



## Αντικειμενοστρεφής Προγραμματισμός Διάλεξη – 12 : ΜΙΚΡΟΕΦΑΡΜΟΓΕΣ (APPLETS)



### ΕΙΣΑΓΩΓΗ ΣΤΑ APPLETS

- Ένα applet είναι ένα πρόγραμμα Java που τρέχει σε ένα πρόγραμμα περιήγησης στο Web (web browser). Ένα applet μπορεί να είναι μια πλήρως λειτουργική εφαρμογή Java, διότι έχει όλο το Java API στη διάθεσή του και κατά την κατασκευή του ο προγραμματιστής μπορεί να χρησιμοποιήσει όλο το API.
- Υπάρχουν όμως και μερικές σημαντικές διαφορές μεταξύ ενός applet και μια αυτόνομης εφαρμογής Java (standalone java application). Οι σπουδαιότερες διαφορές είναι οι εξής:

## ΔΙΑΦΟΡΕΣ APPLETS ΚΑΙ STANDALONE (1/2)

- Ένα applet είναι μία Java class που κάνει extend (δηλαδή κληρονομεί) την java.applet.Applet class.
- Δεν καλείται ποτέ η μέθοδος main() σε ένα applet και ως εκ τούτου δεν υπάρχει κώδικας της main().
- Τα Applets έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να ενσωματώνονται μέσα σε μια σελίδα HTML.
- Όταν ένας χρήστης προβάλλει μια σελίδα HTML που περιέχει ένα applet, ο κώδικας για το applet κατεβαίνει στον υπολογιστή του χρήστη και τρέχει από εκεί.
- Χρειάζεται μία Java Virtual Machine (JVM) για να δούμε το applet. Η JVM μπορεί είτε να είναι ένα plug-in του Web browser είτε ένα εντελώς ξεχωριστό περιβάλλον εκτέλεσης προγραμμάτων.

## ΔΙΑΦΟΡΕΣ APPLETS ΚΑΙ STANDALONE (2/2)

- Η JVM στο μηχάνημα του χρήστη δημιουργεί ένα στιγμιότυπο της κλάσης applet και μπορεί να καλέσει διάφορες μεθόδους κατά τη διάρκεια της ζωής του applet.
- Τα Applets έχουν αυστηρούς κανόνες ασφαλείας που επιβάλλονται από το Web Browser. Η ασφάλεια ενός applet συχνά αναφέρεται και ως ασφάλεια sandbox, συγκρίνοντας το applet με ένα παιδί που παίζει σε ένα sandbox με διάφορους κανόνες που πρέπει να ακολουθούνται.
- Εάν τα applets για να τρέξουν χρειάζονται και άλλες κλάσεις τότε όλες μαζί και μαζί με το applet μπορούν να κατεβούν σε ένα Java Archive (JAR) αρχείο.

## Διάρκεια Ζωής ενός Applet (1/2)

- Πέντε μέθοδοι είναι ότι μας δίνει η **Applet class** με βάση τις οποίες μπορούμε να δημιουργήσουμε και να λειτουργήσουμε ένα **applet**:
  - **init**: Αυτή η μέθοδος προορίζεται για ό, τι προετοιμασία χρειάζεται να γίνει πριν την εκτέλεση του **applet**.
  - **start**: Αυτή η μέθοδος καλείται αυτόματα αμέσως μόλις το πρόγραμμα περιήγησης καλεί τη μέθοδο **init**. Καλείται, επίσης, κάθε φορά που ο χρήστης επιστρέφει στη σελίδα που περιέχει το **applet** αφού πάει σε άλλες σελίδες.
  - **stop**: Αυτή η μέθοδος καλείται αυτόματα όταν ο χρήστης φύγει από τη σελίδα στην οποία βρίσκεται η μικροεφαρμογή. Μπορεί, επομένως, να κληθεί επανειλημμένα στο ίδιο **applet**.

