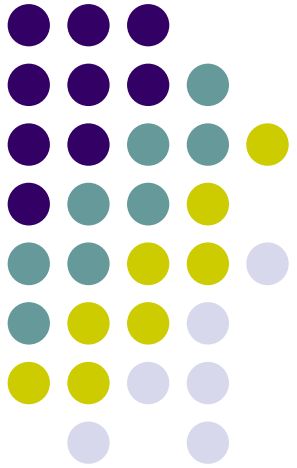
Τεχνολογία και Προγραμματισμός Κινητών Συσκευών

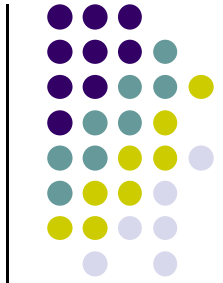
2b



Διεπαφή Χρήστη

Ιωάννης Έλληνας

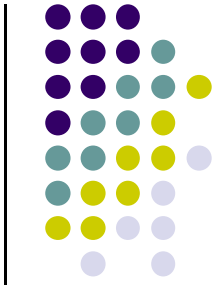
Διεπαφή χρήστη



- **Όψεις (Views ή widgets) – Πλήκτρο, πεδίο κειμένου, κλπ.**
- **Διατάξεις οθόνης (Layouts) – Σύνολο όψεων**
- **Δημιουργία όψεων με το αρχείο *activity_main.xml* στον φάκελο *res/layout*.**

```
@Override
public void onCreate(Bundle savedInstanceState)
{
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
}
```

Widgets

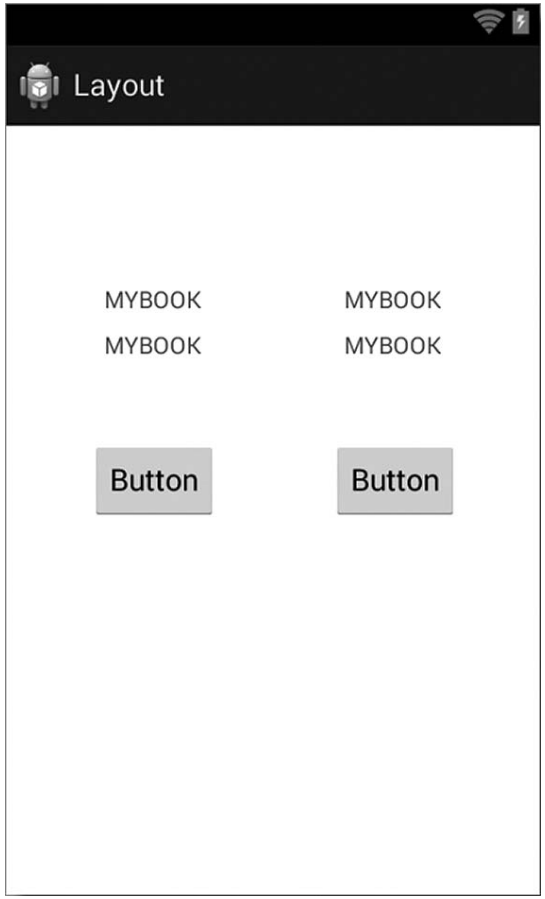


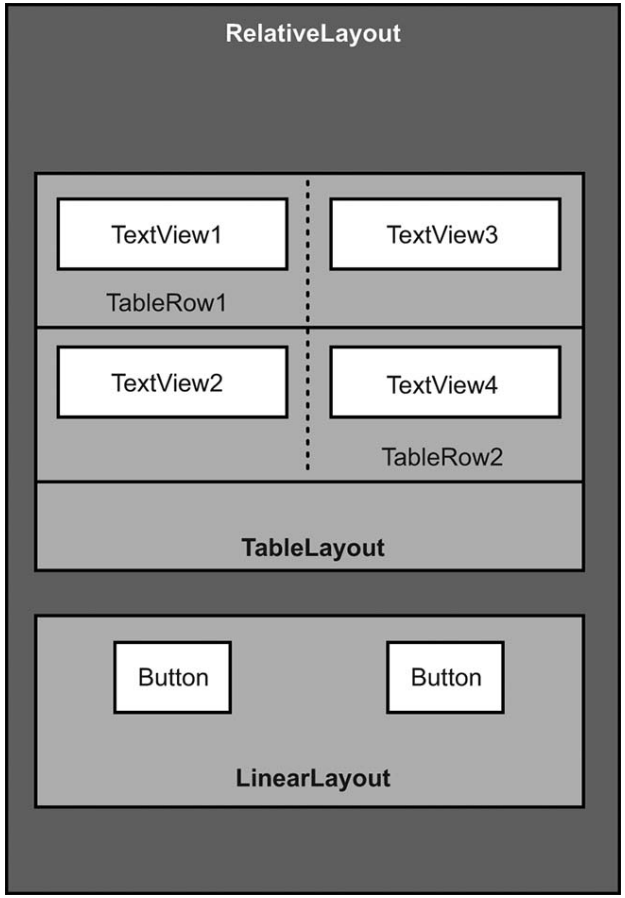
- **TextView** – Πεδίο κειμένου το οποίο μπορεί να είναι με πολλές γραμμές, μεταβλητής γραμματοσειράς και με αναδίπλωση λέξεων.
- **EditText** – Πεδίο εισαγωγής κειμένου το οποίο μπορεί να είναι με πολλές γραμμές και με αναδίπλωση λέξεων.
- **ListView** – Ανήκει στα ViewGroup και αποτελείται από πολλά TextView σε διάταξη λίστας.
- **Spinner** – Συνδυασμός πεδίου κειμένου και λίστας, όπου η επιλογή ενός στοιχείου της λίστας εμφανίζεται στο πεδίο κειμένου.
- **Button** – Πλήκτρο
- **CheckBox** – Πλήκτρο δύο καταστάσεων οι οποίες εμφανίζονται σαν επιλογή (check) ή μη επιλογή (uncheck).
- **RadioButton** – Ομάδα πλήκτρων με δύο καταστάσεις, από τα οποία μόνο ένα μπορεί να είναι επιλεγμένο.

