

Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

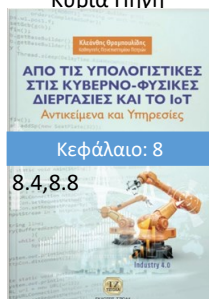
Διαχείριση Συμβάντων Event Handling

Java

High-level programming
language



Κύρια Πινή

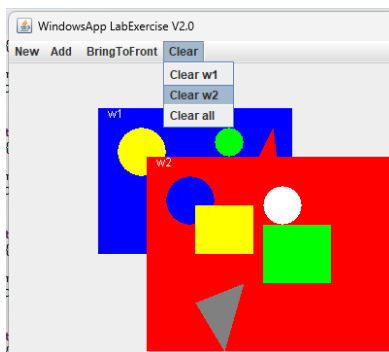


Kleanthis Thramboulidis

Prof. of Software and System Engineering

University of Patras

<https://sites.google.com/site/thramboulidiskleanthis/>



Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

Wikipedia

Designed by: James Gosling

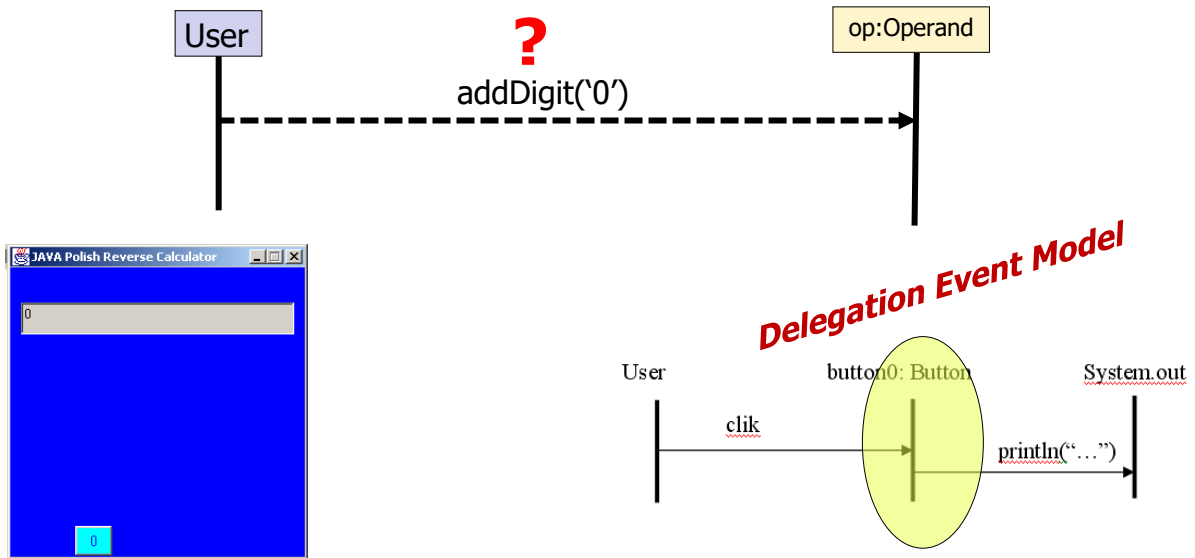
First appeared: May 23, 1995; 27 years ago

Paradigm: Multi-paradigm: generic, object-oriented (class-based), functional, imperative, reflective, concurrent

Οργάνωση Διάλεξης

- **Βασικές έννοιες**
 - Delegation Event Model
 - Event Handling στο Goody's
- Event Handling in Java Gui
 - ActionListener Interface
 - Design For Flexibility and Maintainability
- WindowsApp με Graphics
- RpnCalculator εναλλακτικές υλοποιήσεις
 - Εσωτερική (inner) κλάση
 - Ανώνυμη (anonymous) κλάση
 - έκφραση Lambda (lambda expression)

Το πρόβλημα ... και η λύση



© 2023 Κλεάνθης Θραμπουλίδης

Event Handling

Διαφάνεια 3

Event Handling

- Είναι ο μηχανισμός ο οποίος μας επιτρέπει να ορίσουμε την απόκριση του συστήματος σε ένα συμβάν (event).
 - Ένα **event δημιουργείται** όταν ο χρήστης επιδράσει σε ένα στοιχείο της γραφικής διεπαφής.
 - Ο κώδικα που θα εκτελεστεί σε απόκριση ενός συμβάντος είναι γνωστός ως διαχειριστής συμβάντος (**event handler**).
- Η Java χρησιμοποιεί το *Delegation Event Model* για να διαχειριστεί συμβάντα στις γραφικές διεπαφές.
 - μας επιτρέπει π.χ. να ορίσουμε την συμπεριφορά που θέλουμε να έχει η εφαρμογή μας στο συμβάν "press and release" ενός στιγμιότυπου της Button στην γραφική διεπαφή της εφαρμογής.
- Βασικό Πλεονέκτημα
 - διαχωρίζει πλήρως **application logic** από **interface logic**

© 2023 Κλεάνθης Θραμπουλίδης

Event Handling

Διαφάνεια 4

Βασικά συνθετικά του event model

■ Event

- Είναι ένα αντικείμενο που περιγράφει αλλαγή κατάστασης του αντικειμένου που το δημιουργεί (event source).
- Δημιουργείται ως αποτέλεσμα της επίδρασης του χρήστη πάνω σε ένα συνθετικό της γραφικής διεπαφής.
- Περιέχει πληροφορία για το συμβάν η οποία είναι χρήσιμη για την απόκριση του συστήματος.