## Διάρκεια Ζωής ενός Applet (2/2)

- **destroy**: Αυτή η μέθοδος καλείται μόνο όταν το πρόγραμμα περιήγησης κλείσει κανονικά. Επειδή τα **applets** είναι προορισμένα να «ζουν» σε μια σελίδα **HTML**, κανονικά δεν θα πρέπει να αφήνουμε τους πόρους πίσω ανεκμετάλλευτους και δεσμευμένους όταν αφήσουμε την ιστοσελίδα που περιέχει το **applet**.
- **paint**: Αυτή η μέθοδος καλείται αμέσως μετά την **start()** και επίσης κάθε φορά που το **applet** χρειάζεται να κάνει **render** τον εαυτό του στον **browser**. Η μέθοδος **paint()** κληρονομεί την **java.awt** και έχει τη μορφή **public void paint(Graphics g)**. Αυτή καλείται κάθε φορά που χρειάζεται να σχεδιαστεί κάτι στην οθόνη, πχ μερική επικάλυψη παραθύρου από άλλο. Η μέθοδος **paint** χρησιμοποιεί ένα αντικείμενο **Graphics** που επιλέγουμε να το λέμε **g**. Η κλάση **Graphics** ορίζεται στο πακέτο **java.awt.Graphics**

## "Hello, World" Applet

- Το παρακάτω είναι ένα απλό applet με την ονομασία HelloWorldApplet.java:

```
import java.applet.*;
import java.awt.*;
public class HelloWorldApplet extends Applet {
    public void paint (Graphics g)
    {
        g.drawString ("Hello World", 25, 50);
    }
}
```

- Ιδιαίτερη προσοχή στα import:
  - java.applet.Applet.
  - java.awt.Graphics.
  - Χωρίς αυτά, ο Java compiler δεν θα μπορούσε να αναγνωρίσει τις κλάσεις Applet και Graphics, στις οποίες αναφέρεται η applet class.

## Η Applet CLASS (1/2)

- Κάθε applet κληρονομεί την κλάση *java.applet.Applet class*. Η βασική Applet class παρέχει όλες τις ουσιαστικές μεθόδους τις οποίες μπορεί η κληρονομούσα Applet class μπορεί να καλεί για να παίρνει πληροφορία και υπηρεσίες από το περιεχόμενο του browser. Τα παραπάνω περιέχουν μεθόδους κάνουν τα εξής
  - Να λαμβάνουν τις παραμέτρους του applet
  - Μπορούμε μάθουμε την τοποθεσία του δικτύου όπου το HTML αρχείο περιέχει το applet
  - Να εκτυπώσουμε το μήνυμα της κατάστασης του browser
  - Μπορούμε να κάνουμε λήψη μίας εικόνας
  - Μπορούμε να κάνουμε λήψη ενός αρχείου ήχου
  - Μπορούμε να εκτελέσουμε ένα αρχείο ήχου (δηλαδή να κάνουμε Play)
  - Μπορούμε να αλλάξουμε το μέγεθος ενός applet (Resize applet)

## Η Applet CLASS (2/2)

- Επιπροσθέτως η Applet class παρέχει μία διεπαφή (interface) με το οποίο μπορούμε να γράψουμε κώδικα για τον browser ώστε να μπορεί να λάβει πληροφορία σχετικά με την εκτέλεση του applet. Πιο συγκεκριμένα μπορούμε να:
  - Ζητήσουμε πληροφορία σχετικά με τα εξής: (author, version, copyright)
  - Ζητήσουμε πληροφορία σχετικά με την περιγραφή των παραμέτρων που αναγνωρίζει το applet
  - Αρχικοποιήσουμε το applet
  - Καταστρέψουμε το applet
  - Αρχίσουμε την εκτέλεση του applet
  - Σταματήσουμε την εκτέλεση του applet
- Η Applet class παρέχει κάποια default ανάπτυξη για καθεμία από αυτές τις μεθόδους. Αυτές οι αναπτύξεις κώδικα μπορούν να γίνουν overridden αν χρειάζεται.

## Πως καλούμε ένα Applet (1/2)

- Ένα applet μπορεί να καλεστεί μέσα από ένα HTML αρχείο και με τη χρήση ενός browser που είναι Java-enabled. Το βασικό tag που χρησιμοποιείται είναι το <applet> tag. Για το "Hello, World" applet δείτε πως γίνεται η ενσωμάτωση:

```
<html>
<title>The Hello, World Applet
</title>
<hr>
<applet code="HelloWorldApplet.class" width="320"
height="120">If your browser was Java-enabled, a "Hello,
World" message would appear here.
</applet>
<hr>
</html>
```

