

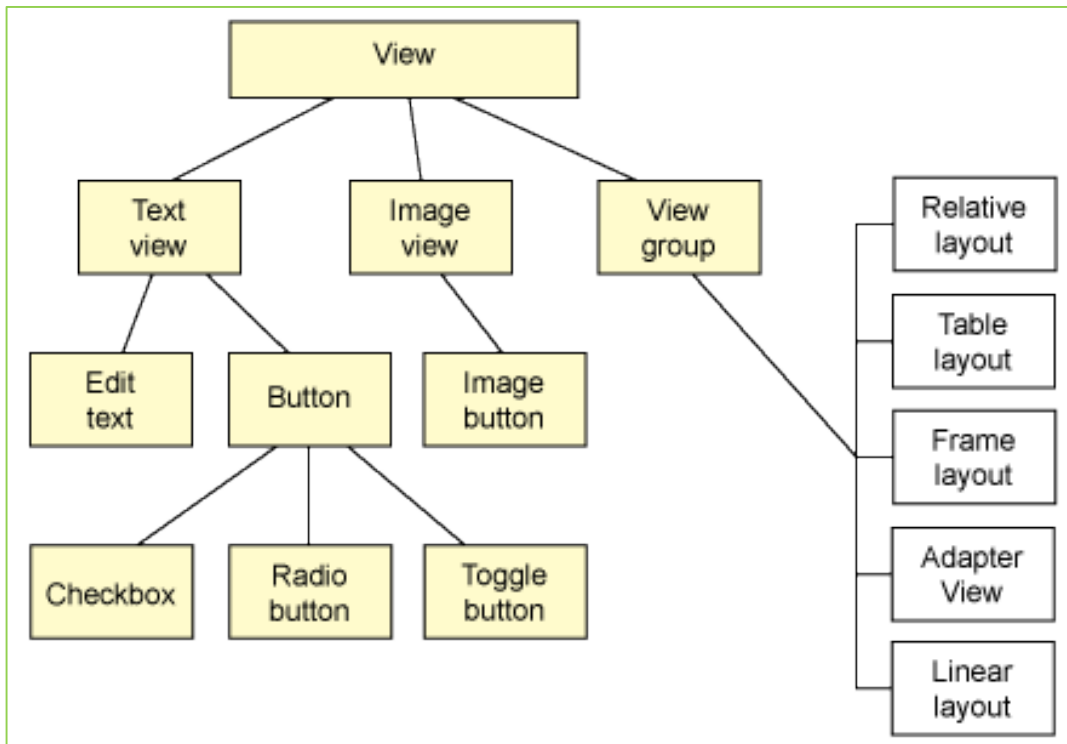
13.UI Layouts and controls

android වලදී **View** හා **ViewGroup** යනුවෙන් class දෙකක් තියෙනවා. තවද අපි දැනගත යුතුම කරුණක් වන්නේ screen එකේ තියෙන සෑම ui element එකක්ම object එකක් වේ. එනම් අපි java file එක ඇතුලෙදි code කරද්දී සියලු ui elements objects ලෙස සැලකිය යුතුයි. so මේවා object නම් ඒවාට අදාළ classes ද පවතී. එනම් Button class එකක්, EditText class එකක් ලෙස සියලු ui elements වලට classes තියෙනවා. අපි screen එකේ එම element එකක් place කරද්දී වෙන්වෙන් එම class එකේ object එකක් සෑදී screen එකේ place වීමයි. මේවා xml file එකේදී tags වශයෙන් තිබ්බත් java file එක තුලදී අපි ඒවාට සැලකිය යුත්තේ objects ලෙසයි.

ඉහත කතා කල සියලුම ui elements වල parent class එකක් පවතී. එනම් android වල සියලු ui elements මගින් inherit කරන් ඉන්න parent class එකක් පවතී. එම parent class එකේ නම වන්නේ **view** යන්නයි. සියලු ui elements වල classes මෙහි child classes වේ.

