

MÔN : CÔNG NGHỆ JAVA

Bài thực hành 8.1as : Xây dựng chương trình giải pt bậc 2 chạy trên android

I. Mục tiêu :

- Giúp SV làm quen với việc sử dụng môi trường lập trình trực quan Android Studio.
- Giúp SV làm quen với qui trình xây dựng chương trình Android dùng giao diện LCDUI (Limited Connected Device User Interface) để chạy trên thiết bị Android.

II. Nội dung :

- Dùng NetBeans để viết và chạy chương trình giải phương trình bậc 2 chạy trên Android dùng giao diện LCDUI.

III. Chuẩn đầu ra :

- Sinh viên nắm vững việc sử dụng môi trường lập trình trực quan Android Studio để viết các chương trình Android dùng giao diện LCDUI.
- Sinh viên nắm vững cấu trúc các thành phần của chương trình Android dùng giao diện LCDUI.

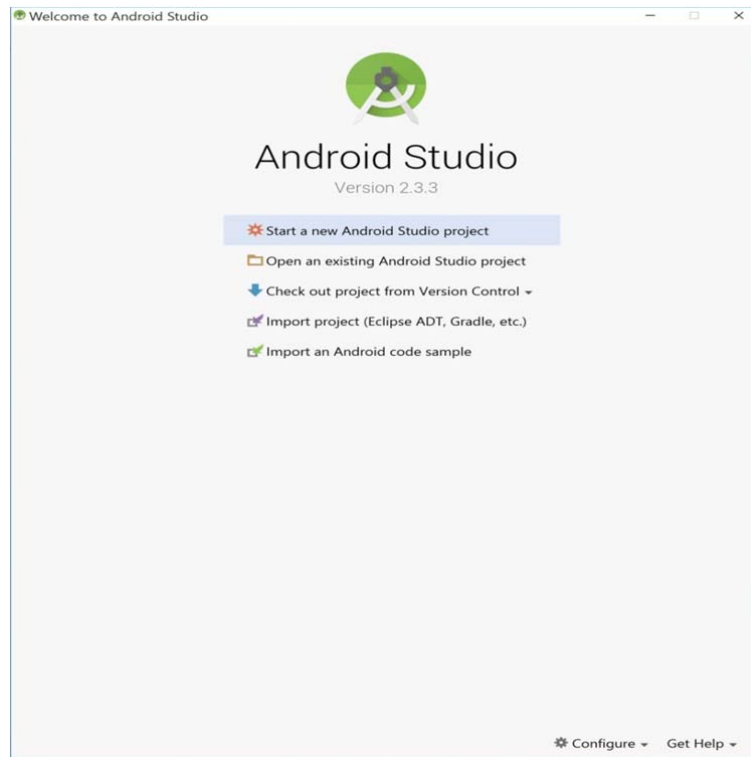
IV. Qui trình viết phần mềm bằng môi trường Android Studio :

IV.1 Cài đặt Android Studio, cài đặt và cấu hình Android SDK (nếu chưa cài) :

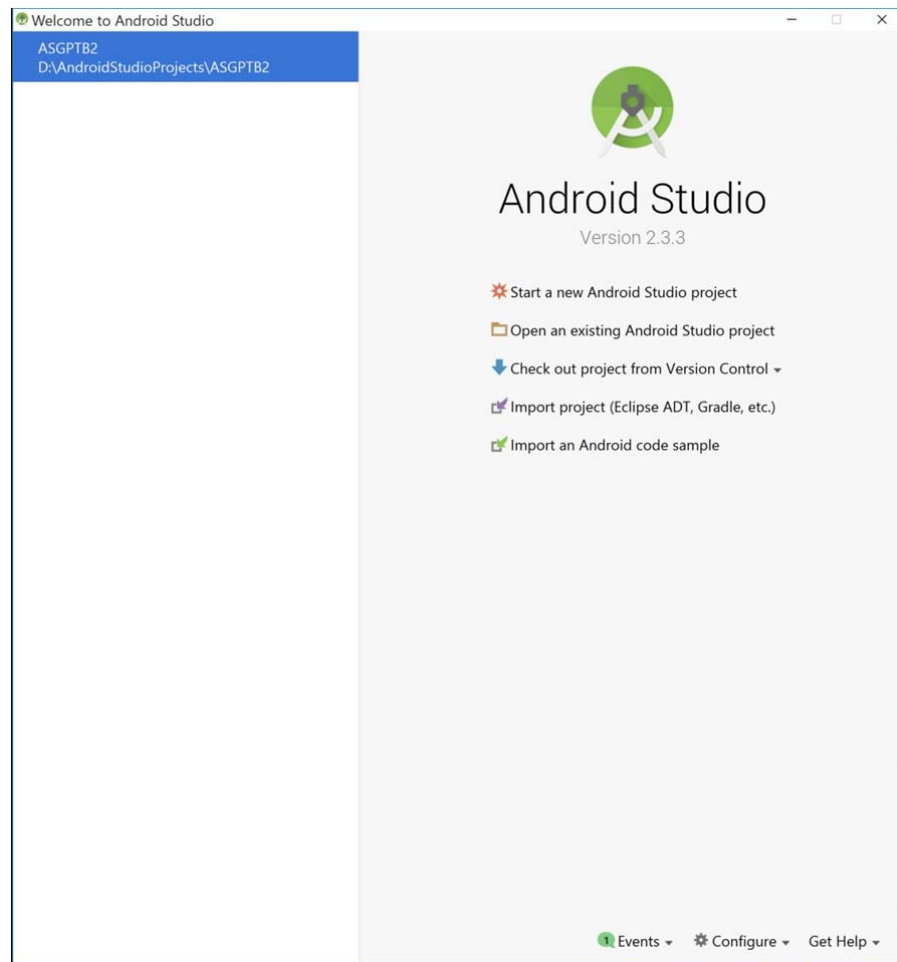
Thực hiện bài thực hành 8.0.

IV.2 Dùng Android Studio xây dựng ứng dụng giải phương trình bậc 2 :

1. Chạy Android Studio 2.3.3, nếu lần đầu được chạy trên máy thì cửa sổ ban đầu được hiển thị như sau :

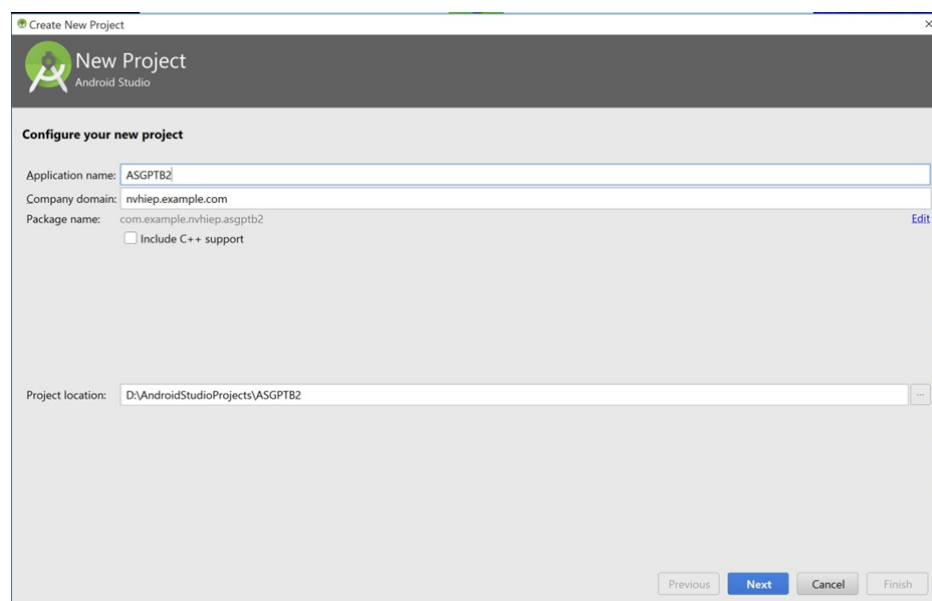


Còn nếu đã chạy nhiều lần rồi thì cửa sổ ban đầu được hiển thị như sau :