## Πως καλούμε ένα Applet (2/2)

- Οι ιδιότητες `code`, `width` και `height` είναι απαραίτητες ιδιότητες και όλες πρέπει να ορίζονται. Επίσης δεν πρέπει να ξεχνάμε να κλείσουμε το `applet-tag` `</applet>`.
- Αν κάποιο `applet` παίρνει παραμέτρους τότε μπορούμε να χρησιμοποιήσουμε διάφορα `<param>` `tags` ανάμεσα στα `tags <applet>` και `</applet>`. Ο `browser` παραβλέπει το κείμενο ή άλλα `tags` τα οποία είναι ανάμεσα στα `applet tags`.
- Οι `browsers` που δεν είναι `Java-enabled` δεν μπορούν να επεξεργαστούν τα `applet tags`. Επομένως οτιδήποτε εμφανίζεται ανάμεσα στα `tags <applet>` και `</applet>` μπορεί να εμφανιστεί σε τέτοιους `browsers`.
- Εάν ένα `applet` βρίσκεται σε κάποιο πακέτο που δεν είναι το `default`, τότε την θέση που βρίσκεται αυτό το πακέτο πρέπει να την αναφέρουμε σαν ιδιότητα χρησιμοποιώντας την τελεία (.) για να διαχωρίζουμε τα διάφορα κομμάτια του πακέτου. Δείτε το εξής παράδειγμα: `<applet code="mypackage.subpackage.TestApplet.class" width="320" height="120">`

## Πως μπορούμε να πάρουμε τα Applet Parameters (1/3)

- Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να κάνουμε ένα `applet` να ανταποκρίνεται σε παραμέτρους ρύθμισης που προσδιορίζονται στο `HTML`-αρχείο. Αυτό το `applet` εμφανίζει ένα μοτίβο σκακιέρας μαύρου χρώματος καθώς επίσης και ένα δεύτερο χρώμα.
- Αυτό το `applet` που ονομάζεται `CheckerApplet` παίρνει τις παραμέτρους στην μέθοδο `init()`. Μπορεί σε μερικές περιπτώσεις να παίρνει τις παραμέτρους στην `paint()` μέθοδο.
- Ο `browser` καλεί την `init()` μία φορά αμέσως μόλις φορτώσει το `applet`. Η `default Applet.init()` όμως δεν κάνει τίποτα . Μπορούμε να κάνουμε όμως `Override` την `init()` και να εισαγάγουμε τον δικό μας κώδικα εφόσον χρειάζεται.
- Η μέθοδος `Applet.getParameter()` λαμβάνει μία παράμετρο αν σε αυτή τη μέθοδο περάσουμε το όνομα της παραμέτρου (πάντα σαν συμβολοσειρά). Αν η τιμή της παραμέτρου είναι αριθμητική πρέπει να την κάνουμε `parse`.

## Πως μπορούμε να πάρουμε τα Applet Parameters (2/3)

```
import java.applet.*;
import java.awt.*;
public class CheckerApplet extends Applet
{
    int squareSize = 50;// initialized to default size
    public void init () {}
    private void parseSquareSize (String param) {}
    private Color parseColor (String param) {}
    public void paint (Graphics g) {}
}
```

## Πως μπορούμε να πάρουμε τα Applet Parameters

(3/3)

```
public void init ()
{
    String squareSizeParam = getParameter ("squareSize");
    parseSquareSize (squareSizeParam);
    String colorParam = getParameter ("color");
    Color fg = parseColor (colorParam);
    setBackground (Color.black);
    setForeground (fg);
}
private void parseSquareSize (String param)
{
    if (param == null) return;
    try {
        squareSize = Integer.parseInt (param);
    }
    catch (Exception e) {
        // Let default value remain
    }
}
```

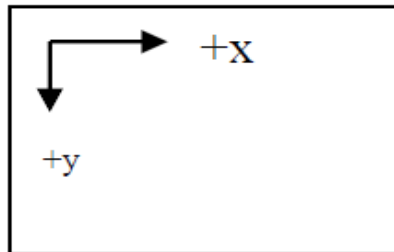
## Γραφικά και χρώμα

- Το πακέτο Abstract Window Toolkit AWT, και το πακέτο swing που είναι νεώτερο του AWT και πιο πλούσιο. Με την εντολή `import java.awt.*;` ή `import javax.swing.*;` κάνω `import` όλων των κλάσεων του πακέτου για να είναι διαθέσιμες στο applet που έχω δημιουργήσει.
- Στο πακέτο awt υπάρχει η τάξη Color με την οποία ρυθμίζουμε τα χρώματα στην οθόνη. (RGB- Red, Green, Blue 0-255). Υπάρχουν και τα έτοιμα χρώματα π.χ. blue, red, white κ.λ.π.
  - Μέθοδοι για ορισμό χρώματος στο
  - Προσκήνιο : `setForeground(Color.gray)`
  - Παρασκήνιο : `setBackground(Color.cyan)`
  - Χρήση : `g.setColor(Color.yellow)`