View class එකට inherit කරන් ඉන්න තවත් වැදගත් class එකක් වේ. එය නම් **ViewGroup** යන්නයි. ViewGroup එකක් යනු අනෙකුත් views (ui elements සියල්ලම view class එකේ sub classes නිසා, සමහර ස්ථානවලදී “ui elements” යන්න සඳහා “views” යන නමද භාවිතා වේ.) අන්තර් ගත කරන් හිටිය හැකි class එකක් වේ. එනම් මෙම ViewGroup class එකේ object එකකට පුළුවන් ui element objects රඳවාගෙන තබා ගන්න. එසේම අවශ්‍ය නම් තවත් ViewGroup object එකක් වුවත් අන්තර් ගත කරන් සිටිය හැකියි. එනම් මේ කියන්නේ screen එකේ layout එක හඳුන්වන elements ගැන වේ. එනම් RelativeLayout, LinearLayout, TableLayout, FrameLayout වැනි class ගැනයි. මේ කියන සියලු classes වල object එකක් මගින් අනෙකුත් views (ui elements) රඳවාගෙන තියාගන්න පුලවන්.

ඉහත කියන ලද classes වල hierarchy එක පහත රූප සටහනෙන් පැහැදිලිව තේරුම් ගත හැක.



අපි කලින් 5 ස් වෙනි පාඩමේදී කතා කලා අපිට අවශ්‍ය විදියට screen එක සකසා ගැනීම සඳහා යම් කිසි ViewGroup එකේ sub class වන RelativeLayout වැනි වැනි element එකක් තුල අවශ්‍ය අනෙකුත් views place

කරලා තියෙන structure එක හදලා xml file එකක් සැදිය යුතුයි කියල. එසේම ඉන් පසු 6වන පාඩමේදී සඳහන් කළා එසේ සාදා ගත් xml file එක අනුව, screen එක display කරවීම සඳහා එම xml file එක, java code එක තුළදී අරගෙන එය screen එකට set කරන්නේ කෙසේද කියල. අවශ්‍ය නම් එසේ වෙනම xml file එකක ui components හා ඒවා තිබෙන structure එක code නොකර app එක run වන අවස්ථාවේදී ඒවා generate වන ලෙස java file එකේ code කරද අපිට display කල හැක.

View Identification

අපි භාවිතා කරන සියලුම view object වලට ඒවා විස්තර කර අපි එක එක attributes යොදනවා. උදාහරණ ලෙස height එක, width එක, color එක වැනි දේවල්. මීට අමතරව අපි සෑම view object එකකටම අනිවාර්යෙන් assign කල යුතු දෙයක් තමා, එම object එක uniquely හඳුනා ගැනීම සඳහා id එකක්. XML tag එකක් තුළ id එකක් assign කිරීම සඳහා භාවිතා කරන syntax එක පහත පරිදි වේ.

```
<ui_element
.....
    android: id = “ @+id / unique_name_for_id ” />
```

මෙහි ඇති @ ලකුණේ හා + ලකුණේ අර්ථයන් පහත පරිදි වේ,

@ ලකුණෙන් කියන්නේ id එකට දීලා තියෙන string එක string එකක් ලෙස නෙමේ, / ලකුණෙන් එහා තියෙන ටික id resource එකක් ලෙස සලකන්න කියලයි. id resource එකක් යනු මෙය R.java file එකට යනවා කියලයි. මේ හේතුව නිසයි අපිට මෙය භාවිතා කරලා element එක ගත හැකි වන්නේත් activity එකට අදාළ java code එකට.

