



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

Προγραμματισμός Διεπαφής Χρήστη

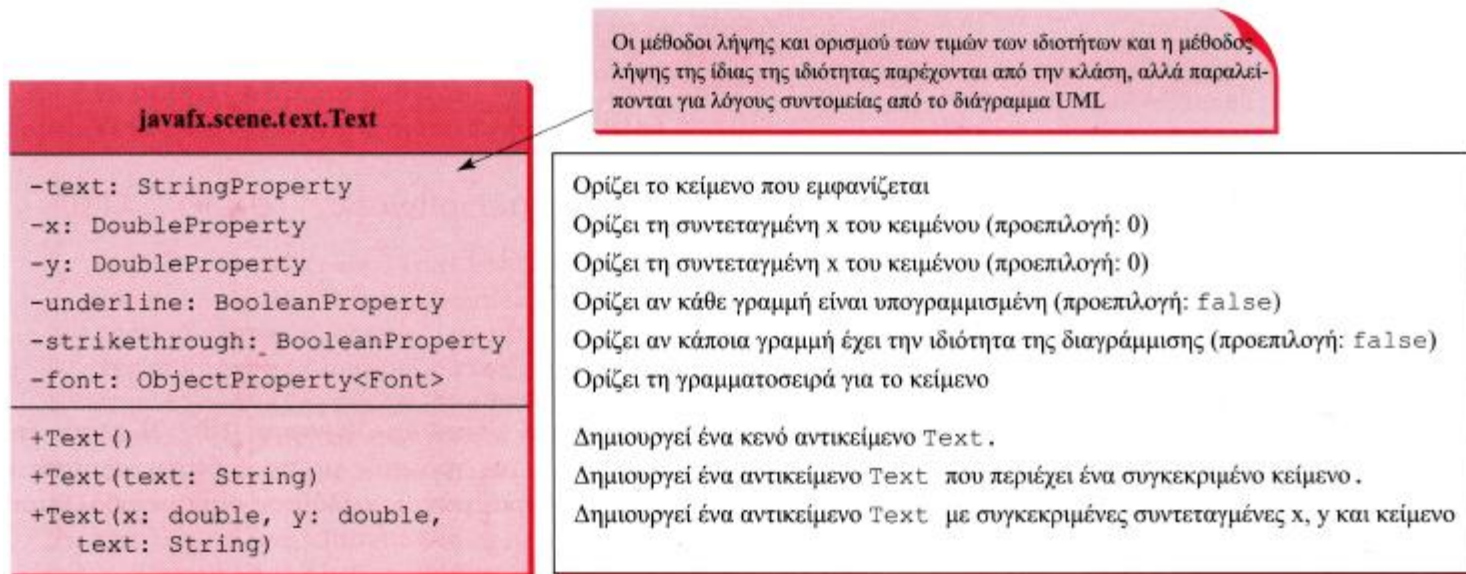
Ενότητα 3: Βασικά στοιχεία του εργαλείου JavaFX Μέρος 2^ο

Κωνσταντίνος Τσίκνας
ktsik@teiemt.gr

Περιεχόμενα ενότητας

- Στύλ συμβολοσειράς. Η κλάση Font.
- Δημιουργία χρωμάτων. Η κλάση Color.
- Εισαγωγή κειμένου. Η κλάση text.
- Σχεδίαση κύκλου. Η κλάση Circle.
- Σχεδίαση γραμμής. Η κλάση Line.
- Σχεδίαση ορθογωνίου. Η κλάση Rectangle.
- Σχεδίαση πολυγώνου – τεθλασμένης γραμμής.
Οι κλάσεις Polygon και Polyline.

