



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

Προγραμματισμός Διεπαφής Χρήστη

Ενότητα 2: Βασικά στοιχεία του εργαλείου JavaFX Μέρος 1^ο

Κωνσταντίνος Τσίκνας
ktsik@teiemt.gr

Περιεχόμενα ενότητας

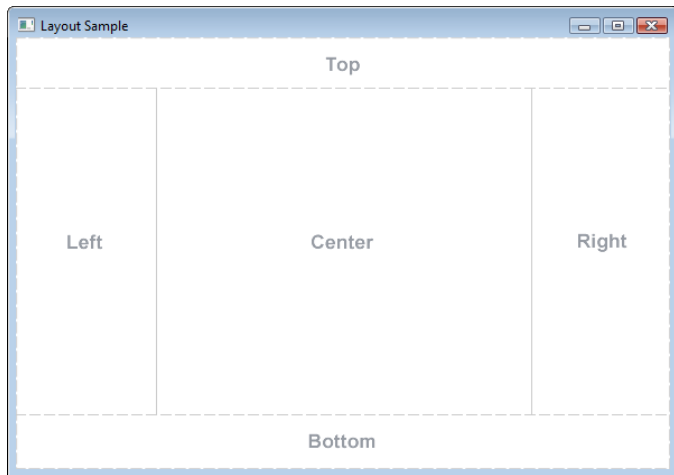
- Εισαγωγή στο JavaFX
- Στοιχεία του εργαλείου JavaFX
- Βασική δομή προγράμματος JavaFX
- Τοποθέτηση Εικόνας: Η κλάση ImageView
- Πλαίσια διάταξης: Η κλάση FlowPane
- Πλαίσια διάταξης: Η κλάση GridPane

Τί είναι το JavaFX

- Βιβλιοθήκη για την ανάπτυξη Γραφικών Εφαρμογών Διεπαφής Χρήστη (GUIs) σε Java. Περιέχει (μεταξύ άλλων):
 - **Πλαίσια διαταξης (*layout paines*)** (π.χ. *BorderPane*, *StackPane*, *GridPane*)
 - **UI controls** (π.χ. Κουμπιά, checkboxes, dropdowns)
 - **Δισδιάστατα και τρισδιάστατα σχήματα (*shapes*)**
 - **Διαγράμματα – γραφήματα (*charts*)**
 - **Image view** (ενσωμάτωση εικόνας)
 - **Web view** (ενσωμάτωση κώδικα html)
 - **CSSs** (Cascading Style Sheets)
- Γραμμένη σε Java – μπορεί να εκτελεστεί σε JVM (Java Virtual Machine), όπως οποιαδήποτε εφαρμογή σε Java
- Έχει ενσωματωθεί στο Netbeans (Από την έκδοση 8.0)

Πλαίσια διάταξης (Layout panes)

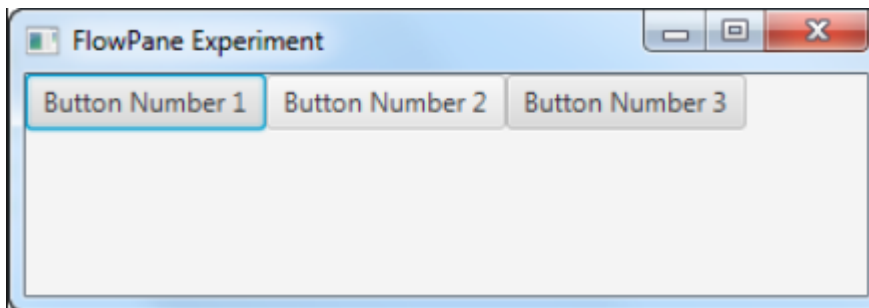
- BorderPane



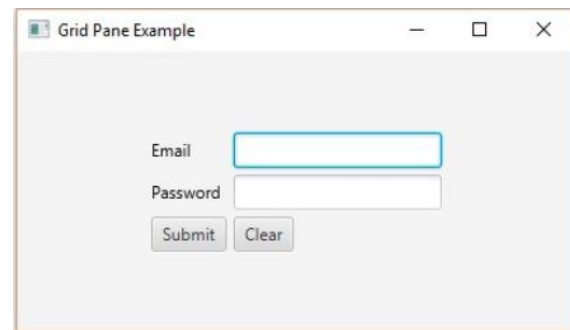
- StackPane



- FlowPane



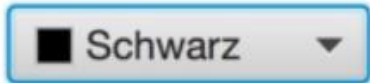
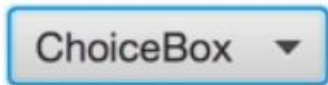
- GridPane