Διατάξεις (Layouts)

- **LinearLayout** – Τοποθετεί κάθε όψη σε μια γραμμή κάθετη ή οριζόντια ανάλογα με τον προσανατολισμό της οθόνης. Με την παράμετρο *weight* ρυθμίζεται ο χώρος που θα καταλάβει το στοιχείο σε σχέση με το διαθέσιμο χώρο της διάταξης.
- **RelativeLayout** – Τοποθετεί τις όψεις σε θέσεις σχετικές με τη μεταξύ τους απόσταση και την απόστασή τους από τα όρια της οθόνης. Είναι η πιο δυναμική διάταξη, ιδιαίτερα για τη σχεδίαση UI που προορίζονται για μια ποικιλία διαστάσεων οθόνης. → **ConstraintLayout**
- **TableLayout** – Τοποθετεί τις όψεις στα κελιά ενός πλέγματος σειρών και στηλών. Κάθε σειρά εκπροσωπείται από το αντικείμενο *TableRow* το οποίο περιέχει μια όψη για κάθε κελί της σειράς.
- **GridLayout** – Δημιουργεί ένα πλέγμα σειρών και στηλών και τοποθετεί τις όψεις στα δημιουργούμενα κελιά. Μια όψη μπορεί να καταλάβει περισσότερα του ενός κελιά δημιουργώντας εύκολα διαφορετικές διατάξεις. Αν ένα κελί πρέπει να παραμείνει κενό, τότε τοποθετείται σε αυτό η κενή όψη (Space view) ή χρησιμοποιείται η παράμετρος *margin*.

