MOBILE PROGRAMMING

Chapter 4

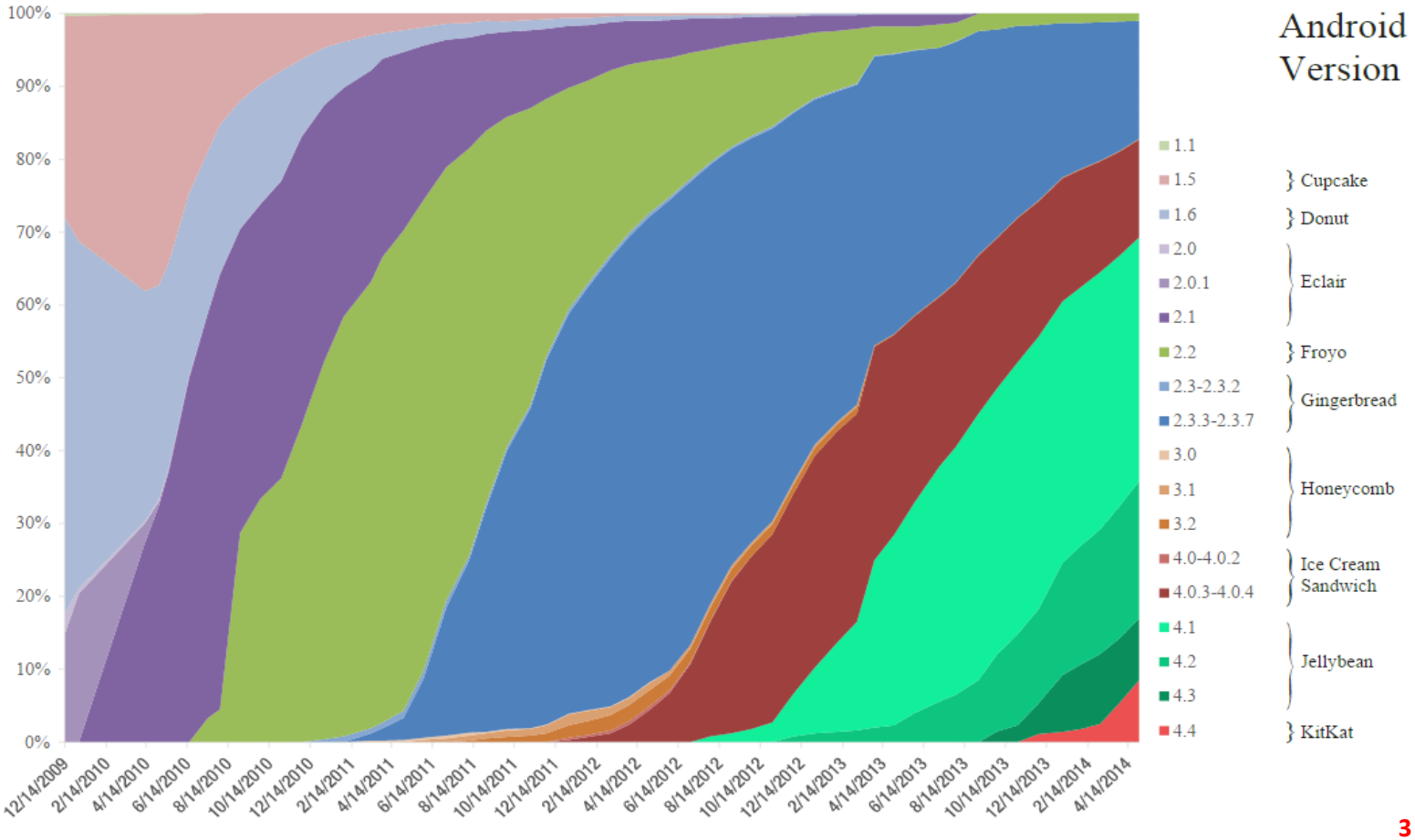
ANDROID PROGRAMMING (BASIC)

Contents

- Android Operating System
- Android Programming
- UI Programming
- Events handling
- Graphics

