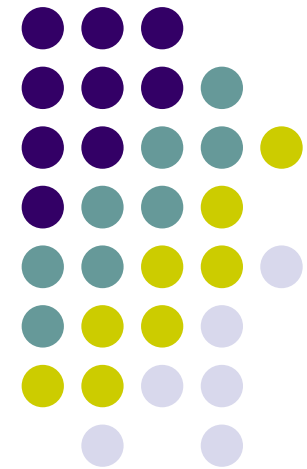


Τεχνολογία και Προγραμματισμός Κινητών Συσκευών

2a



Δραστηριότητες
Ιωάννης Έλληνας



Περιεχόμενα

- Δραστηριότητες (Activities)
- Activity life cycle
- Παράδειγμα



Εφαρμογές & Δραστηριότητες

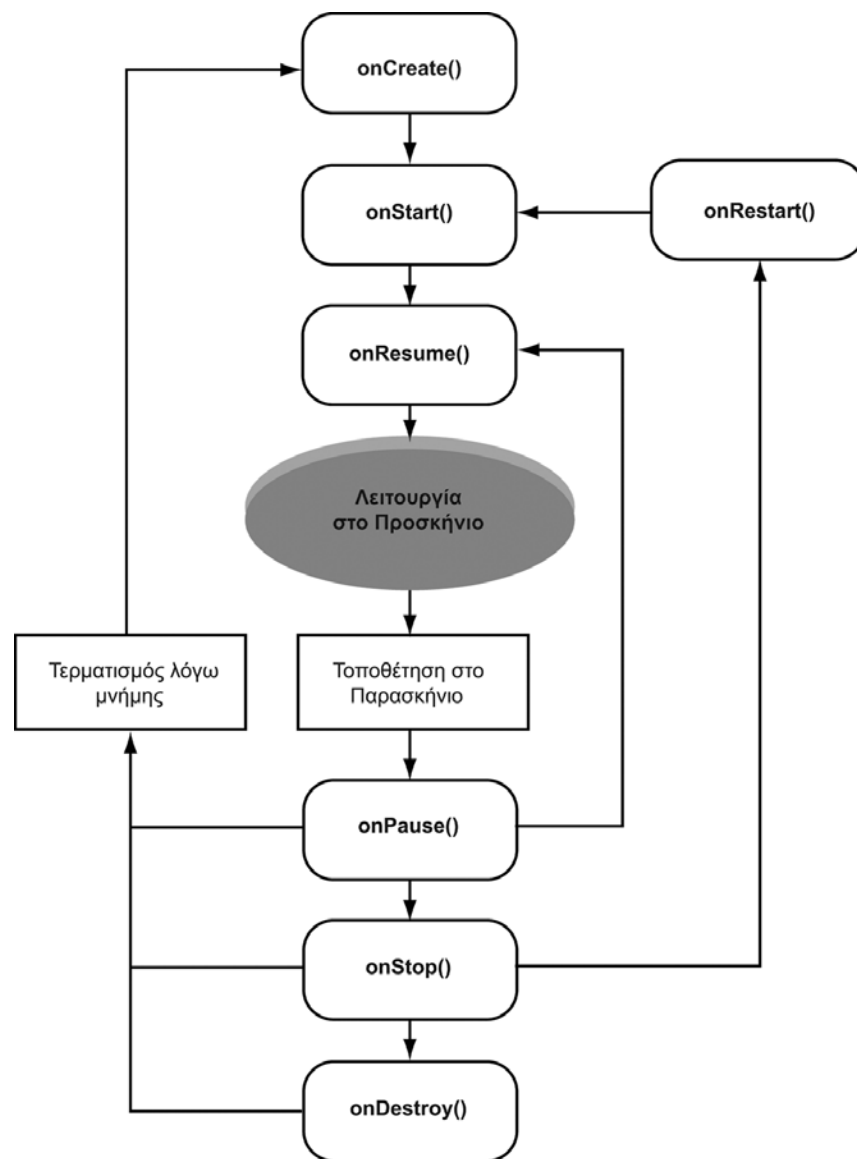


- Το λειτουργικό σύστημα χειρίζεται τις εφαρμογές σαν ανεξάρτητες διαδικασίες διαθέτοντας την εικονική μηχανή Dalvik για κάθε μια και λαμβάνοντας πάντοτε υπ' όψη τους περιορισμένους πόρους μνήμης που διαθέτει μια κινητή συσκευή.
- Μια διαδικασία, που φιλοξενεί μια εφαρμογή, μπορεί να είναι στο προσκήνιο (Foreground Process), οπότε έχει την υψηλότερη προτεραιότητα, να είναι μόνο ορατή (Visible Process), να αποτελεί υπηρεσία που εκτελείται στο παρασκήνιο ανεξάρτητα από την τρέχουσα εφαρμογή (Service Process), να εκτελείται στο παρασκήνιο (Background Process) ή να είναι κενή (Empty Process), οπότε έχει τη χαμηλότερη προτεραιότητα.
- Η πιο σημαντική συνιστώσα μιας εφαρμογής είναι η δραστηριότητα (Activity) η οποία παρέχει το UI (User Interface) σε μια συσκευή Android με μια επέκταση στην κλάση Activity.
- Η δραστηριότητα είναι σε μια από τρεις καταστάσεις: **active, pause, stop**.
- **Active** = Ορατή και έχει αλληλεπίδραση με τον χρήστη.
- **Pause** = Μερικά ορατή (επικαλυπτόμενη από άλλη). Είναι στη μνήμη και μπορεί να επανέλθει στο προσκήνιο.
- **Stop** = Μη ορατή. Το λειτουργικό μπορεί να τη διαγράψει για απελευθέρωση πόρων.