Σχεδίαση Οθόνης με το Graphical Layout

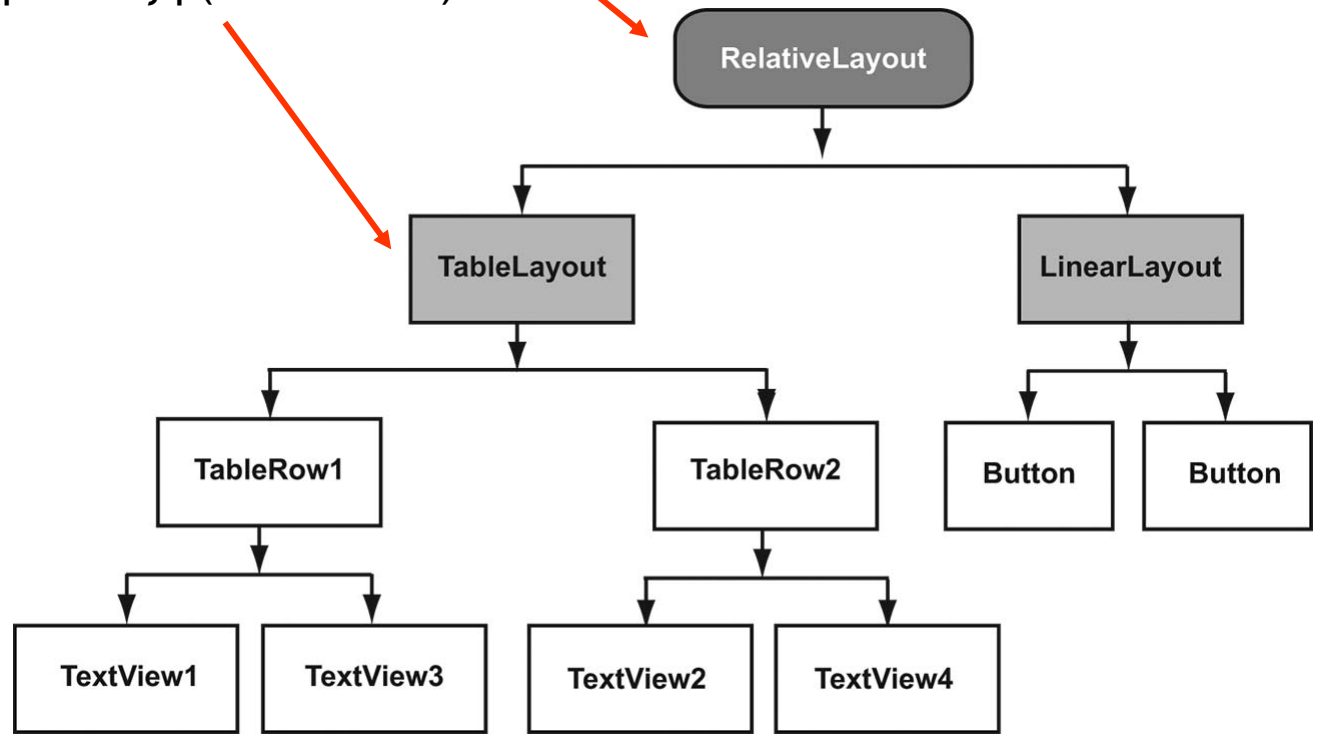




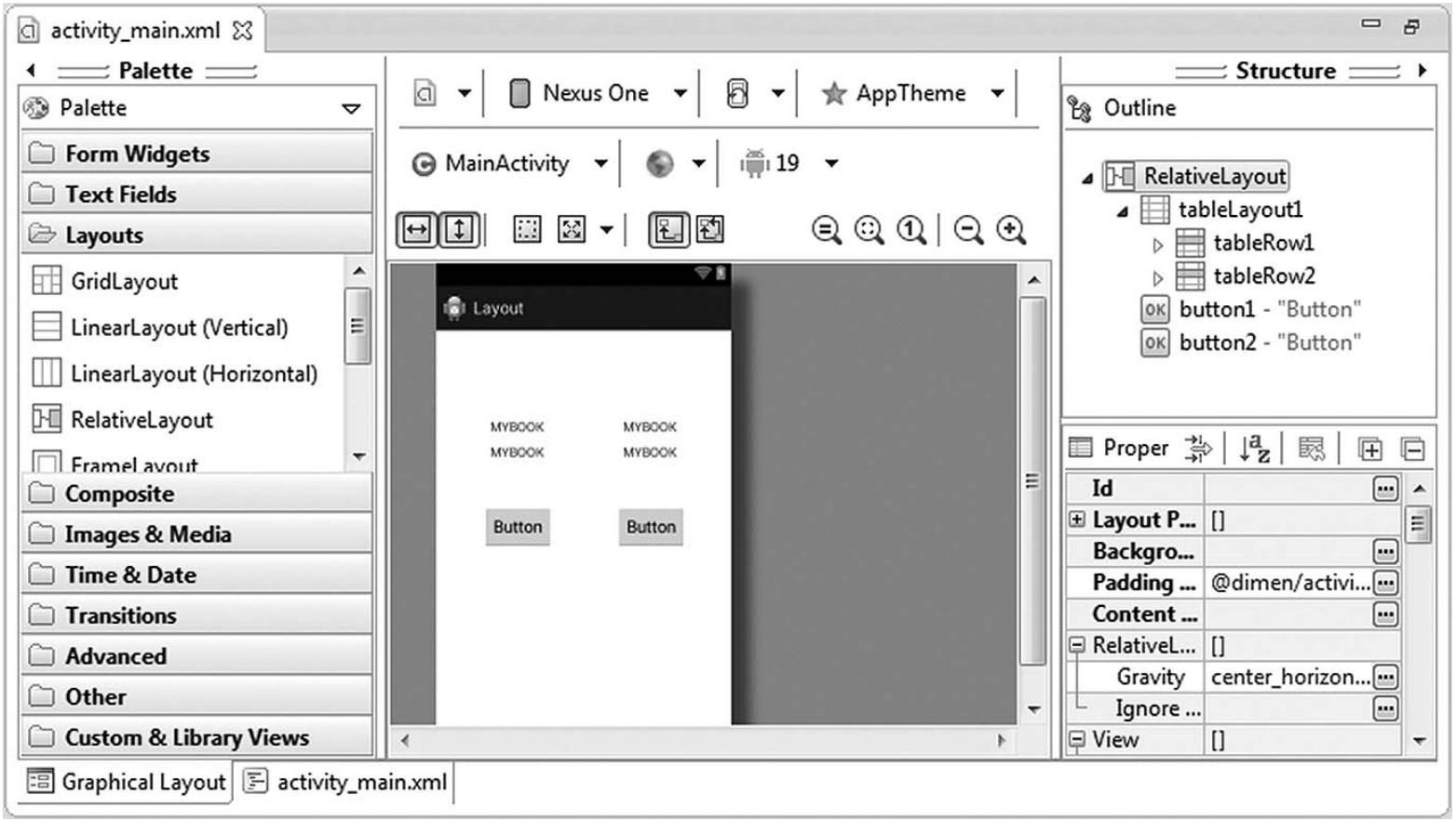
Ιεραρχική δομή

Γονική διατάξη (Parent View)

Παιδική διάταξη (Child View)

