

Bài tập 8: Các kiểu lập trình sự kiện trong Android

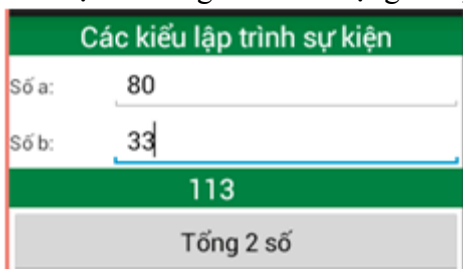
Bài tập này Tôi sẽ trình bày 6 kiểu lập trình sự kiện trong Android. Tôi sẽ sử dụng một số Control căn bản để Demo, đặc biệt là **Button** dùng để tạo sự kiện.

1. **Onclick in XML**
2. **Inline anonymous listener**
3. **Activity is listener**
4. **Listener in variable**
5. **Explicit listener class**
6. **View Subclassing**

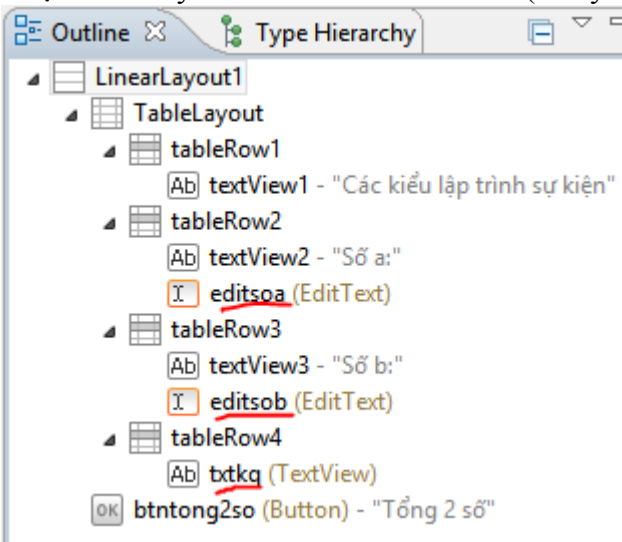
- Tôi sẽ lần lượt đưa ra 6 ví dụ khác nhau cho 6 kiểu lập trình sự kiện ở trên, các bạn hãy cố gắng theo dõi, nó rất quan trọng để làm các bài tập tiếp theo.

1. Onclick in XML:

- Ví dụ 1: Đơn giản chỉ là cộng 2 số, bạn thiết kế giao diện như bên dưới:



- Khi nhấn vào nút “Tổng 2 số”, chương trình sẽ xuất kết quả như hình bên trên : $80+33=113$
- Bạn xem Layout Outline để dễ thiết kế (chú ý là bạn có thể bỏ LinearLayout1 đi):



- Chú ý là ta sử dụng Onclick in XML:

```
<Button
    android:id="@+id/btntong2so"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="btn_tong2so"
    android:text="Tổng 2 số" />
```

Trong đoạn lệnh ở trên thì ta sử dụng **android:onClick="btn_tong2so"**, tức là ta đã gán một sự kiện click cho Button này, sự kiện này tên là **btn_tong2so**. Ta cần khai báo một hàm **btn_tong2so** ở trong Activity class như hình bên dưới:

```

10 public class MainActivity extends Activity {
12*   protected void onCreate(Bundle savedInstanceState) {
16*   public void btn_tong2so(View v)
17   {
18       EditText edita=(EditText)
19           findViewById(R.id.editsoa);
20       int a=Integer.parseInt(edita.getText()+"");
21
22       EditText editb=(EditText) findViewById(R.id.editsob);
23
24       int b=Integer.parseInt(editb.getText()+"");
25
26       TextView txtkq=(TextView) findViewById(R.id.txtkq);
27       txtkq.setText((a+b)+"");
28   }
30*   public boolean onCreateOptionsMenu(Menu menu) {}
35 }

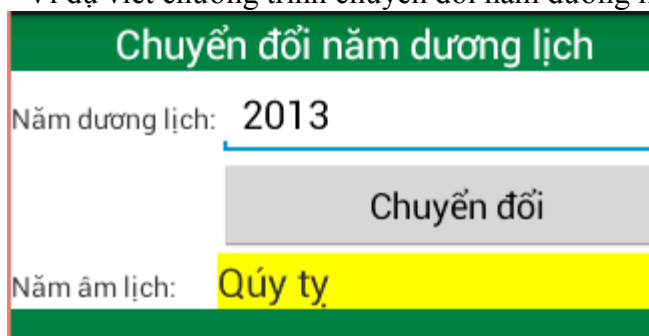
```