Κύκλος ζωής δραστηριότητας



- Οι καταστάσεις των Activities διατηρούνται στο Activity Stack.
- Κύρια δραστηριότητα ή κύριο νήμα (main thread) ή νήμα διεπαφής χρήστη (UI thread).
- Ενέργεια MAIN – κατηγορία LAUNCHER στο αρχείο Μανιφέστου. Κορυφή σωρού και σπρώχνεται προς τα κάτω αν ενεργοποιηθεί νέα δραστηριότητα.
- **Ενεργή (Active)** – Η δραστηριότητα βρίσκεται στην κορυφή του σωρού και εκτελείται ενημερώνοντας την οθόνη και διασφαλίζοντας τη διεπαφή με τον χρήστη. Έχει την υψηλότερη προτεραιότητα και επομένως τη μικρότερη πιθανότητα τερματισμού από το σύστημα για ανεύρεση πόρων.
- **Παύση (Pause)** - Η δραστηριότητα είναι ορατή αλλά επικαλύπτεται από άλλη δραστηριότητα η οποία διασφαλίζει τη διεπαφή με το χρήστη. Διατηρείται στη μνήμη και έρχεται πάλι στο προσκήνιο όταν τερματιστεί η δραστηριότητα που την καλύπτει και επιστρέψει στην κορυφή του σωρού.
- **Στάση (Stop)** - Η δραστηριότητα δεν είναι ορατή επειδή επικαλύπτεται από άλλες δραστηριότητες. Διατηρείται στη μνήμη αλλά με μικρή προτεραιότητα και επομένως υπάρχει μεγάλη πιθανότητα τερματισμού της από το σύστημα για ανεύρεση πόρων μνήμης. Αν δεν τερματιστεί, μπορεί να επανέλθει στο προσκήνιο μετά τον τερματισμό των άλλων δραστηριοτήτων.
- **Τερματισμός (Destroy)** - Η δραστηριότητα τερματίζεται είτε με επιλογή του χρήστη είτε από το σύστημα για να απελευθερώσει πόρους μνήμης. Διαγράφεται από το σωρό και η επαναφορά της μπορεί να γίνει με επανεκκίνησή της από την εφαρμογή.



