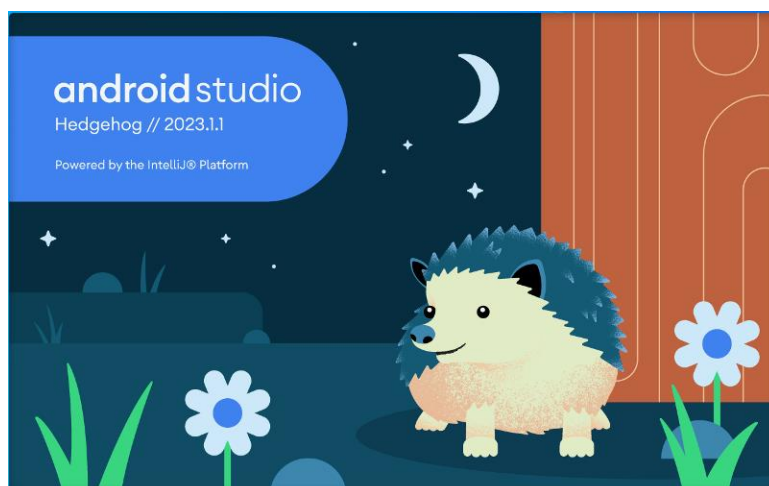
I. Mục tiêu buổi thực hành

- Tạo dự án ứng dụng di động bằng java trên Android Studio.
- Xây dựng Activity với các kiểu Layout:
 - + ConstraintLayout.
 - + LinearLayout.
 - + TableLayout.
 - + FrameLayout.
- Tạo giao diện gồm các view:
 - + TextView.
 - + Edit text.
 - + Button.
 - + ImageView.
 - + ImageButton.
 - + CheckBox.
 - + RadioButton, ...

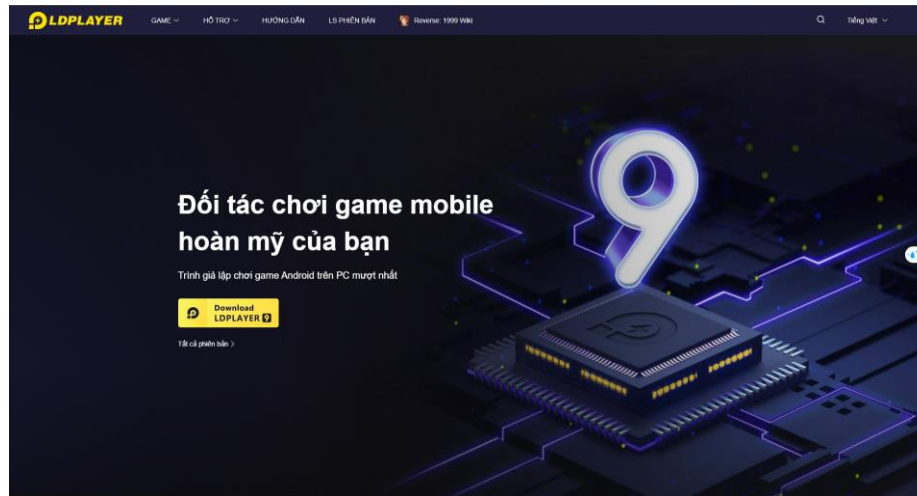
II. Triển khai buổi thực hành

Nội dung 1: Thông số phần mềm được sử dụng trong bài thực hành

- Hệ điều hành: Windows 10 Pro 22H2.
- Phiên bản phần mềm Android Studio: Android Studio Hedgehog | 2023.1.1 Patch 2



- Phần mềm máy điện thoại ảo giả lập trên máy tính: LDPlayer 9. Phiên bản android trên LDPlayer 9 là Android 9.