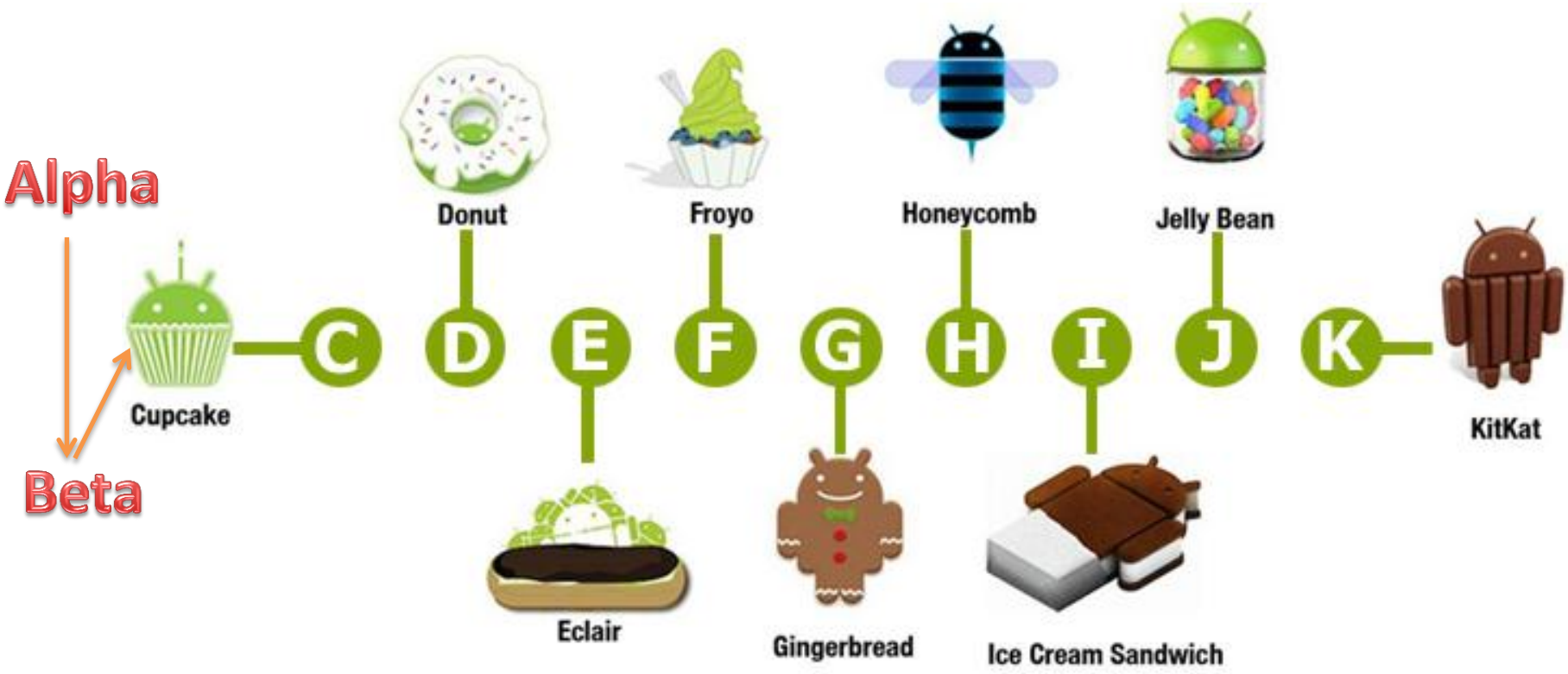
4.1- Android OS

- **History**
  - Versions as a revolution



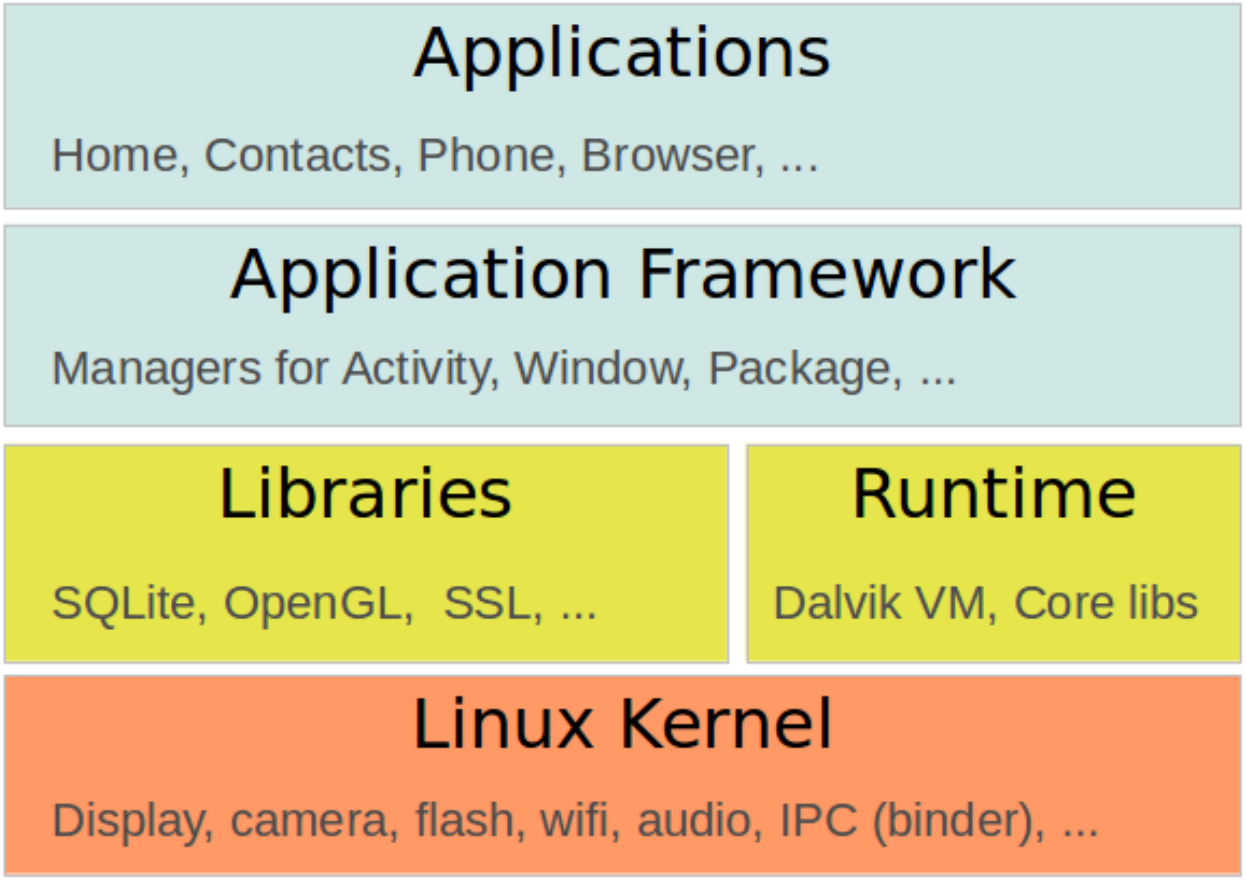
4.1- Android OS

- **History**
  - Code Name of versions are foodie



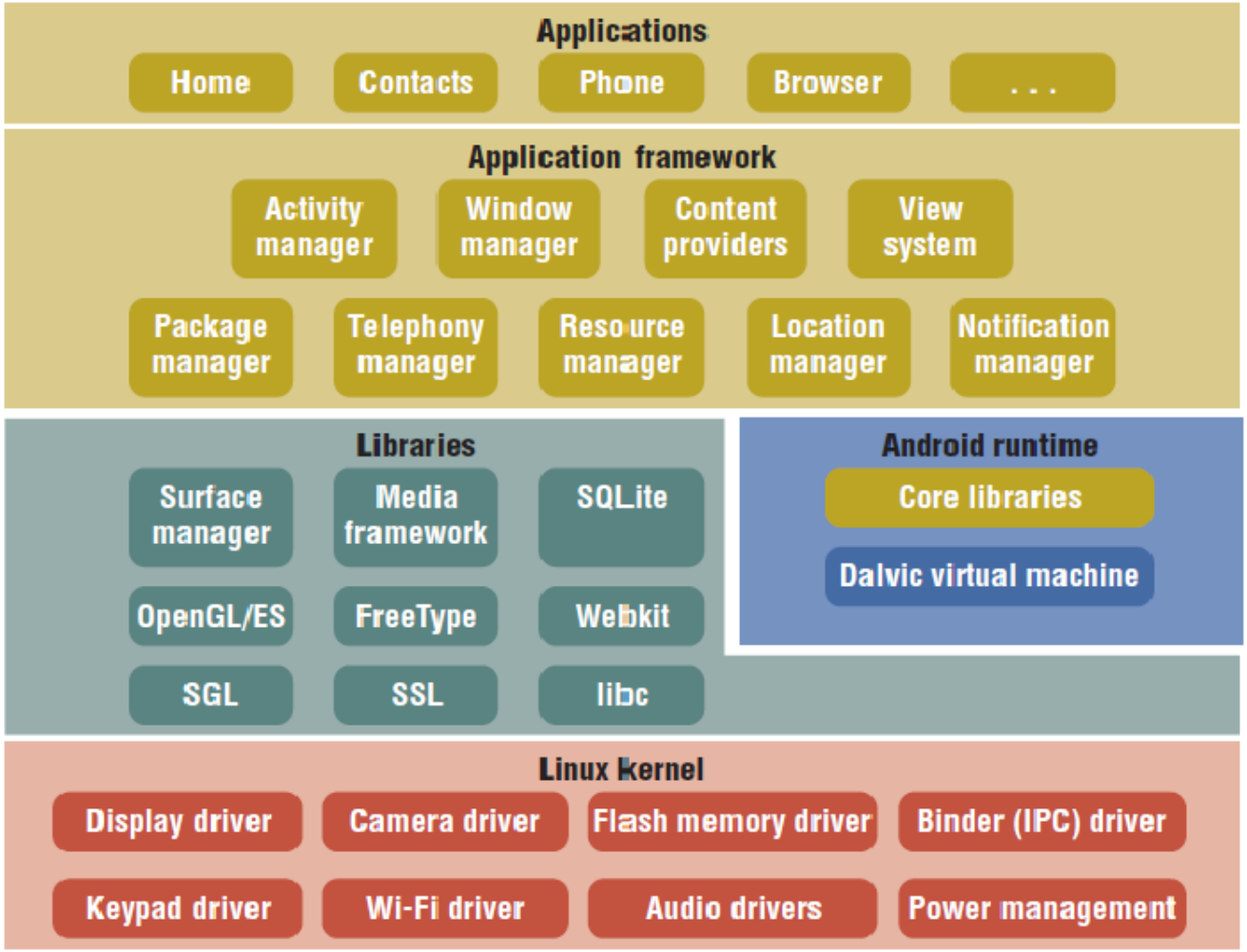
4.1- Android OS

- **Architecture**
  - 4 layers



4.1- Android OS

- **Architecture**
  - Details

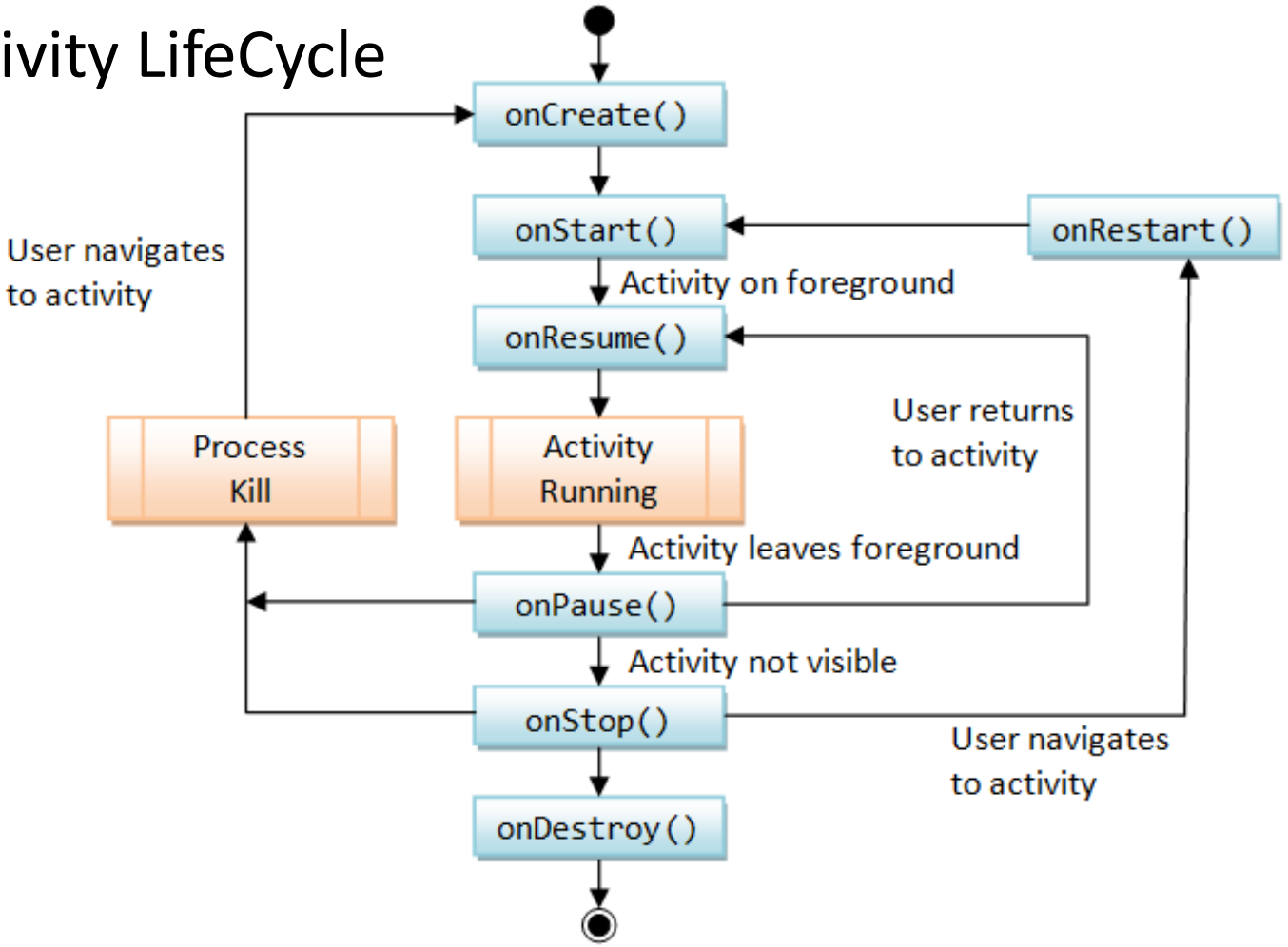


4.2- Android programming

- **Parts of android applications**
  - Activities (at least):
    use View for GUI
  - Intents
  - Services
  - Broadcast
    and Receivers
  - Content Providers

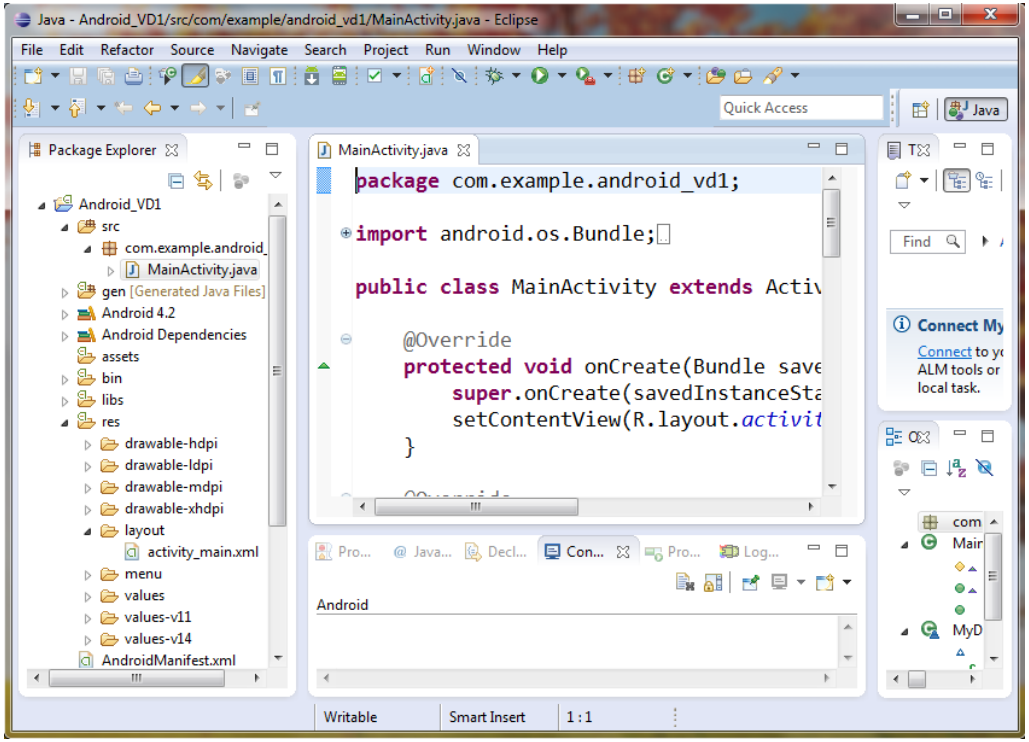
4.2- Android programming