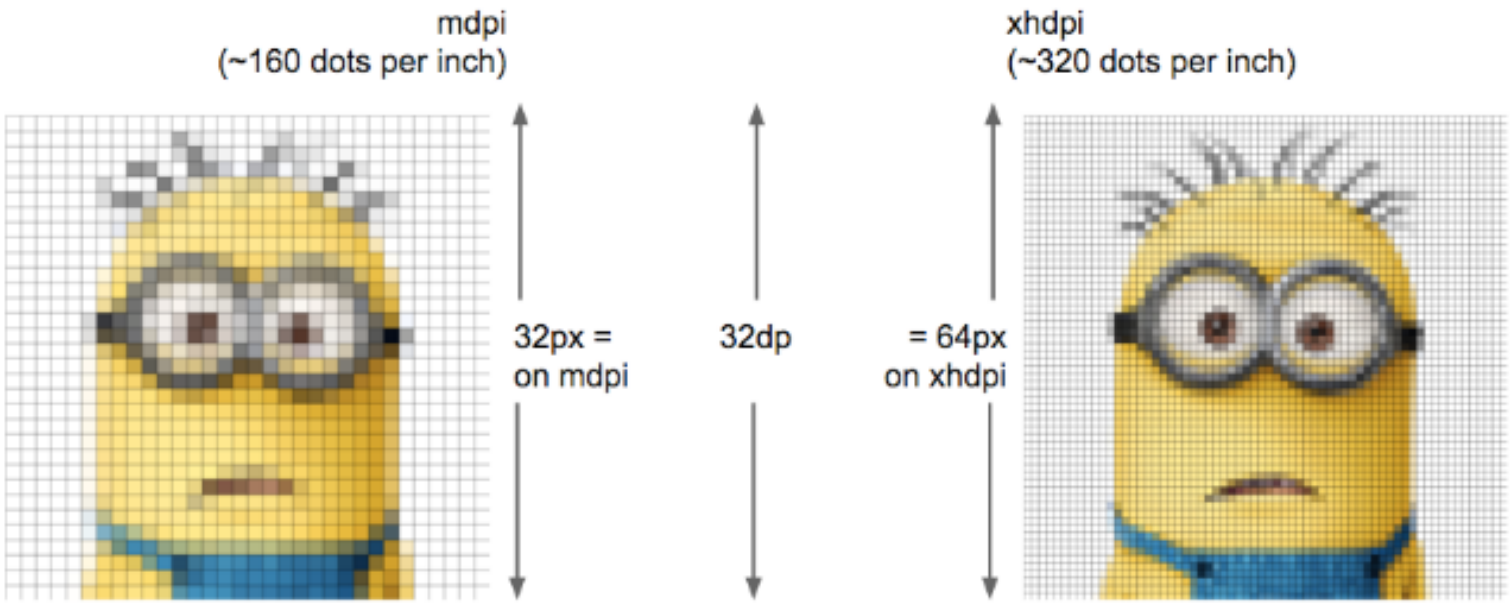


Γραφική Σχεδίαση Οθόνης



Μονάδες μέτρησης

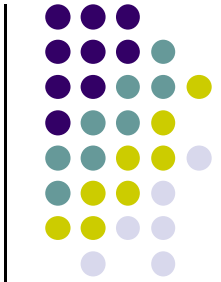
- **in** – ίντσες.
- **mm** – χιλιοστά.
- **pt** – σημεία (points). Σε μια ίντσα υπάρχουν 72 σημεία.
- **dp** – density independent pixels. Αντιπροσωπεύει τη φυσική πυκνότητα της οθόνης σε σχέση με μια οθόνη 160 dpi. Ένα dp είναι ένα pixel σε μια οθόνη 160 dpi. Η σχέση μεταξύ pixel και dp είναι: **px = dp * (dpi / 160)**. Σε μια οθόνη των 320 dpi, ένα dp αντιστοιχεί σε 2 pixels.
- **sp** – scale independent pixels. Μοιάζει με το dp αλλά εξαρτάται και από το μέγεθος του font που επιλέγει ο χρήστης. Συνήθως χρησιμοποιείται στον καθορισμό του μεγέθους ενός font.
- **px** – pixels στην οθόνη. Δεν συνιστάται η χρήση του γιατί διαφορετικές οθόνες έχουν διαφορετικό αριθμό pixels στην ίντσα.



px = dp * (dpi / 160)

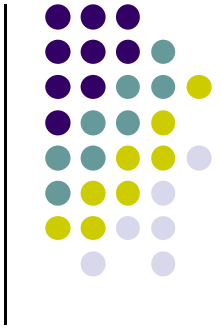
float px = sp * getResources().getDisplayMetrics().scaledDensity;

Ιδιότητες



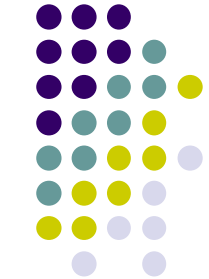
- **Προσανατολισμός (Orientation)** – Δείχνει αν η διάταξη είναι σειρά ή στήλη. Μπορεί να γίνει γραφικά ή χρησιμοποιώντας την ιδιότητα ***android:orientation*** στο αρχείο xml ή τη μέθοδο *setOrientation()* στον κώδικα της java.
- **Γέμισμα μοντέλου (Fill Model)** – Η τοποθέτηση όψεων στην οθόνη πρέπει να συνοδεύεται από την ιδιότητα του μεγέθους τους σε σχέση με τις διαστάσεις της γονικής διάταξης. Αυτό δηλώνεται με τα ***android:layout_width*** και ***android:layout_height*** τα οποία δέχονται τις εξής τιμές:
  - Απόλυτες διαστάσεις σε pixels (π.χ. 8 px).
  - Την τιμή ***wrap_content***, η οποία περιορίζει την αντίστοιχη διάσταση στο διάστημα που καταλαμβάνει η όψη. Αν το πλάτος αυτό δεν χωράει στην οθόνη, τότε πραγματοποιείται αναδίπλωση λέξεων (word wrap).
  - Την τιμή ***fill_parent***, η οποία επεκτείνει την αντίστοιχη διάσταση στο διάστημα που καταλαμβάνει η όψη.

Ιδιότητες



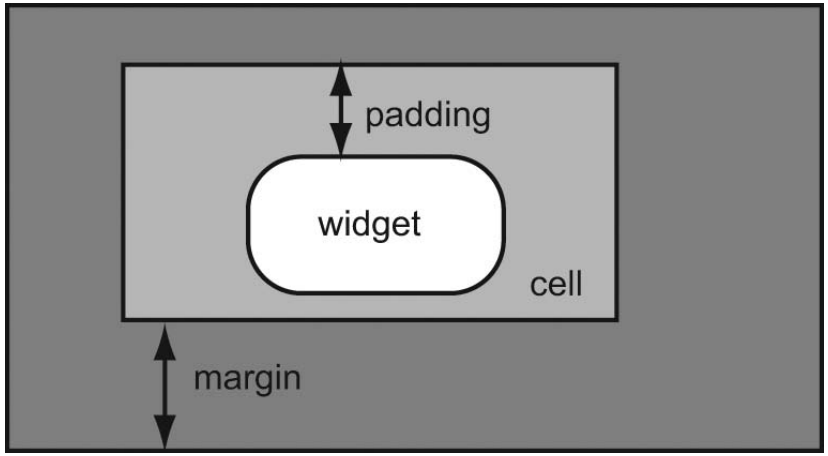
- **Βάρος (Weight)** – Το βάρος επιτρέπει την κατανομή ενός διαστήματος στις όψεις που πρόκειται να το καταλάβουν. Αφού τοποθετηθούν οι προηγούμενες ιδιότητες της όψης android:layout_width=”fill_parent” και android:layout_height=”fill_parent”, το βάρος σε μια σειρά ή στήλη καθορίζεται από την ιδιότητα **android:layout_weight=”n”.** Εάν n=1 για δύο widget, το ελεύθερο διάστημα θα μοιραστεί ομοιόμορφα, ενώ αν n=1 για το ένα και n=2 για το άλλο τότε το δεύτερο θα καταλάβει διπλάσιο χώρο από το πρώτο.
- **Βαρύτητα (Gravity)** – Τα αντικείμενα τοποθετούνται σε μια διάταξη με ευθυγράμμιση αριστερά και επάνω. Για μια διαφορετική ευθυγράμμιση χρησιμοποιείται η ιδιότητα **android:layout_gravity** (ή *setGravity()* στη java) με τιμές κεντραρίσματος (**center_Horizontal** ή **center_Vertical**) και ευθυγράμμισης αριστερά ή δεξιά (**align_Left** ή **align_Right**).