Μέθοδοι δραστηριότητας



- **onCreate()** – Είναι η πρώτη μέθοδος που διαχειρίζεται τη δημιουργία μιας δραστηριότητας. Η μέθοδος καλείται με την παράμετρο “Bundle savedInstanceState”, η οποία αφορά τις πληροφορίες που έχει αποθηκεύσει το σύστημα από προηγούμενη κλήση της δραστηριότητας (π.χ. τη διάταξη οθόνης, δεδομένα πεδίων κειμένου, κλπ). Σε αυτήν γίνεται αρχικοποίηση δεδομένων, η σύνδεση με τους πόρους της εφαρμογής και η δημιουργία της διάταξης οθόνης.
- **onRestart()** – Είναι η μέθοδος που καλείται μετά από επανεκκίνηση της δραστηριότητας από την κατάσταση Στάσης.
- **onStart()** - Είναι η μέθοδος που καλείται αμέσως μετά από τις δύο προηγούμενες και αφορά την έναρξη της διαδικασίας εμφάνισης του UI στον χρήστη. Ακολουθείται από την *onResume()* εάν γίνει ορατή στον χρήστη ή την *onStop()* εάν καλυφθεί από άλλη δραστηριότητα.
- **onResume()** - Είναι η μέθοδος που καλείται αμέσως μετά από την προηγούμενη εφόσον αυτή τοποθετηθεί στην κορυφή του σωρού και διασφαλίζει τη διεπαφή με τον χρήστη. Δεσμεύει πόρους και εκκινεί ήχο και βίντεο.
- **onPause()** - Είναι η μέθοδος που καλείται μόλις κάποια άλλη δραστηριότητα την επικαλύψει αλλά είναι ακόμα ορατή στον χρήστη. Ακολουθείται από την *onResume()* εάν επανέλθει στο προσκήνιο και γίνει ορατή στον χρήστη ή την *onStop()* εάν παύσει να είναι ορατή στον χρήστη. Απελευθερώνει πόρους, σώζει δεδομένα και σταματά ήχο και βίντεο.
- **onStop()** - Είναι η μέθοδος που καλείται μόλις παύσει να είναι ορατή στον χρήστη. Ακολουθείται από την *onRestart()* εάν επανέλθει στο προσκήνιο ή την *onDestroy()* εάν τερματιστεί.
- **onDestroy()** - Είναι η μέθοδος που καλείται όταν τερματίζεται η λειτουργία μιας δραστηριότητας είτε επειδή έχει εκτελέσει την αποστολή της είτε γιατί το λειτουργικό σύστημα θέλει πόρους μνήμης είτε γιατί έχουν αλλάξει οι απαιτήσεις λειτουργίας της δραστηριότητας (π.χ η διάταξη οθόνης έχει γίνει οριζόντια αντί για κάθετη).

Ενεργοποίηση



- Δήλωση στο αρχείο μανιφέστου.

```
<intent-filter>  
  <action android:name = "android.intent.action.MAIN" />  
  <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>
```

- Χρησιμοποιώντας τη μέθοδο **startActivity()** η οποία δέχεται σαν παράμετρο μια πρόθεση. Η Activity2 πρέπει να δηλώνεται στο μανιφέστο.

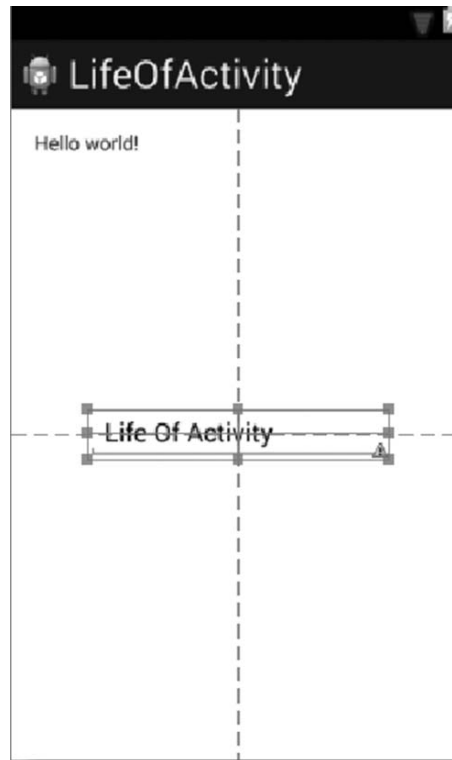
```
Intent intent = new Intent(getApplicationContext(), Activity2.class);  
startActivity(intent);
```

- Μερικές φορές μια δραστηριότητα πρέπει να ενεργοποιήσει μια άλλη δραστηριότητα και να περιμένει τα αποτελέσματα μετά την εκτέλεσή της. Σε αυτήν την περίπτωση χρησιμοποιείται η μέθοδος **Activity.startActivityForResult()**. Τα αποτελέσματα θα ληφθούν από τη μέθοδο επανάκλησης **onActivityResult()**.

Παράδειγμα



- **Project = LifeOfActivity**
- **Δημιουργία διάταξης οθόνης με το Graphical Layout.**





```
package gr.mybook.lifeofactivity;

import android.os.Bundle;
import android.app.Activity;
import android.util.Log;
import android.view.Menu;

public class MainActivity extends Activity {
    private static final String TAG = "gr.mybook.lifeofactivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.i(TAG, "onCreate callback method");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.i(TAG, "onResume callback method");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.i(TAG, "onRestart callback method");
    }
```

```
    @Override
    protected void onStart() {
        super.onStart();
        Log.i(TAG, "onStart callback method");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.i(TAG, "onPause callback method");
    }
```



```
@Override
protected void onStop() {
    super.onStop();
    Log.i(TAG,"onStop callback method");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.i(TAG,"onDestroy callback method");
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    Log.i(TAG,"onSaveInstanceState callback method");
}

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    Log.i(TAG,"onRestoreInstanceState callback method");
}
}
```

Αποτελέσματα



- Συνδέεται κινητή συσκευή Android και εκτελείται η εφαρμογή.