- **Parts of android applications**
  - Activity LifeCycle



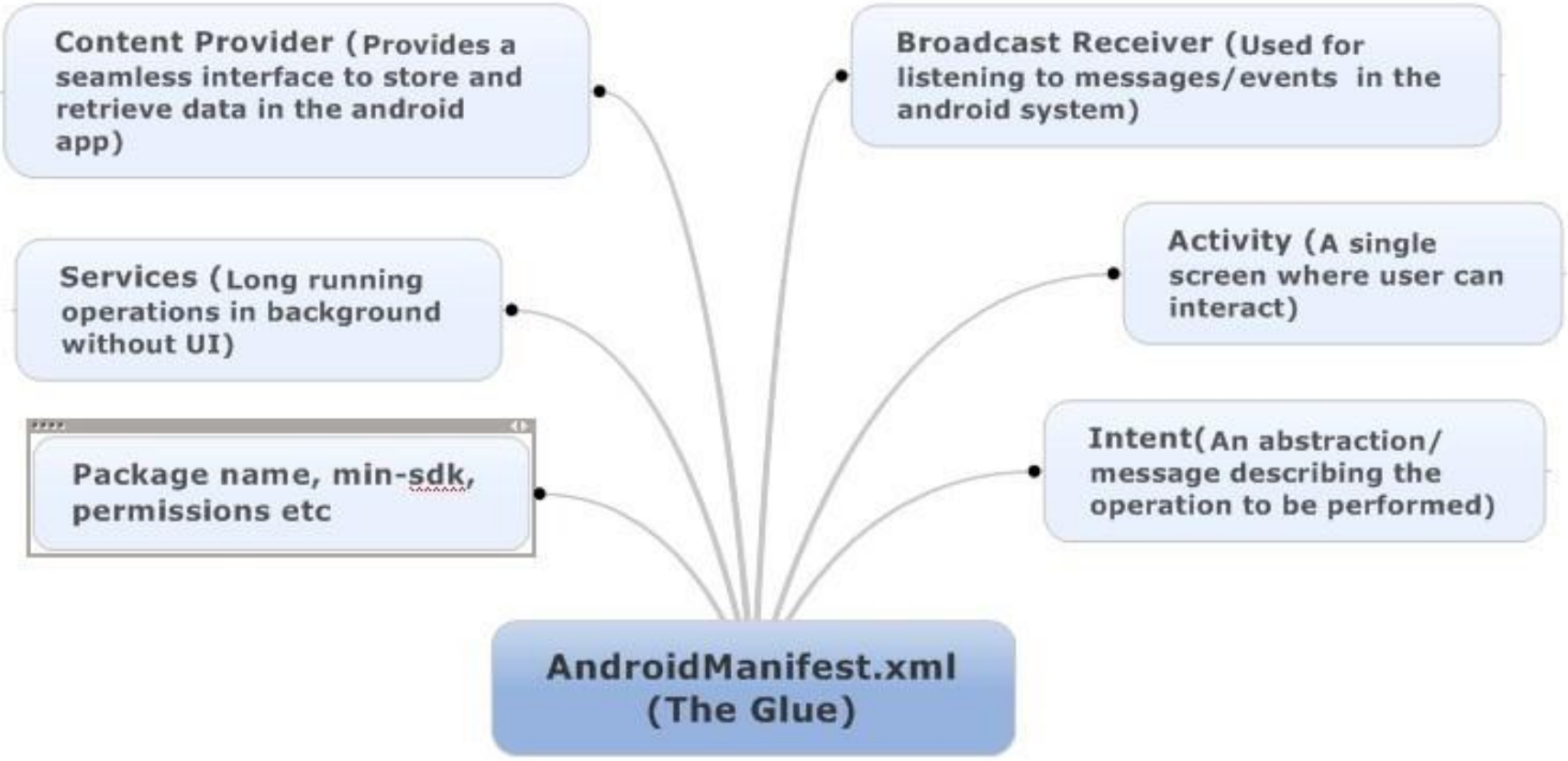
4.2- Android programming

- **IDE for android programming**
  - JDK + ADT {SDK, Eclipse}
  - An application has:
    - src
    - res/layout
    - res/menu
    - res/value
    - res/drawable
    - gen, bin, lib, assets



4.2- Android programming

- **IDE for android programming**
  - AndroidManifest.xml



4.2- Android programming

– Example of an "AndroidManifest.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.example.apk_vd1"
    android:versionCode="1"
    android:versionName="1.0" xmlns:android="http://schemas.android.com/apk/res/an