■ Event Source (object that "fires" events)

- Το αντικείμενο που δημιουργεί το event. Δημιουργεί το event όταν η κατάσταση του αλλάξει.
- Ένα σύνολο από κλάσεις της βασικής βιβλιοθήκης είναι Event Sources, π.χ. Button, JButton,

■ Event Listener (object interested in events)

- Γνωστός και ως **Event Handler**.
- Είναι το αντικείμενο στο οποίο έχει ανατεθεί η αρμοδιότητα της απόκρισης σε συμβάν.
- Όταν ενημερωθεί για το συμβάν (λάβει το event) ενεργοποιείται για να υλοποιήσει την απόκριση του συστήματος στο συμβάν.

Μηχανισμός Event Handling – Βασική αρχή



Ο Νικολάου μόλις έδωσε την παραγγελία του στην Μαίρη. Για να μην περιμένει όρθιος μέχρι αυτή να ετοιμαστεί ζητά από την Μαίρη να τον ειδοποιήσει. Απαραίτητη προϋπόθεση είναι:

- η γνωστοποίηση του ονόματος του (της αναφοράς του) στην Μαίρη και
- η γνώση που θα πρέπει να έχει ο Νικολάου για το πως θα αντιδράσει όταν η Μαίρη του αποστείλει το μήνυμα «Νικολάου έτοιμος».

Ο Νικολάου κάθεται σε ένα τραπέζι και αναμένει την ειδοποίηση της Μαίρης. Όταν η παραγγελία του Νικολάου ετοιμασθεί (συμβεί το event για το οποίο έκανε register ο Νικολάου) η Μαίρη δημιουργεί το μήνυμα «παραγγελία έτοιμη» και το αποστέλλει στον Νικολάου.

Ο Νικολάου **πρέπει να γνωρίζει πως αντιδρά στο event** «παραγγελία έτοιμη».

Οργάνωση Διάλεξης

- Βασικές έννοιες
 - Delegation Event Model
 - Event Handling στο Goody's
- **Event Handling in Java Gui**
 - ActionListener Interface
 - Design For Flexibility and Maintainability
- WindowsApp με Graphics
- RpnCalculator εναλλακτικές υλοποιήσεις
 - Εσωτερική (inner) κλάση
 - Ανώνυμη (anonymous) κλάση
 - έκφραση Lambda (lambda expression)

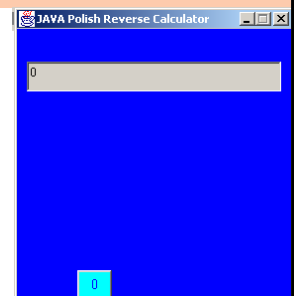
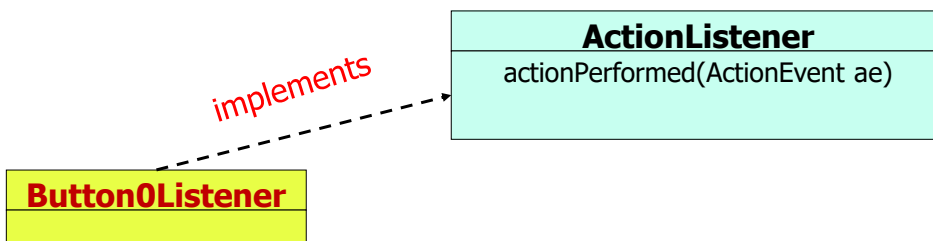
Event Handling in Java (1/2)

Στην τεκμηρίωση της κλάσης Button αναφέρονται τα παρακάτω:

Εάν μια εφαρμογή θέλει να εκτελέσει κάποιες ενέργειες όταν θα πατηθεί και αφεθεί ένα στιγμιότυπο της κλάσης Button **θα πρέπει** :

A) να ορίσει μια κλάση που θα κάνει implements το interface ActionListener

```
class Button0Listener implements ActionListener{
```



Event Handling in Java (2/2)

Β) να δημιουργήσει ένα στιγμιότυπο της κλάσης αυτής

Γ) να κάνει register το στιγμιότυπο της Button0Listener στο στιγμιότυπο της Button. Να ενημερώσει δηλαδή το στιγμιότυπο της Button για το στιγμιότυπο της Button0Listener.