Application	Tag	Text
	<code>gr.mybook.lifeofactivity</code>	<code>onCreate callback method</code>
	<code>gr.mybook.lifeofactivity</code>	<code>onStart callback method</code>
	<code>gr.mybook.lifeofactivity</code>	<code>onResume callback method</code>

- Η οθόνη σβήνει:

Application	Tag	Text
	<code>gr.mybook.lifeofactivity</code>	<code>onCreate callback method</code>
	<code>gr.mybook.lifeofactivity</code>	<code>onStart callback method</code>
	<code>gr.mybook.lifeofactivity</code>	<code>onResume callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onSaveInstanceState callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onPause callback method</code>



- Πλήκτρο HOME για επαναφορά οθόνης:

Application	Tag	Text
	<code>gr.mybook.lifeofactivity</code>	<code>onCreate</code> callback method
	<code>gr.mybook.lifeofactivity</code>	<code>onStart</code> callback method
	<code>gr.mybook.lifeofactivity</code>	<code>onResume</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onSaveInstanceState</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onPause</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onResume</code> callback method

- Πατώντας το πλήκτρο “HOME”, η εφαρμογή εισέρχεται σε κατάσταση στάσης και εκτελείται η μέθοδος `onStop` αφού πρώτα αποθηκευτεί η τρέχουσα κατάσταση και εκτελεστεί η μέθοδος παύσης

Application	Tag	Text
	<code>gr.mybook.lifeofactivity</code>	<code>onCreate</code> callback method
	<code>gr.mybook.lifeofactivity</code>	<code>onStart</code> callback method
	<code>gr.mybook.lifeofactivity</code>	<code>onResume</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onSaveInstanceState</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onPause</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onResume</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onSaveInstanceState</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onPause</code> callback method
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onStop</code> callback method



- Ενεργοποίηση της εφαρμογής ξανά (onRestart):

Application	Tag	Text
	gr.mybook.lifeofactivity	onCreate callback method
	gr.mybook.lifeofactivity	onStart callback method
	gr.mybook.lifeofactivity	onResume callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onSaveInstanceState callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onPause callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onResume callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onSaveInstanceState callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onPause callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onStop callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onRestart callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onStart callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onResume callback method

- Η εφαρμογή εκτελείται και αμέσως τερματίζεται με το πλήκτρο επιστροφής:

Application	Tag	Text
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onCreate callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onStart callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onResume callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onPause callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onStop callback method
gr.mybook.lifeofactivity	gr.mybook.lifeofactivity	onDestroy callback method



- Η εφαρμογή εκτελείται και η συσκευή περιστρέφεται οριζόντια:

Application	Tag	Text
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onCreate callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onStart callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onResume callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onSaveInstanceState callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onPause callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onStop callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onDestroy callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onCreate callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onStart callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onRestoreInstanceState callback method</code>
<code>gr.mybook.lifeofactivity</code>	<code>gr.mybook.lifeofactivity</code>	<code>onResume callback method</code>

Άσκηση Πράξης



- Δημιουργείστε το project myActivity στο οποίο θα υπάρχουν όλες οι καταστάσεις της κύριας δραστηριότητας.
- Δημιουργείστε TAG παρακολούθησης των καταστάσεων της κύριας δραστηριότητας με το αντίστοιχο φίλτρο στο LogCat.
- Συνδέστε μια κινητή συσκευή και κάνετε όλες τις παρατηρήσεις που έχουν αναφερθεί στη διάλεξη.
- Καταγράψτε όλες τις περιπτώσεις που εμφανίζονται στο παράθυρο LogCat στην εργασία σας σχολιάζοντας την κάθε περίπτωση.

- Δείτε από το δίκτυο τι είναι το *Toast Message*.
- Ενσωματώστε ένα τέτοιο μήνυμα στη μέθοδο onCreate() το οποίο να απεικονίζει το μήνυμα “The activity started”.
- Κάντε screen capture και βάλτε την εικόνα στην εργασία σας.