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@android:drawable/btn_star"
        android:label="VD1">
        <activity
            android:name="com.example.apk_vd1.MainActivity"
            android:label="VD1">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

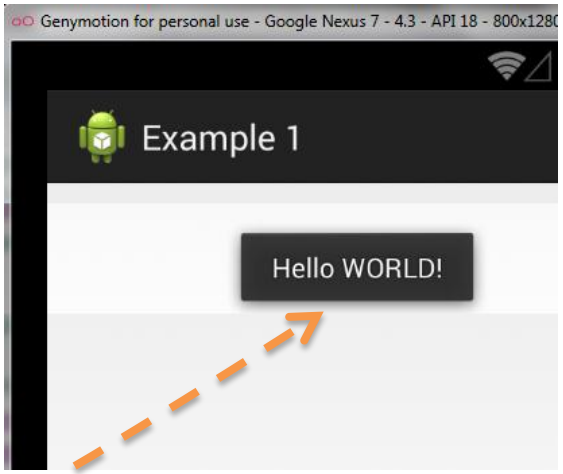
Class name of activity

4.2- Android programming

- **Example of an application**

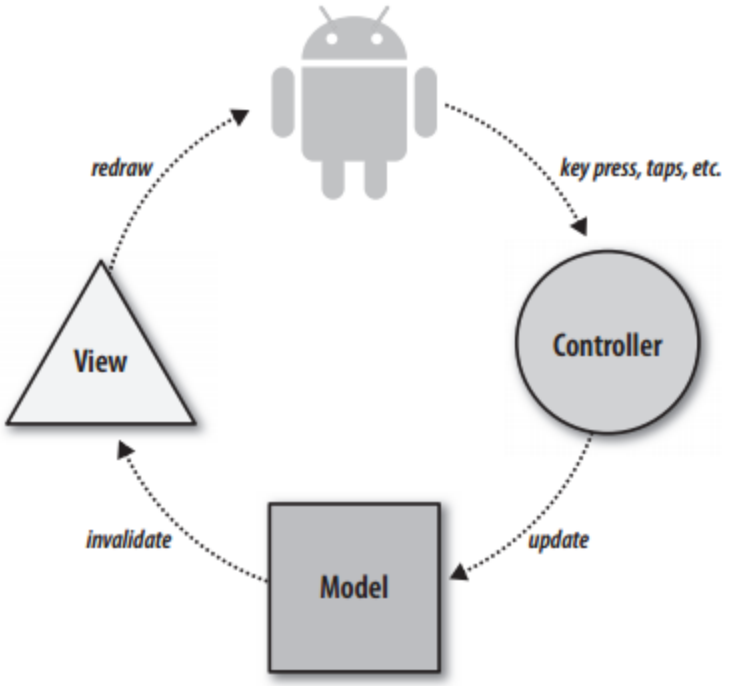
Toast_Example.java

```
1  package com.example.vd1;
2
3  import android.os.Bundle;
4  import android.widget.Toast;
5  import android.app.Activity;
6
7  public class Toast_Example extends Activity {
8      protected void onCreate(Bundle ts) {
9          super.onCreate(ts);
10         Toast.makeText(this, "Hello WORLD!",
11                 Toast.LENGTH_LONG).show();
12     }
13 }
```