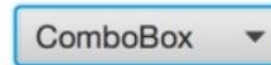
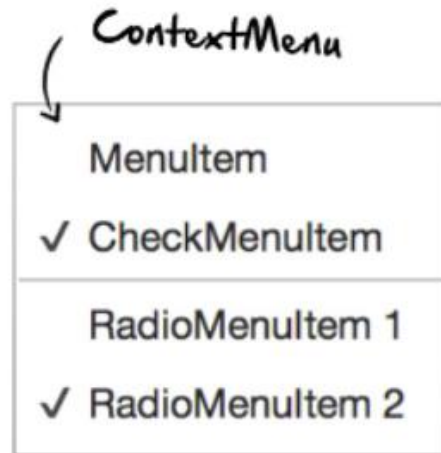
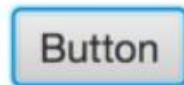
UI Controls



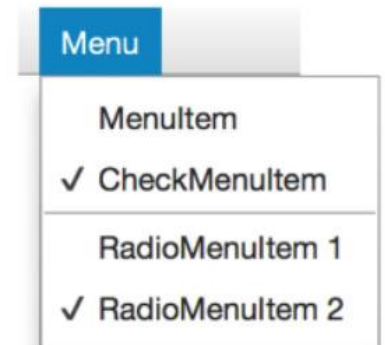
Accordion



ColorPicker



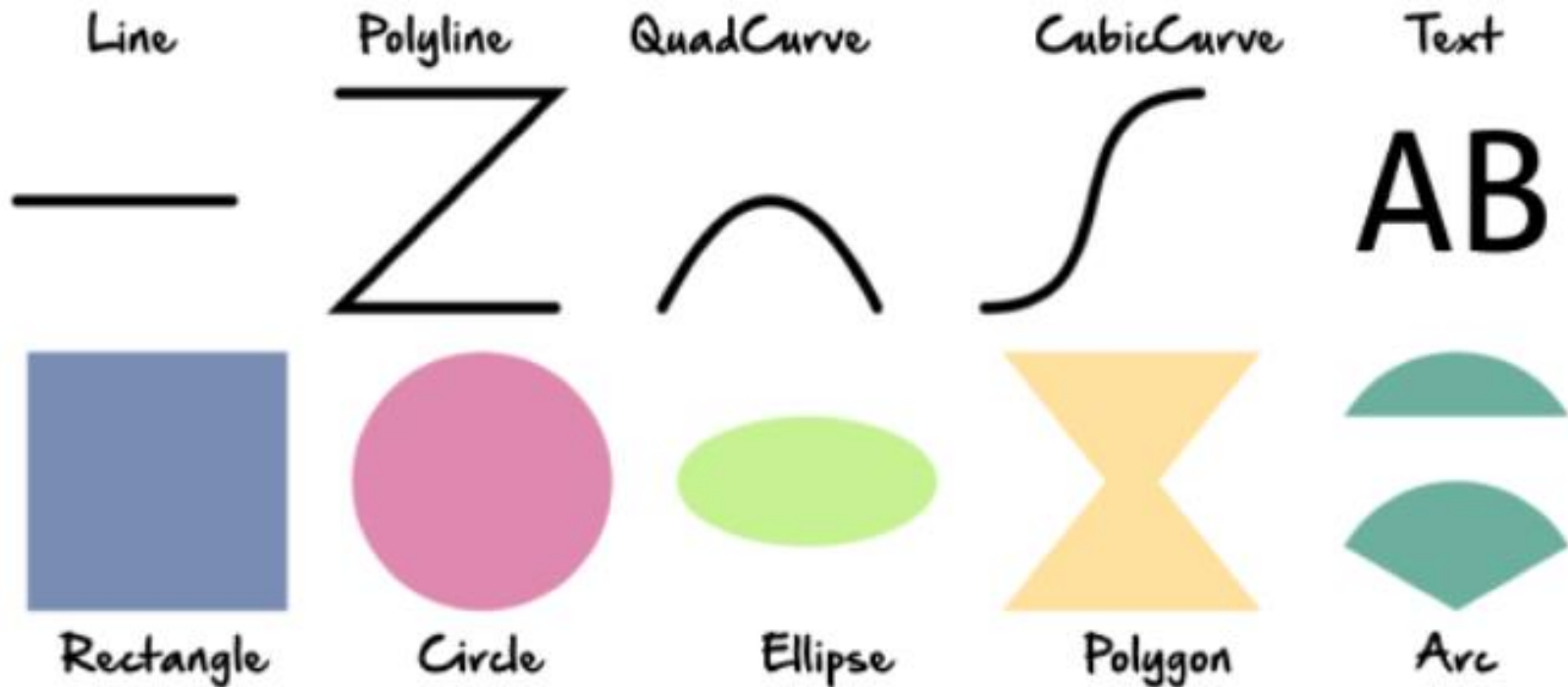
Label



1/10

Pagination

Σχήματα (Shapes)