Εισαγωγή κειμένου. Η κλάση text

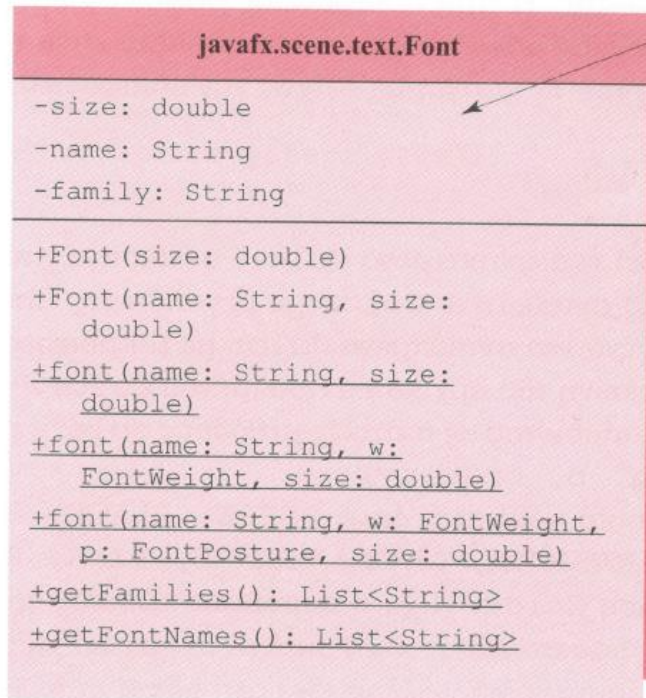


Παράδειγμα:

```
Text text1 = new Text(20, 20, "Programming in JavaFx is fun!");
text1.setFill(Color.RED);
text1.setUnderline(true);
text1.setStrikethrough(true);
```

Στυλ συμβολοσειράς. Η κλάση Font

- Περιγράφει ένα όνομα συμβολοσειράς, ένα στυλ και ένα μέγεθος
- Ένα αντικείμενο της κλάσης Font κατασκευάζεται χρησιμοποιώντας του δομητές του.
`Font myFont = Font.font("Times New Roman", FontWeight.BOLD, FontPosture.ITALIC, 20);`



Οι μέθοδοι λήψης των τιμών των ιδιοτήτων παρέχονται από την κλάση, αλλά παραλείπονται για λόγους συντομίας από το διάγραμμα UML.

Το μέγεθος αυτής της γραμματοσειράς

Το όνομα αυτής της γραμματοσειράς

Η οικογένεια αυτής της γραμματοσειράς

Δημιουργεί ένα αντικείμενο `Font` με το καθορισμένο μέγεθος.

Δημιουργεί ένα αντικείμενο `Font` με το καθορισμένο πλήρες όνομα και μέγεθος.

Δημιουργεί ένα αντικείμενο `Font` με το καθορισμένο όνομα και μέγεθος.

Δημιουργεί ένα αντικείμενο `Font` με το καθορισμένο όνομα, στυλ και μέγεθος.

Δημιουργεί ένα αντικείμενο `Font` με το καθορισμένο όνομα, στυλ, τοποθέτηση και μέγεθος.

Επιστρέφει μία λίστα ονομάτων οικογενειών γραμματοσειρών.

Επιστρέφει μία λίστα πλήρων ονομάτων γραμματοσειρών, όπου συμπεριλαμβάνεται το στυλ και οι οικογένεια.

Δημιουργία χρωμάτων. Η κλάση Color

- Χρησιμοποιείται για τη δημιουργία χρωμάτων
- Ένα αντικείμενο της κλάσης Color κατασκευάζεται χρησιμοποιώντας του δομητές του. π.χ.

```
Color myColor = new Color(0.5, 0.5, 0.5, 0.1)
```

javafx.scene.paint.Color

-red: double
-green: double
-blue: double
-opacity: double

+Color(r: double, g: double, b: double, opacity: double)
+brighter(): Color
+darker(): Color
+color(r: double, g: double, b: double): Color
+color(r: double, g: double, b: double, opacity: double): Color
+rgb(r: int, g: int, b: int): Color
+rgb(r: int, g: int, b: int, opacity: double): Color

Οι μέθοδοι λήψης των τιμών των ιδιοτήτων παρέχονται από την κλάση, αλλά παραλείπονται για λόγους συντομίας από το διάγραμμα UML.