- Phiên bản JDK: JDK 19

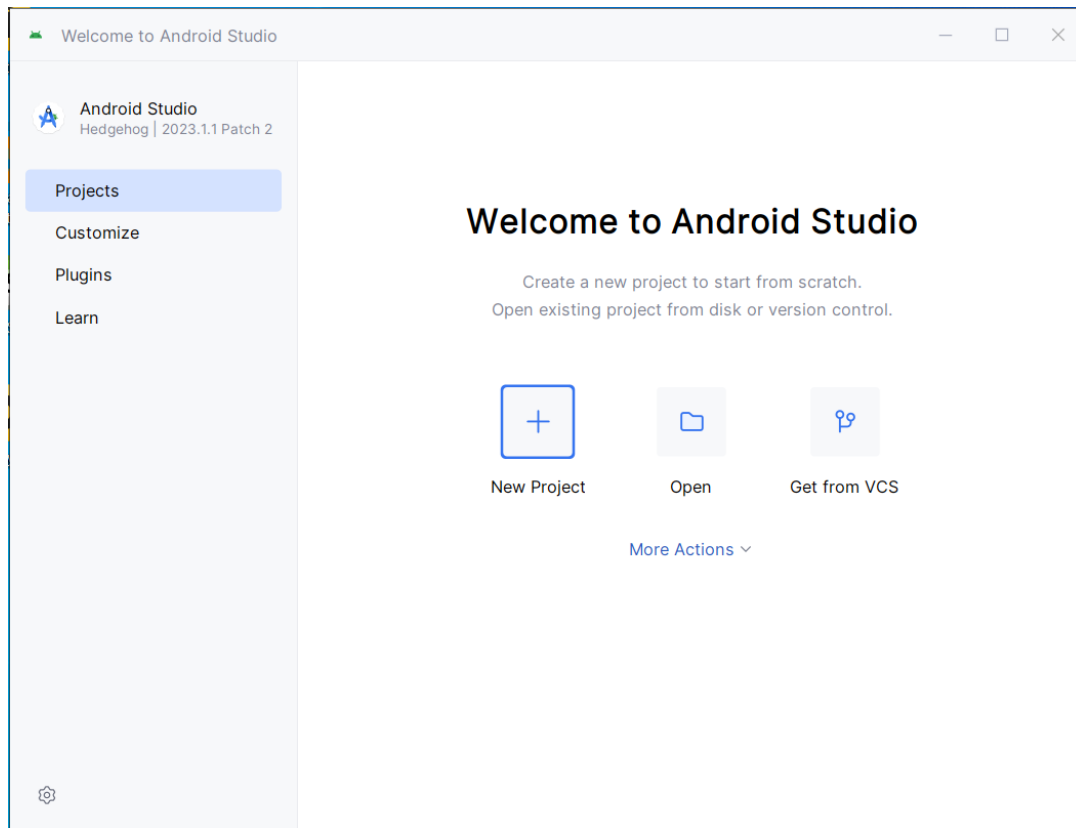
```
Command Prompt
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hvtuan>java --version
java 19.0.2 2023-01-17
Java(TM) SE Runtime Environment (build 19.0.2+7-44)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)

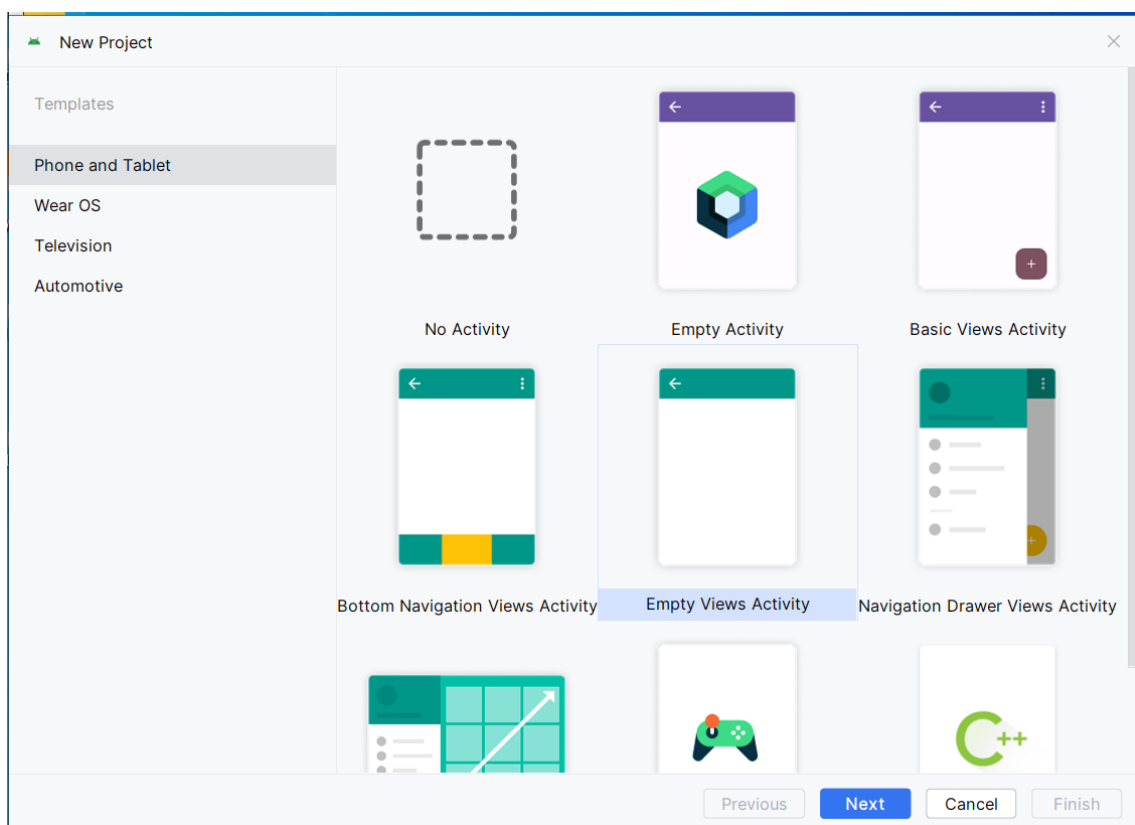
C:\Users\hvtuan>
```

Nội dung 2: Tạo project đầu tiên trên Android Studio

- Tại cửa sổ Welcome của phần mềm Android Studio, ta nhấn chọn New Project



- Có nhiều Template dự án, ta chọn loại “Empty View Activity” rồi chọn “Next” để tiếp tục.



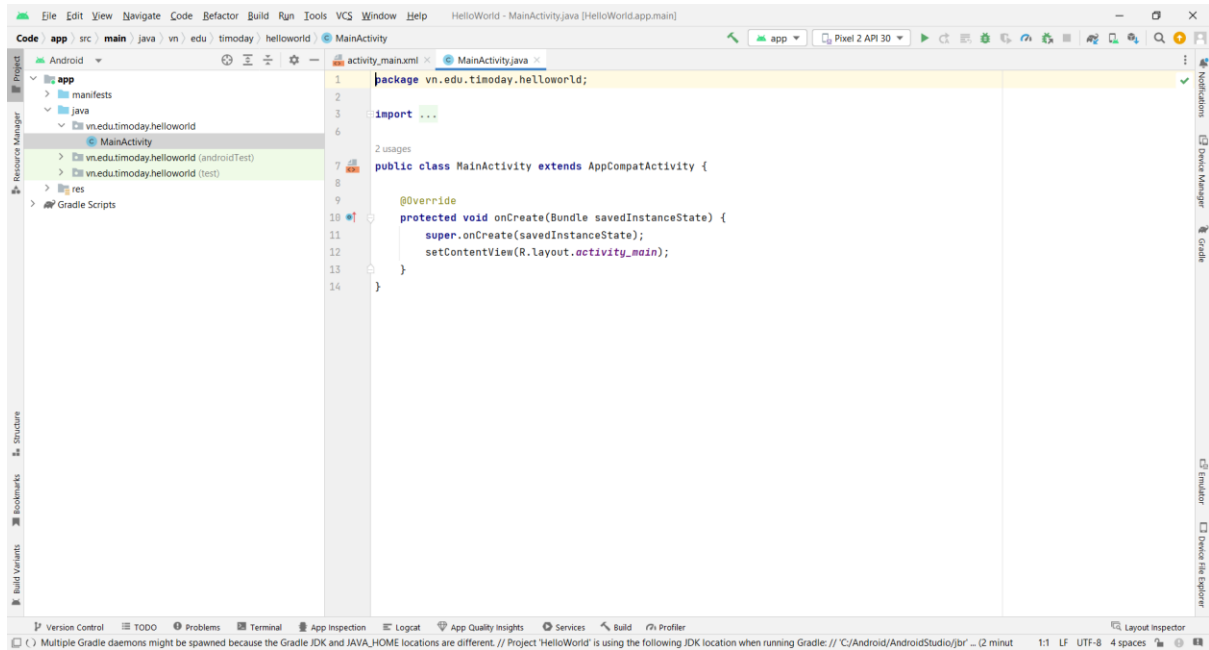
- Màn hình Cấu hình dự án được hiện ra như hình bên dưới:

Tên chức năng	Ý nghĩa
Name	Tên ứng dụng (dự án). Đặt tên có ý nghĩa.
Package name	Quan trọng, được coi như là định danh cho ứng dụng. Liên quan đến vấn đề pháp lý bản quyền khi tải APK lên Google Play.
Save location	Nơi lưu trữ dự án.
Language	Chọn ngôn ngữ lập trình để lập trình xử lý nghiệp vụ. Có 2 tùy chọn là Java hoặc Kotlin.
Minimum SDK	Cho biết ứng dụng được tạo ra sẽ hỗ trợ tối đa đến hệ điều hành cũ nhất là bao nhiêu.

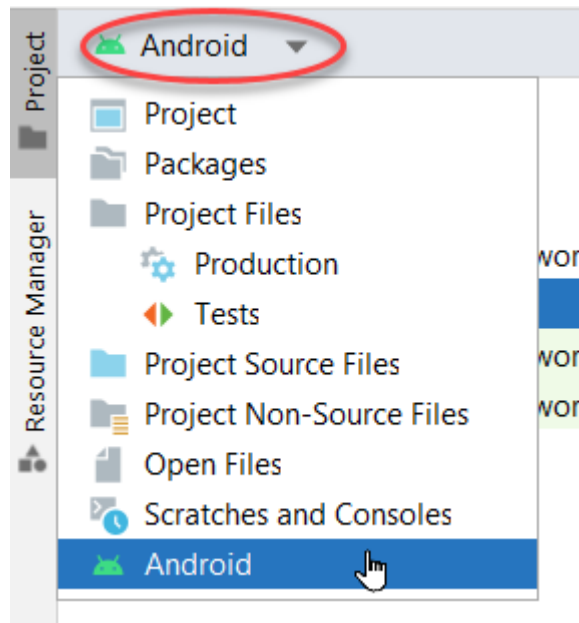
- Nhấn chọn Finish để hoàn tất cấu hình và phần mềm bắt đầu tạo dự án android theo các tùy chọn đã thiết lập.

Nội dung 3: Tổng quan về project

- Ta quan sát vào màn hình project sau khi được tạo như sau:

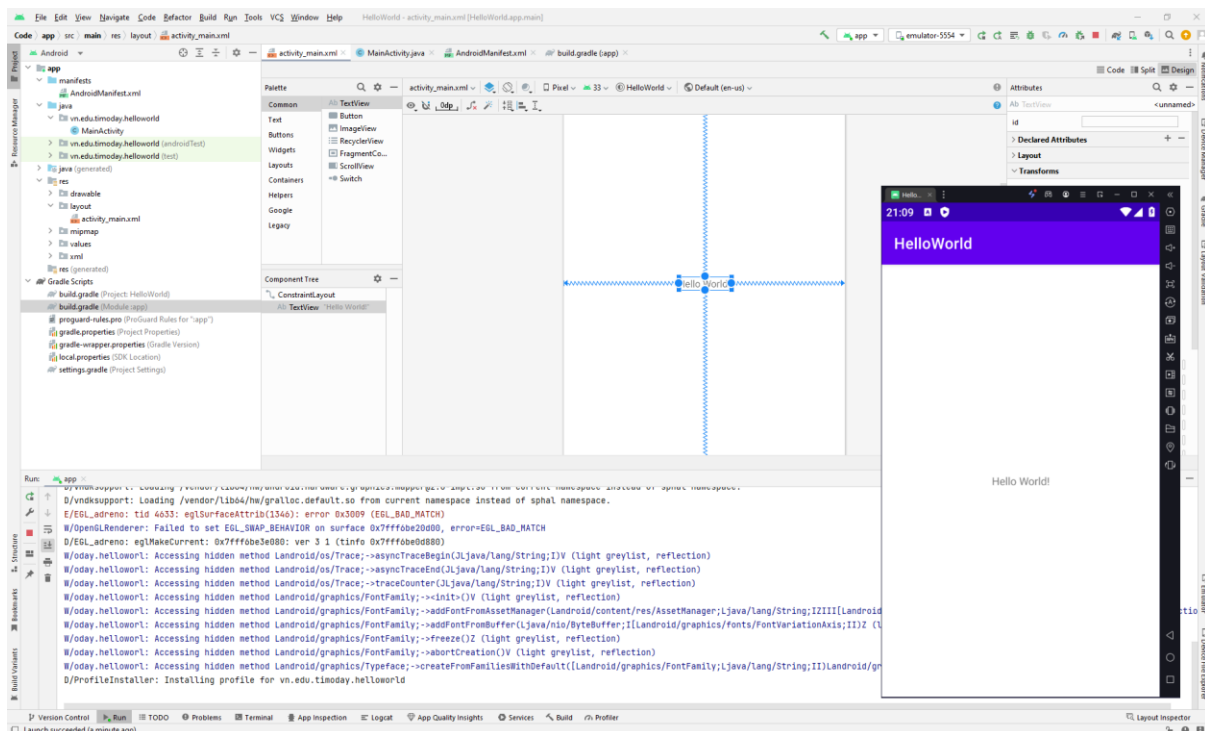


- Cửa sổ được chia thành 2 phần chính: nửa bên trái và nửa bên phải.
 - + Nửa bên phải là Editor window: là nơi ta soạn thảo lệnh lập trình xử lý nghiệp vụ cho hệ thống. Ngoài ra, đây còn là nơi bạn thiết kế giao diện, kéo thả và sắp xếp các thành phần đáp ứng cho hệ thống.
 - + Nửa bên trái là khu vực hiển thị cấu trúc của dự án. Có nhiều cách thức hiển thị cấu trúc dự án khác nhau:



