



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

## Προγραμματισμός Διεπαφής Χρήστη

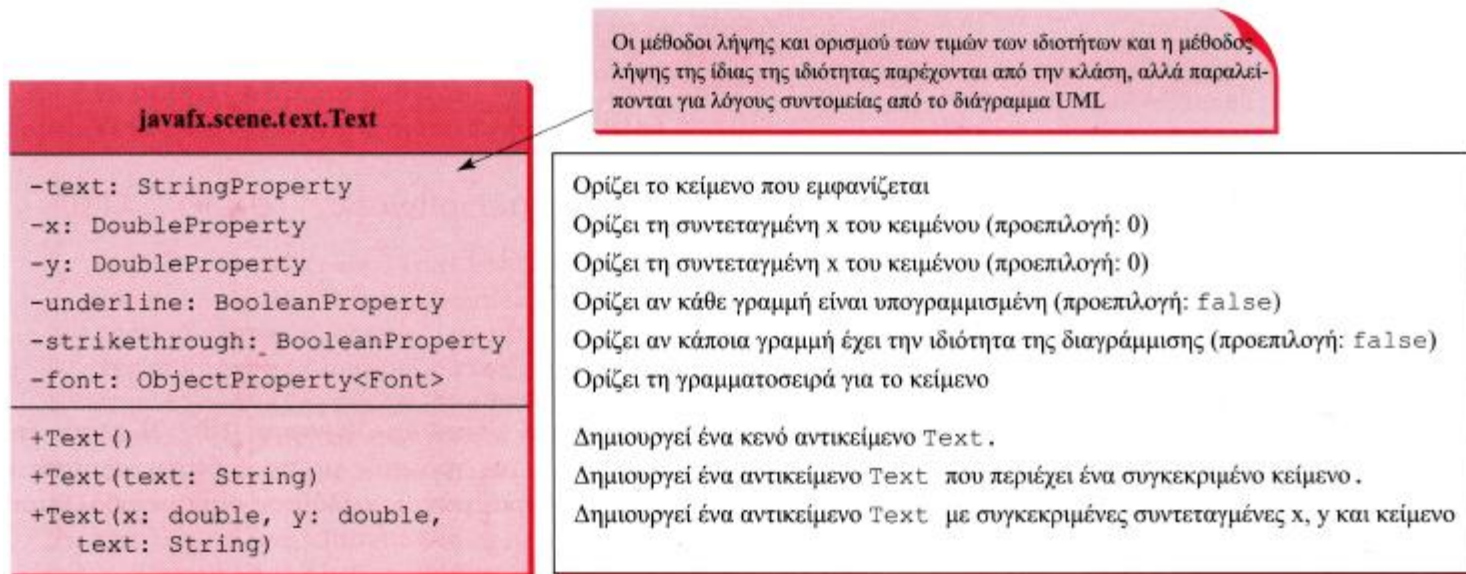
### Ενότητα 3: Βασικά στοιχεία του εργαλείου JavaFX Μέρος 2<sup>ο</sup>

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, Polyline.

# Εισαγωγή κειμένου. Η κλάση text

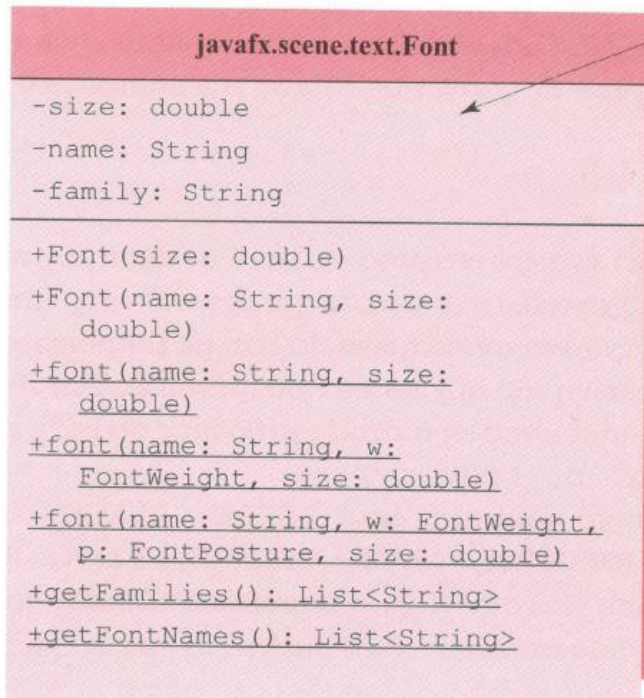


## Παράδειγμα:

```
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) {

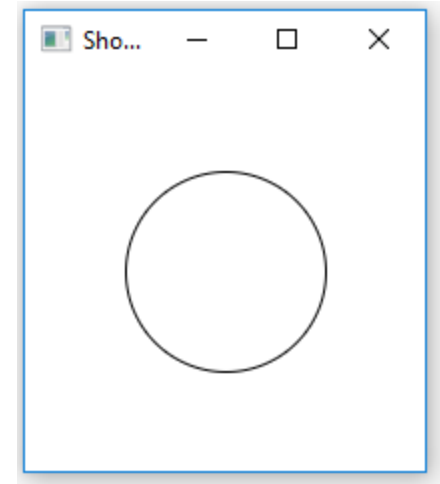
        launch(args);
    }
}
```