Η τιμή red αυτού του χρώματος (από 0.0 ως 1.0).

Η τιμή green αυτού του χρώματος (από 0.0 ως 1.0).

Η τιμή blue αυτού του χρώματος (από 0.0 ως 1.0).

Η αδιαφάνεια αυτού του χρώματος (από 0.0 ως 1.0).

Δημιουργεί ένα αντικείμενο color με καθορισμένες τιμές των ιδιοτήτων red, green, blue, και opacity.

Δημιουργεί ένα αντικείμενο color η οποία είναι μία φωτεινότερη έκδοση αυτού του color.

Δημιουργεί ένα αντικείμενο color η οποία είναι μία σκοτεινότερη έκδοση αυτού του color.

Δημιουργεί ένα αδιαφανές αντικείμενο color με καθορισμένες τιμές των ιδιοτήτων red, green, blue, και opacity.

Δημιουργεί ένα αντικείμενο color με καθορισμένες τιμές των ιδιοτήτων red, green, blue, και opacity.

Δημιουργεί ένα αντικείμενο color με καθορισμένες τιμές των ιδιοτήτων red, green, και blue, στο διάστημα από 0 μέχρι 255.

Δημιουργεί ένα αντικείμενο color με καθορισμένες τιμές των ιδιοτήτων red, green, και blue, στο διάστημα από 0 μέχρι 255 και δοθείσα τιμή της ιδιότητας opacity.

Παράδειγμα χρήσης text, font και color

```
public class ShowText extends Application {

    @Override
    public void start(Stage stage){
        Text text1 = new Text(20, 20, "Programming in JavaFx \nis fun!");
        text1.setFont(Font.font("Courier", FontWeight.BOLD, FontPosture.ITALIC, 15));
        text1.setFill(Color.RED);
        text1.setUnderline(true);
        text1.setStrikethrough(true);

        Pane pane = new Pane();
        pane.getChildren().add(text1);

        Scene scene = new Scene(pane, 250, 50);

        stage.setTitle("Show Text");
        stage.setScene(scene);
        stage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

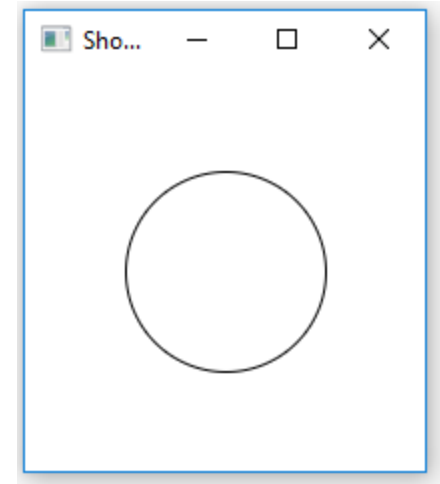
        launch(args);
    }
}
```