- Ta cần nắm cấu trúc của một dự án android để dễ dàng làm việc khi lập trình android:

- + AndroidManifest: tập tin rất quan trọng, dùng để thiết lập việc cấp quyền, các Service, các màn hình Activity, ... cho ứng dụng.
- + java/vn.edu.timoday.helloworld: tất cả các lớp Activity xử lý nghiệp vụ người dùng.
- + Res/layout: tất cả các giao diện đều nằm trong mục này. Tức là một màn hình sẽ gồm 2 file là file code (MainActivity) và file giao diện (activity_main.xml).
- + Res/drawable và Res/mipmap: ta có thể đưa hình ảnh vào ứng dụng thông qua hai thư mục là drawable và mipmap.
- + Res/values: nơi để tạo ra các tài nguyên như chuỗi, màu, độ phân giải, ...
- + Gradle Scripts: cấu hình biên dịch. Giúp ta hiệu chỉnh các thông số khi biên dịch phần mềm, cho phép nhúng các thư viện bên ngoài vào ứng dụng.
- Ta nhấn chọn Run để chạy dự án và xem kết quả:

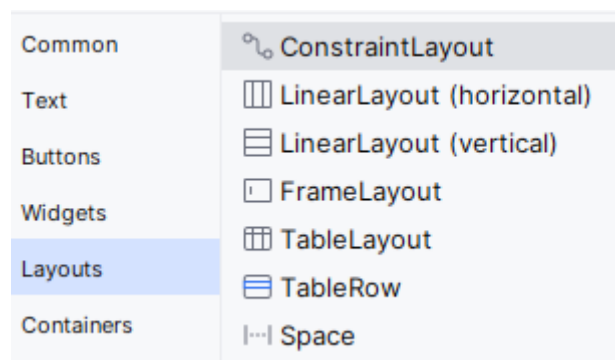


Để hiểu khi thiết kế màn hình ứng dụng để tương tác với người dùng, ta cần phải có kiến thức về cách sắp xếp, bố trí các đối tượng trên giao diện cho khoa học, đáp ứng được nhu cầu của người khách hàng. Mục này sẽ trình bày các Layout thường dùng như: ConstraintLayout, LinearLayout, TableLayout, ... Các

đối tượng mà người dùng tương tác trên giao diện được gọi là View hay Control. Mục sau sẽ trình bày một số control cơ bản như TextView để hiển thị dữ liệu, EditText để nhập liệu, Button để nhấn nhằm ra lệnh, ImageView để hiển thị hình ảnh, ...

Nội dung 4: Tìm hiểu về Layout trong Android

- **Layout** là nơi để tổ chức, sắp xếp các control/view trên giao diện và thường được sử dụng đầu tiên để triển khai việc thiết kế giao diện. Các loại Layout được bố trí trong Palette/Layouts:



4.1. LinearLayout

Loại Layout này được dùng để xây dựng giao diện bố trí theo một chiều duy nhất (chiều dọc – Vertical hoặc chiều ngang – Horizontal).

Ta sử dụng thuộc tính “android:orientation” để điều chỉnh hướng bố trí control:

```
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

</LinearLayout>
```

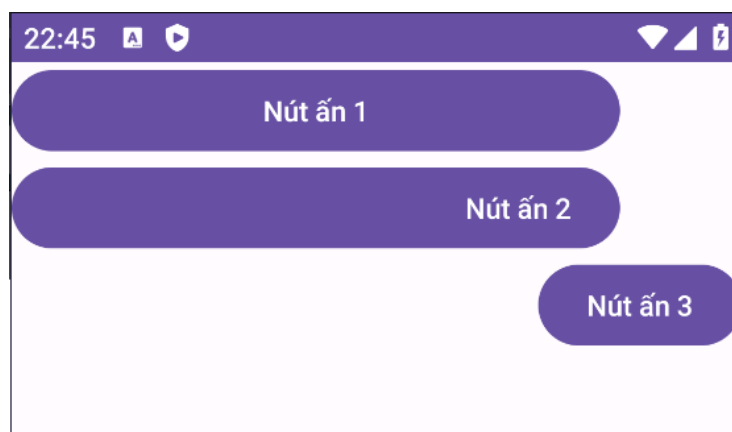
Mặc định, các thành phần bên trong của LinearLayout có vị trí là top left. Ta có thể sử dụng thuộc tính Gravity để thiết lập lại vị trí (có thể tổ hợp lại với ký hiệu).

Các thuộc tính của LinearLayout:

Thuộc tính	Mô tả
------------	-------

android:id	ID là duy nhất để nhận diện Layout
android:baselineAligned	Giá trị boolean “true” hoặc “false” và ngăn Layout căn chỉnh các baseline của view con.
android:baselineAlignedChildIndex	Khi một Linear Layout là một phần của Layout khác được căn chỉnh, nó có thể xác định view con nào của nó để căn chỉnh.
android:divider	Đây là drawable để sử dụng như một vertical divider giữa các button. Bạn sử dụng một giá trị màu trong dạng “#rgb”, “#argb”, “#rrggbb”, hoặc “#aarrggbb”.
android:gravity	set vị trí đặt nội dung của view, trên cả hai tọa độ X và Y. Giá trị có thể là top, bottom, left, right, center, center_vertical, center_horizontal ...
android:orientation	Xác định hướng sắp xếp và bạn sẽ sử dụng “horizontal” cho một hàng, “vertical” cho một cột. Nếu không chỉ định thì mặc định là horizontal.
android:weightSum	Tính tổng weight của các view con.