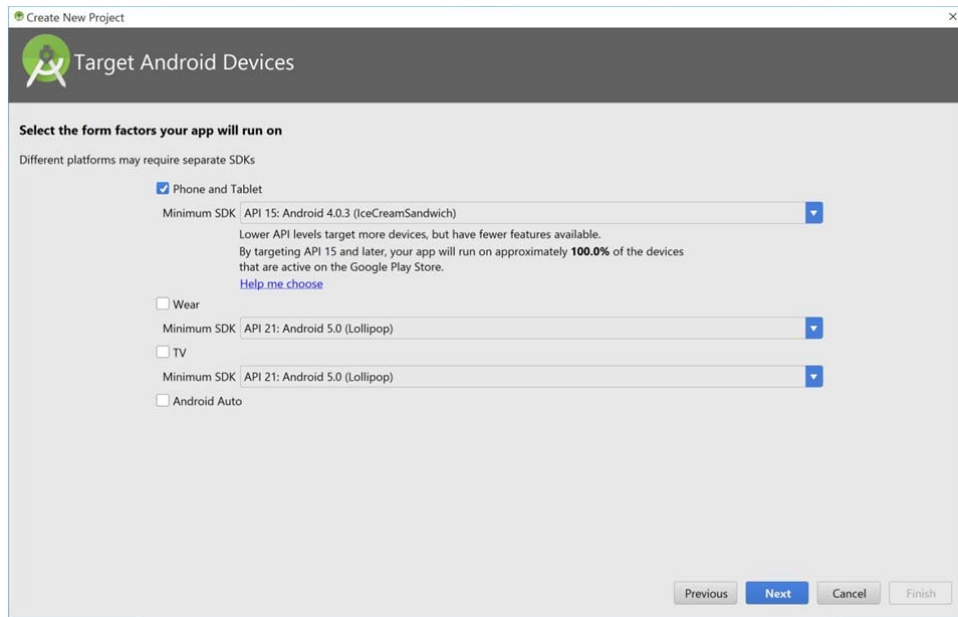


Ta thấy bên trái cửa sổ là danh sách từ 1 tới n project đã làm trước đây, nếu muốn tiếp tục làm việc với project nào thì chọn project đó để mở lại nó.

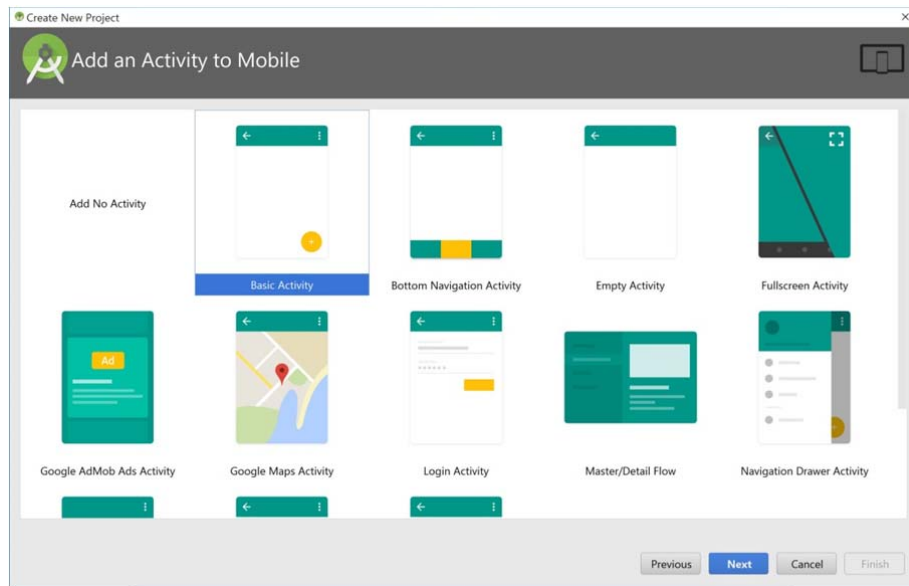
2. Trong trường hợp của mình, ta chọn mục “Start a new Android Studio project” để tạo mới project quản lý phần mềm mới do ta sắp xây dựng. Cửa sổ New Project sau đây sẽ hiển thị :



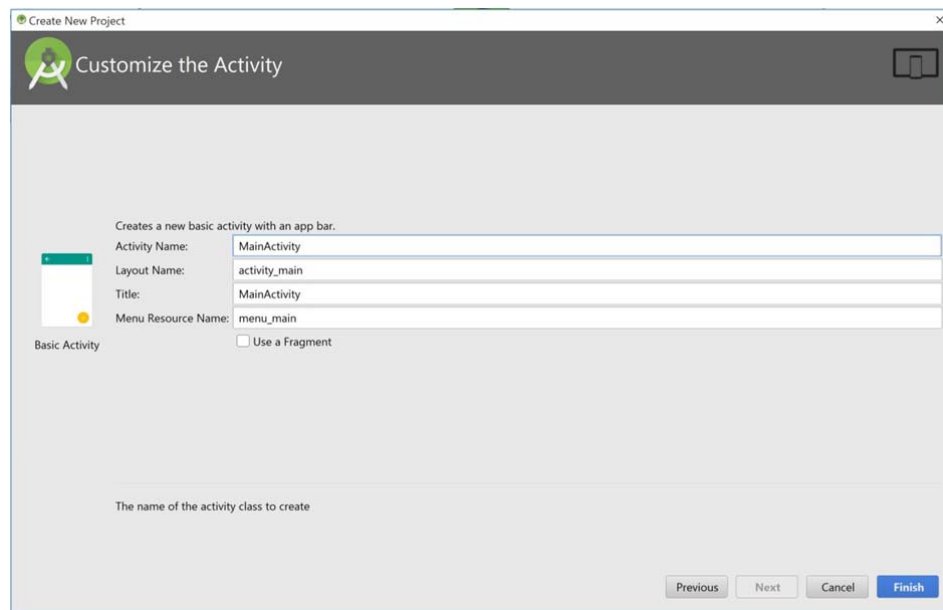
3. Ta nhập tên ứng dụng cần viết, tên domain của công ty, vị trí thư mục trên đĩa chứa project rồi click button Next để tiếp tục. Máy sẽ hiển thị cửa sổ sau



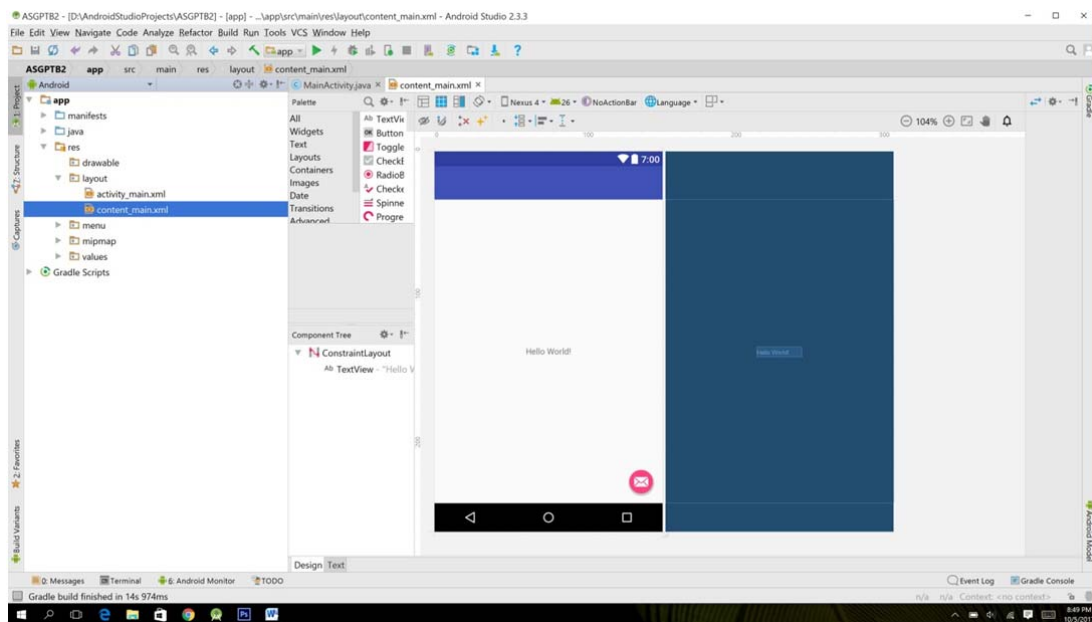
4. Máy yêu cầu ta chọn loại thiết bị dự định sẽ chạy chương trình của mình viết (Phone, Wear, TV hay Android Auto). Ở đây ta sẽ chọn mục Phone rồi chọn version SDK tối thiểu theo yêu cầu, nhưng tốt nhất là dùng version mặc định. Sau đó ta click button Next để tiếp tục, máy sẽ hiển thị cửa sổ yêu cầu chọn loại form giao diện cần dùng như sau :



