

# Προγραμματισμός II

13/5/2020

Γραφικές διεπαφές χρήστη

# Γραφικές διεπαφές χρήστη

- Σε αυτό το μάθημα θα γίνει μια εισαγωγή στη σχεδίαση γραφικών διεπαφών χρήστη (Graphical User Interfaces-GUI).
- Θα εστιάσουμε στη χρήση της βιβλιοθήκης Swing (πακέτο `java.swing`).
- Η Swing προσφέρει βελτιωμένες κλάσεις γραφικών συστατικών σε σχέση με την προγενέστερη βιβλιοθήκη Abstract Windowing Toolkit (`java.awt`).
- Ωστόσο, εξακολουθούν να χρησιμοποιούνται κάποια στοιχεία από την τελευταία.

# Παράθυρα

```
import javax.swing.*;
```

```
public class FirstAttempt
```

```
{
```

```
    public static void main (String[] args)
```

```
    {
```

```
        JFrame window=new JFrame("My first window");
```

```
        window.setSize(100,100);
```

```
        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        window.setVisible(true);
```

```
    }
```

```
}
```

# Παράθυρα

- Η μέθοδος **setSize** ορίζει το αρχικό μέγεθος του παραθύρου.
- Η μέθοδος **setDefaultOperation** ορίζει, μέσω της σταθεράς **EXIT\_ON\_CLOSE** πως όταν πιέζεται το κουμπί [X] του παραθύρου θα πρέπει να τερματίζεται η λειτουργία της εφαρμογής.
- Η μέθοδος **setVisible** δίνει την εντολή να εμφανιστεί το παράθυρο (ή να κρυφτεί σε περίπτωση που το όρισμά της είναι **false**).
- Αυτός δεν είναι ο ενδεδειγμένος τρόπος κατασκευής παραθύρων.

# Παράθυρα

```
import javax.swing.*;  
  
public class MyWindow extends JFrame  
{  
    JButton b1=new JButton("OK");  
  
    public MyWindow(String title)  
    {  
        super(title);  
        setSize(100,100);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setVisible(true);  
  
        //Add button to window  
        add(b1);  
    }  
}
```

# Παράθυρα

- Αυτός ήταν ο προτεινόμενος τρόπος κατασκευής παραθύρων.
- Παρουσιάστηκε η κατασκευή ενός παραθύρου με ένα κουμπί (το οποίο δεν εκτελεί κάποια ενέργεια).
- Ένα παράθυρο πρέπει να κατασκευάζεται ως νέα κλάση η οποία είναι υποκλάση της JFrame.
- Τα περιεχόμενα του παραθύρου (τα οποία ονομάζονται συστατικά-components) πχ κουμπιά, πρέπει να δηλώνονται και να δημιουργούνται ως μεταβλητές υπόστασης της κλάσης παραθύρου.

# Παράθυρα

- Η συναρμολόγηση του παραθύρου από τα συστατικά του πρέπει να γίνεται μέσα στη μέθοδο δημιουργό της κλάσης παραθύρου.
- Για να εμφανιστεί το παράθυρο της προηγούμενης κλάσης, αρκεί να δημιουργηθεί ένα αντικείμενο της κλάσης `myWindow`.

```
import class SecondAttempt  
{  
    public static void main (String[] args)  
    {  
        MyWindow w=new MyWindow("My second window");  
    }  
}
```



# Συναρμολόγηση παραθύρου

- Για να συναρμολογηθεί ένα παράθυρο από διάφορα συστατικά, πρέπει πρώτα να δημιουργήσουμε τα αντικείμενα των συστατικών και μετά να τα προσθέσουμε στο παράθυρο (όπως προστέθηκε το κουμπί στο προηγούμενο παράδειγμα).
- Πριν τη συναρμολόγηση, θα πρέπει να προσδιοριστεί ο τρόπος με τον οποίο θα διατάσσονται (layout) στο παράθυρο.
- Υπάρχουν τρία είδη διατάξεων:
  - Διάταξη ροής
  - Διάταξη πλέγματος
  - Διάταξη περιθωρίου

