



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Οντοκεντρικός Προγραμματισμός

Ενότητα 4: JAVA: ΔΙΕΠΑΦΕΣ ΧΡΗΣΤΗ, ΓΡΑΦΙΚΑ, APPLETS

Γραφικά, Java Applets

ΔΙΔΑΣΚΟΝΤΕΣ: Ιωάννης Χατζηλυγερούδης, Χρήστος
Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΓΡΑΦΙΚΑ με JAVA2D

ΧΡΗΣΗ ΓΡΑΦΙΚΩΝ

- Μέσω της Java2D:
 - Ένα σύνολο κλάσεων που υποστηρίζει υψηλής ποιότητας 2D γραφικά, εικόνες, χρώμα και κείμενο
 - Περιλαμβάνει κλάσεις από awt και swing
- Βασική κλάση: Graphics2D
 - Επέκταση της Graphics για καλύτερης ποιότητας γραφικά



ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ GRAPHICS

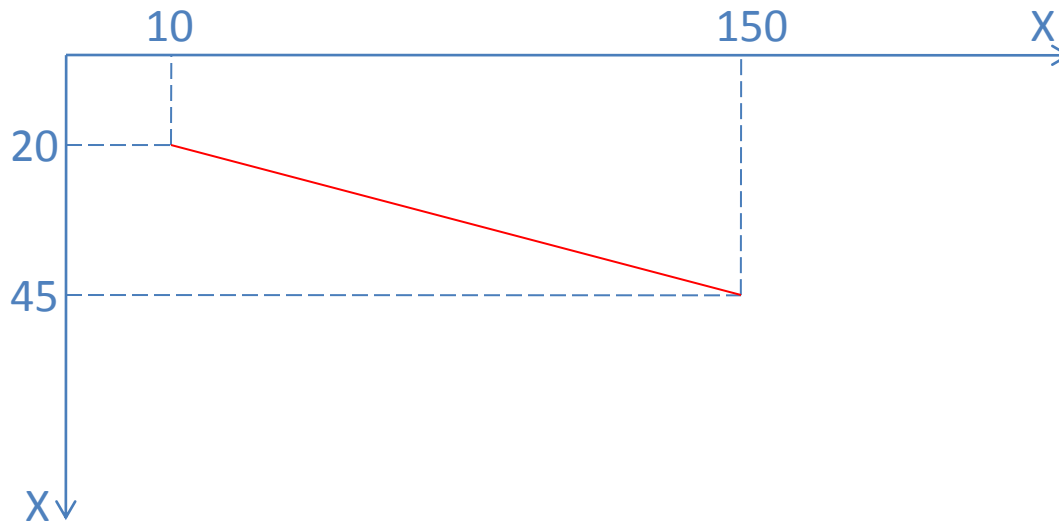
Σχήματα

- Ένα σχήμα μπορεί να είναι filled ή unfilled, ανάλογα με τη μέθοδο
- Οι παράμετροι των μεθόδων καθορίζουν συντεταγμένες και μεγέθη



ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ GRAPHICS

■ Γραμμές



```
g.drawLine(10, 20, 150, 45);
```

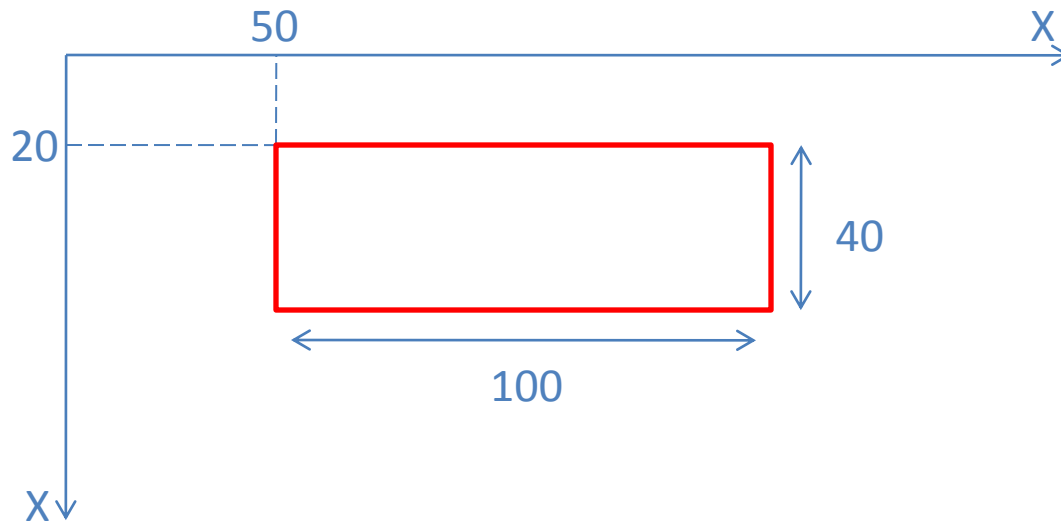
ή

```
g.drawLine(150, 45, 10, 20);
```



ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ GRAPHICS

■ Ορθογώνια

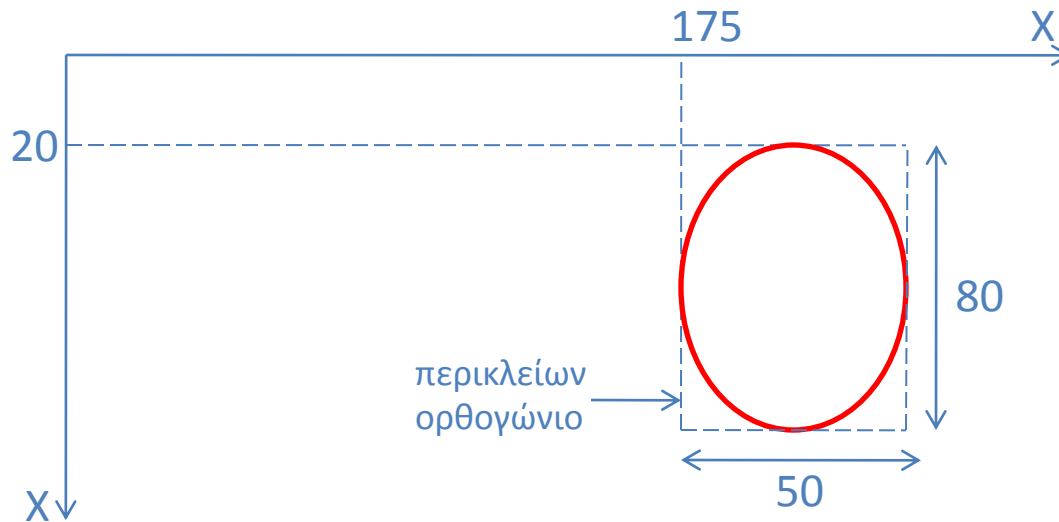


```
g.drawRect (50, 20, 100, 40) ;
```



ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ GRAPHICS

■ Ελλείψεις



```
g.drawOval(175, 20, 50, 80);
```

Ένα τόξο (*arc*) είναι ένα τμήμα μιας έλλειψης

```
g.drawArc(175, 20, 50, 80, 0, 45);
```

ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ GRAPHICS

Μέθοδοι Σχεδίασης Σχημάτων (υποσύνολο):

`drawLine(startX, startY, endX, endY)`

`drawRect(startX, startY, width, height)`

`fillRect(startX, startY, width, height)`

`drawRoundRect(startX, startY, width, height, arcWidth, arcHeight)`

`fillRoundRect(startX, startY, width, height, arcWidth, arcHeight)`

`draw3DRect(startX, startY, width, height, boolean value)`

`fill3DRect(startX, startY, width, height, boolean value)`

`drawPolygon(arrayX, arrayY, array length)`

`fillPolygon(arrayX, arrayY, array length)`

`drawPolyline(arrayX arrayX, arrayY arrayY, array length)`

`fillPolyline(arrayX arrayX, arrayY arrayY, array length)`

`drawOval(TopCornerX, TopCornerY, width, height)`

`fillOval(TopCornerX, TopCornerY, width, height)`

`drawArc(TopCornerX, TopCornerY, width, height, arcStart, arcStop)`

`fillArc(TopCornerX, TopCornerY, width, height, arcStart, arcStop)`

Μέθοδοι αντιγραφής - καθαρισμού περιοχών:

`copyArea(StartX, StartY, Width, Height, dX, dY)`

`clearRect(StartX, StartY, Width, Height)`



ΚΛΑΣΗ COLOR

- Ένα χρώμα ορίζεται με χρήση ενός στιγμιότυπου της κλάσης Color
 - `Color myColor = new Color(200, 90, 30);`
- Κάθε drawing surface έχει ένα **background color**
- Κάθε graphics context έχει ένα τρέχον **foreground color** και **font** (type, size).
- Και τα δύο μπορούν να αλλάξουν δυναμικά
 - `setBackground(c)`
 - `g.setColor(c)` `g.setFont(f)`
- Μερικά predefined colors:

Object	RGB Value
Color.black	0, 0, 0
Color.blue	0, 0, 0
Color.cyan	0, 255, 255
Color.orange	255, 255, 0
Color.white	255, 255, 255
Color.yellow	255, 255, 0