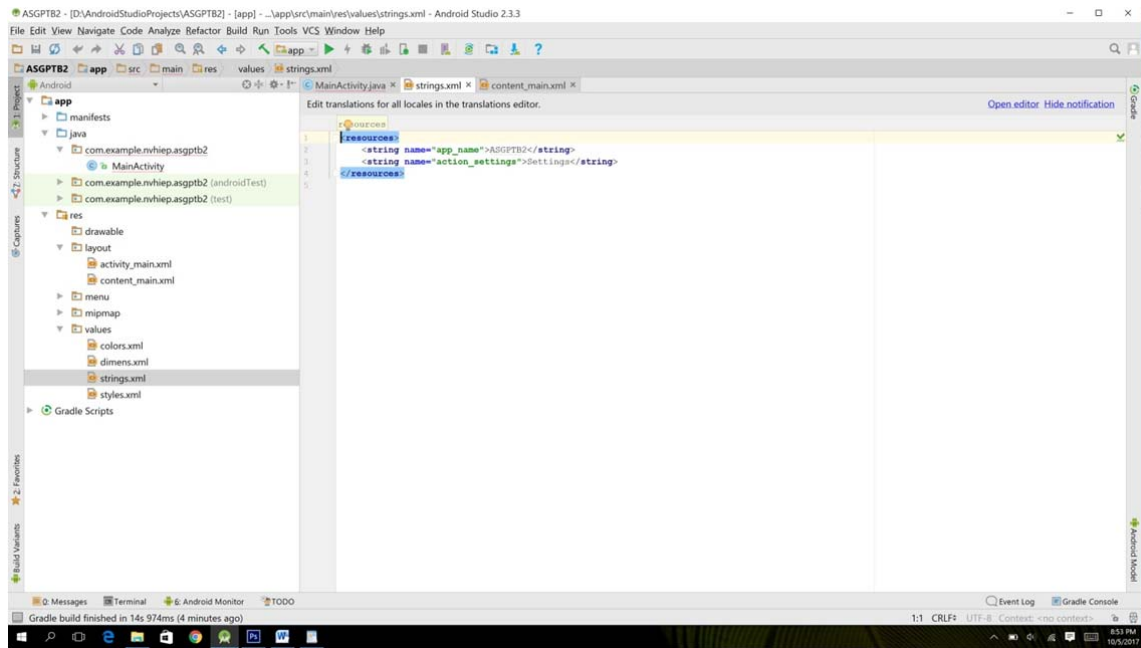
5. Do ta muốn viết code để tạo form giao diện nên chỉ cần chọn mục “Basic Activity” rồi click button Next để tiếp tục, máy sẽ hiển thị cửa sổ sau :



6. Máy yêu cầu ta khai báo tên form chính của ứng dụng cùng tên các đối tượng phục vụ nó. Hãy đặt tên cho form chính theo ngữ nghĩa chức năng rồi click button Finish để tiếp tục. Máy sẽ tạo project theo các thông tin mà ta thiết lập từ đầu đến giờ. Sau một thời gian nhất định (tùy cấu hình phần cứng máy), máy sẽ hiển thị cửa sổ làm việc như sau :



7. Ta nói rộng mục app.res.values ra rồi mở file strings.xml, máy sẽ hiển thị nội dung file này như sau :



8. Ta thêm các lệnh XML định nghĩa các tài nguyên chuỗi cần dùng trong chương trình như sau :

```
<resources>
    <string name="app_name">ASGPTB2</string>
    <string name="action_settings">Settings</string>
    <string name="strBatdau">Bắt đầu giải</string>
    <string name="strA">Nhập a:</string>
    <string name="strB">Nhập b:</string>
    <string name="strC">Nhập c:</string>
    <string name="strVonghiem">Phương trình vô nghiệm</string>
    <string name="strConghiem">Phương trình có 2 nghiệm thực :</string>
</resources>
```

Lưu ý chỉ thêm các lệnh màu xanh dương, còn các lệnh màu nâu phải giữ nguyên.

9. Mở rộng mục app.java.com.example.nvhiep.asgptb2 bạn sẽ thấy file MainActivity.java (tên file này là tên form chính của chương trình mà ta thiết lập ở bước 6). File này chứa mã nguồn của class miêu tả form chính. Hãy viết code cho file này để xây dựng giao diện cho form và thực hiện giải phương trình bậc 2 như sau :

```
package com.example.nvhiep.asgptb2;
//import các tài nguyên cần dùng (phần máy tạo sẵn)
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
//import các tài nguyên cần dùng (phần ta viết thêm vì code của ta cần dùng chúng)
import android.app.Activity;
import android.view.ViewGroup.*;
import android.widget.*;
import static android.view.ViewGroup.LayoutParams.FILL_PARENT;
import static android.view.ViewGroup.LayoutParams.WRAP_CONTENT;
import static android.widget.LinearLayout.VERTICAL;;
```

```

//đặc tả chi tiết chương trình Android giải phương trình bậc 2
public class MainActivity extends AppCompatActivity {
    //định nghĩa các đối tượng giao diện
    TextView lblA;
    TextView lblB;
    TextView lblC;
    EditText txtA;
    EditText txtB;
    EditText txtC;
    Button btnStart;
    TextView lblKetqua;
    TextView lblX1;
    TextView lblX2;
    String strVonghiem, strConghiem;
    String strA, strB, strC;
    String strBatdau;
    //định nghĩa các biến cần dùng
    double a, b, c;
    double delta;
    double x1, x2;

    //Hàm nhập 3 tham số a, b, c của phương trình
    private void NhapABC() {
        //mã hóa 3 tham số a, b, c từ chuỗi thành số thực
        a = Double.valueOf(txtA.getText().toString());
        b = Double.valueOf(txtB.getText().toString());
        c = Double.valueOf(txtC.getText().toString());
    }

    //Hàm giải phương trình bậc 2
    private void GiaiPT() {
        //tính biệt số delta
        delta = b * b - 4 * a * c;
        //kiểm tra biệt số delta
        if (delta >= 0) {
            //tính 2 nghiệm thực
            x1 = (-b + Math.sqrt(delta)) / 2 / a;
            x2 = (-b - Math.sqrt(delta)) / 2 / a;
        }
    }

    //Hàm xuất kết quả của phương trình bậc 2
    private void XuatKetqua() {
        //kiểm tra biệt số delta để ra quyết định xử lý
        if (delta >= 0) { //trường hợp có 2 nghiệm thực
            lblKetqua.setText(strConghiem);
            lblX1.setText("x1 = " + x1);
            lblX2.setText("x2 = " + x2);
        } else { //trường hợp vô nghiệm
            lblKetqua.setText(strVonghiem);
            lblX1.setText("");
            lblX2.setText("");
        }
    }
}