## Σύστημα συντεταγμένων

- Η σχεδίαση ενός αντικειμένου στην οθόνη, γίνεται μέσω μεθόδων της κλάσης Graphics. Όλες οι μέθοδοι σχεδίασης παίρνουν ορίσματα που αφορούν σημεία, γωνίες ή αρχικά σημεία των αντικειμένων στο σύστημα συντεταγμένων μιας applet. Το σύστημα συντεταγμένων έχει την αρχή του (0,0) στην αριστερή άνω γωνία. Οι θετικές τιμές του x βρίσκονται στα δεξιά και οι θετικές y τιμές είναι προς τα κάτω.

0,0





## Γεωμετρικά σχήματα

- Μέθοδοι για να ζωγραφίσουμε γεωμετρικά σχήματα
  - `g.drawLine(x1,y1,x2,y2)` -> ζωγραφίζει ένα ευθύγραμμο τμήμα από το σημείο `x1,y1` ως το σημείο `x2,y2`
  - `g.drawRect(x1,y1,πλάτος, ύψος)` -> Παραλληλόγραμμο
  - `g.drawOval(x1,y1,πλάτος, ύψος)` -> έλλειψη
  - `g.drawArc(x1,y1,πλάτος, ύψος, αρχική γωνία, άνοιγμα γωνίας)` -> Τόξο
  - `g.drawPolygon(int xpoints[],int ypoints[],int points)` -> Πολύγωνο

## Πως εμφανίζουμε εικόνες σε applet

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

public class ImageDemo extends Applet
{
    private Image image;
    private AppletContext context;
    public void init()
    {
        context = this.getAppletContext();
        String imageURL = this.getParameter("image");
        if(imageURL == null)
        {
            imageURL = "java.jpg";
        }
        try
        {
            URL url = new URL(this.getDocumentBase(),
            imageURL);
            image = context.getImage(url);
        }
        catch(MalformedURLException e)
        {
            e.printStackTrace();
            // Display in browser status bar
            context.showStatus("Could not load
            image!");
        }
        public void paint(Graphics g)
        {
            context.showStatus("Displaying image");
            g.drawImage(image, 0, 0, 200, 84, null);
            g.drawString("www.javalicence.com", 35, 100);
        }
    }
}
```

## Άσκηση 1

- Να γράψετε ένα applet ShowColors το οποίο να τυπώνει ένα μήνυμα με πολλούς συνδυασμούς χρωμάτων.

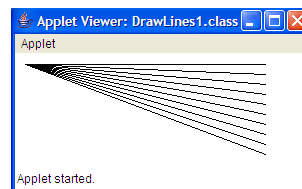
```
// ShowColors.java
import java.awt.*;
import java.applet.*; // εισαγωγή πακέτου για χρήση των applets
public class ShowColors extends Applet { //επεκτείνω την υπάρχουσα κλάση Applet
    // προσοχή το όνομα του πακέτου είναι applet ενώ
    // το όνομα της κλάσης που επεκτείνω είναι Applet

    public void init() { }
    public void paint(Graphics g) {
        for (int j=0; j<10; j++) {
            // set new drawing color using integers
            int rc = (int)(Math.random()*256);
            int gc = (int)(Math.random()*256);
            int bc = (int)(Math.random()*256);
            Color col = new Color(rc,gc,bc);
            g.setColor( col );
            g.drawString( "Hello ! Welcome to Java programming with applets ", 130, 30*j );
        }
    }
}
```

## Άσκηση 2

- Να γράψετε ένα applet DrawLines1 το οποίο να σχεδιάζει 10 ευθύγραμμα τμήματα όπως στο παρακάτω σχήμα.

