



# Αντικειμενοστρεφής Προγραμματισμός

Ενότητα 9: Γραφικές Διεπαφές Χρήστη

Γρηγόρης Τσουμάκας, Επικ. Καθηγητής  
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ  
ΑΚΑΔΗΜΑΙΚΑ  
ΜΑΘΗΜΑΤΑ



# Γραφικές Διεπαφές Χρήστη



**Τα παραδείγματα κώδικα που χρησιμοποιούνται σε κάποιες από τις ακόλουθες διαφάνειες μπορούν να βρεθούν στον παρακάτω σύνδεσμο:**

# Βασικές Αρχές

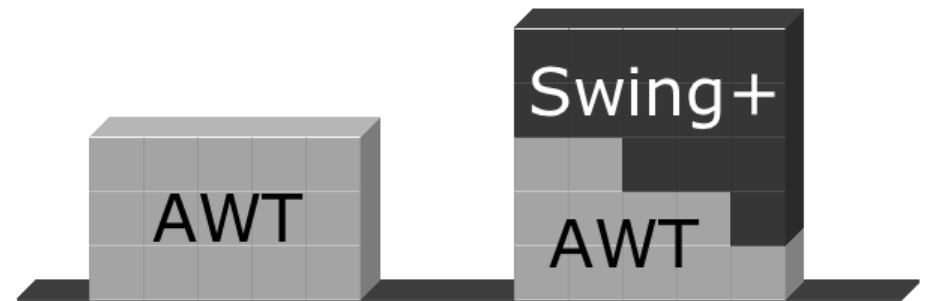
- Συστατικά.
  - Τα βασικά συστατικά στοιχεία μιας διεπαφής, όπως παράθυρα, κουμπιά, μενού, κ.α.
- Διαρρύθμιση.
  - Χωροθέτηση των συστατικών με τη βοήθεια διαχειριστών διαρρύθμισης.
- Γεγονότα.
  - Αλληλεπίδραση του χρήστη με την εφαρμογή, όπως πάτημα κουμπιού, επιλογή μενού, κ.α.



# Βιβλιοθήκες AWT και Swing

- AWT.
  - 1<sup>ης</sup> γενιάς.
  - Button, Frame, ...
  - Heavyweight.
- Swing.
  - 2<sup>ης</sup> γενιάς.
  - JButton, JFrame, ...
  - Lightweight.

- JavaFX.
  - 3<sup>ης</sup> γενιάς.
  - Πλούσιες διεπαφές τύπου Flash / Flex.
  - Lightweight.



# Συστατικά (Components)

- Συστατικά.
  - Τα βασικά συστατικά στοιχεία μιας διεπαφής, όπως παράθυρα, κουμπιά, μενού, κ.α.
  - Κλάσεις που κληρονομούν την *JComponent*.
- Έτοιμα συστατικά.
  - Μεγάλος αριθμός έτοιμων συστατικών, τα οποία συνδυάζονται για την κατασκευή διεπαφών.
- Δημιουργία συστατικών.
  - Μπορούν να κατασκευαστούν νέα συστατικά με βάση τα υπάρχοντα (κληρονομικότητα).





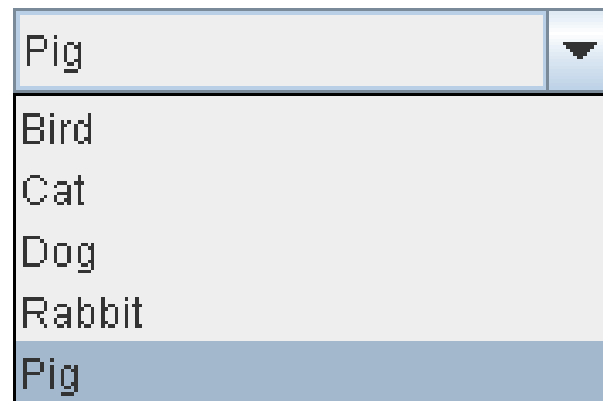
# Συστατικά (1/2)