Graphics 2D

- Σχεδιασμός γραφικών σε ένα JPanel

```
Public void paintComponent (Graphics comp){  
    Graphics2D comp2D = (Graphics2D) comp;  
    Font myFont = new Font("Arial", Font.ITALIC, 14);  
    comp2D.setFont(myFont);  
    comp2D.setColor(Color.orange);  
    comp2D.drawString("Hello !!! ", 100, 50);  
}
```

- Η μέθοδος `paintComponent()` καλείται αυτόματα κάθε φορά που ο υποδοχέας πρέπει να σχεδιαστεί ξανά
- Συνήθης μέθοδος: Δημιουργούμε μια υποκλάση της `JPanel` και υπερσκελίζουμε την `paintComponent()`.



Graphics2D

Καθορισμός πένα σχεδίασης

- Τα αντικείμενα που προσδιορίζουν σχήματα ανήκουν στο πακέτο κλάσεων `java.awt.geom`.
- Όλα τα σχήματα σχεδιάζονται με γραμμές που έχουν πλάτος ένα pixel
- Η αλλαγή του πάχους των γραμμών γίνεται με την δημιουργία μιας «πένας» ως εξής:

```
BasicStroke brush = new BasicStroke(5);  
comp2D.setStroke(brush);
```

- Επίσης μπορούμε με την `BasicStroke` να καθορίσουμε το στυλ τελειώματος μια γραμμής και το στυλ της ένωσης ανάμεσα σε δυο ευθύγραμμα τμήματα



Graphics2D

Δημιουργία Αντικειμένων Σχεδίασης

- Γραμμές:

```
Line2D.Float line1 = new Line2D.Float( 40F, 200F, 70F, 130F);  
comp2D.draw(line1);
```

- Ορθογώνια:

```
Rectangle2D.Float rect1 = new Rectangle2D.Float (10F,20F,50F,60F);  
comp2D.draw(rect1);
```

- Ορθογώνια (με στρογγυλεμένες γωνίες):

```
Rectangle2D.Float.rect2 = new Rectangle2D.Float(10F,20F,50F,60F, 15F, 15F);  
comp2D.draw(rect2);
```

- Μετά την δημιουργία του σχήματος αυτό μπορεί να σχεδιαστεί με τις μεθόδους `draw()` και `fill()`.

- Η `fill()` γεμίζει ένα σχήμα με μοτίβα γεμίσματος ή με ένα μόνο χρώμα.



Graphics2D

Δημιουργία Αντικειμένων Σχεδίασης

- Κύκλοι ή ελλείψεις:

```
Ellipse2D.Float circle = new Ellipse2D.Float(50F, 100F, 20F, 20F);
```

- Τόξα:

```
Arc2D.Float arc = new Arc2D.Float (10F,20F,30F,30F,45F,120F, Arc2D.Float.OPEN);
```

- Η σχεδίαση πολυγώνου γίνεται ως:

```
GeneralPath poligono = new GeneralPath(); // Δημιουργία  
poligono.moveTo(10F, 20F) // Αρχικό σημείο  
poligono.lineTo(20F, 10F) //Επόμενο σημείο  
poligono.lineTo(30F, 60F) //Επόμενο σημείο  
...  
poligono.closePath(); //Κλείσιμο πολυγώνου
```



JAVA APPLETS

JAVA APPLETS

- Ένα **Java application** είναι ένα αυτόνομο (stand-alone) πρόγραμμα που διαθέτει μια `main` μέθοδο.
- Ένα **Java applet** είναι ένα πρόγραμμα που θα πρέπει να μεταφερθεί μέσω του Web σε ένα **web browser** και να εκτελεστεί στα πλαίσια του.
- Ένα applet μπορεί επίσης να εκτελεσθεί και με τη χρήση του **appletviewer** tool του Java Software Development Kit
- Ένα applet δεν έχει main μέθοδο
- Αντίθετα, υπάρχουν μια σειρά από μεθόδους που εξυπηρετούν συγκεκριμένους σκοπούς.



