

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Χρόνης Βελέντζας

Γραφικά

Στόχος αυτής της ενότητας είναι η εισαγωγή στην κατασκευή εφαρμογών. Παρουσιάζεται η δομή και η μεθοδολογία ανάπτυξης εφαρμογών, ο τρόπος με τον οποίο μπορούμε να εισάγουμε παραμέτρους σε αυτές, αλλά και ο τρόπος με τον μπορούμε να σχεδιάζουμε σε αυτές.

10.1 Χαρακτηριστικά εφαρμογών

Μια εφαρμογή μπορεί να εκτελεστεί σε οποιαδήποτε λειτουργικό σύστημα, με την προϋπόθεση ότι έχουμε εγκαταστήσει το JRE της Java. Συνήθως έχει τα εξής χαρακτηριστικά:

1. Κατασκευάζεται ως υποκλάση της κλάσης **JFrame**.
2. Στην μέθοδο **main** αρχικοποιούμε τις απαραίτητες μεταβλητές και την υποκλάση της **JFrame**.
3. Επιτρέπει είσοδο/έξοδο από το σύστημα αρχείων του χρήστη.
4. Έχει μεθόδους δημιουργούς και θα πρέπει να καλέσουμε τον δημιουργό της **JFrame**.

10.2 Μεθοδολογία ανάπτυξης εφαρμογών για σχεδιασμό γραφικών

Για την ανάπτυξη μιας κλάσης εφαρμογής ακολουθούμε τα εξής βήματα, σε αντιστοιχία με τον κώδικα του Σχ. 1:

1. Ορίζουμε την κλάση της εφαρμογής ως υποκλάση της κλάσης **JFrame** του πακέτου **javax.Swing**, δηλ. προσθέτουμε **extends JFrame** στην επικεφαλίδα της κλάσης (Σχ. 1- γραμμή 4).
2. Δηλώνουμε ως **μεταβλητές υπόστασης** τις μεταβλητές που επιθυμούμε να προσπελαύνονται από όλες τις μεθόδους της κλάσης (Σχ. 1-γραμμή 6).
3. Αρχικοποιούμε τις μεταβλητές υπόστασης στη μέθοδο **main** (Σχ. 1- γραμμή 10).
4. Κατασκευάζουμε μια μέθοδο **paint** μέσα στην οποία πραγματοποιούμε το

σχεδιασμό των γραφικών (Σχ. 1-γραμμές 14-18).

```
SimpleApp.java > ...
1  import javax.swing.JFrame;
2  import java.awt.*;
3
4  public class SimpleApp extends JFrame {
5
6      String s;
7
8      SimpleApp(String atitle) {
9          super(atitle);
10         s = "Hello App World!";
11     }
12
13     public void paint(Graphics g) {
14         super.paint(g);
15         g.setColor(Color.RED);
16
17         // Print string at coordinates x=100, y=100
18         g.drawString(s, x: 100, y: 100);
19     }
20
21     Run | Debug
22     public static void main(String[] args) {
23         SimpleApp frame = new SimpleApp(atitle: "Simple App");
24         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25
26         frame.setSize(width: 300, height: 300);
27         frame.setVisible(b: true);
28     }
29 }
```

Σχήμα 1. Κλάση Java που υλοποιεί εφαρμογή που τυπώνει στην οθόνη το αλφαριθμητικό “Hello App World!”

10.3 Σχεδιασμός γραφικών

Σε μια εφαρμογή μπορούμε να σχεδιάζουμε διάφορα σχήματα μέσω μεθόδων που παρέχει η κλάση Graphics, π.χ. παραλληλόγραμμα (rectangles), ελλείψεις (ovals), τόξα ελλείψεων (arcs), πολύγωνα (polygons) κ.α. (Σχ.2).

```

SimpleGraphicsApp.java > ...
1  import javax.swing.JFrame;
2  import java.awt.*;
3  public class SimpleGraphicsApp extends JFrame {
4      SimpleGraphicsApp(String atitle) {
5          super(atitle);
6      }
7      public void paint(Graphics g) {
8          super.paint(g);
9          // Make a new color object c
10         Color c = new Color(r: 100, g: 0, b: 100);
11         // Set object c as the current drawing color
12         g.setColor(c);
13         // Draw a straight line from point (30,140) to point (100,200)
14         g.drawLine(x1: 30, y1: 140, x2: 100, y2: 200);
15         // Set predefined object RED as current color
16         g.setColor(Color.RED);
17         // Draw an empty rectangle at point (60,130) having width 80 and height 100
18         g.drawRect(x: 60, y: 130, width: 80, height: 100);
19         // Set predefined object GREEN as current color
20         g.setColor(Color.GREEN);
21         // Draw an empty ellipse inscribed within a rectangle at (60, 130) of width 80 and height 200
22         g.drawOval(x: 60, y: 130, width: 80, height: 200);
23         // Draw a filled rectangle and a filled ellipse
24         g.fillRect(x: 90, y: 120, width: 60, height: 50);
25         g.fillOval(x: 70, y: 120, width: 200, height: 20);
26         // Draw an arc of the ellipse defined by the first four arguments, from 0 to 180 degrees
27         g.drawArc(x: 10, y: 120, width: 30, height: 40, startAngle: 0, arcAngle: 180);
28         // Make a new font object f
29         Font f = new Font(name: "Times new Roman", Font.BOLD, size: 30);
30         // Set f as the current font
31         g.setFont(f);
32         // Write a string at coordinates (10,20)
33         g.drawString(str: "test string", x: 10, y: 20);
34         // Draw a polygon with points given as (px,py)
35         int[] px = { 20, 50, 70, 30, 10 };
36         int[] py = { 200, 220, 240, 130, 107 };
37         Polygon p = new Polygon(px, py, px.length);
38         g.drawPolygon(p);
39     }
40     public static void main(String[] args) {
41         SimpleGraphicsApp frame = new SimpleGraphicsApp(atitle: "Simple App");
42         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
43         frame.setSize(width: 300, height: 300);
44         frame.setVisible(b: true);
45     }
46 }