```

```

//tác vụ nhập dữ liệu, giải phương trình, hiển thị kết quả
public void btnStart_Click(View view) {
    //1. Nhập 3 tham số a, b, c của phương trình
    NhapABC();
    //2. Giải phương trình
    GiaiPT();
    //3. Hiển thị kết quả
    XuatKetqua();
}

//tác vụ onCreate của class Activity miêu tả ứng dụng
@Override
public void onCreate(Bundle savedInstanceState) {
    //gọi tác vụ của cha thực hiện trước
    super.onCreate(savedInstanceState);
    //xác định các chuỗi được dùng trong ứng dụng
    strA = getResources().getString(R.string.strA);
    strB = getResources().getString(R.string.strB);
    strC = getResources().getString(R.string.strC);
    strBatdau = getResources().getString(R.string.strBatdau);
    strVonghiem = getResources().getString(R.string.strVonghiem);
    strConghiem = getResources().getString(R.string.strConghiem);
    //tạo và kết hợp cửa sổ giao diện cho ứng dụng
    setContentView(CreateView());
}

//tác vụ tạo giao diện cho ứng dụng
private LinearLayout CreateView() {
    //tạo Layout LinearLayout miêu tả toàn bộ cửa sổ
    LinearLayout linearLayout = new LinearLayout(this);
    LayoutParams params = new LayoutParams(FILL_PARENT, WRAP_CONTENT);
    linearLayout.setLayoutParams(params);
    linearLayout.setOrientation(VERTICAL);
    //tạo 1 TableLayout chứa 3 hàng 2 cột để chứa 3 textbox nhập liệu a,b,c
    TableLayout tl = new TableLayout(this);
    tl.setPadding(0, 10, 0, 0);
    tl.setStretchAllColumns(true);
    //tạo hàng đầu để nhập tham số a
    TableRow tableRow = new TableRow(this);
    tableRow.setPadding(0, 10, 0, 0);
    //tạo 1 TextView để hiển thị thông báo yêu cầu nhập tham số a
    lblA = new TextView(this); lblA.setText(strA);
    tableRow.addView(lblA);
    //tạo 1 EditText để người dùng nhập tham số a
    txtA = new EditText(this);
    tableRow.addView(txtA);
    //add hàng nhập tham số a vào tableLayout
    tl.addView(tableRow);
    //tạo hàng thứ 2 để nhập tham số b
    tableRow = new TableRow(this);
    tableRow.setPadding(0, 10, 0, 0);
    //tạo 1 TextView để hiển thị thông báo yêu cầu nhập tham số b
    lblB = new TextView(this); lblB.setText(strB);
    tableRow.addView(lblB);

```

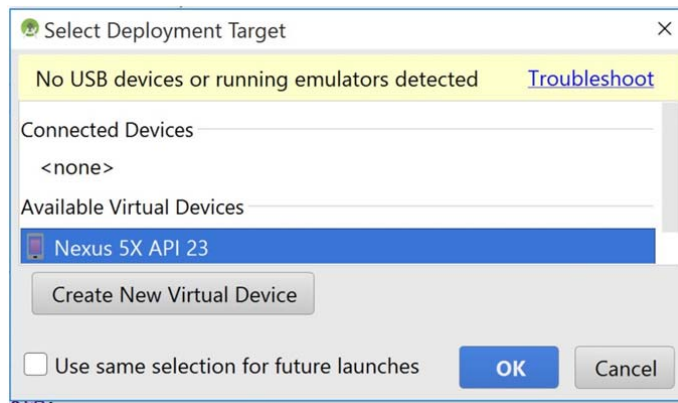
```

//tạo 1 EditText để người dùng nhập tham số b
txtB = new EditText(this);
tableRow.addView(txtB);
//add hàng nhập tham số b vào tableLayout
tl.addView(tableRow);
//tạo hàng thứ 3 để nhập tham số c
tableRow = new TableRow(this);
tableRow.setPadding(0, 10, 0, 0);
//tạo 1 TextView để hiển thị thông báo yêu cầu nhập tham số c
lblC = new TextView(this); lblC.setText(strC);
tableRow.addView(lblC);
//tạo 1 EditText để người dùng nhập tham số c
txtC = new EditText(this);
tableRow.addView(txtC);
//add hàng nhập tham số b vào tableLayout
tl.addView(tableRow);
//add table vào Layout
linearLayout.addView(tl);
//tạo button và add vào linearLayout
btnStart = new Button(this); btnStart.setText(strBatdau);
linearLayout.addView(btnStart);
//thiết lập đối tượng xử lý sự kiện Click vào button
btnStart.setOnClickListener(submitListener);
//tạo 3 TextView hiển thị kết quả và add vào linearLauout
lblKetqua = new TextView(this); linearLayout.addView(lblKetqua);
lblX1 = new TextView(this); linearLayout.addView(lblX1);
lblX2 = new TextView(this); linearLayout.addView(lblX2);
return linearLayout;
}

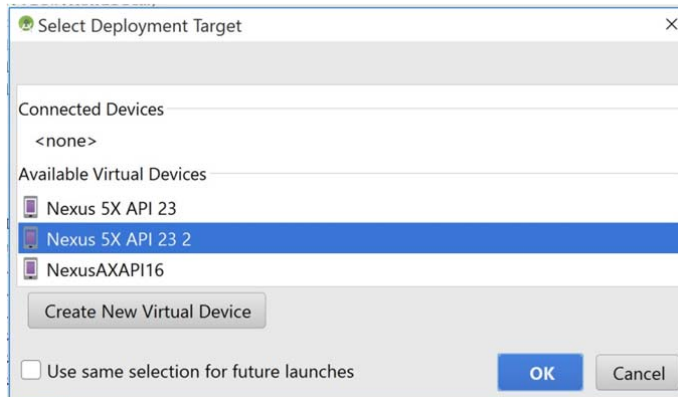
//tạo đối tượng xử lý sự kiện Click chuột trên button
private View.OnClickListener submitListener = new View.OnClickListener() {
    public void onClick(View view) {
        //gọi hàm giải phương trình bậc 2
        btnStart_Click(view);
    }
};
}

```

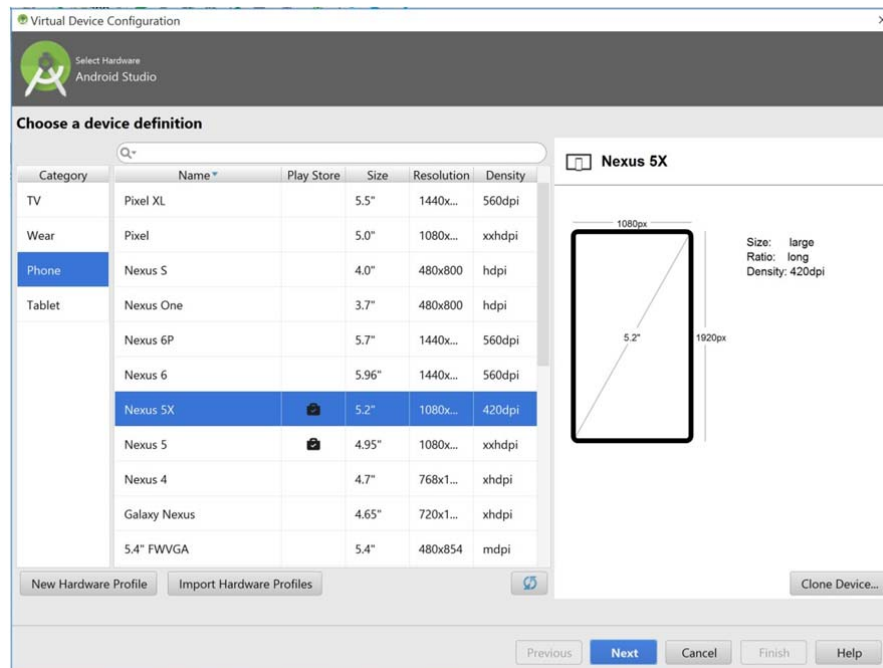
10. Chọn menu Build.Rebuild Project để dịch chương trình, nếu có lỗi máy sẽ báo và ta phải sửa lỗi cho đến hết.
11. Chọn menu Run.Run app để chạy thử chương trình. Nếu ta chưa nối smartphone với máy tính (thông qua cổng USB) thì máy sẽ hiển thị cửa sổ sau :