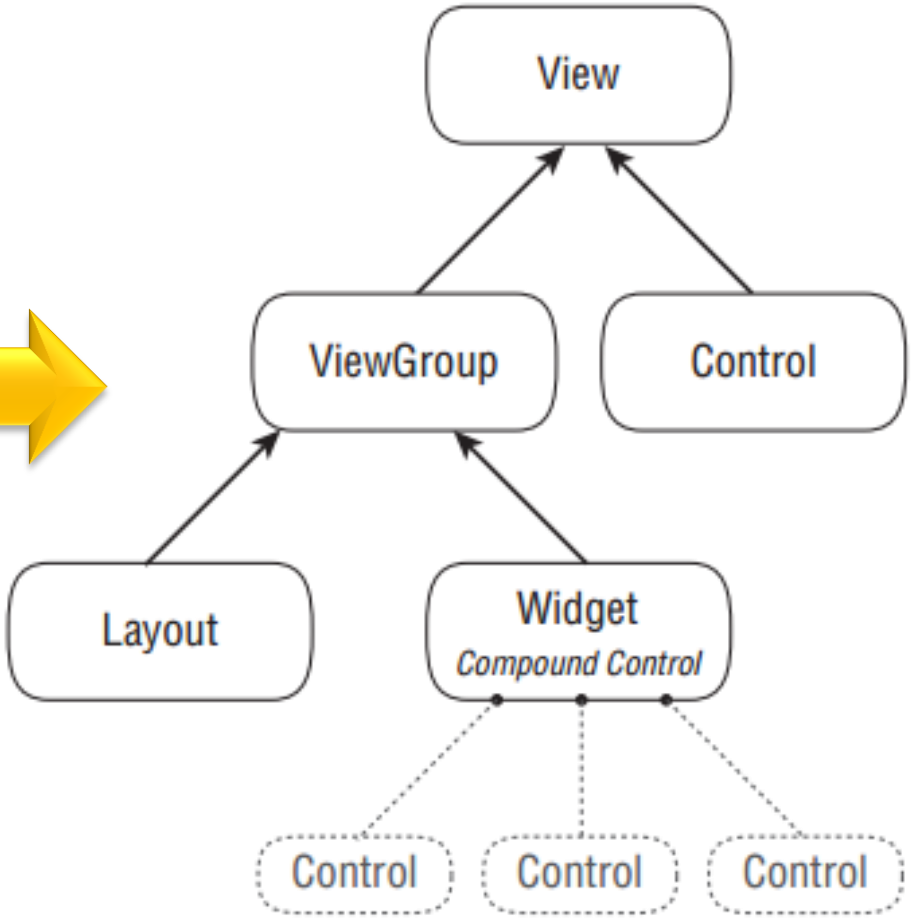


4.3- UI programming

- **MVC for UI**

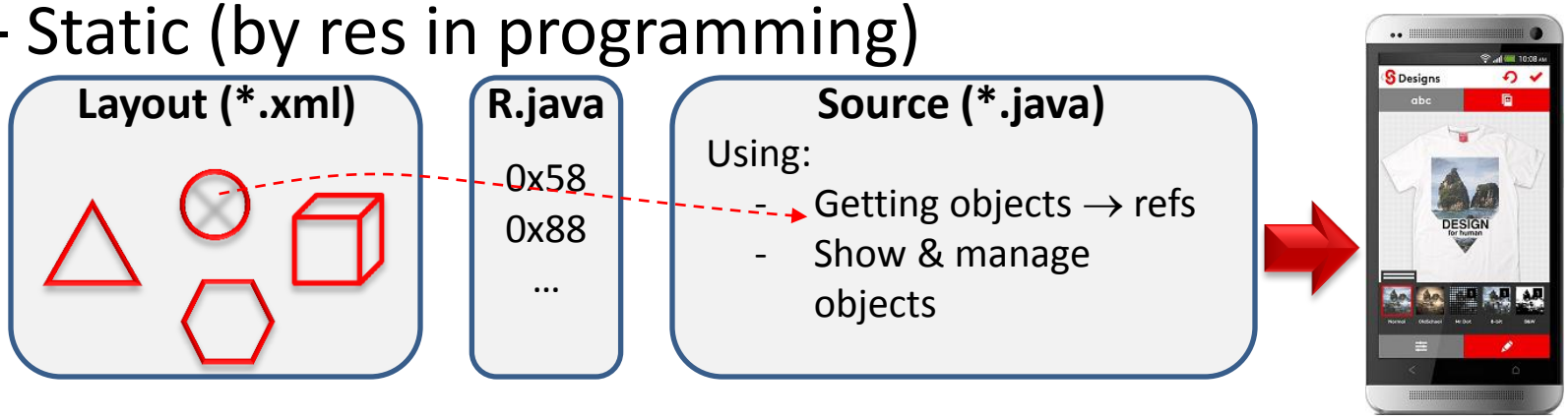


Hierarchy of view classes

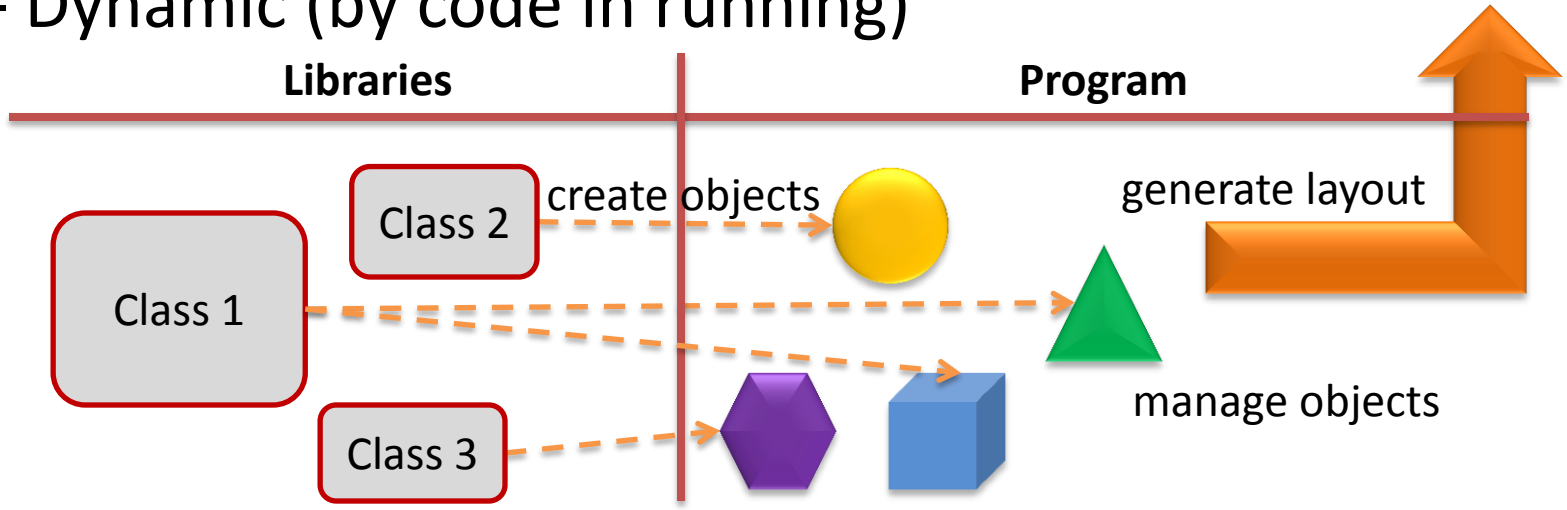


4.3- UI programming

- **Create UI**
  - Static (by res in programming)



  - Dynamic (by code in running)



4.3- UI programming

- **Create UI by layout (*.xml)**
  - Step1) Design a layout (*.xml) using tools in IDE

    *drag & drop or edit in *.xml file*
  - Step2) Show it to screen (in an Activity)

    setContentView(R.layout.♥);

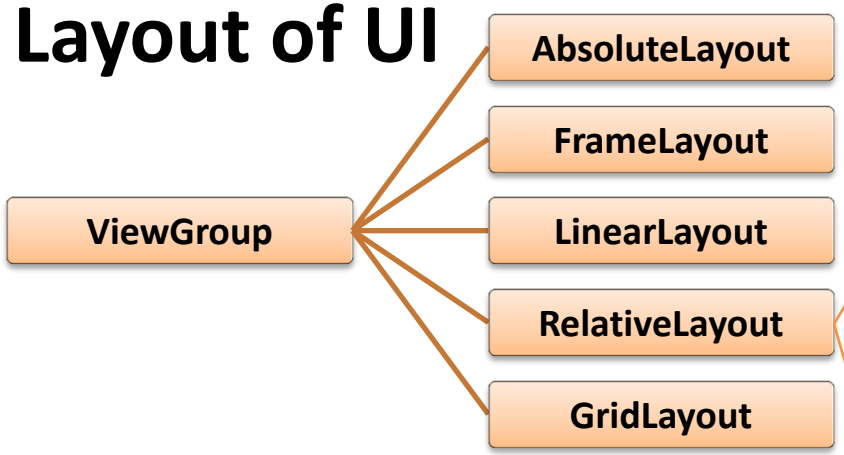
    It generates all objects in the layout
  - Step3) Get objects to references

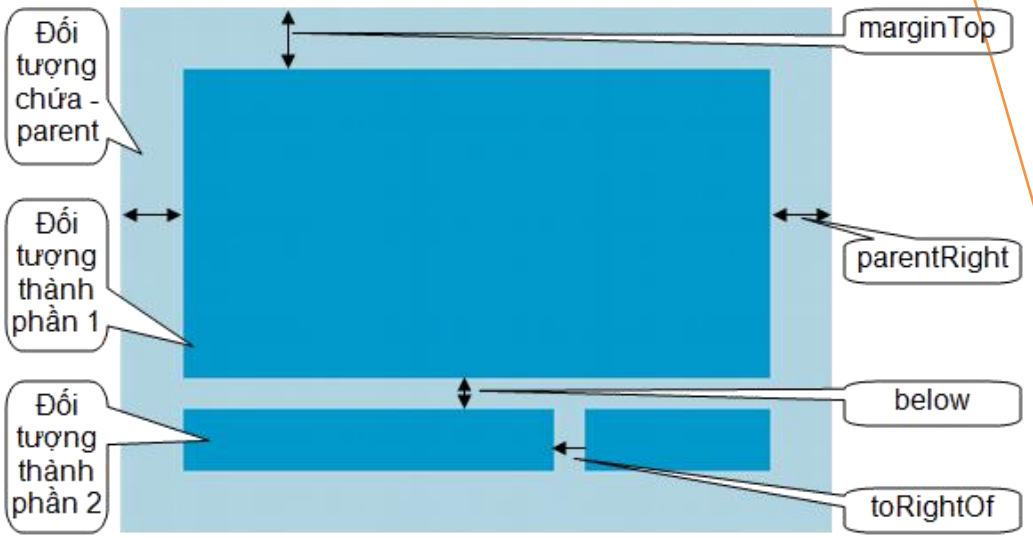
    ♦ = (*TheClass*)findViewById(R.id.♣);
  - Step4) Manage/using objects

    ♦.some_methods**(** params **);**

4.3- UI programming

- **Layout of UI**





LayoutParams

Layout above	
Layout align baseline	
Layout align bottom	
Layout align left	
Layout align parent botton	
Layout align parent left	
Layout align parent right	
Layout align parent top	
Layout align right	
Layout align top	
Layout align with parent if	
Layout below	
Layout center horizontal	
Layout center in parent	
Layout center vertical	
Layout height	wrap_content
Layout margin	
Layout margin bottom	
Layout margin left	
Layout margin right	
Layout margin top	
Layout to left of	
Layout to right of	
Layout width	wrap_content

4.3- UI programming

- **Example of layout XML**

*gui1.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas
3      android:layout_width="match_parent"
4      android:layout_height="match_parent" >
5
6      <Button
7          android:id="@+id/button1"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="Tính kết quả" />
11
12 </RelativeLayout>
```

setContentView(*R.layout.gui1*);

*R.java

```
1  package com.example.vd1;
2
3  public final class R {
4      public static final class id {
5          public static final int button1=0x7f080002;
6      }
7      public static final class layout {
8          public static final int gui1=0x7f030001;
9      }
10 }
```

4.3- UI programming

- **Create UI by coding**
  - Step1) Create an layout object

```
∇ = new  TheLayoutClass( Context  ♥ );
```

  - Step2) Add the layout to screen with its params