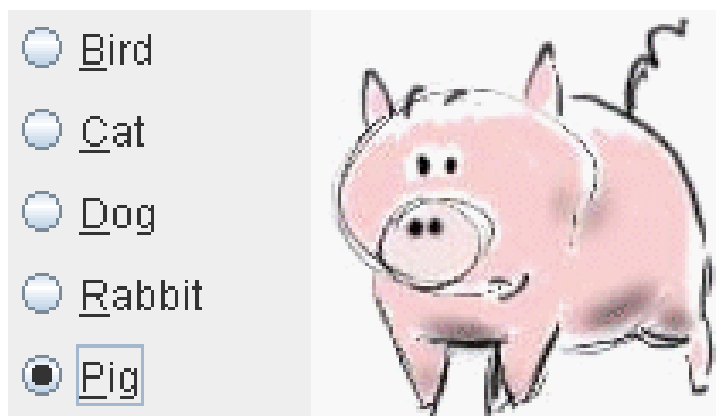
JButton



JCheckBox



JComboBox

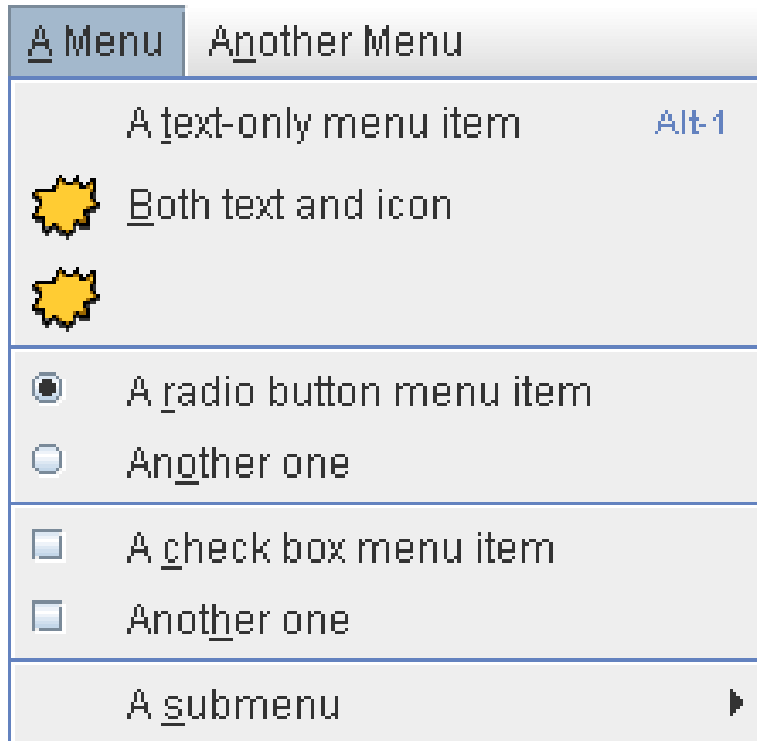


JRadioButton

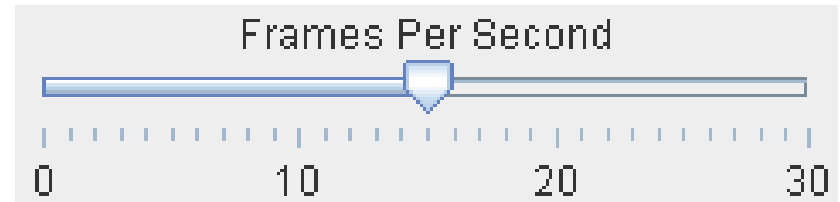


JList

# Συστατικά (2/2)



JMenu



JSlider



JPasswordField



JTextField



JSpinner

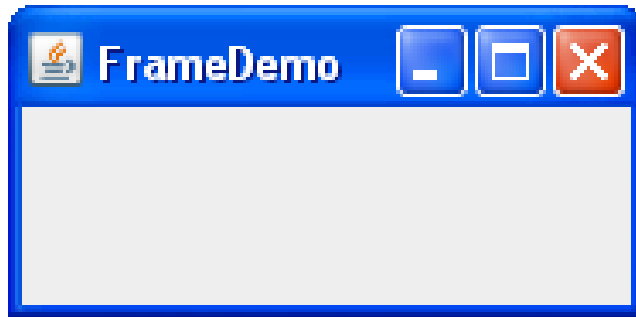


# Δοχεία (Containers)

- Δοχεία.
  - Συστατικά το οποία μπορούν να περιέχουν άλλα συστατικά, εκτός από τα λεγόμενα δοχεία «ρίζας».
  - Προσθήκη ενός συστατικού με τη μέθοδο *add*.
- Ιεραρχία συμπερίληψης.
  - Δέντρο συστατικών, το οποίο ξεκινάει από ένα δοχείο «ρίζας»: *JFrame*, *JDialog* και *JApplet*.
  - Κάθε συστατικό της ιεραρχίας, πλην της ρίζας, περιέχεται σε ένα και μόνο άλλο συστατικό.



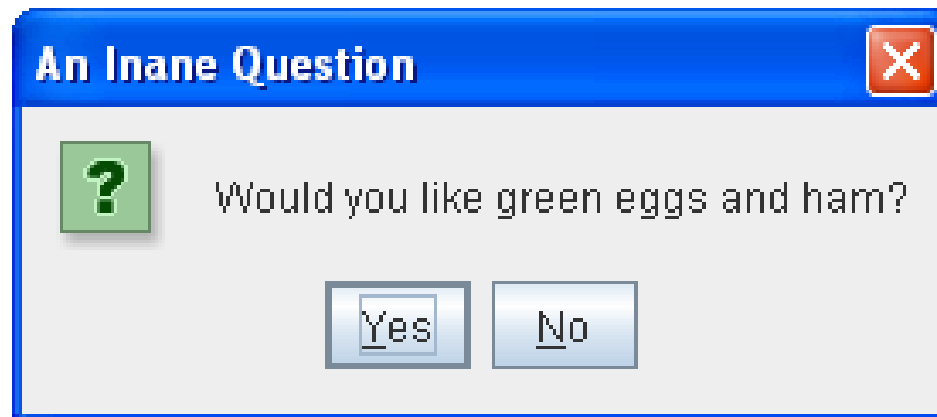
# Δοχεία «Ρίζας»