Khi chạy ứng dụng bạn sẽ được kết quả như bên dưới:



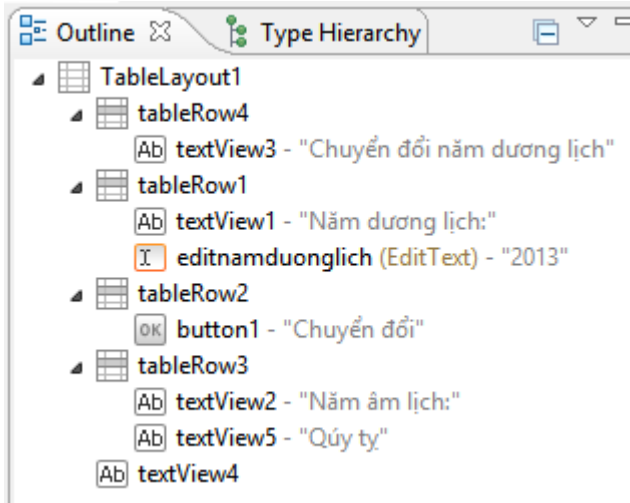
2. Inline anonymous listener

- Ví dụ viết chương trình chuyển đổi năm dương lịch qua năm âm lịch như hình bên dưới:



- Khi người sử dụng nhập vào EditText giá trị là 1 năm Dương Lịch bất kỳ nào đó rồi nhấn nút “Chuyển đổi”, chương trình sẽ chuyển năm dương lịch thành năm âm lịch. Trong ví dụ trên nếu người sử dụng nhập 2013 thì sẽ ra năm âm lịch là “Quý Tỵ”.

- Chú ý là ta tạo một anonymous listener, trước tiên bạn hãy xem Outline XML để cho dễ bề thiết kế:



- Để chuyển từ năm dương lịch sang năm âm lịch bạn cần biết một số thông tin sau:

Năm Âm = Can + Chi;

Can = Năm dương % 10 :

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mẹo	Thìn	Tỵ	Ngọ	Mùi

Chi = Năm dương % 12 :

String can,chi;

```
switch(namduong%10){
    //lấy can trong này dựa vào bảng số 1
}
switch(namduong%12){
    //lấy chi trong này dựa vào bảng số 2
}
```

Ghép can và chi ta được năm âm lịch

- Bây giờ ta tiến hành gán sự kiện cho nút “Chuyển đổi” (ở đây id Tôi để là **button1**), mở Activity class lên vào sửa lệnh như bên dưới:

```

public class MainActivity extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnDoi=(Button) findViewById(R.id.button1);
        btnDoi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                //Buoc 1. Lấy giá trị nhập từ EditText
                //Buoc 2. ép giá trị nhập vào về số nguyên
                //Buoc 3. có số nguyên là đại diện cho năm dương
                //      áp dụng hướng dẫn tính Can + Chi ở trên
                //      để tìm ra năm âm lịch
            }
        });
    }
    public boolean onCreateOptionsMenu(Menu menu) {

```

- Bạn tự đưa lệnh vào Bước 1, Bước 2, bước 3 ở trên. Cách lấy dữ liệu nhập vào từ EditText đã hướng dẫn ở phần **OnClick in XML**, làm theo cái này để lấy được giá trị là năm dương lịch ra, sau đó lấy năm này xử lý theo bảng **Can** và **Chi** như hướng dẫn thì Ta sẽ ra được năm Âm lịch tương ứng.

3. Activity is listener:

- Trong cách viết sự kiện này thì **Activity sẽ implements interface** có kiểu sự kiện (rất nhiều loại interface). Ở đây Tôi chỉ ví dụ trường hợp cho Button các trường hợp khác các bạn tự tìm hiểu và suy luận ra.

- Tôi sẽ có một bài ví dụ nhỏ sau: Hãy xây dựng ứng dụng tính **Chỉ số khối cơ thể** -chữ viết tắt **BMI (Body Mass Index)** - được dùng để đánh giá mức độ gầy hay béo của một người. Chỉ số này có thể giúp xác định một người bị bệnh **béo phì** hay bị bệnh **suy dinh dưỡng**.