Σχεδίαση κύκλου. Η κλάση Circle

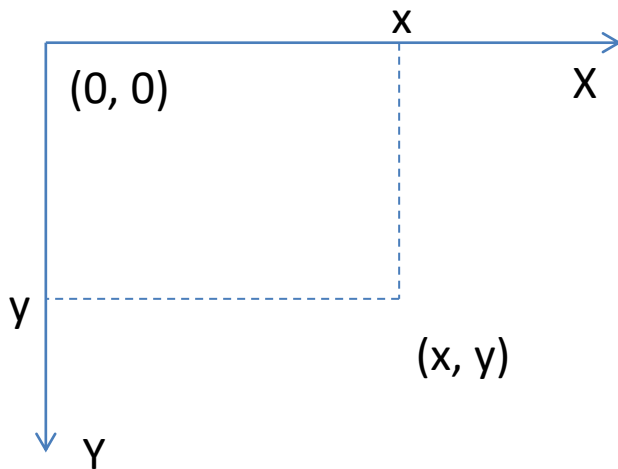
- Ο παρακάτω κώδικας εμφανίζει έναν κύκλο στο κέντρο του πλαισίου

```
8 import javafx.application.Application;
9 import javafx.stage.Stage;
10 import javafx.scene.Scene;
11 import javafx.scene.layout.StackPane;
12 import javafx.scene.paint.Color;
13 import javafx.scene.shape.Circle;
14
15 public class ShowCircle extends Application {
16
17     @Override
18     public void start(Stage myStage){
19         //Δημιουργεί έναν κύκλο και ορίζει τις ιδιότητές του
20         Circle circle = new Circle();
21         circle.setCenterX(100);
22         circle.setCenterY(100);
23         circle.setRadius(50);
24         circle.setStroke(Color.BLACK);
25         circle.setFill(Color.WHITE);
26
27         //Δημιουργεί ένα πλαίσιο στο οποίο τοποθετείται ο κύκλος
28         StackPane myPane = new StackPane();
29         myPane.getChildren().add(circle);
30
31         Scene myScene = new Scene(myPane, 200, 200);
32
33         myStage.setTitle("ShowCircle");
34         myStage.setScene(myScene);
35         myStage.show();
36     }
37 }
```

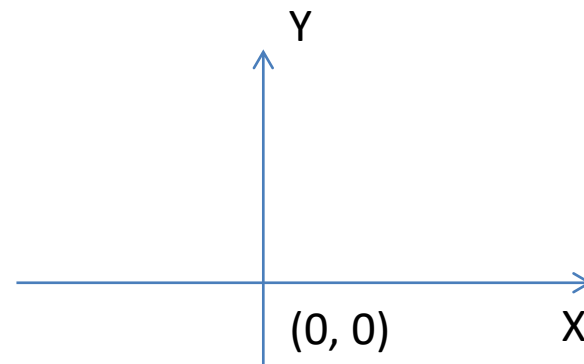


Επεξήγηση κώδικα. Το σύστημα συντεταγμένων της Java

- Το πρόγραμμα δημιουργεί έναν αντικείμενο `Circle` και ορίζει το κέντρο του στο σημείο $(100, 100)$ (μέθοδοι **`setCenterX()`**, **`setCenterY()`** γραμμές 24 -26)
- Η τιμή της ακτίνας του κύκλου ορίζεται ίση με 50 (μέθοδος **`setRadius()`**, γραμμή 23)
- Η γραμμή του κύκλου παίρνει χρώμα μαύρο (μέθοδος **`setStroke()`**, γραμμή 24)
- Ο κύκλος γεμίζει με χρώμα ή να παραμείνει λευκός (μέθοδος **`setFill()`** γραμμή 25)
- Σημειώνεται ότι το σύστημα συντεταγμένων της Java είναι διαφορετικό από το καρτεσιανό σύστημα συντεταγμένων.
- Στο σύστημα συντεταγμένων της Java ο άξονας του x αυξάνεται από τα αριστερά προς τα δεξιά, ενώ ο άξονας των y αυξάνεται προς τα κάτω.



Σύστημα συντεταγμένων Java



Καρτεσιανό σύστημα συντεταγμένων

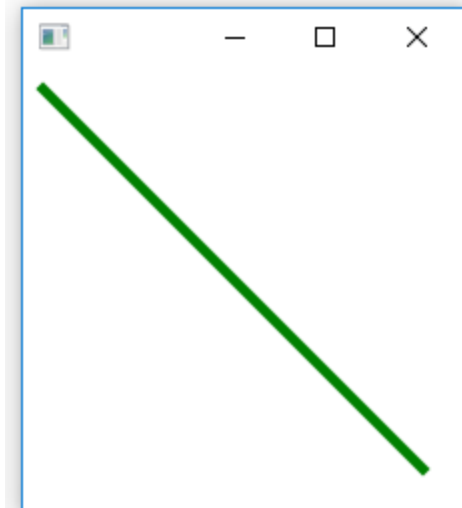
Σχεδίαση γραμμής. Η κλάση Line

Line(double startX, double startY, double endX, double endY)