JFrame



JApplet



JDialog



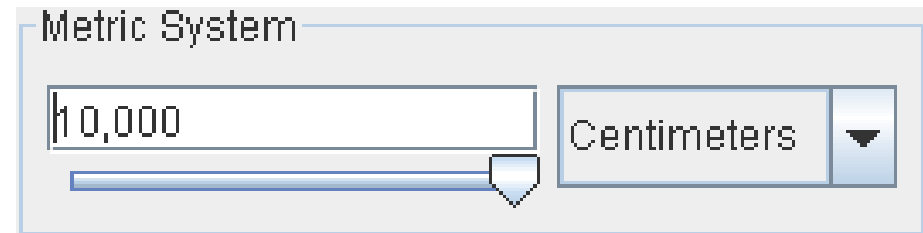
# Άλλα Δοχεία



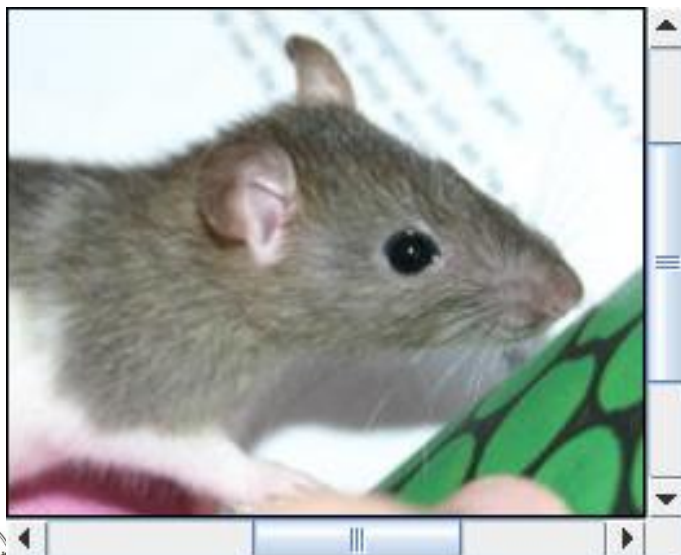
JTabbedPane



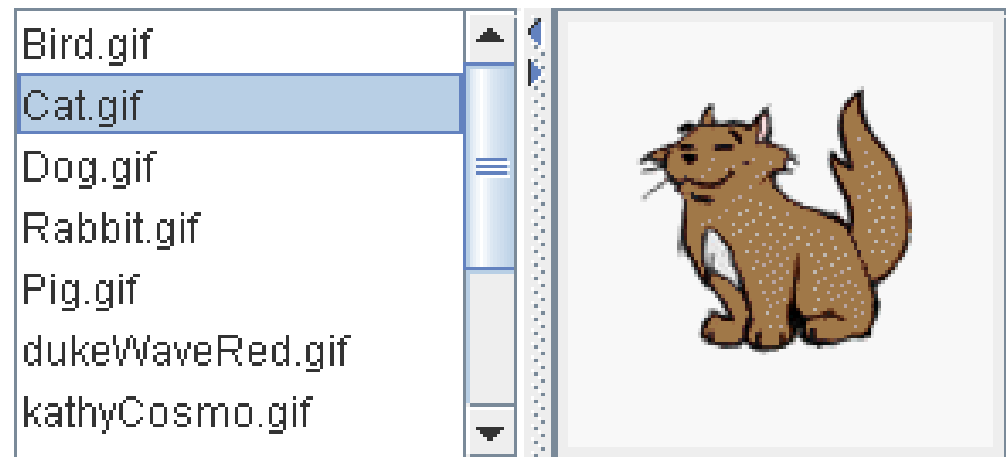
JToolBar



JPanel

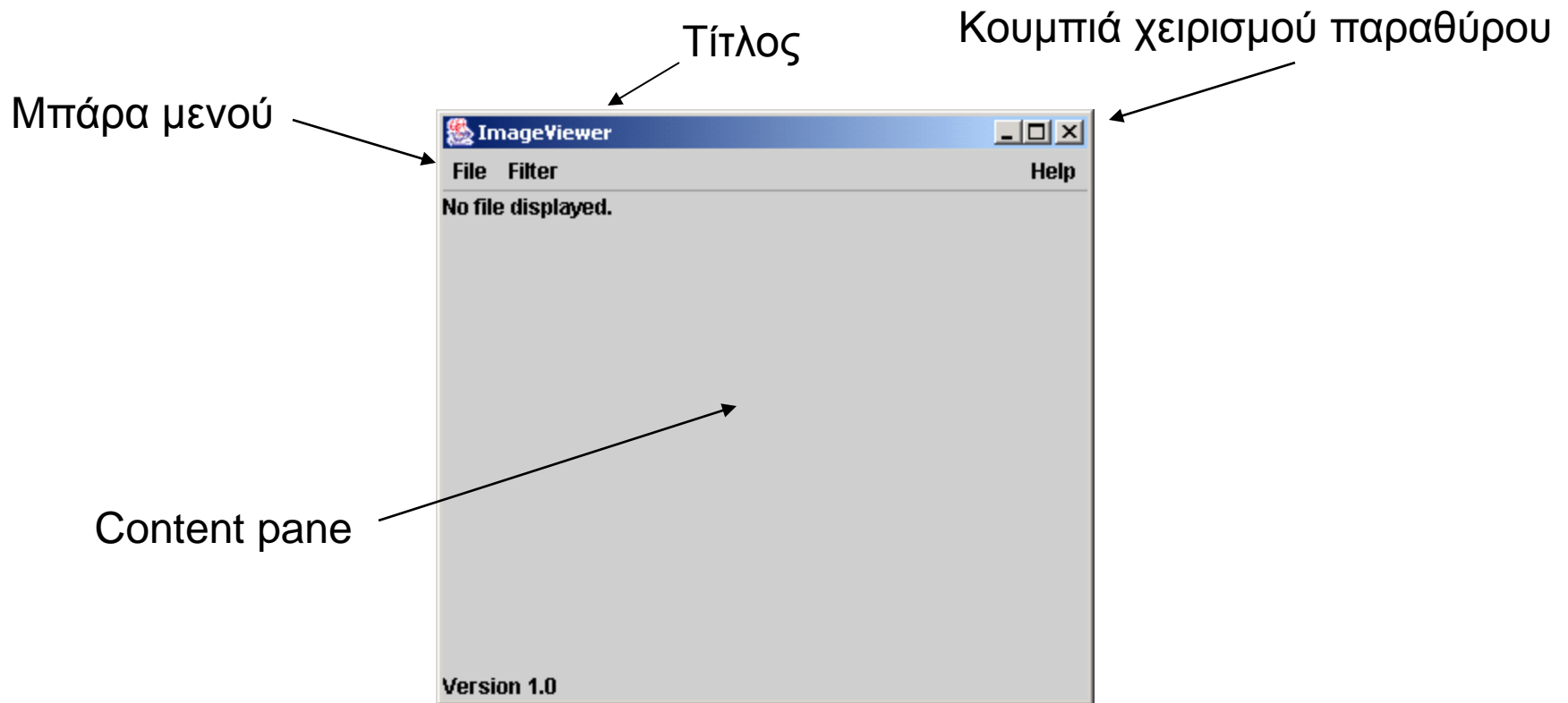


JScrollPane



JSplitPane

# Στοιχεία ενός Παραθύρου JFrame



# Δημιουργία Παραθύρου: Τρόπος A

```
public class FrameA {  
    private JFrame frame;  
  
    public FrameA() { makeFrame(); }  
    private void makeFrame() {  
        frame = new JFrame("Frame A");  
        // ρυθμίσεις παραθύρου  
        frame.setVisible(true);  
    }  
}
```

Κλάση, η οποία περιλαμβάνει ένα αντικείμενο της κλάσης JFrame



# Δημιουργία Παραθύρου: Τρόπος Β

```
public class FrameB extends JFrame {  
    public FrameB() { makeFrame(); }  
    private void makeFrame() {  
        setTitle("Frame B");  
        // άλλες ρυθμίσεις παραθύρου  
        setVisible(true);  
    }  
}
```

Κλάση, η οποία επεκτείνει  
την κλάση JFrame





# JFrame (1/3)

- setTitle(String title).
  - Κείμενο επάνω στην μπάρα του παραθύρου.
  - Μπορεί να δοθεί και στον κατασκευαστή.
- Μέγεθος παραθύρου.
  - Διαστάσεις σε pixel με *setSize(int width, int height)*.
  - Προσαρμογή στα περιεχόμενα με *pack()*.
- setVisible(boolean b).
  - Για να εμφανιστεί το παράθυρο στην οθόνη.
  - Αρχικά είναι αόρατο, ώστε να μην φανεί στον χρήστη η διαδικασία προσθήκης συστατικών.



# JFrame (2/3)

- Τερματισμός με κλείσιμο παραθύρου.
  - Αυτό θέλουμε συνήθως να συμβεί όταν κλείνουμε το βασικό παράθυρο μιας εφαρμογής.
    - `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
  - Η προκαθορισμένη συμπεριφορά είναι να κρύβεται, δηλαδή καλείται η `setVisible(false)`.
- `setResizable(boolean b)`.
  - Καθορίζει αν θα επιτρέπεται η αλλαγή μεγέθους του παραθύρου (προκαθορισμένη τιμή `true`).