JAVA APPLETS

Βήματα δημιουργίας ενός Java Applet

- Γράφουμε τον πηγαίο κώδικα, π.χ. TestApplet.java
- Μεταφράζουμε το Applet για να παραχθεί το αρχείο TestApplet.class
- Ενσωματώνουμε το Applet σε μια ιστοσελίδα χρησιμοποιώντας (κατ' ελάχιστο) την ετικέτα <Applet> η οποία έχει την εξής (ελάχιστη) μορφή:
<applet code= TestApplet.class width=πλάτος height=ύψος></applet>
- Ένα Applet εκτελείται όταν φορτώνεται η ιστοσελίδα που το περιέχει σε κάποιον browser



JAVA APPLETS

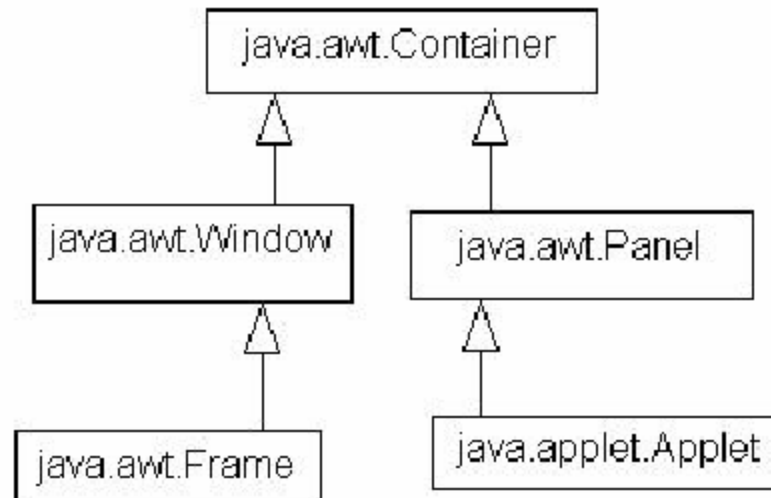
HTML αρχείο:

```
<HTML>
  <HEAD>
    <TITLE>Test Applet</TITLE>
  </HEAD>
  <BODY>
    <APPLET CODE= " TestApplet.class"
              WIDTH=200 HEIGHT=200>
    </APPLET>
  </BODY>
</HTML>
```



JAVA APPLETS

Υποκλάση της κλάσης
`java.applet.Applet`



JAVA APPLETS

HelloWorld:

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends Applet{
    public void paint(Graphics g){
        g.drawString("Hello world!", 50, 25);
    }
}
```



ΚΥΚΛΟΣ ΖΩΗΣ ΕΝΟΣ APPLET

1. Ο browser δημιουργεί ένα στιγμιότυπο της κλάσης Applet με χρήση του default constructor
2. Το applet εμφανίζεται στην ιστοσελίδα στη θέση και με το μέγεθος που καθορίζεται στον HTML κώδικα
3. Ο browser εκτελεί την **init()** μέθοδο του applet
4. Ο browser εκτελεί την **start()** μέθοδο του applet
5. Ο browser εκτελεί την **paint()** μέθοδο του applet
6. Το applet είναι ενεργό και «τρέχει»
7. Ο browser καλεί την **paint()** κάθε φορά που το applet πρέπει να «ξαναζωγραφιστεί»
8. Ο browser εκτελεί την **stop()** μέθοδο όταν ο χρήστης φεύγει από την ιστοσελίδα ή το applet είναι έτοιμο να καταστραφεί
9. Ο browser εκτελεί την **destroy()** μέθοδο αμέσως πριν καταστρέψει το applet



ΒΑΣΙΚΕΣ ΜΕΘΟΔΟΙ APPLET

- **init()** . Αρχικοποίηση – Εκτελείται μία φορά όταν δημιουργείται το applet. Χρησιμοποιείται για αρχικοποίηση, π.χ. μεταβλητών του applet.
- **start()** . Εκκίνηση εκτέλεσης – Εκτελείται την πρώτη φορά αλλά και κάθε φορά που ο έλεγχος επανέρχεται από τον browser στην HTML σελίδα που περιέχει το applet (εκκίνηση threads, animation, playing sound, κλπ.)
- **stop()** . Τερματισμός εκτέλεσης – Πριν την καταστροφή του applet και όταν ο χρήστης εγκαταλείπει την ιστοσελίδα. Τυπικά, τερματίζεται ότι ξεκίνησε στη start().
- **destroy()** . Καταστροφή στιγμιότυπου – garbage collection.



ΒΑΣΙΚΕΣ ΜΕΘΟΔΟΙ APPLET

- `paint(Graphics g)` . Ζωγράφισε το applet – Εκτελείται κάθε φορά που πρέπει να ζωγραφιστεί το περιεχόμενο του applet
 - Μετά την `start()`
 - Όταν το παράθυρο έρχεται στο προσκήνιο
 - Όταν το ζητάει το πρόγραμμα
- Η `paint()` κληρονομείται από την κλάση `java.awt.Container`
- Η παράμετρος `java.awt.Graphics` αναπαριστά το τμήμα της οθόνης που το applet μπορεί να ζωγραφίσει



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.1



Σημείωμα Αναφοράς

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