LinearLayout hỗ trợ thuộc tính “android:gravity” để căn chỉnh chữ trong View. Thuộc tính “android:layout_gravity” để căn chỉnh vị trí tương đối của View trong Layout. Xem xét ví dụ sau:




```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:width="300dp"
    android:text="Nút ấn 1" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:width="300dp"
    android:gravity="right|center"
    android:text="Nút ấn 2"/>

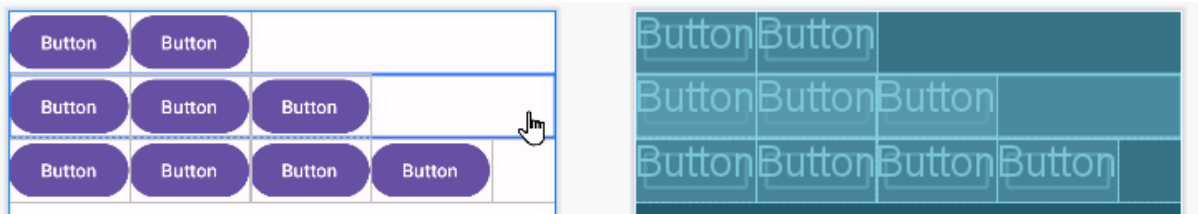
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nút ấn 3"
    android:layout_gravity="right"/>

```

4.2. TableLayout

TableLayout là loại layout được kế thừa từ LinearLayout, nó cho phép hiển thị các đối tượng (View) trên nhiều dòng (TableRow). Mỗi dòng chứa nhiều View, mỗi View là một cột.

Đặc biệt, TableLayout lấy dòng có nhiều View nhất làm số lượng cột cho nó.



```

2  <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6  >    <TableRow...>
22
23 >    <TableRow...>
45
46 >    <TableRow...>
74 </TableLayout>

```

0	1		
0	1	2	
0	1	2	3

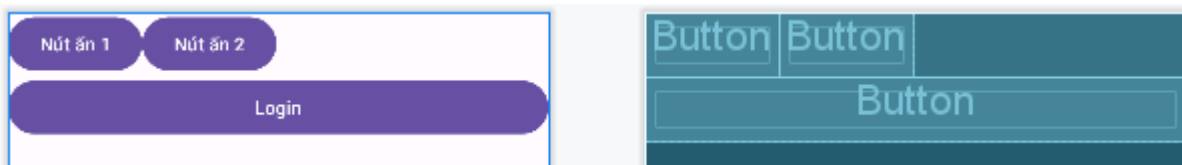
⇒ Trong TableLayout, các cột được đánh chỉ số 0, 1, 2, ... Như vậy theo như hình minh họa trên thì layout này có 4 cột và 3 dòng.

Các thuộc tính của TableLayout:

Thuộc tính	Mô tả
android:id	ID là duy nhất để nhận diện Layout
android:collapseColumns	Xác định index của các cột (bắt đầu từ 0). Các index cột được phân biệt riêng rẽ bởi dấu phẩy.
android:shrinkColumns	Xác định index của các cột (bắt đầu từ 0) để co lại. Các index cột được phân biệt riêng rẽ bởi dấu phẩy.
android:stretchColumns	Xác định index của các cột (bắt đầu từ 0) để kéo ra. Các index cột được phân biệt riêng rẽ bởi dấu phẩy.

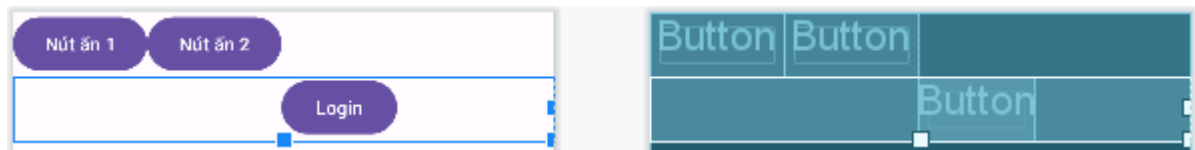
⇒ Để giãn đều các control, các cell, ta dùng thuộc tính “android:stretchColumns” và đặt giá trị là “*” để cấu hình cho toàn bộ các cột hoặc nhập số để cấu hình cho cột được chỉ định.

TableLayout còn cho phép ta Merge (trộn) các ô trong TableRow bằng cách sử dụng thuộc tính “android:layout_span” cho đối tượng View trong TableRow.



```
2 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:stretchColumns="2">
6
7 > <TableRow...>
23
24 <TableRow
25     android:layout_height="wrap_content"
26     android:layout_width="wrap_content" >
27
28     <Button
29         android:id="@+id/btnLogin"
30         android:layout_width="match_parent"
31         android:layout_height="match_parent"
32         android:text="Login"
33         android:layout_span="3"/>
34 </TableRow>
```

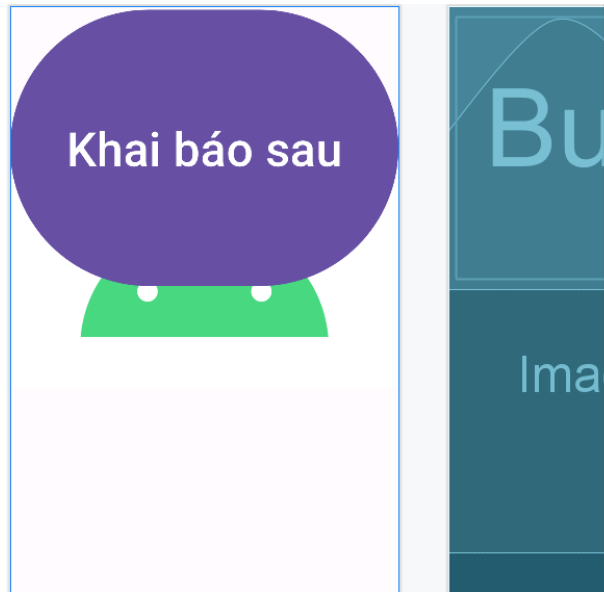
Thuộc tính `android:layout_column`: mặc định mỗi View nằm trong một cell nếu ta muốn chỉ ra View nằm ở một cột cụ thể thì dùng thuộc tính “`layout_column`”.



```
7      <TableRow
8          android:layout_width="match_parent"
9          android:layout_height="match_parent">
10
11          <Button
12              android:id="@+id/button"
13              android:layout_width="wrap_content"
14              android:layout_height="wrap_content"
15              android:text="Nút ấn 1" />
16
17          <Button
18              android:id="@+id/button2"
19              android:layout_width="wrap_content"
20              android:layout_height="wrap_content"
21              android:text="Nút ấn 2" />
22      </TableRow>
23
24      <TableRow
25          android:layout_height="wrap_content"
26          android:layout_width="wrap_content" >
27
28          <Button
29              android:id="@+id/btnLogin"
30              android:layout_width="match_parent"
31              android:layout_height="match_parent"
32              android:text="Login"
33              android:layout_column="2"/>
34      </TableRow>
--
```