# JFrame (3/3)

- `setLocation(int x, int y)`.
  - Συντεταγμένες επάνω αριστερά κορυφής.
- Κεντράρισμα στην οθόνη.
  1. `setLocationRelativeTo(null)`;
  2. `Toolkit t = Toolkit.getDefaultToolkit();`  
`Dimension d = t.getScreenSize();`  
`int x = (d.width-frame.getWidth())/2;`  
`int y = (d.height-frame.getHeight())/2;`  
`frame.setLocation(x, y);`

frames



# Η Κλάση JLabel

- Λειτουργία.
  - Απλά περιλαμβάνει κείμενο ή/και εικόνες.
- Κατασκευαστές.
  - `JLabel lab1 = new JLabel();`
  - `JLabel lab2 = new JLabel("Ε2");`
- Ανάθεση συμβολοσειράς.
  - `setText(String text).`



# Διαρρύθμιση Δοχείου

- Απόλυτη διαρρύθμιση.
  - Τα συστατικά τοποθετούνται σε συγκεκριμένες συντεταγμένες και έχουν συγκεκριμένο μέγεθος.
- Διαχειριστής διαρρύθμισης (layout manager).
  - Υλοποιεί τη διεπαφή *LayoutManager*.
  - Καθορίζει τη θέση (και το μέγεθος) των συστατικών μέσα σε ένα δοχείο.
  - Ανατίθεται σε ένα δοχείο καλώντας τη μέθοδο *setLayout* του δοχείου.



# Flow Layout

- Τοποθετεί τα συστατικά σε μια γραμμή, χωρίς να αλλάζει το μέγεθός τους.
  - Αν ο οριζόντιος χώρος δεν επαρκεί, χρησιμοποιεί περισσότερες γραμμές, ενώ αν είναι μεγαλύτερος η προκαθορισμένη στοίχιση είναι στο κέντρο.
- Αποτελεί την προκαθορισμένη διαρρύθμιση στο δοχείο *Jpanel*.

FlowLayout.java



# JButton (1/3)

- Λειτουργία.
  - Κουμπί, το πάτημα του οποίου προκαλεί ένα γεγονός τύπου *ActionEvent*.
- Κατασκευαστές.
  - `JButton b1 = new JButton();`
  - `JButton lab2 = new JButton("Κουμπί 2");`
- Ανάθεση συμβολοσειράς.
  - `setText(String text).`

Buttons1.java



# Διαχείριση Γεγονότων

- Γεγονότα (events).
  - Αντικείμενα με πληροφορίες που αφορούν την αλληλεπίδραση του χρήστη με κάποιο συστατικό.
  - Π.χ. πάτημα κουμπιού (ποντίκι ή πληκτρολόγιο), κίνηση ποντικιού επάνω από ένα συστατικό.
- Διαχείριση γεγονότων (event handling).
  - Σύνδεση των παραγόμενων γεγονότων με αντικείμενα που ακούνε για γεγονότα (listeners) και παρέχουν κώδικα διαχείρισής τους.
  - Προγραμματισμός καθοδηγούμενος από γεγονότα.





# JButton (2/3)

- Προσθήκη διαχειριστή για το *ActionEvent*.
  - `addActionListener(ActionListener a)`
- Η διεπαφή *ActionListener* ορίζει τη μέθοδο.
  - `public void actionPerformed(ActionEvent e) {`  
    // κώδικας απόκρισης στο γεγονός  
    }
- Προσθήκη διαχειριστών.
  - Ένας διαχειριστής για πολλά γεγονότα.
  - Ξεχωριστός διαχειριστής για κάθε γεγονός.



# JButton (3/3)

- Εντολή δράσης (action command).
  - Συμβολοσειρά που περιέχεται στο *ActionEvent* και βοηθά στην αναγνώριση της προέλευσης του.
  - Αν δεν οριστεί, είναι το κείμενο επάνω στο κουμπί.
- Ορισμός εντολής δράσης σε JButton.
  - `setActionCommand(String command)`.
- Λήψη εντολής δράσης από *ActionEvent*.
  - `String getActionCommand()`.

Buttons2.java



# Φωλιασμένες Κλάσεις (1/2)

```
public class Enclosing {  
    ...  
    private class Nested {  
        ...  
    }  
}
```

- Αποτελούν μέλη της κλάσης που τις περικλείει.
  - Πρόσβαση: private, public, protected, default.



# Φωλιασμένες Κλάσεις (2/2)

- Στατικές κλάσεις.
  - `Enclosing.Nested sn;`  
`sn = new Enclosing.Nested();`
- Μη-στατικές κλάσεις (εσωτερικές).
  - Έχουν πρόσβαση στα υπόλοιπα μέλη των κλάσεων που τις περικλείουν.
  - `Enclosing e = new Enclosing();`  
`Enclosing.Nested n = e.new Nested();`

Enclosing.java, NestedClasses.java, Buttons3.java



# Ιδιαίτερες Εσωτερικές Κλάσεις

- Τοπικές κλάσεις.
  - Εσωτερικές κλάσεις που δηλώνονται εντός ενός block κώδικα χωρίς προσδιοριστή πρόσβασης.
- Ανώνυμες εσωτερικές κλάσεις.
  - Τοπικές κλάσεις χωρίς όνομα που υλοποιούν ένα interface ή επεκτείνουν κάποια κλάση.
  - Ορισμός ανώνυμης κλάσης και ταυτόχρονη δημιουργία ανώνυμου αντικειμένου με *new*.