```
♦ = new   LayoutParams( … );
♦.some_fields = some_values;
theContext.setContentView( ∇, ♦ );
```

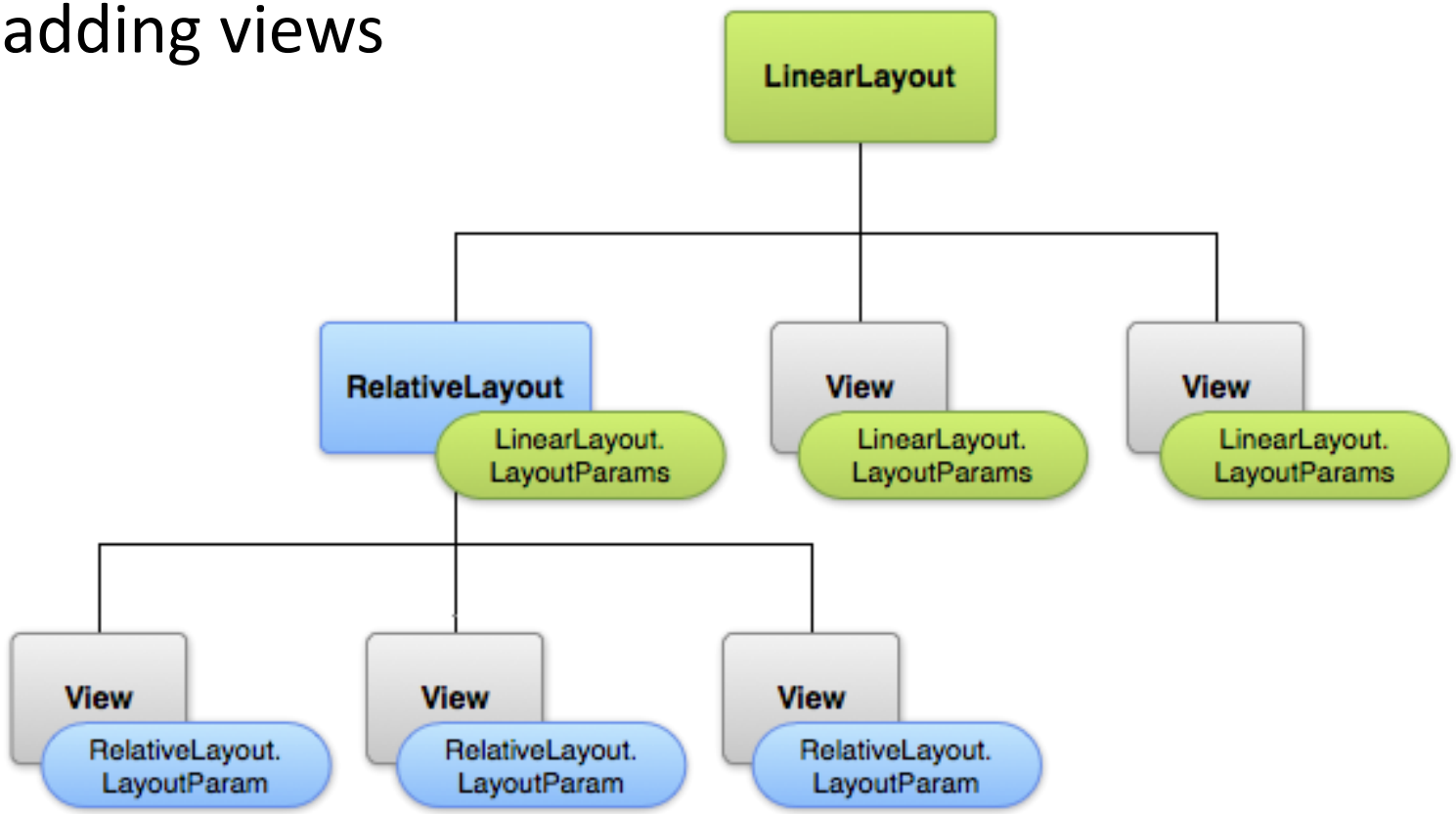
  - Step3) Add views to the layout with its params

```
♣ = new ?.LayoutParams( … );
♣.some_fields = some_values;
∇.addView( a_view, ♣ );
```

Inner class of the layout ∇

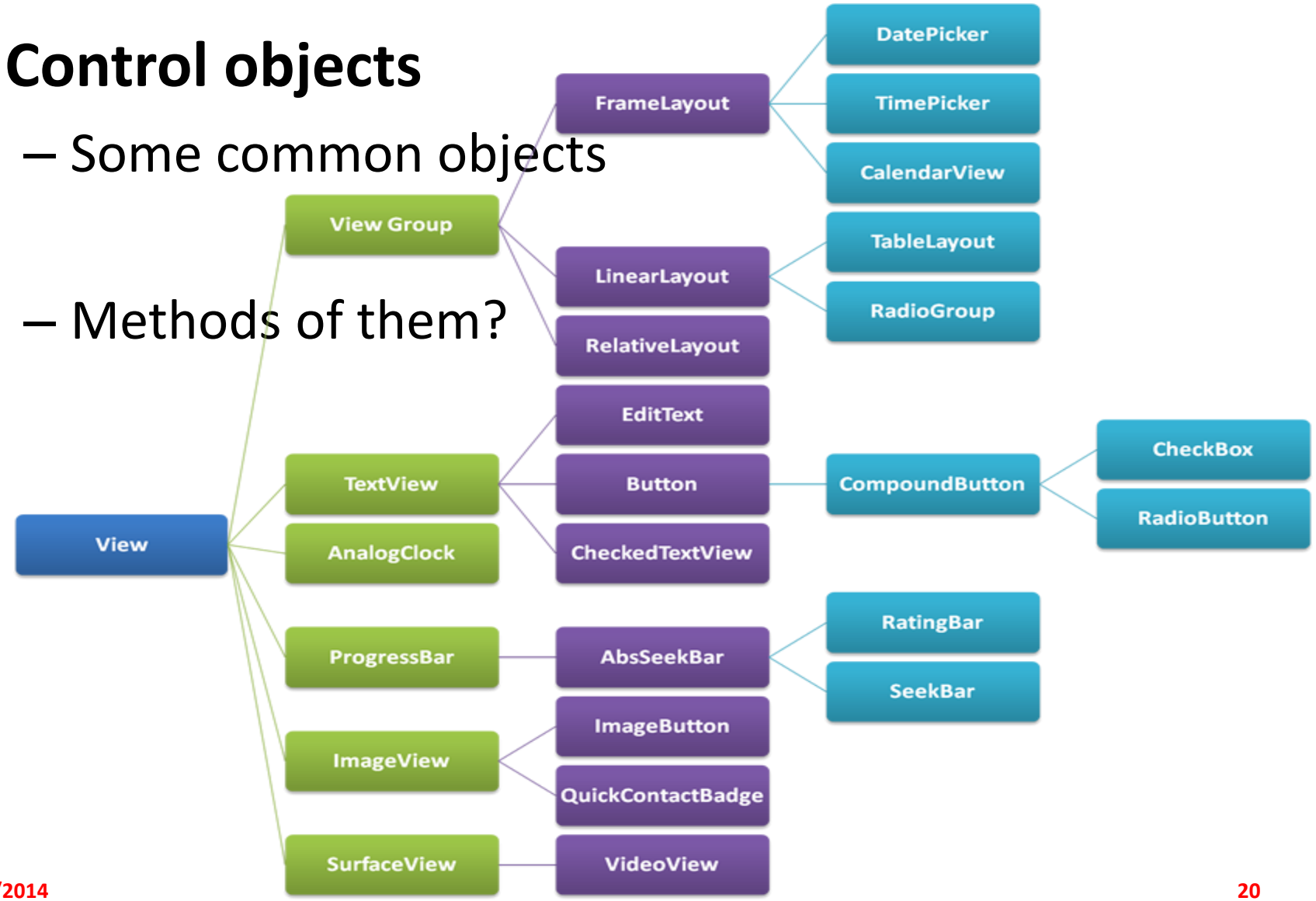
4.3- UI programming

- **Layout parameters**
  - A layout has an inner class with its attributes for adding views



4.3- UI programming

- **Control objects**
  - Some common objects
  - Methods of them?



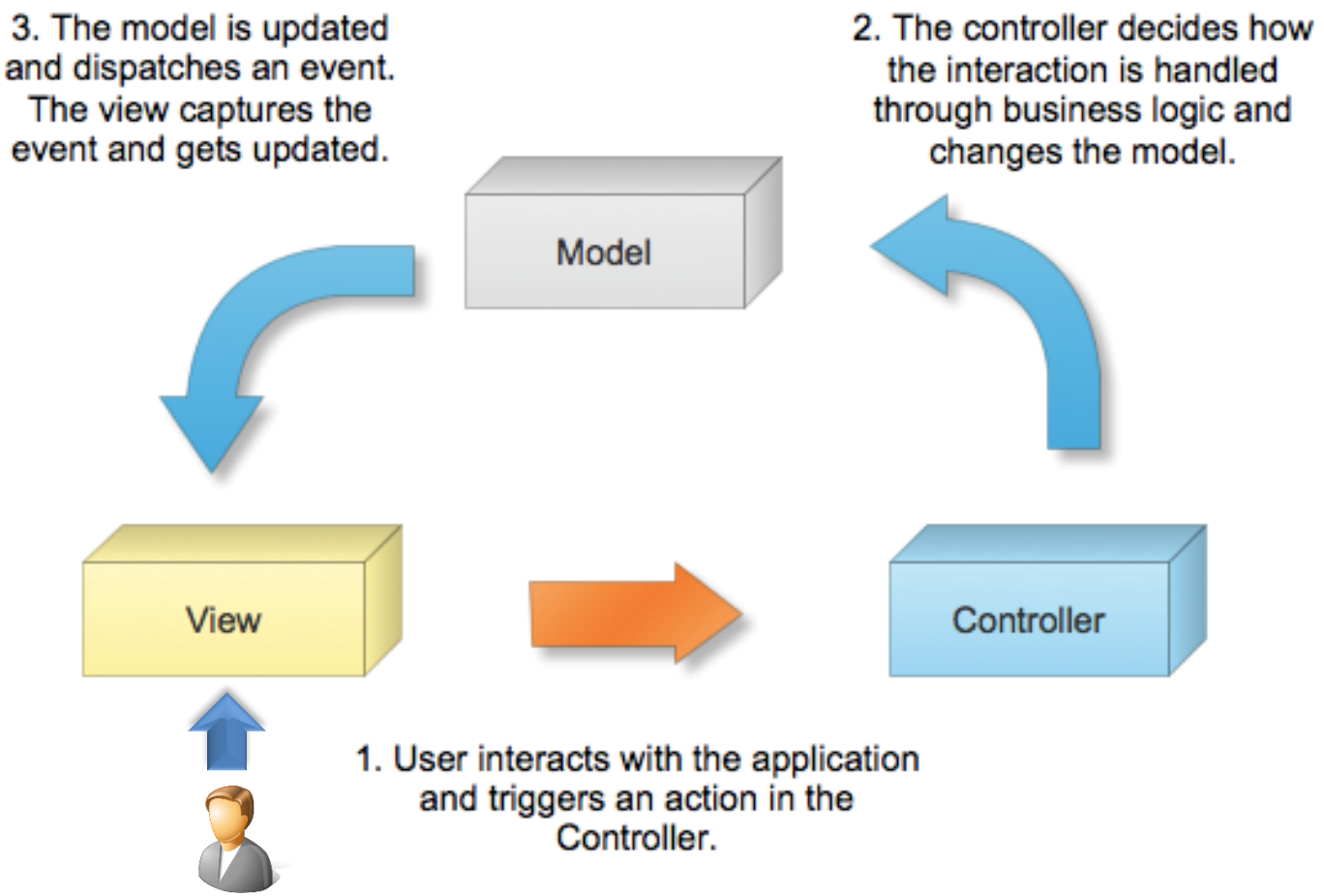
4.3- UI programming

- **Example of UI programming**