Ιδιότητες



- **Γέμισμα (Padding)** – Τα widgets βρίσκονται μέσα σε ένα νοητό κουτί και τοποθετούνται το ένα δίπλα στο άλλο χωρίς πρόβλεψη κάποιας απόστασης μεταξύ τους. Η ιδιότητα που δημιουργεί τη θέση του widget μέσα στο κουτί του είναι η **android:padding** (ή *setPadding()* στη java). Μπορεί να έχει τη μορφή **paddingTop, paddingBottom, paddingLeft, paddingRight** και οι τιμές που μπορεί να πάρει μπορεί να είναι σε pixels (π.χ. 4 px).
- **Περιθώριο (Margin)** – Η θέση ενός widget ως προς κάτι άλλο εξωτερικά του καλείται περιθώριο. Το περιθώριο καθορίζεται με την ιδιότητα **android:layout_margin** και μπορεί να έχει τη μορφή **marginTop, marginBottom, marginLeft, marginRight** και οι τιμές που μπορεί να πάρει είναι σε dp (π.χ. 10 dp).

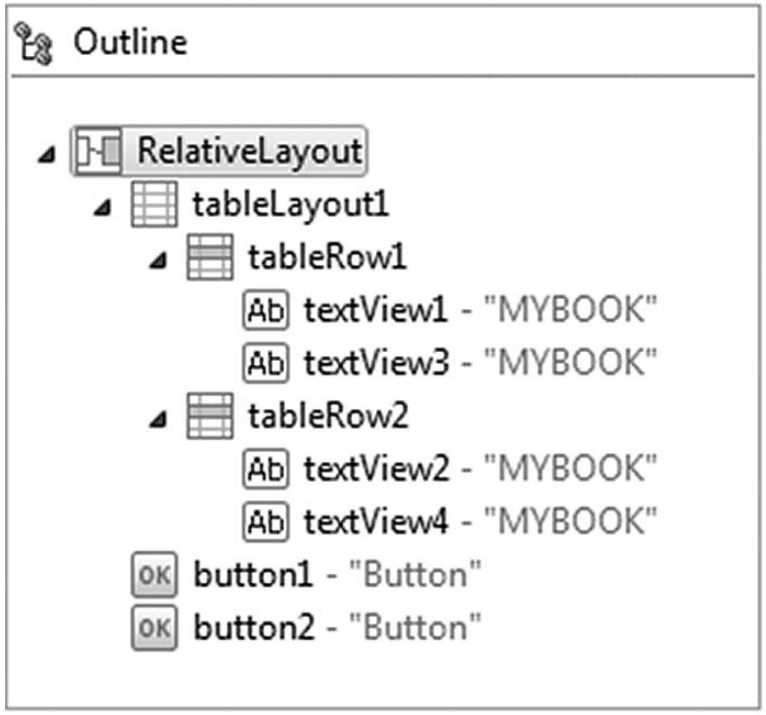
Ιδιότητες



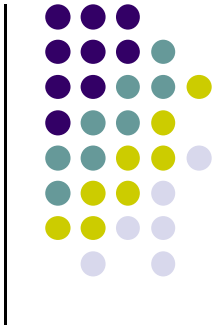
- Μεταβολή κειμένου στο πεδίο κειμένου (TextView), στο πεδίο εισαγωγής κειμένου (EditText) και στο πλήκτρο (Button):
  android:text=”Name” ή **android:text=”@string/txt”**
  όπου “txt=Name” στο αρχείο *strings.xml*.
- Ταυτότητα widget (ID) για χειρισμό τους στον κώδικα java:
  android:id=”@+id/identity”.

Παράδειγμα “Layout”

- Δημιουργία της προηγούμενης διάταξης οθόνης με το Graphical Layout του Eclipse.



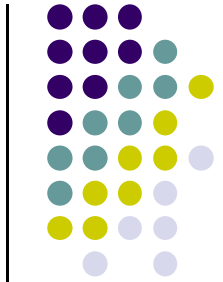
Περιγραφικό αρχείο



```
<TableLayout
  android:id = "@+id/tableLayout1"
  android:layout_width = "fill_parent"
  android:layout_height = "wrap_content"
  android:layout_alignParentTop = "false"
  android:layout_centerHorizontal = "true"
  android:layout_marginTop = "80dp" >
  <TableRow
    android:id = "@+id/tableRow1"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content" >
    <TextView
      android:id = "@+id/textView1"
      android:layout_width = "wrap_content"
      android:layout_height = "wrap_content"
      android:layout_weight = "0.5"
      android:gravity = "center"
      android:text = "@string/text1" />
```

```
    <TextView
      android:id = "@+id/textView3"
      android:layout_width = "wrap_content"
      android:layout_height = "wrap_content"
      android:layout_weight = "0.5"
      android:gravity = "center"
      android:text = "@string/text1" />
  </TableRow>
  <TableRow
    ……
  </TableRow>
</TableLayout>
  <Button
        ……      />
  <Button
        ……      />
</RelativeLayout>
```