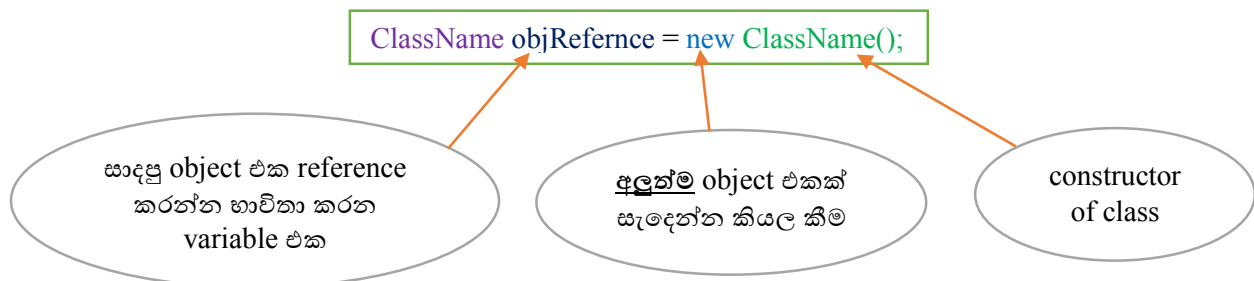
+ ලකුණෙන් කියන්නේ / ලකුණෙන් ඉදිරියෙන් තියෙන නම භාවිතා කර අලුත්ම resource එකක් එකක් හදලා එය resources වලට **add** වෙන්න කියලයි. (id/ ලෙස තිබෙන නිසා id resource ගොඩටයි add වෙන්නේ)

උදා:

```
<Button android:id= “@+id/button1”
    android:layout_hight= “wrap_content”
    android:layout_width= “wrap_content”
    android:text= “Click me” />
```

XML file එකේ ui element එකක් java code එකට ගැනීම

ඇත් අපි බලමු ඉහත සාදා ගත් button element එක activity එකට අදාළ java code එකට ගන්නේ කෙසේද කියල. මුලින්ම අපි කිව්ව මේ සෑම view එකක්ම android වලදී සලකන්නේ object එකක් ලෙසයි. අපි element එකක් screen එකට ඇදලා දාල හරි, xml වලින් code කරලා හරි නිපදවුවා යනු, object එකක් සෑදුවා යන්නයි. එලෙසම අපි දන්නවා java වලදී සාමාන්‍යයෙන් object එකක් සාදන්නේ පහත පරිදියි.



නමුත් දැන් මෙහිදී new කියන විද්‍යාය සහිතව අලුත් object එකක් සෑදීම කල යුතු නැ.මොකද අපි ui element එකක් screen එකට ඇදලා දාපුහම හෝ xml code එකෙහි element tag එකක් type කලා යනු object එකක් සෑදීමයි කලේ.දැන්කිරීමට ඇත්තේ එම සාදපු object එක refer කිරීම සඳහා variable එකක් සෑදීමයි.(refer කරන්න variable එකක් පමණයි හදන්නේ,එනම් එම object එක තියෙන place එකේ address එක තියාගෙන ඉන්න variable එකකුයි හදන්නේ,මොකද object එකක් රඳවාගෙන ඉන්න පුළුවන් variables නැ.object යනු size එකෙන් විශාල memmory කොටස් වේ)

object එක refer කරන්න ඉන්න හදන variable එකේ type එක විය යුත්තේද එම object එකේ type එකමයි.(හෝ ඊට ඉහල parent class එකක type එකක් වුනත් අවුලක් නැ.නමුත් android වලදී අපි එතරම් දුරට යන්නේ නැ.එම object එකේ වර්ගයේම variable එකකුයි හදන්නේ).

so ඉහත උදාහරණයේ පරිදි නම් object එකේ type එක වන්නේ Button යන්නයි.so reference variable එකේ type එකද Button විය යුතුයි.එනම් පහත පරිදියි.එහි reference variable එකේ නම වන්නේ btn1 යන්නයි.

Button btn1;

දැන් අපිට කිරීමට ඇත්තේ කලින් සාදන ලද object එක අරගෙන මෙයට සමාන කිරීමයි.එවිට මෙම btn1 කියන variable එකෙන් එම object එක දිහා බලාගෙනම සිටීමට පටන් ගනු ඇත.එනම් btn1 මගින් එම object එක refer කරන්න සිටිනු ඇත.සාදපු object එක ගැනීම සඳහා දැන් එහි id එක භාවිතා කල හැක.එය resource වල id කොටසේ ඇත.