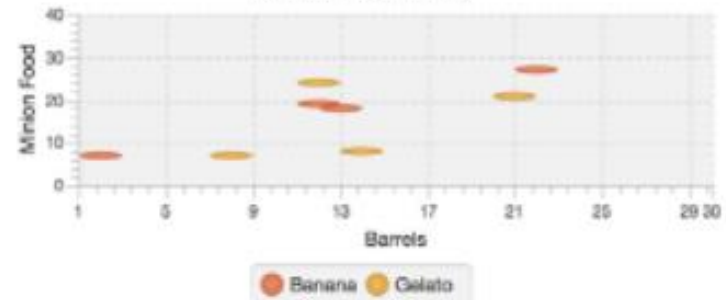


Γραφήματα (Charts)

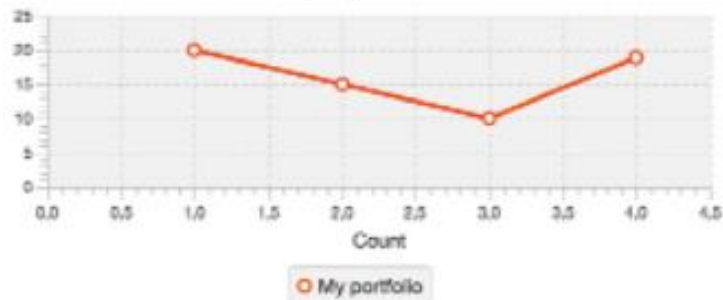
Ui Toolkits



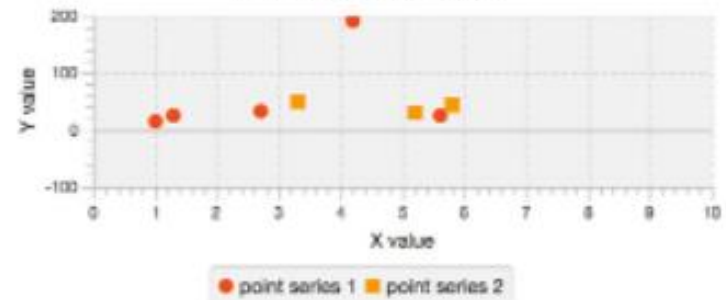
Minion Monitoring



Some Data



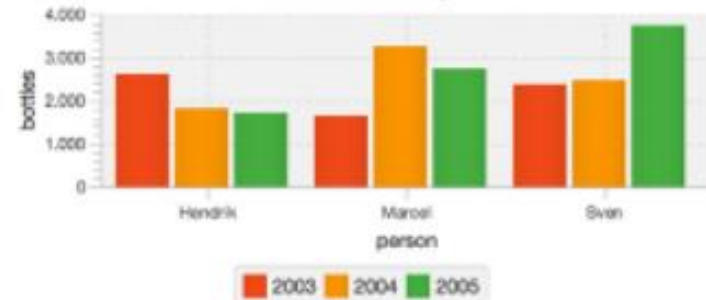
Some Random Points



Bug Monitoring



Beer Summary



Βασική δομή προγράμματος JavaFX (1/3)