Σχεδιασμός γραμμής από το σημείο (startX, startY) μέχρι το σημείο (endX, endY)

Παράδειγμα:

```
19 public class ShowLine extends Application {  
20  
21     @Override  
22     public void start(Stage myStage){  
23         Line line = new Line(10, 10, 200, 200);  
24         Pane pane = new Pane();  
25  
26         line.setStrokeWidth(5);  
27         line.setStroke(Color.GREEN);  
28         pane.getChildren().add(line);  
29  
30         Scene scene = new Scene(pane, 220, 220);  
31         myStage.setScene(scene);  
32         myStage.show();  
33     }
```

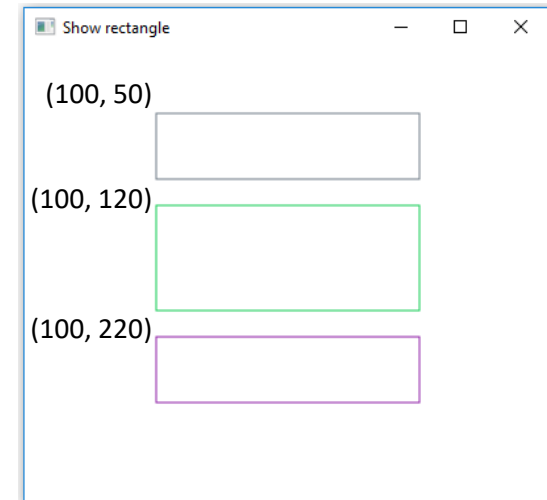


Σχεδίαση ορθογωνίου: Η κλάση Rectangle (1)

Rectangle(double x, double y, double width, double height)
Δημιουργία ορθογωνίου με διαστάσεις width x height και τοποθέτησή του στο σημείο x, y

Παράδειγμα:

```
public void start(Stage stage){  
    21  
    22    Pane pane = new Pane();  
    23    /*δημιουργία ορθογωνίου με διαστάσεις 200 x 50 (πλάτος x ύψος)  
    24    και τοποθέτησή του στο σημείο (100, 50) */  
    25    Rectangle r1 = new Rectangle(100, 50, 200, 50);  
    26    r1.setStroke(Color.color(Math.random(), Math.random(), Math.random()));  
    27    r1.setFill(Color.WHITE);  
    28    pane.getChildren().add(r1);  
    29  
    30    Rectangle r2 = new Rectangle(100, 120, 200, 80);  
    31    r2.setStroke(Color.color(Math.random(), Math.random(), Math.random()));  
    32    r2.setFill(Color.WHITE);  
    33    pane.getChildren().add(r2);  
    34  
    35    Rectangle r3 = new Rectangle(100, 220, 200, 50);  
    36    r3.setStroke(Color.color(Math.random(), Math.random(), Math.random()));  
    37    r3.setFill(Color.WHITE);  
    38    pane.getChildren().add(r3);  
    39  
    40  
    41    Scene scene = new Scene(pane, 400, 350);  
    42    stage.setTitle("Show rectangle");  
    43    stage.setScene(scene);  
    44    stage.show();  
    45 }
```



Σχεδίαση ορθογωνίου: Η κλάση Rectangle (2)