```

Σχήμα 2. Εφαρμογή που καλεί δημοφιλείς μεθόδους της κλάσης Graphics για ζωγραφική.

10.4 Φόρτωση και εμφάνιση εικόνων

Σε μια εφαρμογή μπορούμε πολύ εύκολα να εμφανίσουμε και μια εικόνα, όπως στο παράδειγμα του Σχ.3.

```

1  import javax.swing.JFrame;
2  import java.awt.*;
3  import java.io.*;
4  import javax.imageio.*;
5
6  public class GraphicsImageApp extends JFrame {
7      static Image img;
8
9      GraphicsImageApp(String atitle) {
10         super(atitle);
11     }
12
13     public void paint(Graphics screen) {
14         super.paint(screen);
15
16         int iWidth = img.getWidth(this);
17         int iHeight = img.getHeight(this);
18         int xPos = 10;
19         // 25%
20         screen.drawImage(img, xPos, y: 10,
21             iWidth / 4, iHeight / 4, this);
22         // 100%
23         xPos += (iWidth / 4) + 10;
24         screen.drawImage(img, xPos, y: 10, this);
25     }
26
27     Run | Debug
28     public static void main(String[] args) {
29         try {
30             img = ImageIO.read(new File(pathname: "UTH.jpg"));
31         } catch (IOException e) {
32             GraphicsImageApp frame = new GraphicsImageApp(atitle: "Simple Graphics App");
33             frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34
35             frame.setSize(width: 1400, height: 700);
36             frame.setVisible(b: true);
37         }
38     }
39 }

```

Σχήμα 3. Παράδειγμα φόρτωσης και εμφάνισης εικόνας σε εφαρμογή .

Στην παραπάνω εφαρμογή η μέθοδος **ImageIO.read()** έχει όρισμα το **όνομα του αρχείου**. Εξ' ορισμού βρίσκει το αρχείο εικόνας στον ίδιο φάκελο που τρέχει η εφαρμογή. Αν εμείς επιθυμούμε να διαβάσει μια εικόνα από άλλο φάκελο, απλά συμπληρώνουμε την πλήρη διαδρομή και το όνομα αρχείου εικόνας. Φυσικά θα πρέπει να χειριζόμαστε τα **πιθανά exceptions**, με τον τρόπο που βλέπετε στην γραμμή 28 του σχήματος 3.

Επίσης η drawImage, στη γραμμή 20 του Σχ.3, μπορεί να κληθεί και με έξι ορίσματα, π.χ.

screen.drawImage (img, 0, 0, 100, 200, this);

η οποία προσδιορίζει ότι η εικόνα θα πρέπει να έχει συγκεκριμένο μέγεθος (ακόμα και αν παραμορφώνεται), δηλαδή, πλάτος 100 και ύψος 200 εικονοστοιχεία.

Ασκήσεις

1. Να υλοποιηθούν και να εκτελεσθούν οι εφαρμογές του Σχ.1,2 και 3.
2. Να κατασκευαστεί μια εφαρμογή που να εκτυπώνει το όνομα του χρήστη μέσα σε ένα παραλληλόγραμμο πλαίσιο. Το όνομα του χρήστη να δίνεται ως παράμετρος στην εκτέλεση της εφαρμογής.
3. Να κατασκευαστεί εφαρμογή που να ζωγραφίζει ένα κυκλικό κουλούρι πάχους τουλάχιστον 20 εικονοστοιχείων. Να τροποποιηθεί ώστε να ζωγραφίζει το σήμα των ολυμπιακών αγώνων.
4. Να κατασκευαστεί εφαρμογή που να ζωγραφίζει 100 κύκλους τυχαίας ακτίνας σε τυχαία σημεία. Να τροποποιηθεί ώστε να ζωγραφίζει N κύκλους.
5. Να κατασκευάσετε ένα τυχαίο πολύγωνο, με τυχαίο πλήθος γωνιών.
6. Να φορτώσετε μια εικόνα ενός προσώπου και χρησιμοποιώντας γραφικά να του «φορέσετε » γυαλιά.