4.3. FrameLayout

FrameLayout được sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị một đối tượng. Vị trí mặc định là top-left trên FrameLayout. Đối tượng (View) khai báo sau sẽ được xếp chồng lên đối tượng khai báo trước.



```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/android_img"
        android:id="@+id/imgIcon"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:height="300dp"
        android:text="Khai báo sau"
        android:textSize="50dp"/>
</FrameLayout>
```

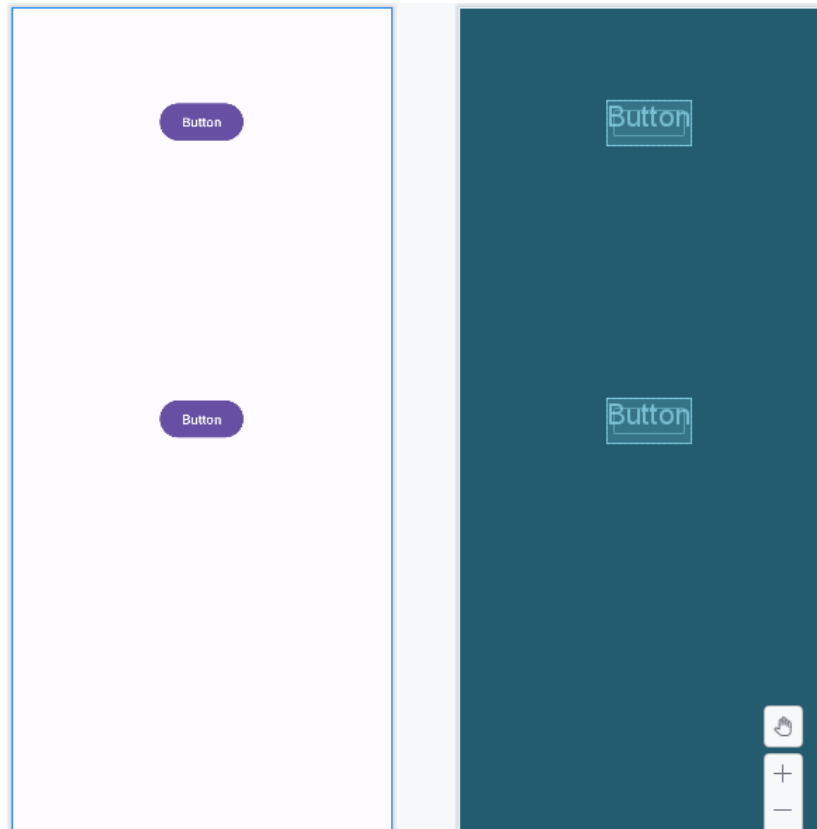
4.4. ConstraintLayout

Một ConstraintLayout là phần mô tả làm thế nào để một View nên được định vị trên màn hình tương đối với các phần tử khác trong layout. Được giới thiệu lần đầu tiên tại sự kiện Google I/O 2016.

Ta có thể xác định một Constraint cho một hay nhiều mặt của một View bằng cách đặt chế độ kết nối bất kỳ sau:

- Điểm neo nằm trên một View khác.
- Một cạnh của layout.
- Một guideline.

Android cung cấp các loại Constraint như: Relative positioning, Margins, Centering positioning, Visibility behavior, Dimension constraints, Chains. Layout này được dùng để thiết kế các màn hình Responsive không phụ thuộc vào độ phân giải.



```
<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">

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="100dp"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

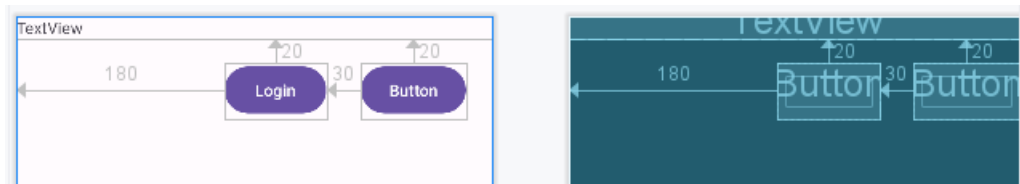
```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

4.5. RelativeLayout

RelativeLayout là loại layout cho phép sắp xếp các đối tượng (view) theo vị trí tương đối giữa các View khác trên giao diện, bao gồm cả đối tượng chứa nó. Thường nó dựa vào Id của các view khác để sắp xếp vị trí tương đối nên khi thiết kế giao diện có sử dụng RelativeLayout, ta cần phải đặt Id chuẩn xác trước rồi mới kéo và sắp xếp các view sau. Nếu ngược lại thì giao diện bị xáo trộn vì view sau không tham chiếu được với Id cũ đã đặt.



```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/txtNo1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/txtNo1"
        android:layout_alignParentStart="true"
        android:layout_marginStart="180dp"
        android:layout_marginTop="20dp"
        android:text="Login" />

    <Button
        android:id="@+id/button6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/txtNo1"
        android:layout_marginStart="30dp"
        android:layout_marginTop="20dp"
        android:layout_toEndOf="@+id/button5"
        android:text="Button" />

</RelativeLayout>

```

Nội dung 5: Tìm hiểu về hàm findViewById

5.1. Quy tắc đặt tên id trong Android (prefix)

Trong chương trình bài học, ta áp dụng quy tắc đặt tên sau:

Tên View	Tiền tố	Ví dụ
Button	btn	btnLogin
TextView	txv	txvUserName
Edit text: <ul style="list-style-type: none"> Nhập chuỗi: Plain Text Mật khẩu: Password ... 	edt	edtPass
RadioButton	rad	radKeyA
ListView	lsv	lsvDanhSach
...		