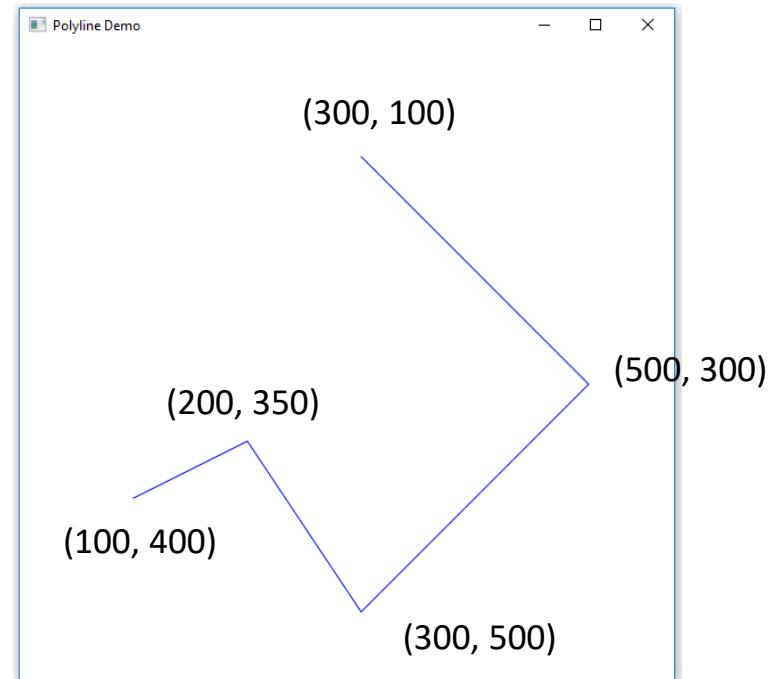
- Βελτιωμένη υλοποίηση: Δημιουργία νέας κλάσης που κληρονομεί την Rectangle

```
17 public class ShowRectangles extends Application{
18
19     @Override
20     public void start(Stage stage){
21
22         Pane pane = new Pane();
23         /*δημιουργία ορθογωνίου με διαστάσεις 200 x 50 (πλάτος x ύψος)
24         και τοποθέτησή του στο σημείο (100, 50) */
25
26         DrawRectangle r1 = new DrawRectangle(100, 50, 200, 50);
27         DrawRectangle r2 = new DrawRectangle(100, 120, 200, 80);
28         DrawRectangle r3 = new DrawRectangle(100, 220, 200, 50);
29         pane.getChildren().addAll(r1, r2, r3);
30
31         pane.getChildren().add(r3);
32         Scene scene = new Scene(pane, 400, 350);
33         stage.setTitle("Show rectangle");
34         stage.setScene(scene);
35         stage.show();
36     }
37
38     public class DrawRectangle extends Rectangle{
39
40         public DrawRectangle(double x, double y, double width, double height) {
41             super(x,y,width,height);
42             setStroke(Color.color(Math.random(), Math.random(), Math.random()));
43             setFill(Color.WHITE);
44         }
45     }
```

Σχεδίαση τεθλασμένης: Η κλάση Polyline

```
20 public class ShowPolyline extends Application{
21
22     @Override
23     public void start(Stage myStage){
24
25         Polyline polyline = new Polyline();
26         polyline.setStroke(Color.BLUE);
27         ObservableList<Double> list = polyline.getPoints();
28
29         //προσθήκη του σημείου (x, y) = (300.0, 100.0)
30         list.add(300.0);
31         list.add(100.0);
32
33         list.add(500.0);
34         list.add(300.0);
35
36         list.add(300.0);
37         list.add(500.0);
38
39         list.add(200.0);
40         list.add(350.0);
41
42         list.add(100.0);
43         list.add(400.0);
44
45         Pane myPane = new Pane();
46         myPane.getChildren().add(polyline);
47
48         Scene myScene = new Scene(myPane, 600, 600);
49         myStage.setTitle("Polyline Demo");
50         myStage.setScene(myScene);
51         myStage.show();
```

- Συνδέει μια ακολουθία σημείων
- Τα σημεία προστίθενται στην `ObservableList<Double>` με τη μέθοδο `add(value)`
- Προσοχή τα σημεία να δηλωθούν σαν `double`