එම නිසා **R.id.button1** යන්න භාවිතයෙන් හා **findViewById()** යන method එක භාවිතයෙන් අපිට එම object එක ලබා ගත හැක.තවද අපි එසේ කලත් මදි.අපි කිවිය යුතුයි මේ ගන්න object කුමන type එකේ object එකක්ද කියල එය assign කිරීමට පුළුම.මොකද කලින් කිව්වා වගේ reference variable එකකට refer කරන්න ඉන්න පුළුවන් එම variable type එකේ object එකක් හෝ එහි type එකේ child class වල objects වුනත්.එම නිසා එය ලබා ගත object එක reference variable එකට assign කිරීමට පුළුම අපි එය කුමන වර්ගයේ object එකක්ද කියලා කියන්න ඕන.මේ ඉහත කියන ලද දේවල් පහත පරිදි කල හැක

(Button) findViewById (R.id.button1)

object එක reference variable එකට assign කිරීමට පුළුම, එය කුමන type එකේ object එකක්ද කියා මෙසේ වරහන් තුල එම type එක දමා කිව යුතුයි.

පහතින් දැක්වෙන්නේ ඉහත සාදපු button1 නම් id එක ඇති object එක java code එකට ගැනීම සඳහා අවශ්‍ය වන complete code statement එකයි.

Button btn1=(Button) findViewById (R.id.button1);

Android UI controls

අපි දැන් දන්නවා View එකක් යනු screen එකක් මත යමක් ඇඳිය හැකි object එකක් කියලා. එනම් මෙම view යටතේ තියෙන sub class වලින් තමා එක එක දේවල් screen එක මත draw කරන්නේ. So අපිට හිතන්න පුළුවන් View යනු screen එක මත කුමක් හෝ අඳින object එකක් කියලා. සහ ඕනෑම ui component එකකට අපි පොදු වශයෙන් කියන්නේ view එකක් කියලයි (මොකද ඒ සියල්ල View class එකේ child class වන නිසා). සහ ViewGroup එකක් යනු එනකුත් views සහ ViewGroup වුවත් රඳවාගෙන තියා ගත හැකි objects සහ මේවාගෙන් තමා user interface එකේ layout එක කොහොමද කියලා define කරන්නේ කියලත් අපි දැන් දන්නවා.

මෙම layout එක සහ එහි ui componets (views) කොහොමද කියලා අපි define කරන්නේ xml file එකකයි. පහත දැක්වෙන්නේ එවැනි layout xml file එකක සරල උදාහරණයකි.

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

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a TextView" />

    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button" />

</LinearLayout>
```

android වලදී user interface එක ගොඩ නැගීම සඳහා, user inputs ලබා ගැනීමට, userට texts පෙන්වීමට, වැනි දෑ කිරීම සඳහා විශාල ui elements ප්‍රමාණයක් සපයනවා. පහතින් ඇත්තේ ඉන් සමහරක් පිළිබඳ කුඩා හැඳින්වීමකි.

- **TextView** - මෙය userට text පෙන්වීම සඳහා භාවිතා කරනවා
- **EditText** – userට අකුරු type කිරීමට ලබා දේ. මෙය ඉහත කියන ලද TextView class එකේ sub class එකකි.
- **AutoCompleteTextView** - මෙය EditText වගේම අකුරු type කරන්න දෙනවා. ඊට අමතරව type කරන අතර තුර suggestions පෙන්වනවා.
- **Button** – user හට press/click කල හැකි button එකක් ලබා දේ.
- **ImageButton** -
- **CheckBox** -
- **ToggleButton** – on/off button එකක්. on ද off ද කියලා light එකක් මගින් පෙන්වනවා.
- **RadioButton** -
- **RadioGroup** - මෙවිට එක radio button එකක් පමණයි select කරලා තියන්න පුළුවන්
- **ProgressBar** - යම් කාර්යයක කොතෙක් දුරක් කරලාද වැනි දේ පෙන්වීම සඳහා භාවිතා කල හැක
- **Spinner** – DropDown list එකක් වේ. මෙම list එකෙන් එකක් userට තෝරා ගැනීමට පුළුවන්.
- **TimePicker** - මෙම view එක මගින් යම් දිනයක කාලය තෝරා ගැනීමට ඉඩ සලස්වයි. 24 පැය ක්‍රමයට හෝ AM/PM ක්‍රමයට.
- **DatePicker** - දවස තෝරාගැනීමට ඉඩ සලස්වයි.