# Διάταξη ροής

- Υλοποιείται μέσω της κλάσης `FlowLayout`.
- Τα συστατικά του παραθύρου διατάσσονται το ένα δίπλα στο άλλο, όσο χωράνε στο πλάτος του παραθύρου, και μετά συνεχίζουν από κάτω.
- Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, πχ

```
setLayout(new FlowLayout());
```

- Και μετά ένα συστατικό `b` μπορεί να προστεθεί με:

```
add(b);
```

# Διάταξη ροής



# Διάταξη πλέγματος

- Υλοποιείται μέσω της κλάσης GridLayout.
- Τα συστατικά του παραθύρου διατάσσονται το ένα μετά το άλλο σε ένα πλέγμα (grid)/πίνακα των διαστάσεων που επιθυμούμε.
- Δεν προσδιορίζουμε σε ποιο κελί του πίνακα θα τοποθετηθεί ένα αντικείμενο.

# Διάταξη πλέγματος

- Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, πχ

**setLayout(new GridLayout(3,2));**

- Αν θέλουμε το πλέγμα να έχει 3 γραμμές και 2 στήλες, και μετά ένα συστατικό μπορεί να προστεθεί με:

**add(b);**

1	2
3	4
5	6

# Διάταξη περιθωρίου

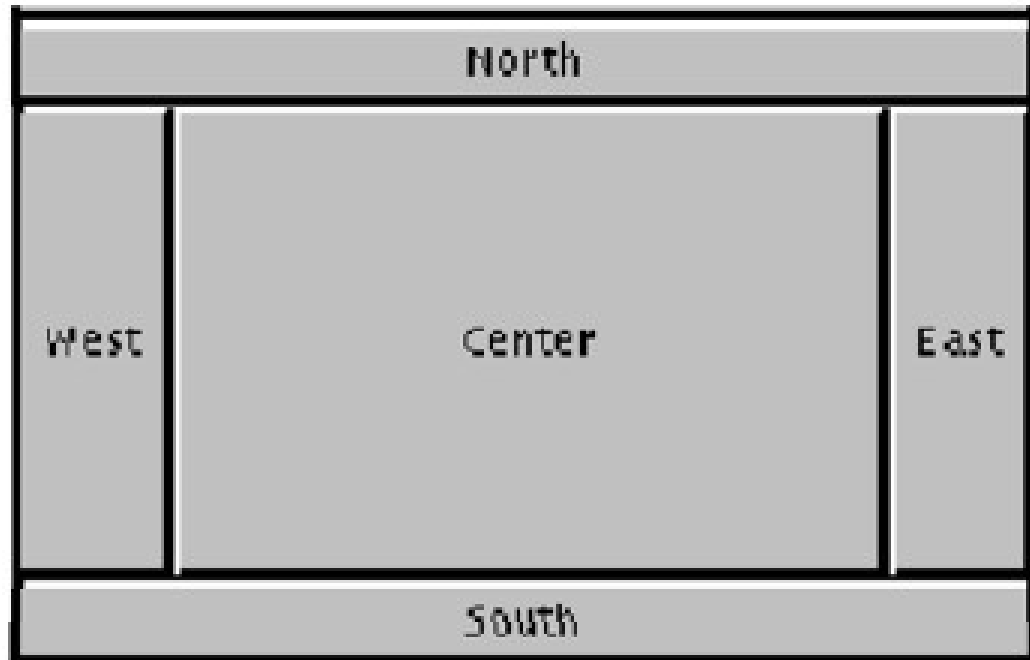
- Υλοποιείται μέσω της κλάσης BorderLayout.
- Τα συστατικά του παραθύρου διατάσσονται σε πέντε δυνατές θέσεις: NORTH, SOUTH, EAST, WEST, CENTER.
- Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, πχ

**setLayout(new BorderLayout());**

- Και μετά ένα συστατικό b μπορεί να προστεθεί με:

**add(b, BorderLayout.NORTH);** (αν επιθυμούμε το b να προστεθεί βόρεια)

# Διάταξη περιθωρίου



# Παράδειγμα

```
import javax.swing.*;
import java.awt.*;

public class YesNo extends JFrame
{
    JButton b1=new JButton("YES");
    JButton b2=new JButton("NO");
    JTextField tf=new JTextField();

    public YesNo()
    {
        super("Yes/No");
```