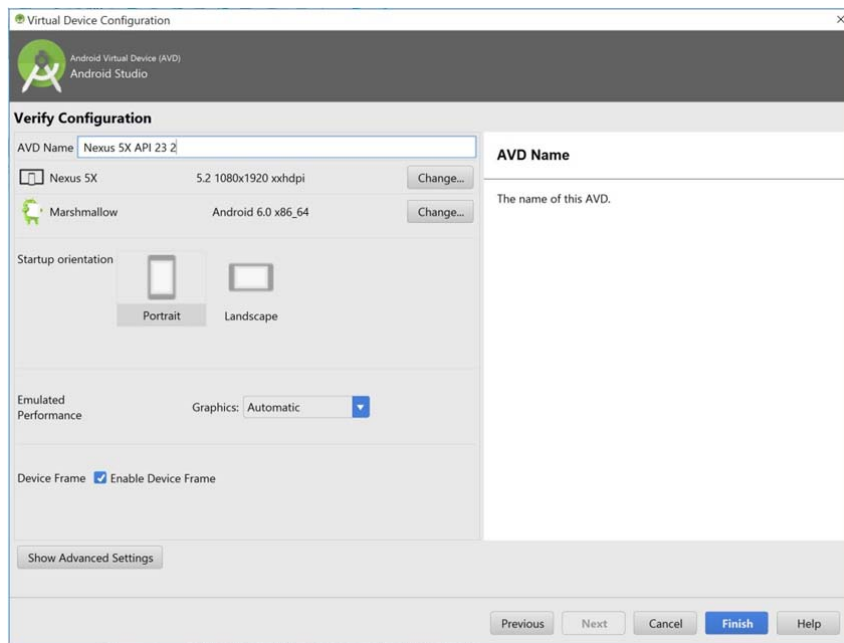
Còn nếu ta đã nối smartphone với máy tính (thông qua cổng USB) thì máy sẽ không hiển thị thông báo "No USB devices..." như cửa sổ sau :



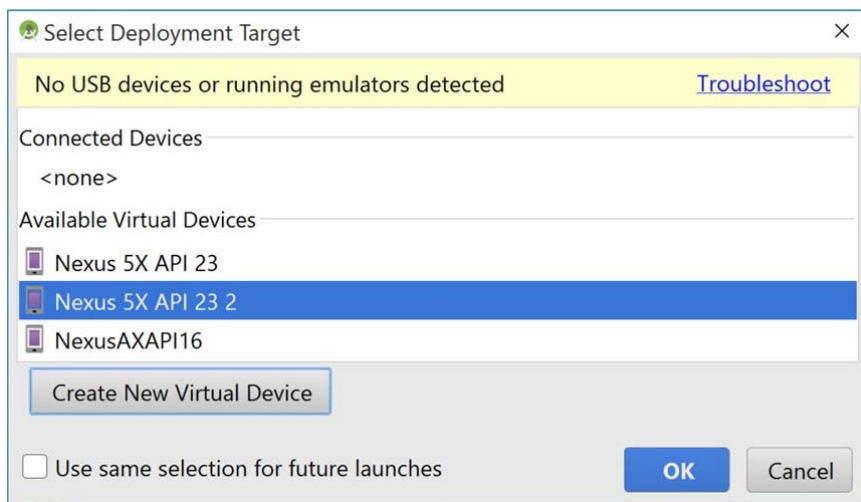
12. Chọn 1 thiết bị ảo để chạy chương trình trong danh sách "Available Virtual Devices". Nếu không có AVD nào thích hợp với yêu cầu của mình thì click vào button "Create New Virtual Device" để tạo 1 AVD mới theo yêu cầu của mình, máy sẽ hiển thị cửa sổ sau :



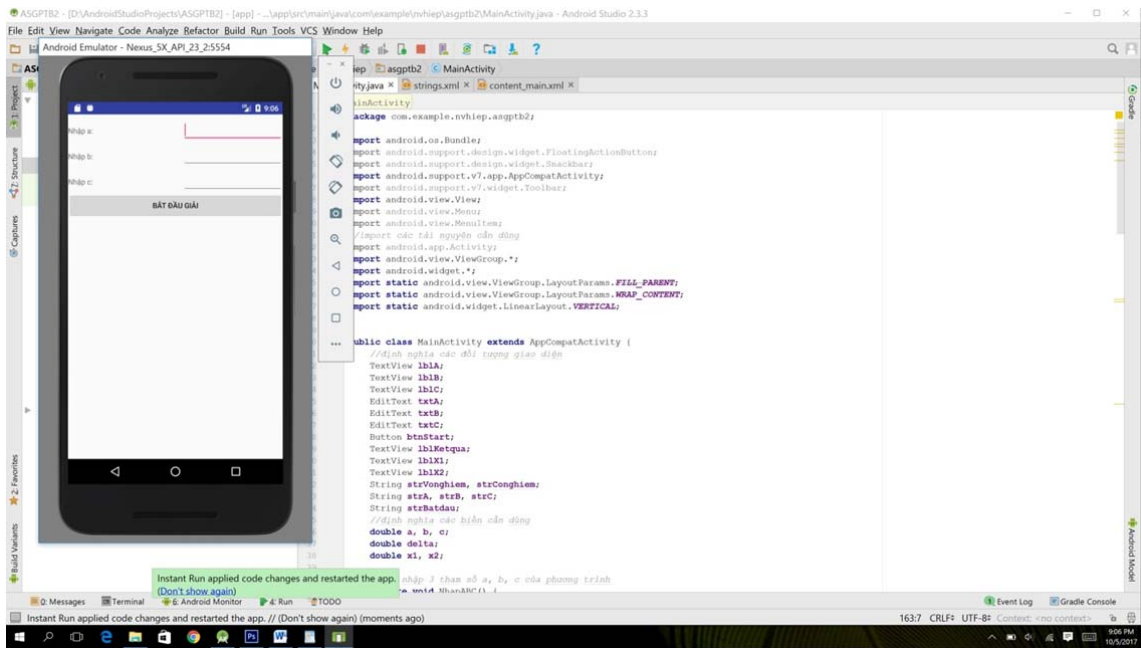
13. Hãy duyệt và chọn 1 thiết bị ảo trong danh sách, nếu không có thiết bị nào phù hợp với yêu cầu chuyên biệt của mình thì click button "New Hardware Profile" để tạo AVD mới. Click button Next để máy chạy tiếp và hiển thị cửa sổ sau :



14. Hãy kiểm tra lại thông tin chi tiết về thiết bị AVD mà mình vừa chọn, nếu đúng thì click button Finish để máy quay về cửa sổ chọn thiết bị AVD như sau :