# Σχεδίαση κύκλου. Η κλάση Circle

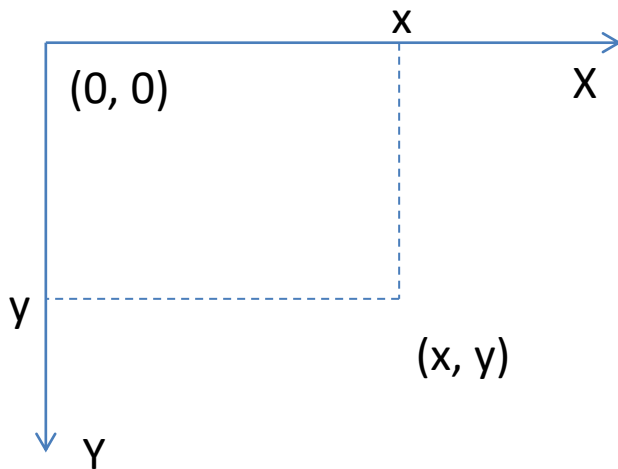
- Ο παρακάτω κώδικας εμφανίζει έναν κύκλο στο κέντρο του πλαισίου

```
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 }
```

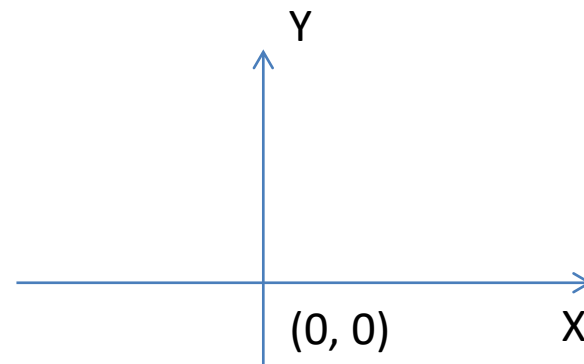


# Επεξήγηση κώδικα. Το σύστημα συντεταγμένων της Java

- 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.



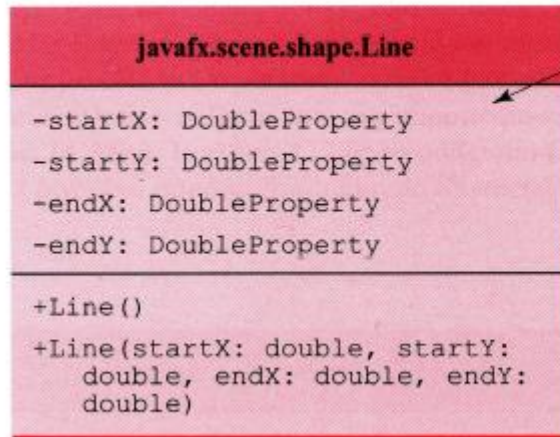
*Σύστημα συντεταγμένων Java*



*Καρτεσιανό σύστημα συντεταγμένων*



# Σχεδίαση γραμμής. Η κλάση Line



Οι μέθοδοι λήψης και ορισμού των τιμών των ιδιοτήτων και η μέθοδος λήψης της ίδιας της ιδιότητας παρέχονται από την κλάση, αλλά παραλείπονται για λόγους συντομίας από το διάγραμμα UML.

Η συντεταγμένη x του σημείου εκκίνησης.

Η συντεταγμένη y του σημείου εκκίνησης.

Η συντεταγμένη x του σημείου τερματισμού.

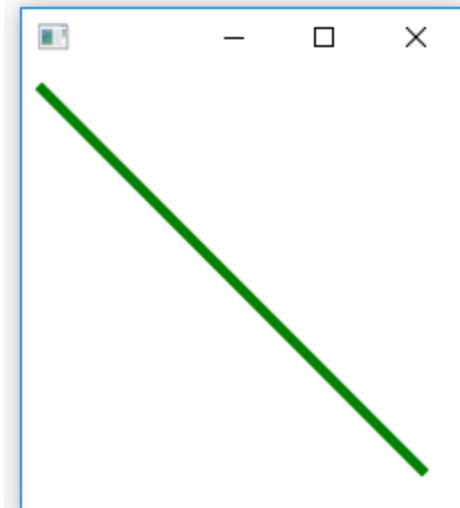
Η συντεταγμένη y του σημείου εκκίνησης.

Δημιουργεί ένα κενό αντικείμενο Line.

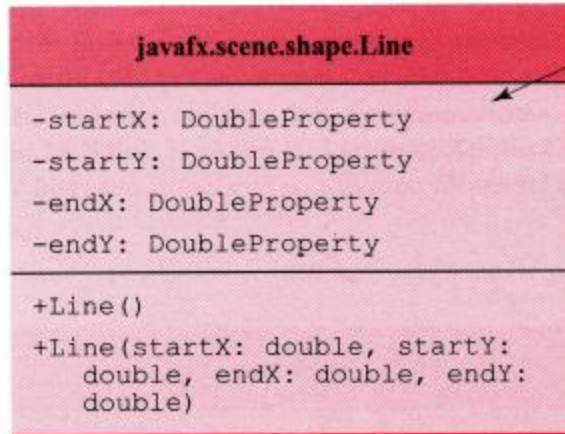
Δημιουργεί ένα αντικείμενο Line με συγκεκριμένα σημεία εκκίνησης και τερματισμού.

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



Οι μέθοδοι λήψης και ορισμού των τιμών των ιδιοτήτων και η μέθοδος λήψης της ίδιας της ιδιότητας παρέχονται από την κλάση, αλλά παραλείπονται για λόγους συντομίας από το διάγραμμα UML.

Η συντεταγμένη x του σημείου εκκίνησης.  
Η συντεταγμένη y του σημείου εκκίνησης.  
Η συντεταγμένη x του σημείου τερματισμού.  
Η συντεταγμένη y του σημείου εκκίνησης.

Δημιουργεί ένα κενό αντικείμενο Line.

Δημιουργεί ένα αντικείμενο Line με συγκεκριμένα σημεία εκκίνησης και τερματισμού.

## 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     }
```