+Cách tính như sau:

Gọi **W** là khối lượng của một người (tính bằng **kg**) và **H** là chiều cao của người đó (tính bằng **m**), chỉ số khối cơ thể được tính theo công thức:

$$BMI = \frac{W}{(H)^2}$$

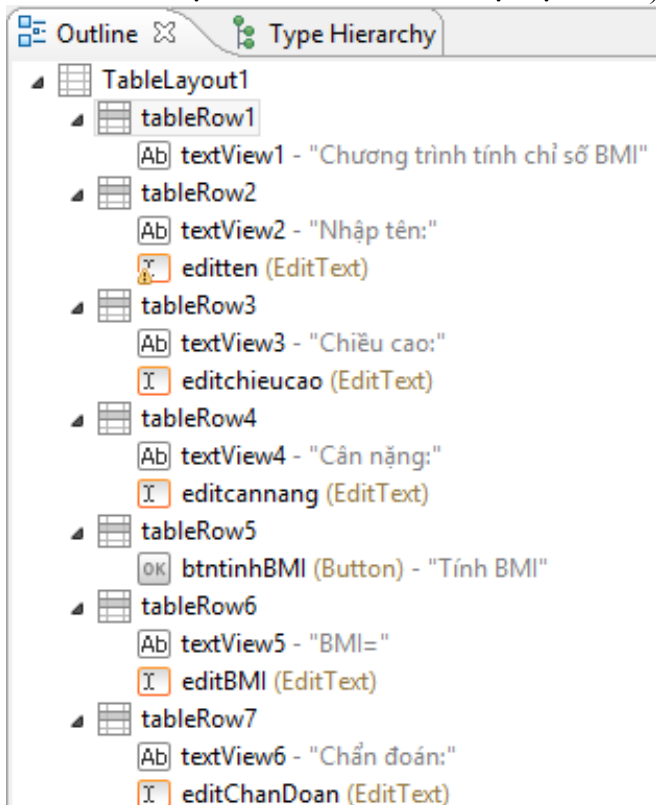
Phân loại để đánh giá như sau:

- BMI < 18: người gầy
- BMI = 18 – 24,9: người bình thường
- BMI = 25 – 29,9: người béo phì độ I
- BMI = 30 – 34,9: người béo phì độ II
- BMI > 35: người béo phì độ III

Tôi sẽ thiết kế giao diện như hình bên dưới và cung cấp Outline, các bạn hãy thiết kế lại để nâng cao kinh nghiệm:



- Bạn thấy đó: Thông số bên trên chính là của chính Tôi, Tên Tôi là Thanh, chiều cao 1.68 mét, cân nặng 58 kg. Khi Tôi chọn “Tính BMI” thì chương trình sẽ tính ra được BMI của Tôi là 20.5 và chẩn đoán là “Bình thường”, tức là Tôi có sức khỏe tốt, dáng người
- Quay trở lại ví dụ ,Bạn xem Outline của giao diện này dưới đây (Tôi cố tình không cung cấp source XML nhằm mục đích kích thích các bạn tự tìm tòi):



- Đây là nội dung Coding trong Activity:

```
import java.text.DecimalFormat;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
publicclass MainActivity extends Activity implements OnClickListener{
    Button btnChandoan;
    EditText editTen,editChieucac,
    editCannang,editBMI,editChandoan;
    @Override
protectedvoid onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btnChandoan=(Button) findViewById(R.id.btntinhBMI);
    btnChandoan.setOnClickListener(this);
    editTen=(EditText) findViewById(R.id.editten);
    editChieucac=(EditText) findViewById(R.id.editchieucac);
    editCannang=(EditText) findViewById(R.id.editcannang);
    editBMI=(EditText) findViewById(R.id.editBMI);
    editChandoan=(EditText) findViewById(R.id.editChanDoan);
}
@Override
publicvoid onClick(View arg0) {
    double H=Double.parseDouble(editChieucac.getText()+"");
    double W=Double.parseDouble(editCannang.getText()+"");
    double BMI=W/Math.pow(H, 2);
    String chandoan="";
    if(BMI<18)
    {
        chandoan="Bạn gầy";
    }
    Else if(BMI<=24.9)
    {
        chandoan="Bạn bình thường";
    }
    Else if(BMI<=29.9)
    {
        chandoan="Bạn béo phì độ 1";
    }
    Else if(BMI<=34.9)
    {
```



```

        chandoan="Bạn béo phì độ 2";
    }
    else
    {
        chandoan="Bạn béo phì độ 3";
    }
    DecimalFormat dcf=new DecimalFormat("#.0");
    editBMI.setText(dcf.format(BMI));
    editChandoan.setText(chandoan);
}
}