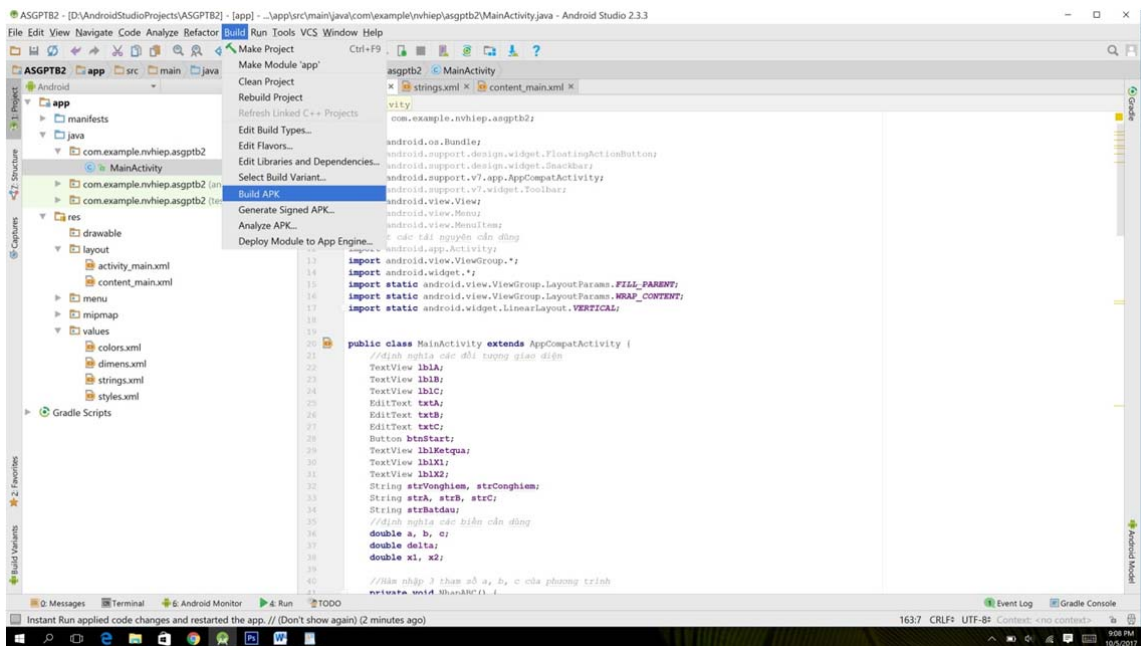


15. Hãy chọn thiết bị AVD trong danh sách và click button OK để chạy chương trình của mình trên thiết bị giả lập này.

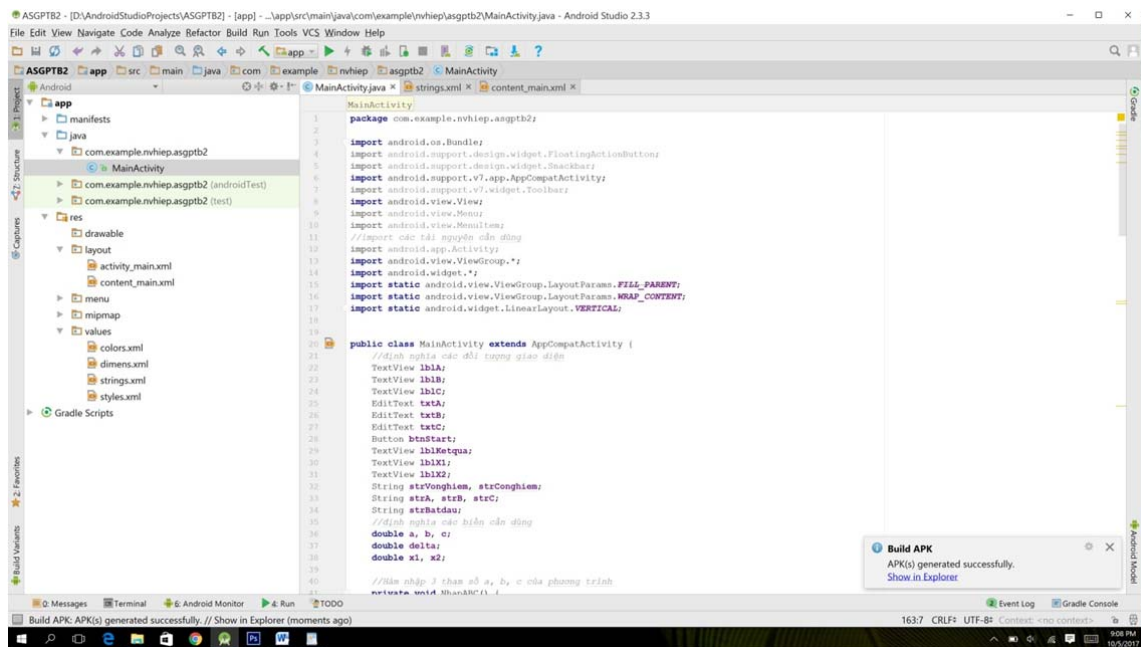


16. Sau một lúc xử lý, máy hiển thị thiết bị AVD và chạy chương trình của ta trên bị thiết giả lập này. Hãy kiểm thử các chức năng của chương trình, nếu chưa đúng với yêu cầu thì đóng cửa sổ giả lập, sửa chữa mã nguồn và chạy lại...

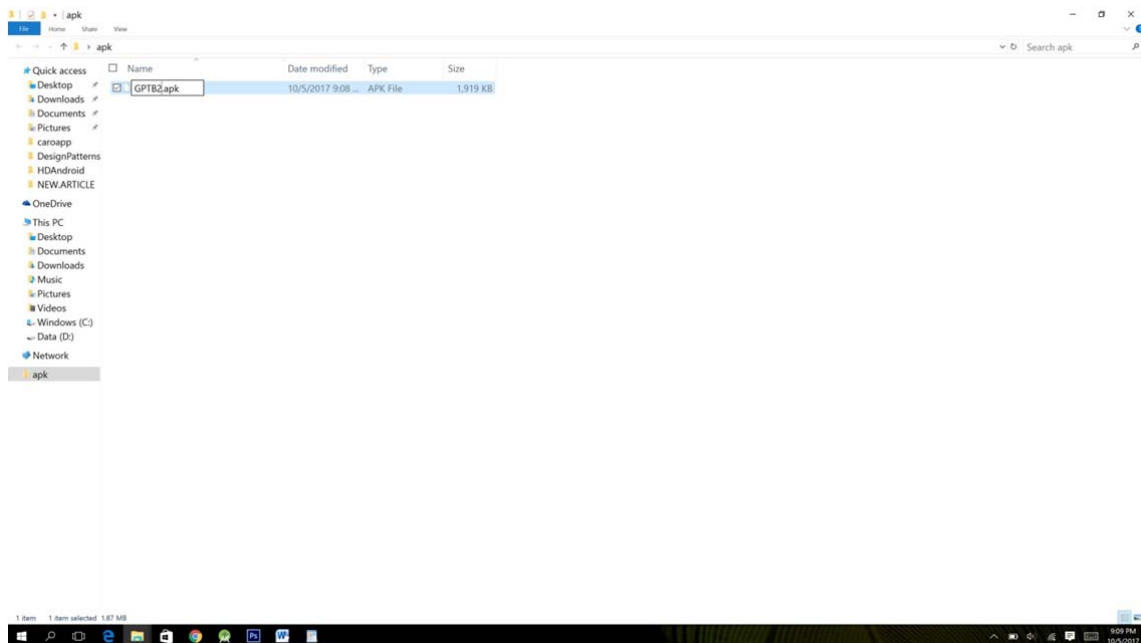
17. Nếu chương trình chạy đúng, ta cũng đóng cửa sổ AVD lại, chọn menu Build.Build APK để máy dịch thật chương trình ra định dạng *.apk theo qui định của Android.