5.2. Ý nghĩa hàm findViewById

Trong lập trình Android, ta có hai nhiệm vụ: phân thiết kế giao diện và xử lý nghiệp vụ. Do vậy để tương tác với các View trên giao diện, ta cần thông qua thuộc tính Id để truy xuất dữ liệu => dùng hàm **findViewById** để tham chiếu.

Khi đặt Id cho các View, cần đặt tên gợi nhớ, có quy tắc. Những Id này được tự động lưu trong lớp R của Android. Vì vậy, để truy xuất đến View, ta cần dùng **R.id.idView**.

Nội dung 6: Tìm hiểu về các View cơ bản trong Android

6.1. Về TextView

Đặc điểm: là View chỉ được dùng để hiển thị dữ liệu, không cho phép chỉnh sửa dữ liệu (nhập liệu từ bàn phím).

Các thuộc tính cơ bản của TextView:

Thuộc tính	Mô tả
android:id	Id của TextView
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Xâu ký tự hiển thị trên giao diện

android:textcolor	Màu chữ
android:textsize	Cỡ chữ
android:backgroud	Màu nền
android:hint	Chữ gợi ý
...	

Giả sử ta có một TextView và đặt Id cho nó là txvMessage.

- Hàm findViewById trong file xử lý nghiệp vụ:

```
txvMessage = (TextView)findViewById(R.id.txvMessage)
```

- Để đặt giá trị xâu văn bản hiển thị trên giao diện:

- + Trong file XML:

```
android:text = “Đang học Android”
```

- + Trong code xử lý nghiệp vụ (java):

```
txvMessage.setText(“Đang học Android”)
```

6.2. Về EditText

Các view Edit Text được kế thừa từ TextView, nhưng khác là EditText View có thể được chỉnh sửa từ người dùng (nhập liệu từ bàn phím).

Các thuộc tính quan trọng của Edit Text là:

Thuộc tính	Mô tả
android:id	Id của EditText
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Xâu ký tự hiển thị trên giao diện
android:textcolor	Màu chữ
android:textsize	Cỡ chữ
android:backgroud	Màu nền
android:hint	Chữ gợi ý
android:inputType	Thiết lập loại dữ liệu được nhập để hiển thị bàn phím cho phù hợp.

...	
-----	--

Để lấy giá trị nội dung văn bản trên EditText, ta dùng hàm `getText()`:

`edtPass.getText().toString()`

6.3. Về Button

Ngoài tính thể hiện nội dung văn bản, hình ảnh thì Button còn cho phép nhận và phản hồi tương tác từ người dùng (là nhấn chuột). Id đặt cho Button là `btn`.

Các thuộc tính/sự kiện quan trọng như:

Thuộc tính/sự kiện	Mô tả
<code>android:id</code>	Id của Button
<code>android:layout_width</code>	Độ rộng
<code>android:layout_height</code>	Độ cao
<code>android:text</code>	Xâu ký tự hiển thị trên giao diện
<code>android:textcolor</code>	Màu chữ
<code>android:textsize</code>	Cỡ chữ
<code>android:background</code>	Màu nền
<code>android:onClick</code>	Gán sự kiện click cho Button
...	

Cách nhanh nhất để lắng nghe sự kiện nhấn vào Button là:

- Trên file XML, thêm thuộc tính “`android:onClick`”:

```
<Button
    android:onClick="login"
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Đăng nhập"
    android:textSize="25dp"
    android:layout_marginTop="30dp"
    android:layout_gravity="right" />
```

- Trên file code java xử lý nghiệp vụ, thêm phương thức xử lý tương ứng:

```
public void login(View view) {
    // lay du lieu tu form
    String username = edtUsername.getText().toString();
    String password = edtPassword.getText().toString();
    txtvKetQua.setText("");

    // kiem tra dang nhap
    if ((username.equals("tuan")) && (password.equals("1234"))) {
        txtvKetQua.setTextColor(Color.RED);
    }
}
```

```

        txvKetQua.setText("Thành công!");
    }
}

```

6.4. Về CheckBox

Là View cho phép người dùng chọn lựa nhiều tùy chọn, hình dạng thường thấy là ô vuông có dấu tích. Id đặt cho CheckBox là chk.

Các thuộc tính/sự kiện quan trọng như:

Thuộc tính/sự kiện	Mô tả
android:id	Id của CheckBox
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Xâu ký tự hiển thị trên giao diện
android:textcolor	Màu chữ
android:textsize	Cỡ chữ
android:background	Màu nền
android:checked	Thiết lập chọn/bỏ chọn checkbox (true/false)
...	

6.5. Về RadioButton

Khác với CheckBox cho phép người dùng chọn nhiều, View RadioButton lại cho người dùng chọn một tùy chọn (dựa trên cha chứa những tùy chọn). Id đặt cho RadioButton là rad.

Các thuộc tính/sự kiện quan trọng như:

Thuộc tính/sự kiện	Mô tả
android:id	Id của RadioButton
android:layout_width	Độ rộng
android:layout_height	Độ cao
android:text	Xâu ký tự hiển thị trên giao diện
android:textcolor	Màu chữ
android:textsize	Cỡ chữ

android:background	Màu nền
android:checked	Thiết lập chọn/bỏ chọn checkbox (true/false)
...	

6.6. Về ImageButton

View này là một button có thể chứa hình ảnh. Id đặt cho ImageButton là imgBtn. Button hoạt động như nào thì ImageButton hoạt động như vậy.

Các thuộc tính/sự kiện quan trọng như:

Thuộc tính/sự kiện	Mô tả
android:id	Id của ImageButton
android:layout_width	Độ rộng
android:layout_height	Độ cao
app:srcCompat	Thiết lập hình ảnh cho ImageButton

6.7. Về ImageView

ImageView dùng để hiển thị hình ảnh. Id đặt cho ImageView là img. Giống như ImageButton, hình ảnh cần phải có trong thư mục drawable để gán ảnh cho đối tượng.

Các thuộc tính/sự kiện quan trọng như:

Thuộc tính/sự kiện	Mô tả
android:id	Id của ImageButton
android:layout_width	Độ rộng
android:layout_height	Độ cao
app:srcCompat	Thiết lập hình ảnh cho ImageButton
scaleType	Thiết lập chế độ hiển thị hình ảnh

Nội dung 7: Câu hỏi lý thuyết và bài tập thực hành

Câu 1: Trình bày lịch sử ra đời hệ điều hành Android.