```

- a) Bạn quan sát Activity này sẽ implements interface **OnClickListener**
- b) Bản thân interface **OnClickListener** có một Abstract Method là **onClick**(View arg0) nên ta phải Override nó.
- c) Để Button có thể hiểu được sự kiện thì Ta phải gọi dòng lệnh: **btnChandoan.setOnClickListener(this)**; bản thân Activity là một sự kiện nên ta dùng **this** để truyền vào hàm.
- d) Bạn quan sát thêm Tôi có sử dụng : **DecimalFormat dcf=new DecimalFormat("#.0");** Mục đích là Tôi định dạng 1 số lẻ thập phân, bạn muốn 2 số lẻ thập phân thì ghi "#.00" hay muốn 3 thì "#.000" ...vân vân.

4) Listener in variable:

- Tương tự như Activity Is listener, nhưng khác ở chỗ thay vì implement interface cho Activity thì nó lại được lưu trữ vào một biến có kiểu Listener trong activity. Làm cách này thì ta có thể chia sẻ chung một biến sự kiện cho các control khác nhau.
- Ví dụ bạn xem hình bên dưới:

```

public class LoginExampleVariableImplementation extends Activity {
    ...

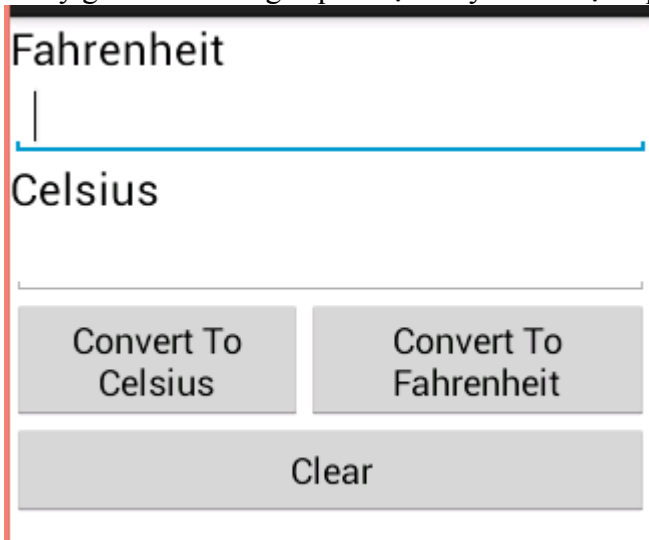
    OnClickListener myClickListener = new OnClickListener() {
        @Override
        public void onClick(View v) {
            if(v==btnLogin) {
                // Check Login
            } else if(v==btnCancel) {
                // Close the application
                finish();
            }
        }
    };

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        // Set Click Listener
        btnLogin.setOnClickListener(myClickListener);
        btnCancel.setOnClickListener(myClickListener);
    }
}

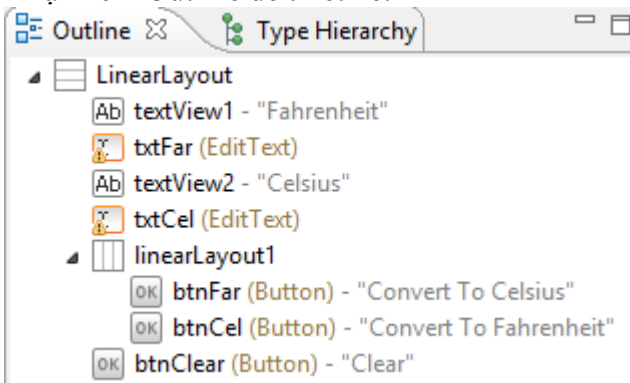
```

- Bạn thấy đó, ở trên khai báo 1 biến có kiểu listener là interface **OnClickListener**.
- Và biến này sẽ được chia sẻ cho 2 Button Login và Cancel.

- Bây giờ Tôi sẽ cung cấp ví dụ Chuyển đổi độ F qua C và ngược lại. Bạn xem giao diện bên dưới:



- Giao diện bên trên sẽ có 3 button. Ta sẽ tạo một biến sự kiện và chia sẻ cho 3 Button ở trên.
- Bạn xem Outline để thiết kế:



- Đây là công thức chuyển đổi:
- Công thức:

$$\begin{aligned} ^\circ\text{C} \times 9/5 + 32 &= ^\circ\text{F} \\ (^{\circ}\text{F} - 32) \times 5/9 &= ^\circ\text{C} \end{aligned}$$

Ví dụ:

Chuyển từ 37°C tới F : $37^{\circ}\text{C} \times 9/5 + 32 = 98.6^{\circ}\text{F}$

Chuyển từ 98.6°F tới C : $(98.6^{\circ}\text{F} - 32) \times 5/9 = 37^{\circ}\text{C}$

-Bạn xem coding Activity:

```
publicclass MainActivity extends Activity {  
    private Button btnFar,btnCel,btnClear;  
    private EditText txtFar,txtCel;  
    private OnClickListener myVarListener=new OnClickListener() {  
        @Override  
        publicvoid onClick(View arg0) {  
            //Lấy F và C từ control ở đây  
  
            if(arg0==btnFar)  
            {  
                //Bạn xử lý chuyển đổi F->C theo công thức  
            }  
        }  
    }  
}
```



```

        else if(arg0==btnCel)
        {
            // Ban xử lý chuyển đổi C->F theo công thức
        }
        else if(arg0==btnClear)
        {
            txtFar.setText("");
            txtCel.setText("");
            txtFar.requestFocus();
        }
    }
};
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnFar = (Button)findViewById(R.id.btnFar);
    btnCel = (Button)findViewById(R.id.btnCel);
    btnClear = (Button)findViewById(R.id.btnClear);
    txtFar = (EditText)findViewById(R.id.txtFar);
    txtCel = (EditText)findViewById(R.id.txtCel);
    btnFar.setOnClickListener(myVarListener);
    btnCel.setOnClickListener(myVarListener);
    btnClear.setOnClickListener(myVarListener);
}
}

```

Bạn tự viết lệnh cho 2 nút chuyển đổi : cách lấy dữ liệu đã hướng dẫn ở những kiểu lập trình sự kiện trước, bắt buộc bạn phải lấy được. Sau khi lấy được thì chỉ cần ráp vào công thức là xong.

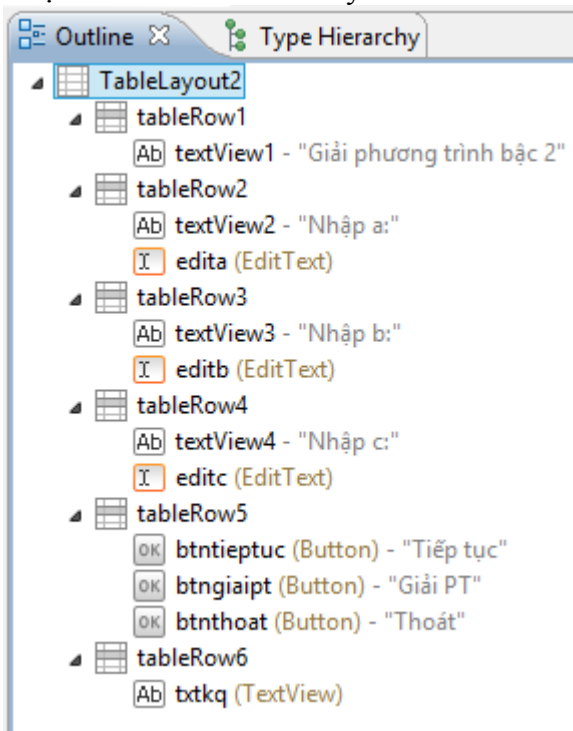
5) Explicit listener class:

- Trường hợp này ta tách riêng một class đóng vai trò là class sự kiện riêng.
- Khi nào lượng coding trong ứng dụng khổng lồ và phức tạp thì ta nên tách class sự kiện riêng để dễ quản lý.

Tôi ví dụ giải phương trình bậc 2, bạn xem giao diện bên dưới:

The screenshot shows an Android application interface for solving quadratic equations. The title bar is green and contains the text "Giải phương trình bậc 2". Below the title bar, there are three input fields labeled "Nhập a:", "Nhập b:", and "Nhập c:". The values entered are 2, 5, and 3 respectively. Below the input fields, there are three buttons: "Tiếp tục", "Giải PT", and "Thoát". At the bottom, there is a green status bar with the text "Pt có 2 No: x1=-1.50; x2=-1.50".