18. Sau khi tạo xong file chương trình mã máy *.apk theo qui định của Android, máy sẽ hiển thị thông báo ở góc dưới phải của cửa sổ chính như sau :



19. Ta click vào link "Show in Explorer" để hiển thị file kết quả. Đặt lại tên file chương trình theo ngữ nghĩa chức năng của nó :

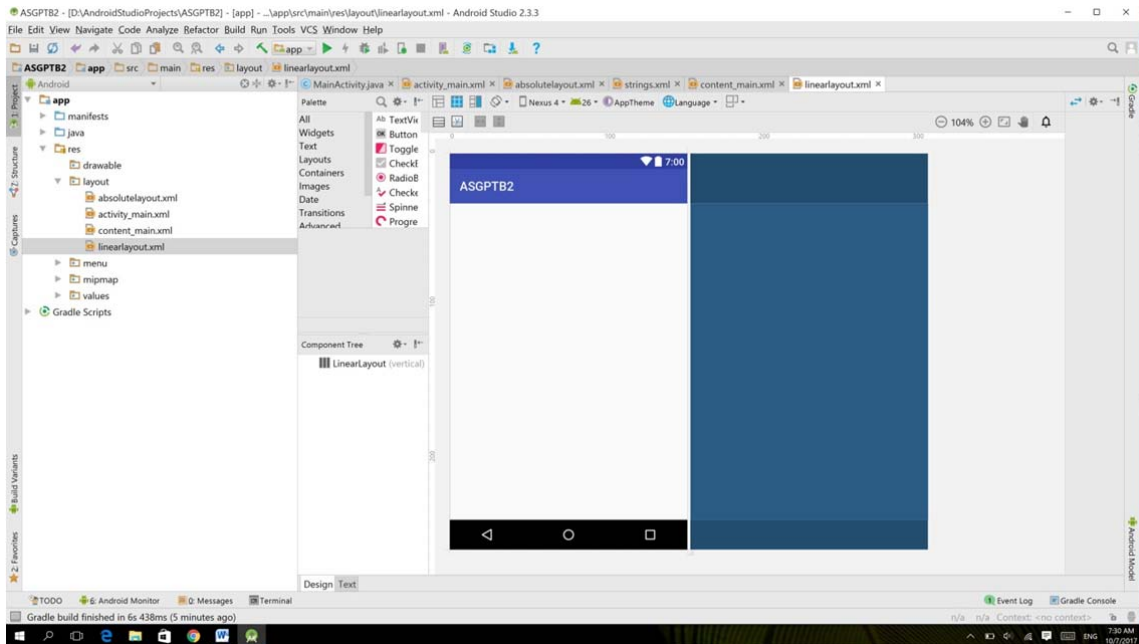


20. Kết nối điện thoại Android của mình vào máy tính (thường qua cổng USB), copy file *.apk vào thư mục thích hợp trên điện thoại.

21. Trên điện thoại, hãy cài đặt chương trình vừa copy và chạy thử trực tiếp trên điện thoại xem nó hoạt động đúng yêu cầu không.

22. Quay lại Project ASGPTB2 đã làm trong các bước trước, mở rộng mục app.res.layout, bạn thấy mục này có sẵn file activity_main.xml. Đây là file XML đặc tả giao diện ứng dụng do máy tạo mặc định, nhưng chúng ta không dùng nó mà lập trình tường minh bằng mã nguồn Java trong hàm onCreateView(). Việc lập trình tường minh giao diện ứng dụng tốn nhiều thời gian, công sức. Do đó ta thử dùng XML đặc tả giao diện cho ứng dụng giải phương trình bậc 2.

23. Ấn phải chuột vào mục layout, chọn chức năng New.Layout resource file để hiển thị cửa sổ "New resource file". Nhập vào mục "File name" tên của file XML cần tạo, thí dụ là linearlayout (lưu ý chỉ được chữ thường), rồi chọn button OK để máy tạo file XML theo yêu cầu. Máy sẽ tạo layout mới theo yêu cầu và hiển thị cửa sổ trực quan của nó như sau :



24. Click phải chuột trên hình trực quan của layout (phần tử đồ họa hình điện thoại có chữ ASGPBT2) để hiển thị menu liên quan, chọn mục "Goto XML" để mở cửa sổ soạn mã XML đặc tả giao diện ứng dụng. Nếu muốn các đối tượng được phân bố theo dạng LinearLayout y như đoạn code Java ở phần trước thì ta có thể đặc tả đoạn XML như sau :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dip"
        android:orientation="horizontal"
        android:weightSum="4"
        >
        <TextView
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/strA" />
        <EditText android:id="@+id/txtA"
            android:textSize="14sp"
            android:layout_weight="3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"> <requestFocus/> </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
```

```

android:layout_height="35dip"
android:orientation="horizontal"
android:weightSum="4"
>
    <TextView
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strB" />
    <EditText android:id="@+id/txtB"
        android:textSize="14sp"
        android:layout_weight="3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="35dip"
    android:orientation="horizontal"
    android:weightSum="4"
    >
        <TextView
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/strC" />
        <EditText android:id="@+id/txtC"
            android:textSize="14sp"
            android:layout_weight="3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

<Button
    android:layout_width="wrap_content"
    android:layout_height="35dip"
    android:text="@string/strBatdau"
    android:layout_gravity="center_horizontal"
    android:onClick="btnStart_Click"
    />
<TextView android:id="@+id/lblKetqua"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
<TextView android:id="@+id/lblX1"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
<TextView android:id="@+id/lblX2"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
</LinearLayout>

```

25. Mở lại cửa sổ soạn mã nguồn của form chính của ứng dụng. Hiệu chỉnh lại hàm onCreate() như sau :

```
//tác vụ onCreate của class Activity miêu tả ứng dụng
@Override
public void onCreate(Bundle savedInstanceState) {
    //gọi tác vụ của cha thực hiện trước
    super.onCreate(savedInstanceState);
    //xác định các chuỗi được dùng trong ứng dụng
    strA = getResources().getString(R.string.strA);
    strB = getResources().getString(R.string.strB);
    strC = getResources().getString(R.string.strC);
    strBatdau = getResources().getString(R.string.strBatdau);
    strVonghiem = getResources().getString(R.string.strVonghiem);
    strConghiem = getResources().getString(R.string.strConghiem);
    //tạo và kết hợp cửa sổ giao diện cho ứng dụng
    setContentView(R.layout.linearlayout);
    //xác định các đối tượng giao diện cần truy xuất
    txtA = (EditText) findViewById(R.id.txtA);
    txtB = (EditText) findViewById(R.id.txtB);
    txtC = (EditText) findViewById(R.id.txtC);
    lblKetqua = (TextView) findViewById(R.id.lblKetqua);
    lblX1 = (TextView) findViewById(R.id.lblX1);
    lblX2 = (TextView) findViewById(R.id.lblX2);
}
```

Lưu ý chỉ thêm các lệnh màu xanh dương, còn các lệnh màu nâu phải giữ nguyên.

26. Bạn có thể xóa đoạn code định nghĩa hàm CreateView và biến submitListener (vì không cần nữa) rồi thực hiện lại các bước 10 đến 21 để biên dịch lại chương trình, chạy thử nó trên môi trường giả lập và cài đặt chương trình vào thiết bị thực.
27. Nếu muốn bố trí các đối tượng giao diện theo kiểu AbsoluteLayout, ấn phải chuột vào mục layout, chọn chức năng New.Layout resource file để hiển thị cửa sổ "New resource file". Nhập vào mục "File name" tên của file XML cần tạo, thí dụ là absolutelayout (lưu ý chỉ được chữ thường), rồi chọn button OK để máy tạo file XML theo yêu cầu. Máy sẽ tạo layout mới theo yêu cầu và hiển thị cửa sổ trực quan của nó y như ở bước 23.
28. Soạn nội dung file như sau để đặc tả giao diện ứng dụng gồm các đối tượng được phân bố theo dạng AbsoluteLayout :

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="35dip"
        android:layout_y="5dip"
        android:layout_x="5dip"
        android:text="@string/strA" />
    <EditText android:id="@+id/txtA"
        android:layout_width="220dip"
        android:layout_height="35dip"
        android:layout_y="5dip"
        android:layout_x="85dip"
```



```

        android:textSize="14sp"
    />