# Παράδειγμα

```
setSize(100,100);  
setDefaultCloseOperation(EXIT_ON_CLOSE);  
setVisible(true);
```

```
setLayout(new GridLayout(3,1));  
add(tf);  
add(b1);  
add(b2);
```

```
}
```

# Παράδειγμα

```
public static void main(String[] args)
{
    YesNo w=new YesNo();
}
}
```

# Εκτέλεση ενεργειών

- Για να έχει ένα GUI λειτουργική αλληλεπίδραση με τον χρήστη, πρέπει οι ενέργειες που εκτελεί ο χρήστης, πχ κλικ σε ένα κουμπί, να γίνονται αντιληπτές από το σύστημα.
- Με άλλα λόγια, το σύστημα πρέπει να «ακούει» τις ενέργειες του χρήστη.
- Αυτό επιτυγχάνεται επιβάλλοντας στην κλάση του παραθύρου να υλοποιεί τη διεπαφή **ActionListener** (πακέτο **java.awt.event**) ή και άλλες διεπαφές που διαθέτει η Java.
- Οι διεπαφές αυτές προσφέρουν μεθόδους, οι οποίες ενεργοποιούνται κάθε φορά που εκτελείται μια ενέργεια.

# Εκτέλεση ενεργειών

- Με αυτόν τον τρόπο μπορεί το σύστημα να ανταποκριθεί στις ενέργειες του χρήστη, πχ όταν κάνει κλικ σε ένα κουμπί, να εκτελεστεί μια ενέργεια.
- Ειδικότερα, για την εκτέλεση ενεργειών που σχετίζονται με την υλοποίηση της διεπαφής `ActionListener` απαιτούνται τρία βήματα:
  - Υλοποίηση της διεπαφής, δηλ. **implements `ActionListener`**.
  - Προσθήκη της δυνατότητας σε ένα αντικείμενο να «ακούγεται», δηλ. σε ένα κουμπί πρέπει να κάνουμε: **`b.addActionListener(this);`**
  - Υλοποίηση της μεθόδου **`actionPerformed`** στην οποία πρέπει να εντοπίζεται από ποιο συστατικό προήλθε η ενέργεια (με **`getSource`**).

## Παράδειγμα (με εκτέλεση ενεργειών)

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
public class YesNo extends JFrame implements ActionListener  
{  
    JButton b1=new JButton("YES");  
    JButton b2=new JButton("NO");  
    JTextField tf=new JTextField();  
  
    public YesNo()  
    {  
        super("Yes/No");
```

# Παράδειγμα

```
super("Yes/No");  
setSize(100,100);  
setDefaultCloseOperation(EXIT_ON_CLOSE);  
setVisible(true);
```

```
setLayout(new GridLayout(3,1));
```

```
add(tf);
```

```
add(b1);
```

```
b1.addActionListener(this);
```

```
add(b2);
```

```
b2.addActionListener(this);
```

```
}
```

# Παράδειγμα

```
public void actionPerformed(ActionEvent e)  
{  
    if (e.getSource()==b1) tf.setText("Yes");  
    if (e.getSource()==b2) tf.setText("No");  
}
```

```
public static void main(String[] args)  
{  
    YesNo w=new YesNo();  
}
```