```
import java.awt.*;
import java.applet.*;
public class DrawLines1 extends Applet {
    public void init() { }
    public void paint(Graphics g) {
        int counter = 1;
        while (counter<=10){
            g.drawLine(10,10,250, counter *10);
            ++counter;
        }
    }
}
```



## Άσκηση 3

- Να γράψετε ένα applet TestPolygon το οποίο να σχεδιάζει ένα τυχαίο πεντάγωνο.

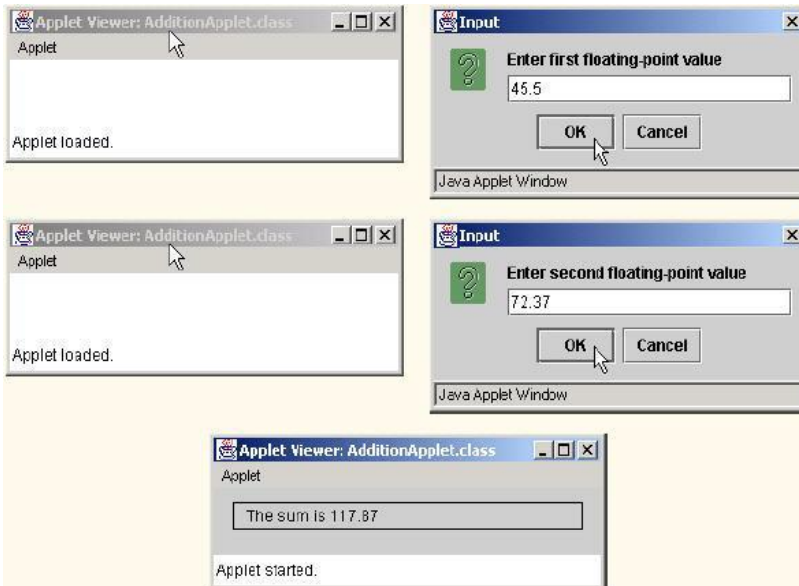
```
import java.awt.*;
import java.applet.*;
public class TestPolygon extends Applet {
    public void init() {
    }
    public void paint(Graphics g) {
        int xpoints[]={30,140,160,130,50};
        int ypoints[]={40,70,100,140,120};
        int points=5;
        g.drawPolygon(xpoints,ypoints,points);
    }
}
```

## Άσκηση 4 (1/3)

- Δημιουργήστε ένα applet που θα προσθέτει 2 Double precision αριθμούς. Αυτό το applet θα ζητά από το χρήστη να εισάγει δύο αριθμούς κινητής υποδιαστολής (δηλαδή, τους αριθμούς με δεκαδικό σημείο, όπως 7.33, 0.0975 και 1000.12345). Η εφαρμογή να υπολογίζει το άθροισμα των τιμών και να εμφανίζει το αποτέλεσμα. Το αρχείο HTML για να φορτώσει αυτό το applet σε appletviewer είναι το παρακάτω:

```
1 <html>
2 <applet code = "AdditionApplet.class" width = "300" height = "65">
3 </applet>
4 </html>
```

## Άσκηση 4 (2/3)



## Άσκηση 4 (3/3)

```
import java.awt.Graphics; // import class Graphics
import javax.swing.*; // import package javax.swing
public class AdditionApplet extends JApplet {
    double sum; // sum of values entered by user
    public void init() {
        String firstNum, secNum;
        double number1, number2;
        firstNum = JOptionPane.showInputDialog( "Enter first
            floating-point value" );
        secNum = JOptionPane.showInputDialog( "Enter second
            floating-point value" );
        number1 = Double.parseDouble( firstNum );
        number2 = Double.parseDouble( secNum );
        sum = number1 + number2;
    }
    public void paint( Graphics g ) {
        super.paint( g );
        g.drawRect( 15, 10, 270, 20 );
        g.drawString( "The sum is " + sum, 25, 25 ); }
}
```

## Άσκηση 5 (1/3)

- Δημιουργήστε ένα applet που θα χρωματίζει το background με χρώμα κυανό. Θα παίρνει τις συντεταγμένες τις οθόνης από το σύστημα και θα δημιουργεί ένα ορθογώνιο χρώματος μπλε γύρο από την οθόνη όπως φαίνεται και στο σχήμα. Στη συνέχεια θα δημιουργεί την πρόταση "I'm a simple applet" σε χρώμα γραμματοσειράς κόκκινο.



## Άσκηση 5 (2/3)

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;
public class SimpleApplet extends Applet {
    String text = "I'm a simple applet";
    public void init() {
        text = "I'm a simple applet";
        setBackground(Color.cyan);
    }
    public void start() {
        System.out.println("starting...");
    }
    public void stop() {
        System.out.println("stopping...");
    }
}
```

## Άσκηση 5 (3/3)

```
public void paint(Graphics g){
    System.out.println("Paint");
    g.setColor(Color.blue);
    g.drawRect(0, 0, getSize().width -1,
               getSize().height -1);
    g.setColor(Color.red);
    g.drawString(text, 15, 25);
}
```