- Khi chọn “Tiếp tục”, chương trình sẽ xóa trắng toàn bộ dữ liệu trên màn hình đồng thời focus tới ô Nhập a.
- Khi chọn “Giải PT”, chương trình sẽ tiến hành lấy thông số a,b,c và tiến hành giải phương trình bậc 2 và cho ra kết quả như hình trên.
- Khi chọn “Thoát”, chương trình sẽ được đóng lại.
- Bạn xem Outline dưới đây:



- Tiến hành coding, bạn mở Activity class và coding như bên dưới:

```
import java.text.DecimalFormat;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
publicclass MainActivity extends Activity {
    Button btnTieptuc,btnGiai,btnThoat;
    EditText edita,editb,editc;
    TextView txtkq;
    @Override
    protectedvoid onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnTieptuc=(Button) findViewById(R.id.btntieptuc);
        btnGiai=(Button) findViewById(R.id.btngiaipt);
        btnThoat=(Button) findViewById(R.id.btnthoat);
        btnTieptuc.setOnClickListener(new MyEvent());
        btnGiai.setOnClickListener(new MyEvent());
        btnThoat.setOnClickListener(new MyEvent());
        edita=(EditText) findViewById(R.id.edita);
        editb=(EditText) findViewById(R.id.editb);
        editc=(EditText) findViewById(R.id.editc);
        txtkq=(TextView) findViewById(R.id.txtkq);
    }
    publicvoid giaiPtb2()
    {
        String sa=edita.getText()+"";
        String sb=editb.getText()+"";
        String sc=editc.getText()+"";
        int a=Integer.parseInt(sa);
        int b=Integer.parseInt(sb);
        int c=Integer.parseInt(sc);
        String kq="";
        DecimalFormat dcf=new DecimalFormat("#.00");
        if(a==0)
        {
            if(b==0)
            {
                if(c==0)
                    kq="PT vô số nghiệm";
                else
                    kq="PT vô nghiệm";
            }
            else
            {
                kq="Pt có 1 No, x="+dcf.format(-c/b);
            }
        }
        else
```

```

{
    double delta=b*b-4*a*c;
    if(delta<0)
    {
        kq="PT vô nghiệm";
    }
    elseif(delta==0)
    {
        kq="Pt có No kép x1=x2="+dcf.format(-b/(2*a));
    }
    else
    {
        kq="Pt có 2 No: x1="+dcf.format((-b-Math.sqrt(delta))/(2*a))+"; x2="+dcf.format((-b-
        Math.sqrt(delta))/(2*a));
    }
}
txtkq.setText(kq);
}
private class MyEvent implements OnClickListener
{
    @Override
    public void onClick(View arg0) {
        if(arg0==btnTieptuc)
        {
            edita.setText("");
            editb.setText("");
            editc.setText("");
            edita.requestFocus();
        }
        elseif(arg0.getId()==R.id.btngiaipt)
        {
            giaiPtb2();
        }
        elseif(arg0.getId()==R.id.btnthoat)
        {
            finish();
        }
    }
}
}

```

- Bạn quan sát coding ở bên trên. Tôi tạo một lớp sự kiện tên là **MyEvent**, control nào muốn được gán sự kiện chỉ cần gọi lệnh giống như Tôi gán cho Button Giải phương trình:

btnGiai.setOnClickListener(new MyEvent());

- Trong hàm **public void** onClick(View arg0) . Bạn để ý là lúc thì Tôi so sánh theo Object, lúc thì Tôi lại lấy Id ra để so sánh. Đây là cố ý của Tôi, Tôi muốn nói rằng các bạn có thể kiểm tra xem Button nào sẽ được chọn trên giao diện bằng cách so sánh đối tượng hoặc lấy Id ra để so sánh (tùy ý đồ lập trình của mỗi người)

- Ở đây Tôi có 1 lời khuyên cho các bạn là khi phải xử lý quá nhiều dòng lệnh (lệnh phức tạp) thì bạn nên viết thành từng hàm riêng, và trong hàm xử lý sự kiện bạn chỉ cần gọi tên hàm mà thôi. Cụ thể là đối với Button "**Giải PT**" thì Tôi lại viết và gọi riêng hàm **giaiPtb2()**, còn đối với Button Tiếp tục và Thoát thì Tôi lại không cần thiết viết hàm vì xử lý quá đơn giản.

- Như vậy các bạn đã được thực hành về các kiểu lập trình sự kiện trong Android, biết được cách lấy dữ liệu từ EditText, biết xử lý định dạng dữ liệu, củng cố thêm được Layout.
 - Trong các bài tập sắp tới các bạn sẽ được thực hành về Toast & Alert Dialog, và rất nhiều các control cơ bản cũng như nâng cao trong Android .
 - Bạn phải hiểu rõ các kỹ thuật lập trình này để tùy vào từng trường hợp cụ thể mà bạn nên quyết định kỹ thuật nào cho phù hợp.
- Chúc các bạn thành công.