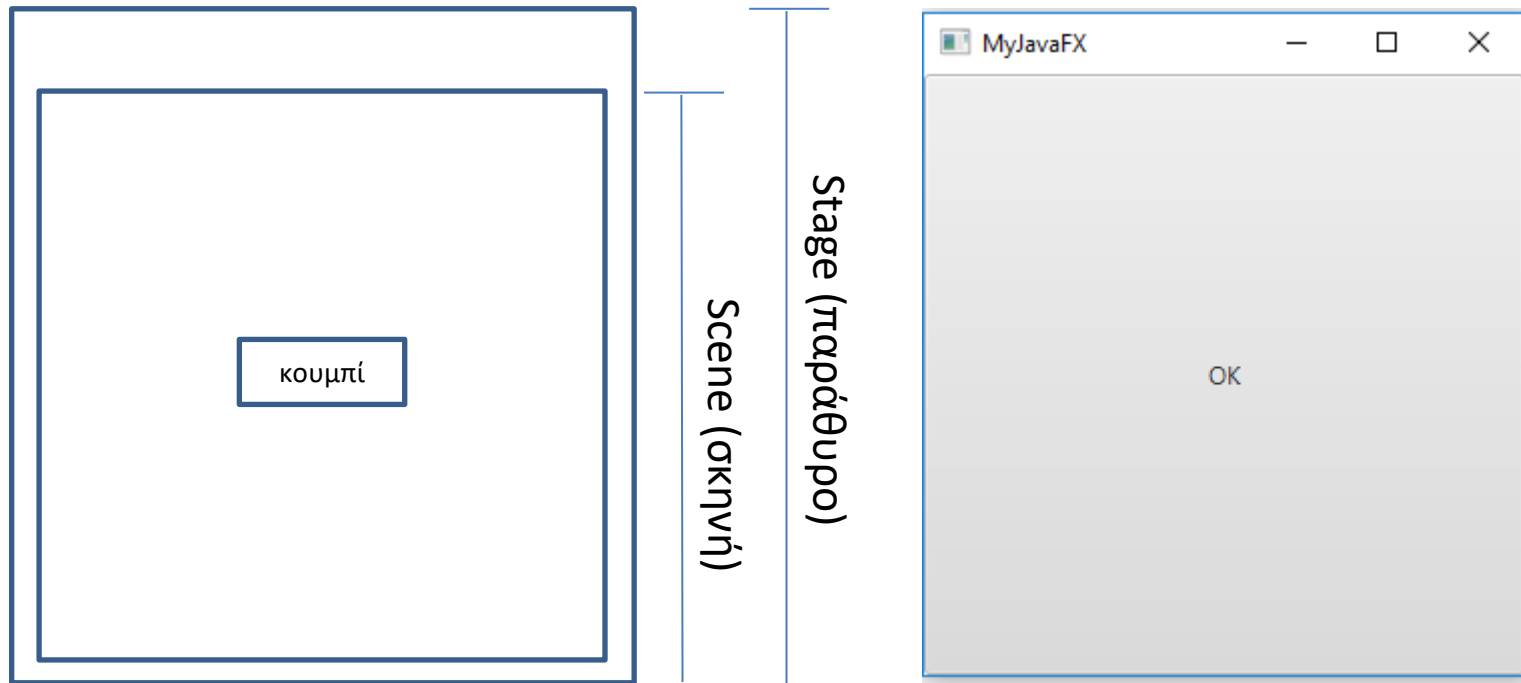
```
1  package myjavafx;
2
3  import javafx.application.Application;
4  import javafx.scene.Scene;
5  import javafx.scene.control.Button;
6  import javafx.stage.Stage;
7
8  public class MyJavaFX extends Application {
9      @Override //Παρακάμπτει τη μέθοδο start της κλάσης Application
10
11      /* Δημιουργεί ένα παράθυρο και τοποθετεί ένα κουμπί σε αυτό */
12      public void start(Stage myStage) {
13          Button btnOK = new Button("OK"); //δημιουργεί κουμπί
14
15          //δημιουργεί σκηνή και τοποθετεί το κουμπί εντός της
16          Scene myScene = new Scene(btnOK, 300, 300);
17          myStage.setTitle("MyJavaFX"); //τίτλος σκηνής
18          myStage.setScene(myScene); //τοποθέτηση σκηνής στο παράθυρο
19          myStage.show(); //εμφάνιση του παραθύρου
20      }
21      /* εκτέλεση της εφαρμογής από τη main */
22      public static void main(String[] args) {
23          Application.launch(args);
24      }
25  }
26
```