```
10 public class VD1 extends Activity {
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         GridLayout g=new GridLayout(this);
14         setContentView(g);
15         Button  b;
16         for(int i=0;i<3;i++)
17             for(int j=0;j<3;j++){
18                 LayoutParams p=new LayoutParams();
19                 b=new Button(this);
20                 b.setText("["+i+","+j+"]");
21                 p.height = LayoutParams.WRAP_CONTENT;
22                 p.width = LayoutParams.WRAP_CONTENT;
23                 p.rightMargin = 5;
24                 p.topMargin = 5;
25                 p.setGravity(Gravity.CENTER);
26                 p.columnSpec = GridLayout.spec(j);
27                 p.rowSpec = GridLayout.spec(i);
28                 g.addView(b, p);
29             }
30     }
31 }
```

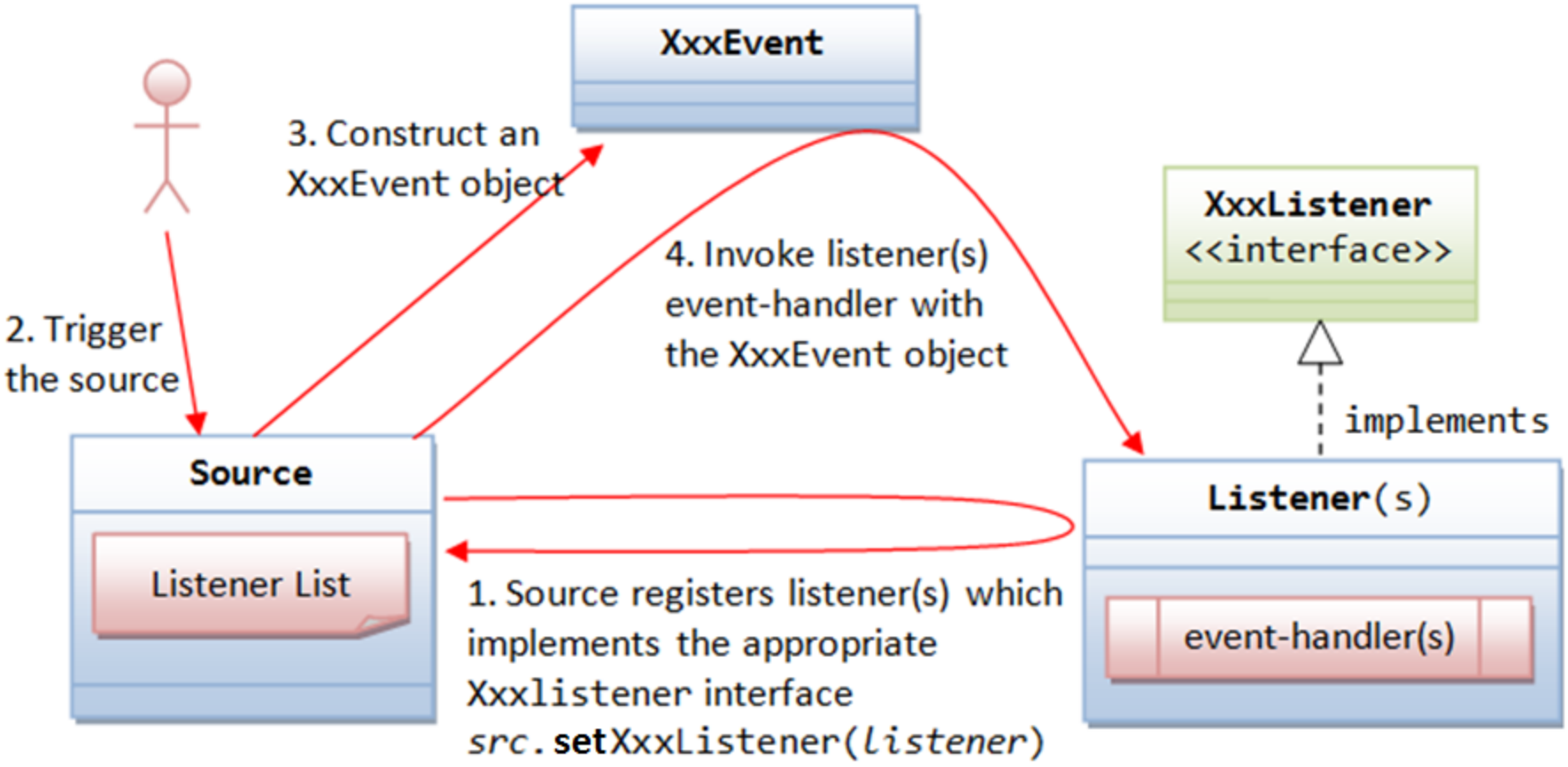
4.4- Events handling

- **MVC in events handling**



4.4- Events handling

- **Method for handling events**



4.4- Events handling

- **Steps of events handling**
  - Step1) Create a listener from interfaces & implements event-handler

```
∇ = new  the…xyz…ListenerInterface(){
      public  void  xyz-Event-Handler( …params… ){
            … do some thing …
      }
};
```

  - Step2) Register the listener to sources

```
theSource.set…xyz…Listener( ∇ );
```

4.4- Events handling

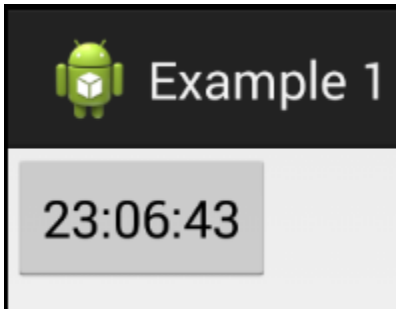
- **Some common events**

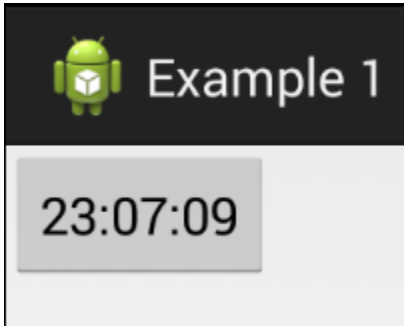
Stt	Sự kiện	Giao diện nghe sự kiện	Hàm xử lý
1	Chọn trên đối tượng	View.OnClickListener	OnClick(View 🏳)
2	Chọn và giữ trên đối tượng	View.OnLongClickListener	OnLongClick(View 🏳)
3	Thay đổi con trỏ hội tụ	View.OnFocusChangeListener	OnFocusChange(View 🏳, boolean hasFocus)
4	Tác động phím	View.OnKeyListener	OnKey(View 🏳, KeyEvent Δ)
5	Chạm trên đối tượng	View.OnTouchListener	OnTouch(View 🏳, MotionEvent Δ)
6	Kéo rê đối tượng	View.OnDragListener	OnDrag(View 🏳, DragEvent Δ)

4.4- Events handling

- **Example of events handling**