Enclosing.java, Buttons4.java



# Εικονίδια

- Εικονίδια.
  - Πολλά συστατικά (π.χ. *JLabel*, *JButton*, κ.α.) μπορούν να διακοσμηθούν με εικονίδια, καλώντας τη μέθοδο `setIcon(Icon icon)`.
- ImageIcon.
  - Κλάση που υλοποιεί τη διεπαφή *Icon* και μπορεί να εμφανίσει εικόνες τύπου GIF, JPEG και PNG.
  - `ImageIcon ico1 = new ImageIcon("java.jpg");`
  - `Url url = getClass.getResource("java.jpg");  
ImageIcon ico2 = new ImageIcon(url);`

IconUsage.java



# Border Layout

- Τοποθετεί τα συστατικά σε μια από 5 περιοχές.
  - PAGE\_START, PAGE\_END, LINE\_START, CENTER, LINE\_END (στατικές σταθερές της BorderLayout).
  - Η περιοχή δίνεται ως 2<sup>η</sup> παρ/τρος στη μέθοδο *add*.
  - Τα συστατικά «πιάνουν» όλο το χώρο της περιοχής.
- Κατά την αλλαγή μεγέθους.
  - Επηρεάζεται αρχικά μόνο η περιοχή *CENTER*.
- Αποτελεί προκαθορισμένη διάταξη στη *JFrame*.

LayoutBorder.java



# Κάποιες Μέθοδοι της JComponent

- (Απ)ενεργοποίηση συστατικού.
  - `setEnabled(false)` ;
- Γραμματοσειρά συστατικού.
  - `setFont(Font.decode("Verdana-bold-20"))` ;
- (Α)διαφάνεια συστατικού.
  - `setOpaque(true)` ;
- Χρώμα πρώτου πλάνου και υποβάθρου.
  - `setForeground(Color c)` ; // `Color.WHITE`
  - `setBackground(Color c)` ; // `Color.BLACK`





# Απόλυτη Διαρρύθμιση

- Καθορισμός.
  - `setLayout(null)` // για το container.
  - `setBounds(x, y, width, height)` // για το συστατικό.
- Μειονέκτημα η μη προσαρμογή σε αλλαγές,
  - Που οφείλονται σε διαφορετικά look-and-feel.
  - Μεγέθους του δοχείου και γραμματοσειρών.
  - Τοπικότητες (π.χ. ανάγνωση από δεξιά).
- Μπορεί να χρειαστεί σε κάποιες περιπτώσεις
  - Εσωτερικά παράθυρα, τα οποία διατάσσει ο χρήστης.

LayoutNull.java



# Grid Layout

- Τα συστατικά χωροθετούνται ως εξής.
  - Σε ένα πλέγμα με συγκεκριμένο αριθμό γραμμών και στηλών, όπου κάθε κελί έχει το ίδιο μέγεθος.
  - Προστίθενται ανά γραμμή από πάνω προς τα κάτω και από αριστερά προς τα δεξιά.
  - Καταλαμβάνουν όλο το χώρο του κελιού.
- Κατασκευαστές.
  - `GridLayout(int rows, int columns)`.
  - `GridLayout(int rows, int columns, int hgap, int vgap)`.

LayoutGrid.java



# JPanel

- JPanel.
  - Δοχείο για ομαδοποίηση ενός αριθμού συστατικών.
  - Προκαθορισμένος διαχειριστής διαρρύθμισης είναι ο Flow Layout.
- Προσθήκη περιγράμματος.
  - Μέσω της μεθόδου *setBorder(Border border)* της κλάσης *JComponent* και χρήση στατικών μεθόδων της κλάσης *BorderFactory*.



# JTextField

- JTextField.
  - Συστατικό που εμφανίζει κείμενο στην οθόνη και επιτρέπει την αλλαγή αυτού του κειμένου.
- Κατασκευαστές.
  - JTextField().
  - JTextField(String) // προκαθορισμένο περιεχόμενο.
  - JTextField(int) // επιθυμητό μήκος σε χαρακτήρες.
  - JTextField(String, int).
- Για εμφάνιση του κειμένου μόνο.
  - `setEditable(false);`

TextField1.java, TextField2.java



# Γεγονότα JTextField

- `ActionEvent` (πάτημα `enter`).
  - Χειρισμός όπως στην περίπτωση του `JButton`.
- Γεγονός `FocusEvent`.
  - Μέθοδοι `focusGained`, `focusLost` της διεπαφής `FocusListener`.
- Γεγονός `DocumentEvent`.
  - Μέθοδοι `insertUpdate`, `removeUpdate`, `changedUpdate` της διεπαφής `DocumentListener`.
  - Προσθήκη χειριστή στο αντικείμενο `Document` ενός `JTextField`.