Βασική δομή προγράμματος JavaFX (2/3)

- **Γραμμή 12:** `public void start(Stage myStage)`
 - παρακάμπτει τη μέθοδο `start(Stage)` που ορίζεται στην κλάση `javafx.application.Application`
 - Δημιουργεί ένα παράθυρο (`stage`)
- **Γραμμή 13:** `Button btnOK = new Button("OK");`
 - Δημιουργία του αντικειμένου `Button`
- **Γραμμή 16:** `Scene myScene = new Scene(btnOK, 300, 300);`
 - Κλήση του δομητή `Scene(node, width, height)` για τη δημιουργία σκηνής διαστάσεων 300 x 300 pixels και την τοποθέτηση ενός κουμπιού (`node`) σε ατήν
- **Γραμμή 18:** `myStage.setScene(myScene);`
 - Τοποθέτηση του σκηνικού εντός του παραθύρου
- **Γραμμή 19:** `myStage.show();`
 - Εμφάνιση του παραθύρου

Βασική δομή προγράμματος JavaFX (3/3)

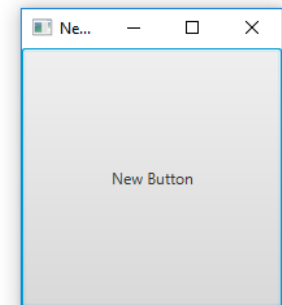
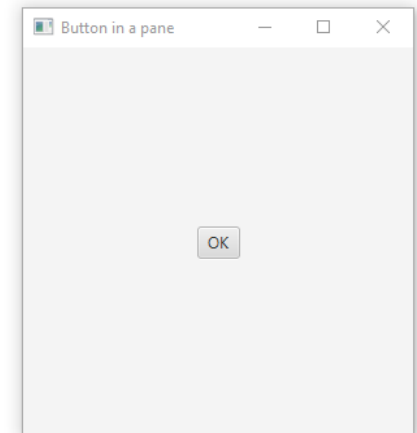
- Κάθε εφαρμογή JavaFX χρειάζεται τουλάχιστον ένα παράθυρο (stage)
- Εντός του παραθύρου υπάρχει η σκηνή που περιέχει έναν ή περισσότερους κόμβους (**nodes**): κουμπί, σχήμα, εικόνα, κ.ο.κ



Εφαρμογή με δύο παράθυρα

- Μπορούμε να δημιουργήσουμε περισσότερα από ένα παράθυρα (stages).
- Ο παρακάτω κώδικας μας δείχνει τον τρόπο

```
17 public class MultipleStageDemo extends Application{
18     @Override
19     public void start(Stage firstStage){
20
21         Button btn = new Button("OK");
22         Scene scene = new Scene(btn, 300, 300);
23
24         firstStage.setTitle("MyJavaFX");
25         firstStage.setScene(scene);
26         firstStage.show();
27
28         //Δημιουργία νέας σκηνής και τοποθέτηση του κουμπιού σε αυτήν
29         Scene newScene = new Scene(new Button("New Button"), 200, 200);
30         //Δημιουργία νέου παραθύρου και τοποθέτηση της σκηνής σε αυτό
31         Stage secondStage = new Stage();
32         secondStage.setTitle("New Window");
33         secondStage.setScene(newScene);
34         secondStage.show();
35     }
36 }
37
38 public static void main(String[] args) {
39
40     Application.launch(args);
41 }
```

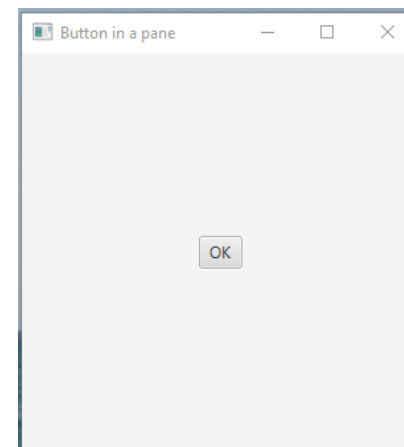


Τοποθέτηση σε πλαίσια (panes)