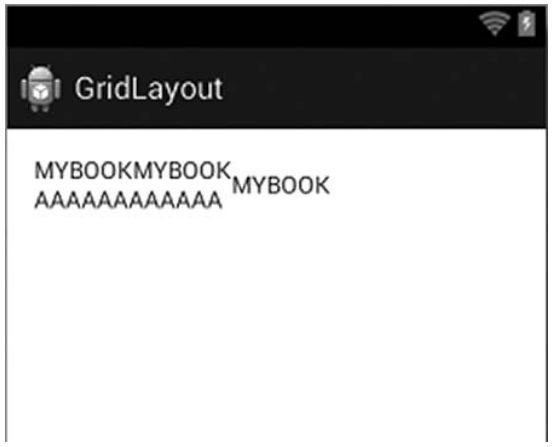
Αρχείο strings.xml

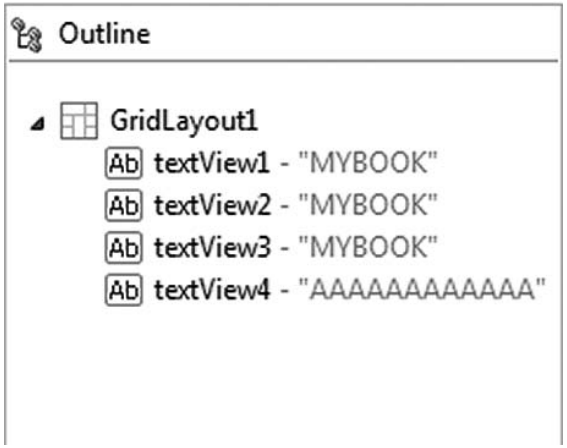


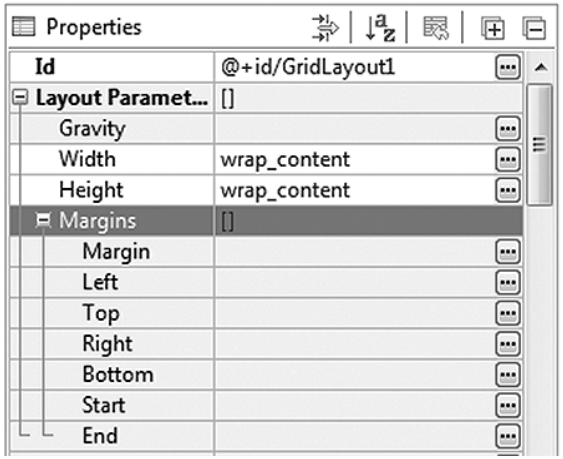
```
<?xml version = "1.0" encoding="utf-8"?>
<resources>

<string name = "app_name">Layout</string>
<string name = "action_settings">Settings</string>
<string name = "text1">MYBOOK</string>
<string name = "btn">Button</string>

</resources>
```

Διάταξη “GridLayout”

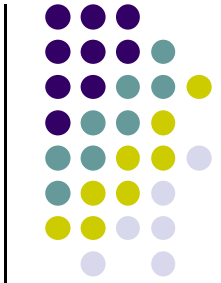
- Δημιουργία του project GridLayout.



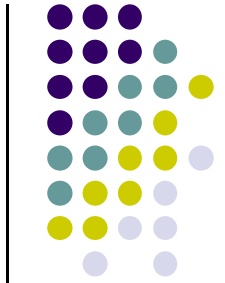




Περιγραφικό αρχείο

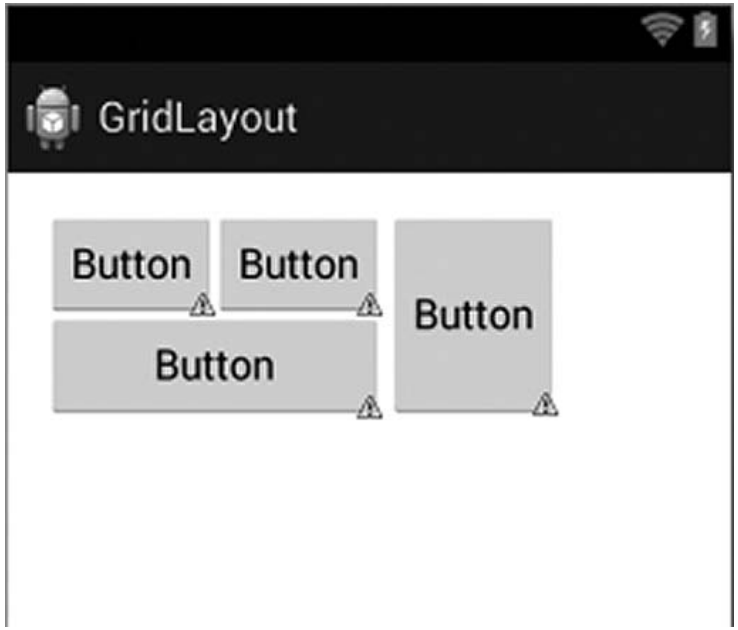


```
<GridLayout
xmlns:android = "http://schemas.android.com/apk/res/android"
xmlns:tools = "http://schemas.android.com/tools"
android:id = "@+id/GridLayout1"
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:columnCount = "3"
android:paddingBottom = "@dimen/activity_vertical_margin"
android:paddingLeft = "@dimen/activity_horizontal_margin"
android:paddingRight = "@dimen/activity_horizontal_margin"
android:paddingTop = "@dimen/activity_vertical_margin"
tools:context = ".MainActivity" >

<TextView
  android:id = "@+id/textView1"
  android:layout_column = "0"
  android:layout_gravity = "left|center_vertical"
  android:layout_row = "0"
  android:text = "@string/text1" />
```