Câu 2: Tại sao vào thời điểm hệ điều hành Android ra đời, có nhiều nền tảng di động khác, vậy sao người ta lại lựa chọn Android để phát triển.

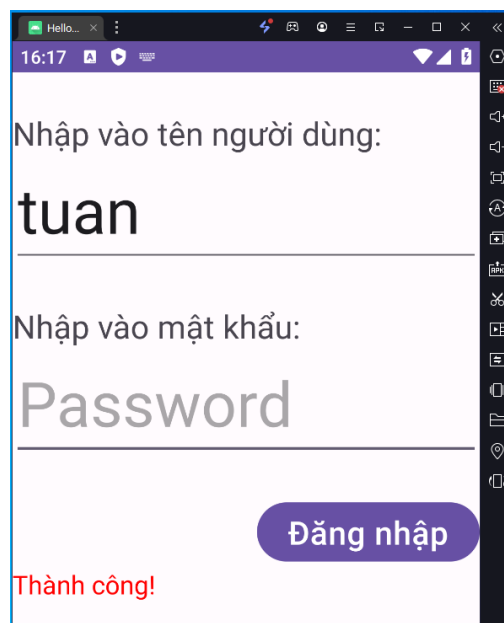
Câu 3: Trình bày về công cụ SDK Manager và cách sử dụng.

Câu 4: Trình bày nguyên tắc chọn min SDK khi tạo một dự án mới.

Câu 5: Tập tin Build.gradle có vai trò gì trong quá trình biên dịch phần mềm.

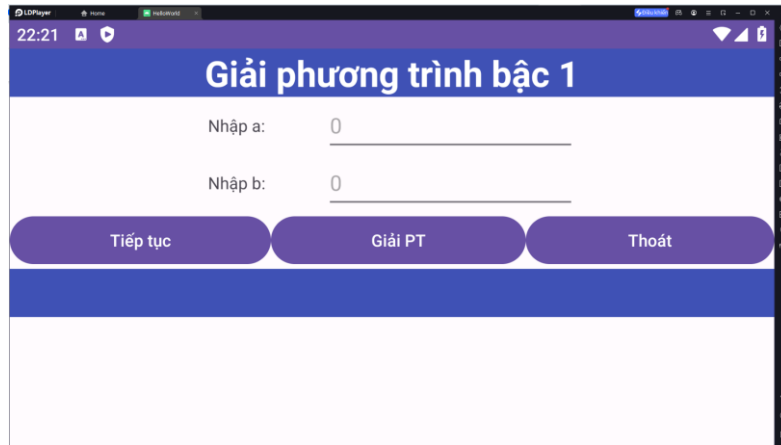
Câu 6: Bản chất tập tin .apk là gì.

Bài 1: Viết chương trình kiểm tra thông tin đăng nhập với giao diện cơ bản dưới đây:



Nút Đăng nhập: kiểm tra Username có là “tuan” và Password có là “1234” hay không. Nếu đúng thì hiển thị “Thành công!” lên một TextView ở dưới như trên hình minh họa.

Bài 2: Viết chương trình giải phương trình bậc 1 với giao diện dưới đây:



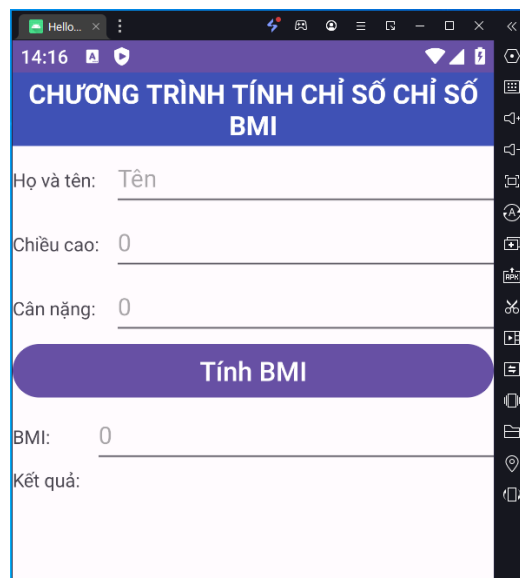
Nút Thoát: nhấn vào để thoát ứng dụng.

Nút Tiếp tục: nhấn vào để xóa dữ liệu cũ trên giao diện và focus vào EditText nhập a.

Nút Giải PT: nhấn vào để giải phương trình. Kết quả được hiển thị ở bên dưới nền màu xanh.

Ứng dụng được chạy trên màn hình nằm ngang.

Bài 3: Viết chương trình tính chỉ số cân đối cơ thể (Body Mass Index - BMI), tham khảo giao diện dưới đây:



Gọi w là khối lượng của một người (tính bằng kg) và h là chiều cao của người đó (tính bằng mét), ta có công thức tính chỉ số BMI như sau:

$$BMI = \frac{w}{h^2}$$

Từ giá trị của chỉ số BMI, ta phân loại ra thành những trường hợp như sau:

- BMI < 18: người gầy.
- BMI từ 18 đến 24.9: người bình thường.
- BMI từ 25 đến 29.9: người béo phì độ I.
- BMI từ 30 đến 34.9: người béo phì độ II.
- BMI trên 35: người béo phì độ III.

Tài liệu tham khảo

1. *Head First: Android Development* (2015), Dawn Griffiths, David Griffiths.
2. *Đề cương chi tiết học phần: Lập trình cho các thiết bị di động* (2019), Trường đại học Cần Thơ.
3. *Đề cương chi tiết học phần: Phát triển ứng dụng di động* (2024), Học viện Nông nghiệp Việt Nam.
4. *Giáo trình Phát triển ứng dụng di động cơ bản bản và nâng cao* (2019), chủ biên Lê Hoàn Sử.
5. Website <https://developer.android.com/>
6. Website: <https://www.geeksforgeeks.org/>

Hà Nội, ngày 08 tháng 03 năm 2024

Biên soạn và tổng hợp

Hoàng Văn Tuấn

----- **Hết** -----