TextField3.java, TextField4.java



# JCheckBox

- JCheckBox.
  - Μια επιλογή που μπορεί να είναι αληθής ή ψευδής.
- Γεγονός ItemEvent.
  - Συμβαίνει με την ενεργοποίηση/απενεργοποίηση.
  - Μέθοδος *itemStateChanged* της διεπαφής *ItemListener*.
- Το γεγονός παρέχει πληροφορίες κατάστασης.
  - Η μέθοδος *e.getStateChange()* επιστρέφει *ItemEvent.SELECTED* ή *ItemEvent.DESELECTED*.

CheckBoxes.java



# JRadioButton και ButtonGroup

- Radio buttons.
  - Ομάδα επιλογών, από τις οποίες, *κατά σύμβαση*, μόνο μία μπορεί να είναι αληθής ανά πάσα στιγμή.
- JRadioButton.
  - Μία από τις επιλογές.
- ButtonGroup.
  - Ομάδα επιλογών (υλοποιεί τη σύμβαση).
- Γεγονότα.
  - Όπως στα JCheckBox, JButton.

RadioButtons.java



# Μενού και Γραμμή Μενού

- JMenuItem.
  - Αντιπροσωπεύει μια γραμμή από μενού επιλογών.
  - Προστίθεται σε ένα δοχείο «ρίζα» με τη μέθοδο `setJMenuBar(JMenuBar menuBar)`.
  - Προσθήκη ενός μενού στη γραμμή μενού με τη μέθοδο `add(JMenu menu)`.
- JMenu.
  - Αντιπροσωπεύει ένα μενού με επιλογές.
  - Εμφανίζεται στη γραμμή μενού μιας εφαρμογής.
  - Κατασκευαστής `JMenu(String text)`.





# Το Βασικό Αντικείμενο ενός Μενού

- JMenuItem.
  - Αντιστοιχεί σε μια επιλογή του μενού.
  - Προσθήκη ενός αντικειμένου μενού σε ένα μενού με τη μέθοδο *add(JMenuItem menuItem)*.
- Κατασκευαστές.
  - JMenuItem(String name);
  - JMenuItem(String name, Icon icon);
  - JMenuItem(Icon icon);
- Γεγονότα.
  - Όπως στο *JButton*.

Menu1.java



# Άλλα Αντικείμενα Μενού

- Γραμμές διαχωρισμού.
  - `addSeparator()`;
- Προσθήκη *JMenu* σε ένα *JMenu*.
  - Δημιουργία υπομενού.
  - Το *JMenu* επεκτείνει την *JMenuItem*, οπότε μπορεί να κληθεί η `add(JMenuItem menuItem)` με ένα μενού ως παράμετρο (κληρονομικότητα).
- Άλλα αντικείμενα.
  - `JRadioButtonMenuItem`, `JCheckBoxMenuItem`.

Menu2.java



# Χρήση Πληκτρολογίου

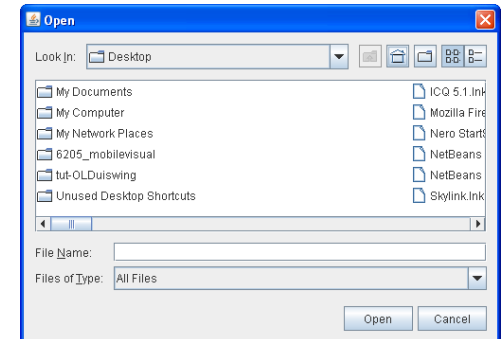
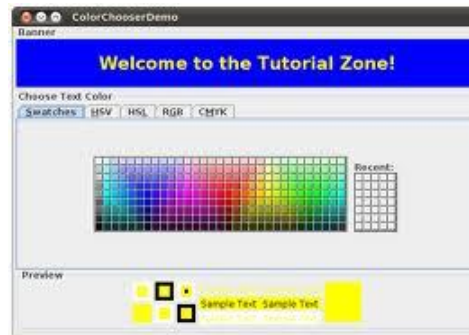
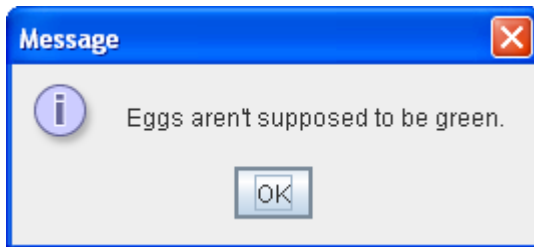
- Μνημονικά (Mnemonics).
  - Επιλογή αντικειμένου του μενού με συνδυασμό ενός ειδικού πλήκτρου (Alt) και ενός κανονικού.
  - Θα πρέπει το αντικείμενο να φαίνεται στην οθόνη.
  - Εφαρμόζεται και σε άλλα συστατικά (π.χ. JCheckBox).
  - `setMnemonic(int i); // KeyEvent.VK_N`
- Επιταχυντές (Accelerators).
  - Επιλογή ενός αντικειμένου (εκτός από υπομενού) με συνδυασμό πλήκτρων, χωρίς απαίτηση να φαίνεται.
  - `setAccelerator(KeyStroke ks);`

Menu3.java



# Παράθυρα Διαλόγου (1/3)

- Υποπαράθυρα, εξαρτώμενα από το παράθυρο (διαλόγου ή κανονικό) που τα δημιούργησε.
  - Καταστρέφονται, ελαχιστοποιούνται και επανέρχονται μαζί με τον δημιουργό τους.
- Συνήθως χρησιμοποιούνται για
  - Εμφάνιση μηνύματος λάθους ή άλλου μηνύματος.
  - Άνοιγμα αρχείου, επιλογή χρώματος, εκτύπωση.
  - Οργάνωση της αλληλεπίδρασης με τον χρήστη σε επί μέρους παράθυρα (π.χ. μενού με επιλογές για καταχώρηση & εύρεση πελάτη, λίστα πελατών, κτλ).