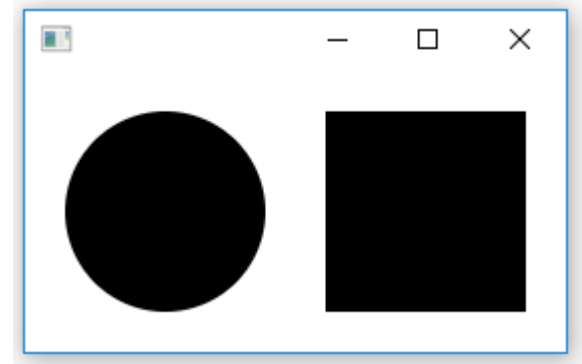


Οριζόντια - κάθετη διάταξη πλαισίων: Οι κλάσεις HBox/VBox

- HBox: Τοποθετεί τα παιδιά της σε οριζόντια γραμμή.
- VBox: Τοποθετεί τα παιδιά της σε κάθετη γραμμή.
- Παρόμοια λειτουργία με τη FlowPane, με τη διαφορά ότι η τελευταία μπορεί να τοποθετεί τα παιδιά της σε πολλαπλές γραμμές ή στήλες, ενώ οι Hbox/Vbox σε μια μόνο γραμμή ή στήλη αντίστοιχα.

Παράδειγμα:

```
20 public class HBoxVBoxDemo extends Application{
21
22     @Override
23     public void start(Stage myStage){
24         HBox hBox = new HBox();
25         hBox.setPadding(new Insets(20, 20, 20, 20));
26         hBox.setSpacing(30);
27         Circle c1 = new Circle(50);
28         Rectangle r1 = new Rectangle(100, 100);
29         hBox.getChildren().addAll(c1,r1);
30
31         Scene scene = new Scene(hBox);
32         myStage.setScene(scene);
33         myStage.show();
34     }
```



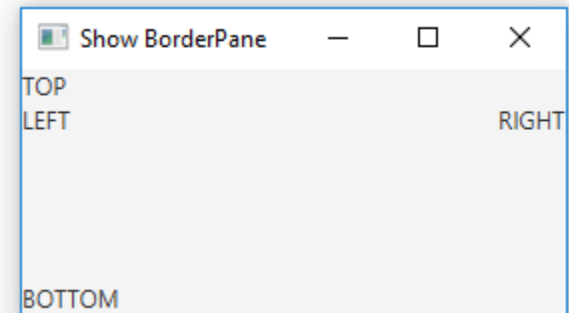
Η κλάση `BorderPane()` (1)

- Τοποθετεί ένα κόμβο στις παρακάτω περιοχές: άνω, κάτω, δεξιά, αριστερή και κεντρική περιοχή.
- Χρησιμοποιεί τις μεθόδους `setTop(node)`, `setBottom(node)`, `setLeft(node)`, `setRight(node)` και `setCenter(node)` αντίστοιχα.

Παράδειγμα

```
14 public class ShowBorderPane extends Application {  
15  
16     @Override  
17     public void start(Stage primaryStage) {  
18         BorderPane borderPane = new BorderPane();  
19  
20         borderPane.setTop(new Label("TOP"));  
21         borderPane.setBottom(new Label("BOTTOM"));  
22         borderPane.setLeft(new Label("LEFT"));  
23         borderPane.setRight(new Label("RIGHT"));  
24  
25         Scene scene = new Scene(borderPane);  
26  
27         primaryStage.setTitle("Show BorderPane");  
28         primaryStage.setScene(scene);  
29         primaryStage.show();  
30     }
```

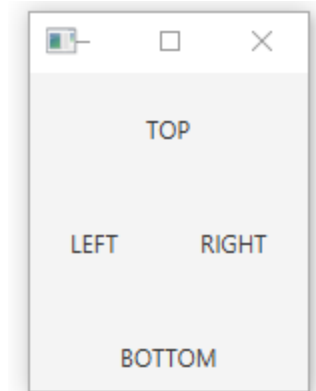
Η τοποθέτηση των κόμβων
κατευθείαν εντός της `BorderPane`
δεν έχει καλό αισθητικό
αποτέλεσμα!