```
}
```

# Εκτύπωση/ανάγνωση κειμένου

- Η μέθοδος **setText** χρησιμοποιείται για την εκτύπωση της λέξης "Yes" ή "No" στο πεδίο κειμένου.
- Αντίστοιχα, μπορεί να χρησιμοποιηθεί και για άλλα συστατικά που περιέχουν κείμενο, πχ **JLabel** (ετικέτες).
- Για να διαβάσει κανείς το κείμενο που περιέχει ένα συστατικό, χρησιμοποιείται η μέθοδος **getText**, πχ **String s=tf.getText();**
- Περισσότερες λεπτομέρειες για άλλες κλάσεις και μεθόδους συστατικών στο API του πακέτου **javax.swing**.



Μικροεφαρμογές (applets)

# Μικροεφαρμογές (applets)

- Κατασκευάζονται όπως και οι εφαρμογές
- Δεν έχουν μέθοδο `main` όπως οι εφαρμογές γραμμής εντολών
- Εκτελούνται μόνο μέσω μιας ιστοσελίδας
- Κανόνες ασφαλείας
  - Δε μπορεί να διαβάσει ή να γράψει αρχεία στο σύστημα αρχείων του χρήστη
  - Δε μπορεί να επικοινωνεί με μια περιοχή στο διαδίκτυο εκτός αυτής που εξυπηρετεί την ιστοσελίδα που περιλαμβάνει τη μικροεφαρμογή
  - Δε μπορεί να εκτελεί προγράμματα στο σύστημα του αναγνώστη
  - Δε μπορεί να φορτώνει προγράμματα αποθηκευμένα στο σύστημα του χρήστη όπως εκτελέσιμα προγράμματα και βιβλιοθήκες κοινής χρήσης

# Δομή μικροεφαρμογών

- Πακέτα που χρησιμοποιούνται
  - `import java.applet.*;`
  - `import java.awt.*;`
- Ο ορισμός της κυρίως κλάσης της μικροεφαρμογής γίνεται με επέκταση της κλάσης `Applet`
  - `public class MyApplet extends java.applet.Applet { όλος ο κώδικάς της }`

# Δομή μικροεφαρμογών

- Μέθοδοι που περιέχει αντί για main():
  - **public void init()**  
{ κώδικας που εκτελείται πριν ξεκινήσει }
  - **public void start()**  
{ κώδικας που εκτελείται μετά την αρχικοποίηση ή αν ξεκινήσει πάλι μετά από διακοπή της }
  - **public void stop()**  
{ κώδικας που εκτελείται όταν διακοπεί η εκτέλεσή της }
  - **public void destroy()**  
{ κώδικας που εκτελείται μόλις πριν κλείσουμε τον εξερευνητή ιστοσελίδων αν (**σπανιότατα**) επιθυμούμε χειροκίνητα να απελευθέρωσουμε δεσμευμένη μνήμη }
  - **public void paint(Graphics g)**  
{ εδώ ο κώδικας που τυπώνει / ζωγραφίζει πράγματα στο παράθυρο της μικροεφαρμογής – Η ΠΛΕΟΝ ΣΗΜΑΝΤΙΚΗ ! }

# Παράδειγμα μικροεφαρμογής

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

public class FirstApplet extends Applet {

    public void paint(Graphics g) {
        g.drawString("Hello World", 25, 50);
    }
}
```

- Αποθηκεύουμε το αρχείο αυτό ως FirstApplet.java
- Το μεταγλωττίζουμε: javac FirstApplet.java
- Παράγεται το αρχείο FirstApplet.class

# Εκτέλεση μικροεφαρμογής

- Σε ένα απλό αρχείο κειμένου γράφουμε τις παρακάτω εντολές HTML και αποθηκεύουμε το αρχείο με προέκταση .html (ή .htm)
- Κώδικας HTML (ιστοσελίδα)
  - `<APPLET CODE="FirstApplet.class" WIDTH=600 HEIGHT=100>`  
`</APPLET>`
- Για να «κρεμάσουμε» μια ιστοσελίδα με μικροεφαρμογή σε ένα διακομιστή πρέπει να αντιγράψουμε σε αυτόν:
  - Την ιστοσελίδα μας (αρχείο .html)
  - Τη μεταγλωττισμένη μικροεφαρμογή μας (αρχείο .class) π.χ. FirstApplet.class

# Παραλλαγές μικροεφαρμογής

- Παραλλαγές HTML

```
<APPLET CODE="FirstApplet.class" WIDTH=600 HEIGHT=100>  
<ALIGN=LEFT VSPACE=50 HSPACE=10>  
</APPLET>
```

- **ALIGN**: Στοιχίση μικροεφαρμογής μέσα στην ιστοσελίδα π.χ. αριστερά
- **VSPACE** και **HSPACE**: Αριθμός pixels που να απέχει η μικροεφαρμογή καθέτως ή οριζοντίως από το κείμενο που μπορεί να την περιβάλλει σε μια ιστοσελίδα