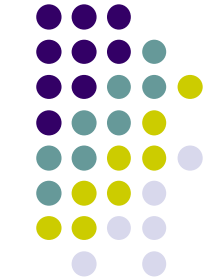


```
<TextView
  android:id = "@+id/textView2"
  android:layout_column = "1"
  android:layout_gravity = "left|top"
  android:layout_row = "0"
  android:text = "@string/text1" />
<TextView
  android:id = "@+id/textView3"
  android:layout_column = "2"
  android:layout_gravity = "center_vertical"
  android:layout_row = "0"
  android:layout_rowSpan = "2"
  android:text = "@string/text1" />
<TextView
  android:id = "@+id/textView4”
  android:layout_column = "0"
  android:layout_columnSpan = "2"
  android:layout_gravity = "fill_horizontal"
  android:layout_row = "1"
  android:text = "@string/text2" />

</GridLayout>
```

Δυνατότητες



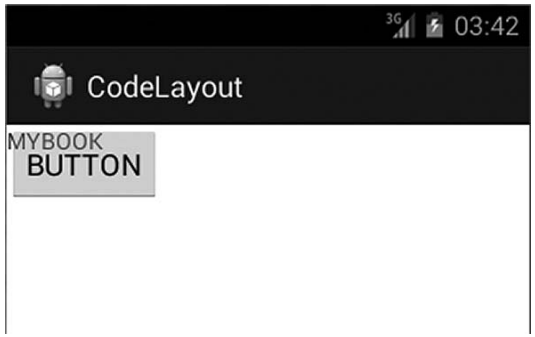
Σχεδίαση με κώδικα java



- Γονική όψη: **setContentView()**
- Επόμενες όψεις: **addView()**
- Δημιουργία και ανάκτηση ταυτότητας όψης: **setID(), getID()**

- Η εμφάνιση των διατάξεων καθορίζεται από παραμέτρους (π.χ. μέγεθος διάταξης σε σχέση με τη γονική διάταξη) ενώ η εμφάνιση των όψεων καθορίζεται από τις ιδιότητες (π.χ. κείμενο, χρώμα υποβάθρου, περιθώρια, κλπ).

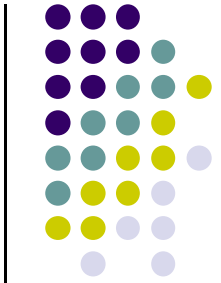
Παράδειγμα “*CodeLayout*”




```
public class MainActivity extends Activity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    RelativeLayout mLayout = new RelativeLayout(this);
    TextView mView = new TextView(this);
    mView.setId(1);
    mView.setText("MYBOOK");
    mLayout.addView(mView);
    Button btn = new Button(this);
    btn.setId(2);
    btn.setText("BUTTON");
    mLayout.addView(btn);
    setContentView(mLayout);
  }
```

```
package com.example.codelayout;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;
```

Τοποθέτηση παραμέτρων



- Δημιουργείται ένα αντικείμενο παραμέτρων για κάθε όψη:

```
RelativeLayout.LayoutParams mViewparam = new RelativeLayout.LayoutParams (
     RelativeLayout.LayoutParams.WRAP_CONTENT,
     RelativeLayout.LayoutParams.WRAP_CONTENT);
mLayout.addView(mView,mViewparam); // or mView.setLayoutParams(mViewparam);

RelativeLayout.LayoutParams btnparam = new RelativeLayout.LayoutParams (
     RelativeLayout.LayoutParams. WRAP_CONTENT,
     RelativeLayout.LayoutParams. WRAP_CONTENT);
mLayout.addView(btn,btnparam); // or btn.setLayoutParams(btnparam);
```

- Προσθήκη παραμέτρων στις όψεις:

```
mViewparam.addRule(RelativeLayout.CENTER_HORIZONTAL);
btnparam.addRule(RelativeLayout.CENTER_HORIZONTAL);
btnparam.addRule(RelativeLayout.CENTER_VERTICAL);
```

Τοποθέτηση παραμέτρων

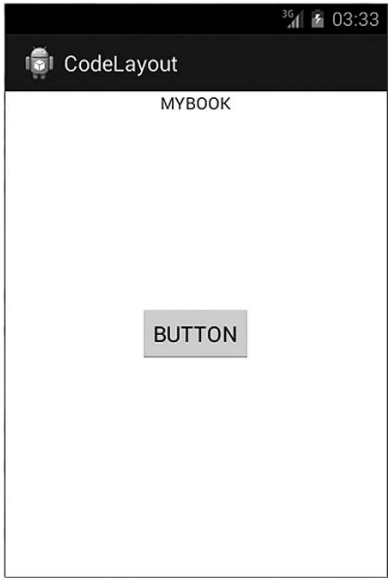
- Πεδίο κειμένου κάτω από το πλήκτρο:

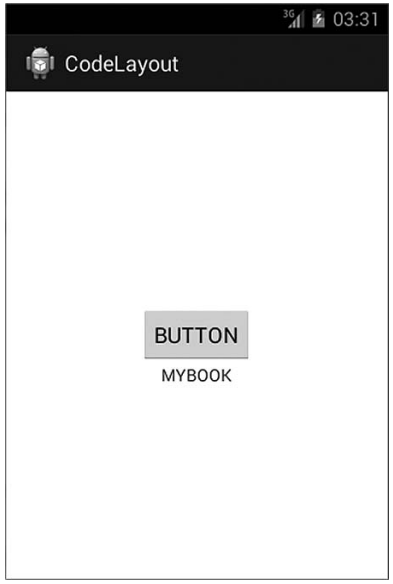
  mViewparam.addRule(RelativeLayout.BELOW, btn.getId());