Η κλάση BorderPane() (2)

Η λύση είναι να τοποθετηθούν οι κόμβοι πρώτα σε ένα άλλο πλαίσιο διάταξης και κατόπιν το τελευταίο να μπει μέσα στο Border pane. Μελετήστε το παρακάτω παράδειγμα:

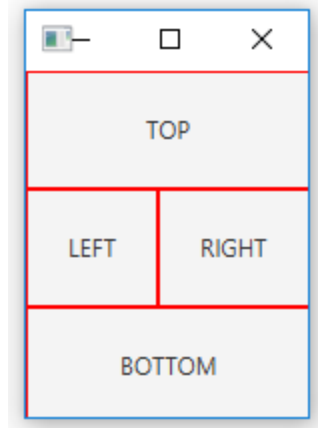
```
16 public class ShowBorderPane extends Application {
17
18     @Override
19     public void start(Stage primaryStage) {
20         BorderPane borderPane = new BorderPane();
21
22         StackPane stackTop = new StackPane(new Label("TOP"));
23         stackTop.setPadding(new Insets(20, 20, 20, 20));
24
25         StackPane stackBottom = new StackPane(new Label("BOTTOM"));
26         stackBottom.setPadding(new Insets(20, 20, 20, 20));
27
28         StackPane stackLeft = new StackPane(new Label("LEFT"));
29         stackLeft.setPadding(new Insets(20, 20, 20, 20));
30
31         StackPane stackRight = new StackPane(new Label("RIGHT"));
32         stackRight.setPadding(new Insets(20, 20, 20, 20));
33
34         borderPane.setTop(stackTop);
35         borderPane.setBottom(stackBottom);
36         borderPane.setLeft(stackLeft);
37         borderPane.setRight(stackRight);
38
39         Scene scene = new Scene(borderPane);
40
41         primaryStage.setTitle("Show BorderPane");
42         primaryStage.setScene(scene);
43         primaryStage.show();
44     }
```



Η κλάση BorderPane() (3)

- Ο κώδικας του προηγούμενου παραδείγματος μπορεί να βελτιωθεί αισθητά χρησιμοποιώντας μια νέα κλάση (CustomisedStackPane) που θα ενσωματώνει τις εντολές για τη δημιουργία του Label εντός του StackPane
- Μελετήστε το παρακάτω παράδειγμα:

```
16 public class ShowBorderPane extends Application {
17
18     @Override
19     public void start(Stage primaryStage) {
20         BorderPane borderPane = new BorderPane();
21
22         borderPane.setTop(new CustomisedStackPane(new Label("TOP")));
23         borderPane.setBottom(new CustomisedStackPane(new Label("BOTTOM")));
24         borderPane.setLeft(new CustomisedStackPane(new Label("LEFT")));
25         borderPane.setRight(new CustomisedStackPane(new Label("RIGHT")));
26
27         Scene scene = new Scene(borderPane);
28
29         primaryStage.setTitle("Show BorderPane");
30         primaryStage.setScene(scene);
31         primaryStage.show();
32     }
33
34     public class CustomisedStackPane extends StackPane {
35
36         public CustomisedStackPane(Label label) {
37             super(label);
38             setPadding(new Insets(20, 20, 20, 20));
39             //δημιουργία περιγράμματος στο stack με κόκκινο χρώμα
40             setStyle("--fx-border-color: red");
41         }
42     }
```



Προστέθηκε εντολή εντός της CustomisedStackPane για τη δημιουργία περιγράμματος με συγκεκριμένο χρώμα