- Τα προηγούμενα προγράμματα εμφανίζουν τα κουμπιά πάντοτε στο κέντρο και καταλαμβάνουν ολόκληρη την περιοχή του παραθύρου.
- Μια καλύτερη προσέγγιση είναι να τοποθετήσουμε τους κόμβους μέσα σε πλαίσια (panes).

```
6 package buttoninpane;
7 import javafx.application.Application;
8 import javafx.stage.Stage;
9 import javafx.scene.Scene;
10 import javafx.scene.control.Button;
11 import javafx.scene.layout.Pane;
12
13 public class ButtonInPane extends Application{
14
15     @Override
16     public void start(Stage myStage){
17
18         Button btnOK = new Button("OK");
19
20         Pane myPane = new Pane();
21         myPane.getChildren().add(btnOK);
22
23         Scene myScene = new Scene(myPane, 300, 300);
24
25         myStage.setTitle("Button in a pane");
26         myStage.setScene(myScene);
27         myStage.show();
28     }
29
30     public static void main(String[] args) {
31         Application.launch(args);
32     }
33
34 }
```

- Οι κόμβοι τοποθετούνται εντός του πλαισίου με το επιθυμητό μέγεθος.
- Το πλαίσιο (pane) τοποθετείται εντός της σκηνής.
- Η `getChildren()` επιστρέφει ένα αντικείμενο τύπου `ObservableList`
- Η μέθοδος `add()` προσθέτει τον κόμβο στη λίστα



Πλαίσια διάταξης: Η κλάση FlowPane

```
7 import javafx.application.Application;
8 import javafx.geometry.Insets;
9 import javafx.geometry.Orientation;
10 import javafx.stage.Stage;
11 import javafx.scene.Scene;
12 import javafx.scene.control.Label;
13 import javafx.scene.control.TextField;
14 import javafx.scene.layout.FlowPane;
15
16 public class ShowFlowPane extends Application{
17     @Override
18     public void start(Stage myStage){
19         //Δημιουργία FlowPane
20         FlowPane myPane = new FlowPane();
21         myPane.setOrientation(Orientation.HORIZONTAL);
22         myPane.setPadding(new Insets(10,10,10,10));
23         myPane.setHgap(5);
24         myPane.setVgap(5);
25
26         //Δημιουργία control nodes και τοποθέτησή τους εντός του pane
27         Label lblName = new Label("First Name:");
28         TextField txtName = new TextField();
29         Label lblMiddleName = new Label("MI:");
30         TextField txtMiddleName = new TextField();
31         txtMiddleName.setPrefColumnCount(1);
32         Label lblLastName = new Label("Last Name:");
33         TextField txtSirName = new TextField();
34         myPane.getChildren().addAll(lblName, txtName, lblMiddleName,
35                                     txtMiddleName, lblLastName, txtSirName);
36
37         Scene myScene = new Scene(myPane, 300, 100);
38         myStage.setScene(myScene);
39         myStage.setTitle("ShowFlowPane");
40         myStage.show();
41     }
42
43     public static void main(String[] args) {
44         Application.launch(args);
45     }
46 }
```

Η FlowPane τοποθετεί τους κόμβους οριζόντια από αριστερά προς δεξιά ή κατακόρυφα, από πάνω προς τα κάτω ακολουθώντας τη σειρά που προστέθηκαν,

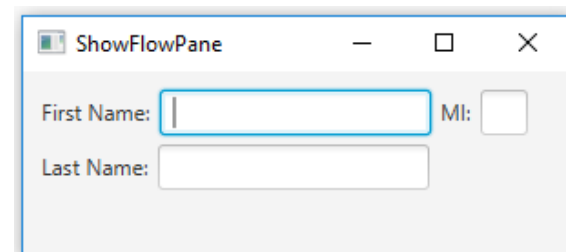
Όταν μια γραμμή ή μια στήλη γεμίσει, τότε ξεκινά μια νέα γραμμή ή στήλη.