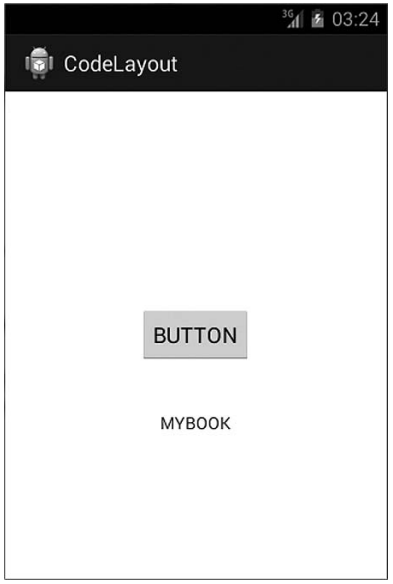
- *Απομάκρυνση από το πλήκτρο:*

  setMargins(left in dp , top in dp , right in dp , bottom in dp);

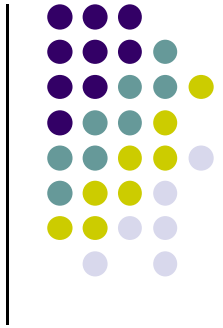
  mViewparam. setMargins(0 , 40 , 0 , 0);







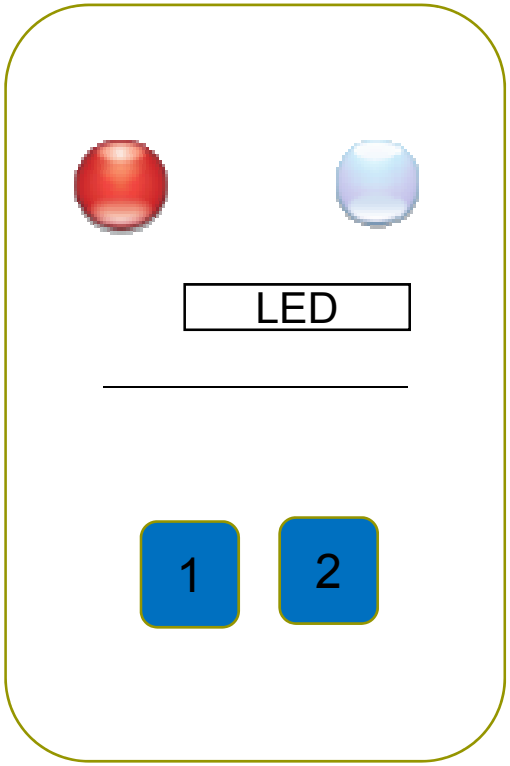
Constraint Layout



Με τη βοήθεια της εφαρμογής “constraint-layout-start” και των σημειώσεων “Σημειώσεις Android” εξετάζονται οι δυνατότητες του Constraint Layout το οποίο θα χρησιμοποιείται εφεξής.

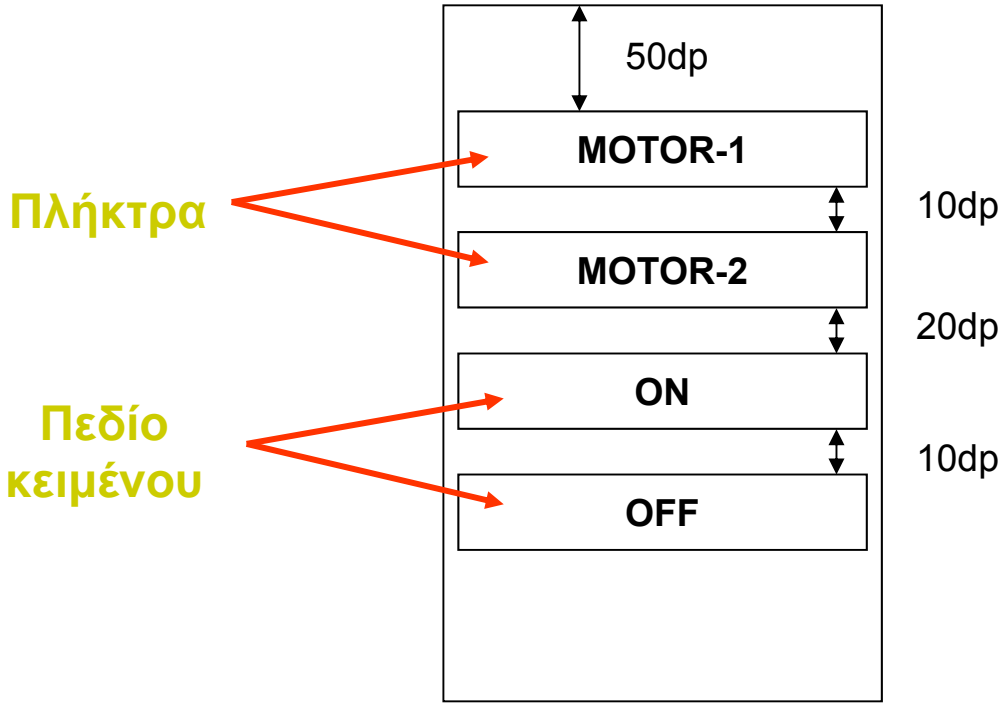
Άσκηση Πράξης

- Να δημιουργηθεί μια νέα εφαρμογή.
- Να σχεδιαστεί με Constraint Layout η εξής οθόνη:



Εργασία

- Να δημιουργηθεί το project MyDisplay το οποίο όταν εκτελεστεί θα μας δώσει την εξής οθόνη χρήστη (με τη βοήθεια του Graphical Layout):



Εργασία

- Γράψτε τον κώδικα του αρχείου activity_main.xml και παρατηρείστε την αντιστοιχία των ιδιοτήτων με τον γραφικό τρόπο σχεδίασης.
- Δημιουργείστε την ίδια οθόνη μόνο με κώδικα java.
- Τοποθετείστε στην εργασία σας τον κώδικα και τις εικόνες που προκύπτουν.
- Συνδέστε το κινητό σας στον υπολογιστή και εκτελέστε την εφαρμογή σας. Πάρτε την εικόνα που παρουσιάζεται στη συσκευή σας (capture) και τοποθετείστε την στην εργασία σας.