# Παράθυρα Διαλόγου (2/3)

- Συνήθως είναι *modal*.
  - Μπλοκάρουν την αλληλεπίδραση με το παράθυρο που τα δημιούργησε.
  - Διακόπτεται η ροή εκτέλεσης στο παράθυρο που το δημιούργησε.
- Κατασκευή διαλόγων με την κλάση *JDialog*.
  - Καθορισμός του «γονέα» του διαλόγου.
  - Καθορισμός του αν θα είναι *modal* ή όχι.

Master1.java, Master2.java







# Παράθυρα Διαλόγου (3/3)

- JFileChooser για επιλογή αρχείου.
  - showOpenDialog(), showSaveDialog()
  - Μη στατικές μέθοδοι.
- JColorChooser για επιλογή χρώματος.
  - showDialog(Component parent, String title, Color i);
  - Στατική μέθοδος που δείχνει το παράθυρο και επιστρέφει το επιλεγμένο χρώμα ή null.

ColorChooser.java



# JOptionPane

- `JOptionPane.showMessageDialog`.    
  - Απλό πληροφοριακό μήνυμα με πολλές παραλλαγές.
- `JOptionPane.showInputDialog`.
  - `showInputDialog("Δώσε το όνομα σου:");`
  - Επιστρέφει το αποτέλεσμα ως `String`.
- `JOptionPane.showOptionDialog`.
  - Διάλογος με πολλές ρυθμίσεις.
- `JOptionPane.showConfirmDialog`.
  - Ρωτάει το χρήστη μια ερώτηση.

`MessageDialog.java, Dialogs.java`



# Διαχείριση Γεγονότων Ποντικιού (1/2)

- interface `MouseListener`.
  - `public void mousePressed(MouseEvent e)`.
  - `public void mouseReleased(MouseEvent e)`.
  - `public void mouseClicked(MouseEvent e)`.
  - `public void mouseEntered(MouseEvent e)`.
  - `public void mouseExited(MouseEvent e)`.
- interface `MouseMotionListener`.
  - `public void mouseDragged(MouseEvent e)`.
  - `public void mouseMoved(MouseEvent e)`.





# Διαχείριση Γεγονότων Ποντικιού (2/2)

- interface `MouseListener`.
  - Συνδυασμός `MouseListener` & `MouseMotionListener`.
- interface `MouseWheelListener`.
  - `public void mouseWheelMoved(MouseWheelEvent e)`.
- abstract class `MouseAdapter`.
  - Παρέχει κενές υλοποιήσεις όλων των παραπάνω διεπαφών διαχείρισης γεγονότων ποντικιού, προκειμένου να διευκολύνει τη διαχείριση ορισμένων μόνο γεγονότων.



# Πληροφορίες MouseEvent

- getX(), getY().
  - Τοποθεσία γεγονότος σε σχέση με το συστατικό.
- getButton().
  - Ποιο κουμπί πατήθηκε.
- getClickCount().
  - Αριθμός των κλικ (π.χ. μονό, διπλό, τριπλό).
- getComponent().
  - Συστατικό στο οποίο έγινε το γεγονός.

MouseEvent.java, ConstrainedResize.java



# Αναδυόμενο Μενού - JPopupMenu

- Παράδειγμα constructor.
  - `JPopupMenu pop = new JPopupMenu();`
- Γεγονότα ποντικιού.
  - Χειρισμός τόσο του *mousePressed* όσο και του *mouseReleased* επειδή το ακριβές γεγονός "ανάδυσης" εξαρτάται από την πλατφόρμα.
  - Ελέγχος για ανάδυση με μέθοδο *isPopupTrigger()* του αντικειμένου *MouseEvent*.
- Εμφάνιση μενού με *show(Component, int, int)*.

PopupMenu.java



# Εποπτική Σχεδίαση Διεπαφής

- Χρήση NetBeans.
  - New File -> Swing GUI Forms -> JFrame Form.
  - New File -> Swing GUI Forms -> JDialog Form.
- Πλεονεκτήματα.
  - Γρηγορότερη ανάπτυξη της διεπαφής.





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ  
ΑΚΑΔΗΜΑΙΚΑ  
ΜΑΘΗΜΑΤΑ



# Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας  
Θεσσαλονίκη, Εαρινό Εξάμηνο 2013-2014



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