```
14 public class Event_Handling_Example extends Activity {
15     Button  button1;
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.ui_layout_example);
19         button1=(Button)findViewById(R.id.button1);
20         button1.setOnClickListener(cListener);
21     }
22     OnClickListener  cListener=new OnClickListener() {
23         public void onClick(View v) {
24             if(v==button1){
25                 Time now = new Time();
26                 now.setToNow();
27                 button1.setText(now.format("%k:%M:%S"));
28             }
29         }
30     };
31 }
```





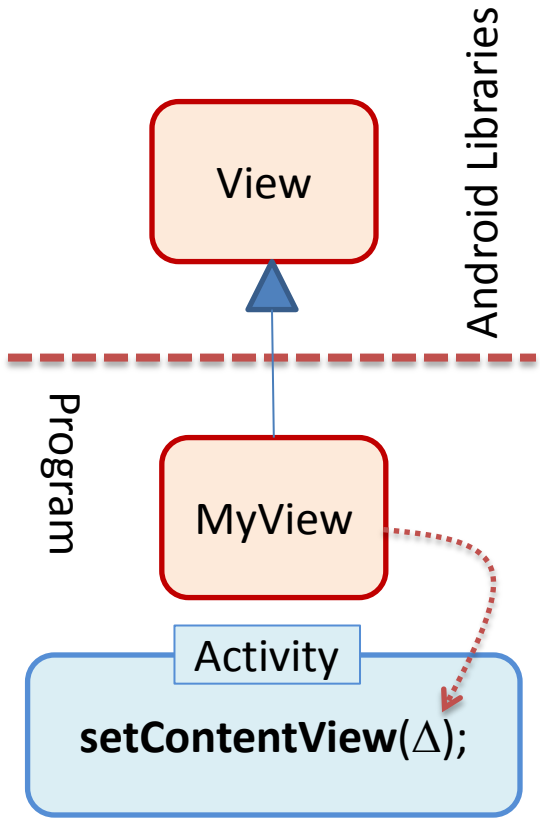
4.5- Graphics

- **Low-level UI programming**
  - To draw all thing on screen
  - We must determine:
    - where to draw
    - pen for using with color & pen style

4.5- Graphics

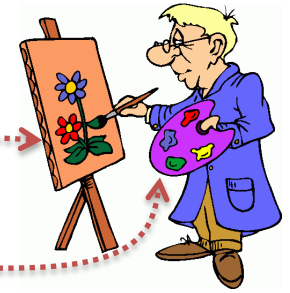
- **Programming with Graphics**
  - extends View for drawing

```
class MyView extends View{

    MyView(Context ct) {
        super(ct);

    }
    protected void onDraw(Canvas  cv){
        super.onDraw(cv);
        …do some thing for drawing here…
    }
}
```



  - create an object & **setContentView** to Activity

4.5- Graphics



- **Programming with Graphics**
  - drawing with "**Canvas**" & "**Paint**"
    - Step1) Create a paint object

Δ=new Paint();

- Step2) Set parameters to the paint for drawing

Δ.set*...xyz...*(…);

- Step3) Draw on canvas object with the paint

***theCanvas*.draw*...xyz...*(… , Δ);**

do this inside "onDraw" method

4.5- Graphics

- **Programming with Graphics**
  - some simple methods

Paint

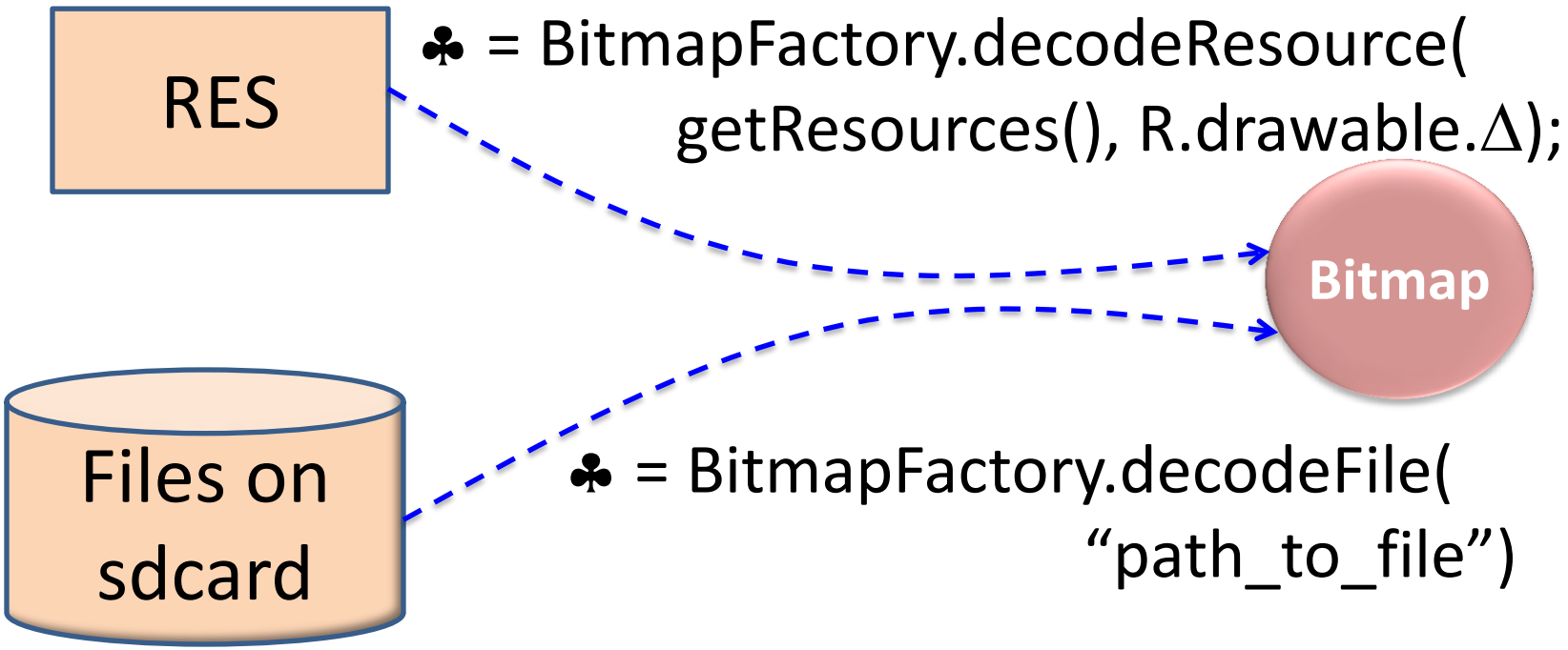
Stt	Tên lệnh	Ý nghĩa
1	*.setAntiAlias(true); *.setDither(true);	Chế độ vẽ nét mịn (đường nét trơn ở các hình)
2	*.setColor(màu);	Màu vẽ & tô
3	*.setStrokeWidth(w);	Độ rộng nét vẽ
4	*.setStyle(style);	Kiểu vẽ, sử dụng Paint.Style.* để thiết lập,…

Canvas

Stt	Tên lệnh	Ý nghĩa
1	*.drawLine(x1,y1, x2,y2, P);	Vẽ đoạn thẳng
2	*.drawRect(x1,y1, x2,y2, P);	Vẽ hình chữ nhật
3	*.drawCircle(x,y, r, P);	Vẽ hình tròn
5	*.drawRGB(r, g, b);	Tô nền của Canvas
6	*.drawRoundRect(RectF, rx,ry, P);	Vẽ hình chữ nhật vát góc theo rx,ry
7	*.drawText(S, x,y, P);	Vẽ văn bản
10	*.drawBitmap(Bitmap, x,y, P);	Vẽ ảnh trong đối tượng Bitmap
11	*.getWidth(); *.getHeight()	Lấy chiều rộng, cao màn hình, …

4.5- Graphics

- **get Bitmap for drawing**



4.5- Graphics

- **Example of graphics**

```
11 public class VD extends Activity {
12         protected void onCreate(Bundle ts) {
13             super.onCreate(ts);
14             setContentView(new MyView(this));
15         }
16 }
```

```
18 class MyView extends View{
19         Paint  p;    RectF  rect;
20         MyView(Context ct) {
21             super(ct);
22             p = new Paint();  rect = new RectF();
23         }
24         protected void onDraw(Canvas  cv){
25             super.onDraw(cv);
26             p.setColor(Color.RED);
27             rect.set(200, 10, 300, 70);
28             cv.drawArc(rect, 45, 270, true, p);
29             p.setStrokeWidth(5);
30             cv.drawLine(0, 0, 500, 200, p);
31             p.setAntiAlias(true);  p.setDither(true);
32             p.setColor(Color.BLUE);
33             cv.drawCircle(100, 100, 30, p);
34             cv.drawLine(500, 0, 0, 200, p);
35         }
36 }
```