Ο προσανατολισμός γίνεται με την `setOrientation()`

Ο δομητής `Insets` ορίζει το μέγεθος του περιγράμματος του πλαισίου (πάνω, δεξιά, κάτω αριστερά)

Οι `Hgap()` / `Vgap()` ορίζουν την απόσταση μεταξύ των κόμβων του πλαισίου

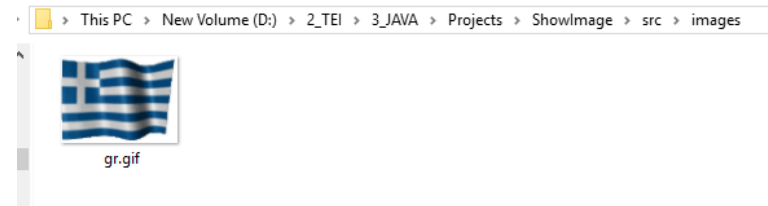
Οι κόμβοι μπορούν να προστεθούν μαζί στο `flowPane` με την `getChildren().addAll()`



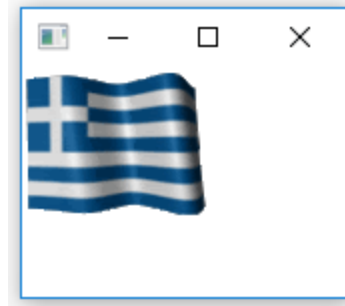
Η κλάση ImageView

```
8 import javafx.application.Application;
9 import javafx.scene.Scene;
10 import javafx.scene.image.ImageView;
11 import javafx.stage.Stage;
12 import javafx.scene.layout.Pane;
13
14 public class ShowImage extends Application {
15
16     @Override
17     public void start(Stage myStage) {
18         Pane pane = new Pane();
19
20         ImageView flag = new ImageView("images\\gr.gif");
21         pane.getChildren().add(flag);
22
23         Scene myScene = new Scene(pane);
24
25         myStage.setTitle("ShowImage");
26         myStage.setScene(myScene);
27         myStage.show();
28     }
29
30     public static void main(String[] args) {
31
32         Application.launch(args);
33     }
34
35 }
```

- Η ImageView είναι ένας κόμβος αναπαράστασης εικόνας
- Η εικόνα του παραδείγματος έχει όνομα gr.gif και βρίσκεται εντός του καταλόγου src\images



- Η εικόνα φορτώνεται σε ένα πλαίσιο με την `getChildren().add()`
- Στη συνέχεια το πλαίσιο τοποθετείται εντός της σκηνής και του παραθύρου



Πλαίσια σχεδίασης. Η κλάση GridPane

```
17 public class ShowLoginGridPane extends Application{
18
19     @Override
20     public void start(Stage myStage){
21         GridPane myGrid = new GridPane();
22         myGrid.setPadding(new Insets(10, 10, 10, 10));
23         myGrid.setVgap(10);
24         myGrid.setHgap(10);
25
26         Label lblName = new Label("Username:");
27         TextField txtName = new TextField();
28         Label lblPwd = new Label("Password:");
29         TextField txtPwd = new TextField();
30         txtPwd.setPromptText("password");
31         Button btnLogin = new Button("Log in");
32
33
34         /* χρήση στατικής μεθόδου setConstraints() για τοποθέτηση στο πλέγμα*/
35         GridPane.setConstraints(lblName, 0, 0);
36         GridPane.setConstraints(txtName, 1, 0);
37         GridPane.setConstraints(lblPwd, 0, 1);
38         GridPane.setConstraints(txtPwd, 1, 1);
39         GridPane.setConstraints(btnLogin, 1, 2);
40         GridPane.setHalignment(btnLogin, HPos.RIGHT);
41         myGrid.getChildren().addAll(lblName, txtName, lblPwd, txtPwd, btnLogin);
42
43         Scene myScene = new Scene(myGrid, 250, 115);
44
45         myStage.setTitle("Login screen");
46         myStage.setScene(myScene);
47         myStage.show();
48     }
49
50     public static void main(String[] args) {
51         Application.launch(args);
52     }
53
54 }
```

- Η GridPane τοποθετεί τους κόμβους σε πλέγμα.
- Η τοποθέτηση γίνεται με τη στατική μέθοδο **setConstraints**(node, colIndex, rowIndex)
 - *colIndex*: ο αριθμός στήλης
 - *rowIndex*: ο αριθμός γραμμή
- Η `getChildren().addAll()` τοποθετεί τους κόμβους στον πλέγμα
- Εναλλακτικά η τοποθέτηση σε κελιά και στο πλέγμα μπορεί να γίνει με κλήση της μεθόδου **add**(node, colIndex, rowIndex)
- Η στατική μέθοδος **setHalignment**(node, HPos) τοποθετεί τον κόμβο στη θέση που προσδιορίζει το δεύτερο όρισμα (HPos).