```

```

    <TextView
        android:layout_width="100dip"
        android:layout_height="35dip"
        android:layout_y="40dip"
        android:layout_x="5dip"
        android:text="@string/strB" />
    <EditText android:id="@+id/txtB"
        android:layout_weight="1"
        android:layout_width="220dip"
        android:layout_height="35dip"
        android:layout_y="40dip"
        android:layout_x="85dip"
        android:textSize="14sp"
    />

```

```

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="35dip"
        android:layout_y="75dip"
        android:layout_x="5dip"
        android:text="@string/strC" />
    <EditText android:id="@+id/txtC"
        android:layout_weight="1"
        android:layout_width="220dip"
        android:layout_height="35dip"
        android:layout_y="75dip"
        android:layout_x="85dip"
        android:paddingTop="2dip"
        android:textSize="14sp"
    />

```

```

    <Button
        android:layout_y="110dip"
        android:layout_x="115dip"
        android:layout_width="wrap_content"
        android:layout_height="35dip"
        android:text="@string/strBatdau"
        android:layout_gravity="center_horizontal"
        android:onClick="btnStart_Click"
    />
    <TextView android:id="@+id/lblKetqua"
        android:layout_y="145dip"
        android:layout_x="5dip"
        android:layout_width="fill_parent"
        android:layout_height="25dip"
        android:text="" />
    <TextView android:id="@+id/lblX1"
        android:layout_y="180dip"
        android:layout_x="5dip"
        android:layout_width="fill_parent"
        android:layout_height="25dip"
    />

```



```

android:text="" />
<TextView android:id="@+id/lblX2"
android:layout_y="215dip"
android:layout_x="5dip"
android:layout_width="fill_parent"
android:layout_height="25dip"
android:text="" />
</AbsoluteLayout>

```

29. Mở lại cửa sổ soạn mã nguồn của ứng dụng. Hiệu chỉnh lại hàng lệnh gọi hàm setContentView() trong hàm onCreate() như sau :

```

//tạo và kết hợp cửa sổ giao diện cho ứng dụng
setContentView(R.layout.absolutelayout);

```

Lưu ý chỉ hiệu chỉnh nội dung màu xanh dương, còn phần màu nâu phải giữ nguyên.

30. Bạn có thể xóa đoạn code định nghĩa hàm CreateView và biến submitListener (vì không cần nữa) rồi thực hiện lại các bước 10 đến 21 để biên dịch lại chương trình, chạy thử nó trên môi trường giả lập và cài đặt chương trình vào thiết bị thực.

31. Nếu muốn bố trí các đối tượng giao diện theo kiểu TableLayout, ấn phải chuột vào mục layout, chọn chức năng New.Layout resource file để hiển thị cửa sổ "New resource file". Nhập vào mục "File name" tên của file XML cần tạo, thí dụ là tablelayout (lưu ý chỉ được chữ thường), rồi chọn button OK để máy tạo file XML theo yêu cầu. Máy sẽ tạo layout mới theo yêu cầu và hiển thị cửa sổ trực quan của nó y như ở bước 23.

32. Soạn nội dung file như sau để đặc tả giao diện ứng dụng gồm các đối tượng được phân bố theo dạng TableLayout :

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    >
    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="20dp"
        android:padding="2dip" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/strA" />
        <EditText android:id="@+id/txtA"
            android:textSize="14sp"
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="20dp"
        android:padding="2dip" >
        <TextView
            android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:text="@string/strB" />
        <EditText android:id="@+id/txtB"
        android:textSize="14sp"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        <requestFocus/>
    </TableRow>
    <TableRow
    android:id="@+id/tableRow3"
    android:layout_marginTop="5dp"
    android:layout_width="wrap_content"
    android:layout_height="20dp"
    android:padding="2dip" >
        <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strC" />
        <EditText android:id="@+id/txtC"
        android:textSize="14sp"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    </TableRow>

    <TableRow
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:padding="2dip" >
        <Button
        android:layout_width="wrap_content"
        android:layout_height="35dip"
        android:text="@string/strBatdau"
        android:layout_weight="1"
        android:layout_column="1"
        android:layout_gravity="center_horizontal"
        android:onClick="btnStart_Click"
        />
    </TableRow>
    <TableRow
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:padding="2dip" >
        <TextView android:id="@+id/lblKetqua"
        android:layout_width="wrap_content"
        android:layout_span="2"
        android:layout_height="25dip"
        android:text="" />
    </TableRow>
    <TableRow
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:padding="2dip" >

```

```

        <TextView android:id="@+id/lblX1"
            android:layout_span="2"
            android:layout_width="wrap_content"
            android:layout_height="25dip"
            android:text="" />
    </TableRow>
    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="25dp"
        android:padding="2dip" >
        <TextView android:id="@+id/lblX2"
            android:layout_span="2"
            android:layout_width="wrap_content"
            android:layout_height="25dip"
            android:text="" />
    </TableRow>
</TableLayout>

```

33. Mở lại cửa sổ soạn mã nguồn của ứng dụng. Hiệu chỉnh lại hàng lệnh gọi hàm setContentView() trong hàm onCreate() như sau :

```

//tạo và kết hợp cửa sổ giao diện cho ứng dụng
setContentView(R.layout.tablelayout);

```

Lưu ý chỉ hiệu chỉnh nội dung màu xanh dương, còn phần màu nâu phải giữ nguyên.

34. Bạn có thể xóa đoạn code định nghĩa hàm CreateView và biến submitListener (vì không cần nữa) rồi thực hiện lại các bước 10 đến 21 để biên dịch lại chương trình, chạy thử nó trên môi trường giả lập và cài đặt chương trình vào thiết bị thực.
35. Nếu muốn bố trí các đối tượng giao diện theo kiểu GridLayout, ấn phải chuột vào mục layout, chọn chức năng New.Layout resource file để hiển thị cửa sổ "New resource file". Nhập vào mục "File name" tên của file XML cần tạo, thí dụ là gridlayout (lưu ý chỉ được chữ thường), rồi chọn button OK để máy tạo file XML theo yêu cầu. Máy sẽ tạo layout mới theo yêu cầu và hiển thị cửa sổ trực quan của nó y như ở bước 23.
36. Soạn nội dung file như sau để đặc tả giao diện ứng dụng gồm các đối tượng được phân bố theo dạng GridLayout :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="135dip"
        >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/strA" />
        <EditText android:id="@+id/txtA"
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strB" />
        <EditText android:id="@+id/txtB"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        <requestFocus/>

        <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strC" />
        <EditText android:id="@+id/txtC"
        android:layout_weight="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    </LinearLayout>

    <Button
    android:layout_width="wrap_content"
    android:layout_height="35dip"
    android:text="@string/strBatdau"
    android:layout_gravity="center_horizontal"
    android:onClick="btnStart_Click"
    />
    <TextView android:id="@+id/lblKetqua"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
    <TextView android:id="@+id/lblX1"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
    <TextView android:id="@+id/lblX2"
    android:layout_width="wrap_content"
    android:layout_height="25dip"
    android:text="" />
</LinearLayout>

```

37. Mở lại cửa sổ soạn mã nguồn của ứng dụng. Hiệu chỉnh lại hàng lệnh gọi hàm setContentView() trong hàm onCreate() như sau :

```

//tạo và kết hợp cửa sổ giao diện cho ứng dụng
setContentView(R.layout.gridlayout);

```

Lưu ý chỉ hiệu chỉnh nội dung màu xanh dương, còn phần màu nâu phải giữ nguyên.

38. Bạn có thể xóa đoạn code định nghĩa hàm CreateView và biến submitListener (vì không cần nữa) rồi thực hiện lại các bước 10 đến 21 để biên dịch lại chương trình, chạy thử nó trên môi trường giả lập và cài đặt chương trình vào thiết bị thực.