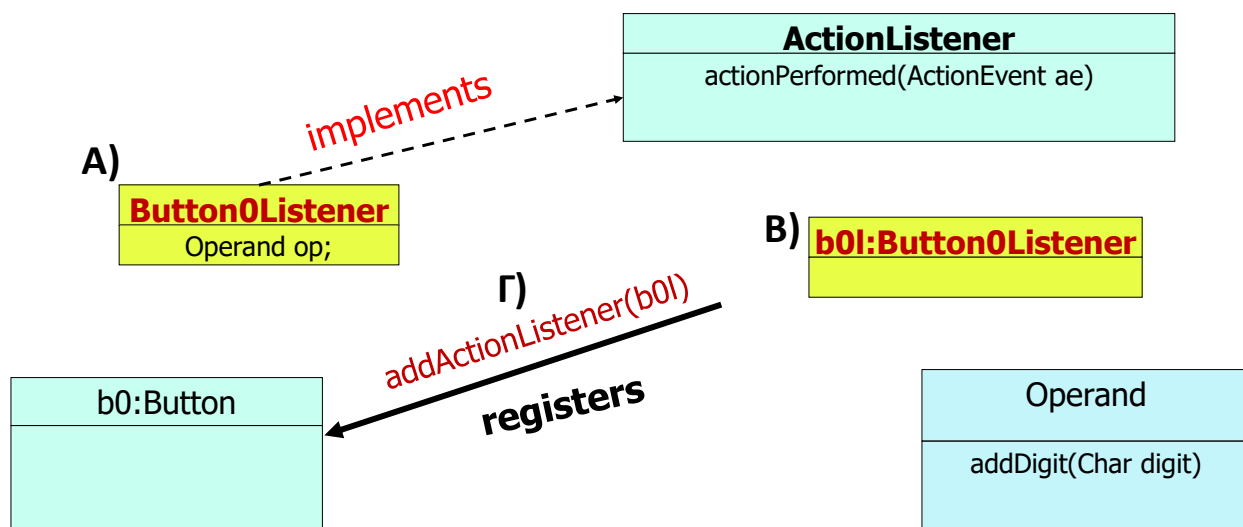
Αυτό γίνεται με την μέθοδο **addActionListener** της Button.

```
public void addActionListener(ActionListener l)
```

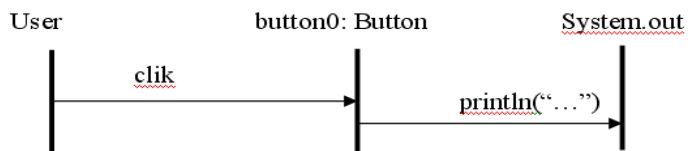
Η μέθοδος «Adds the specified action listener to receive action events from this button. Action events occur when a user presses or releases the mouse over this button. If l is null, no exception is thrown and no action is performed.»



Button0 Event Handling key constructs



Button0 pressed - Event Handling



1. Ορίζουμε την κλάση Button0Handler

```

class Button0Handler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        System.out.println("button0 was pressed");
    }
}
  
```

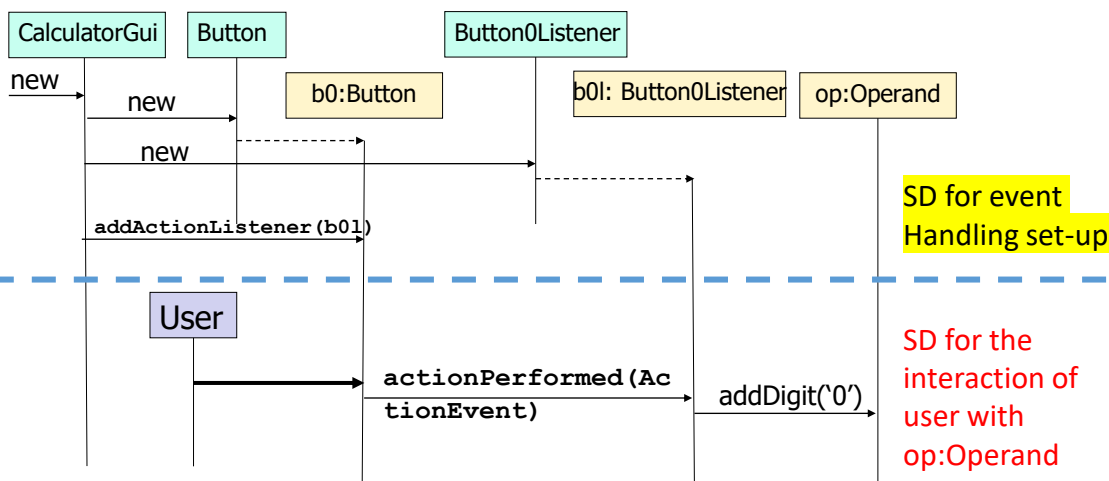
2. Δημιουργούμε ένα στιγμιότυπο της κλάσης Button0Handler

3. Το κάνουμε register στο πλήκτρο 0 χρησιμοποιώντας την μέθοδο addActionListener της Button.

```

button0.addActionListener(new Button0Handler());
  
```

Event Handling -Sequence Diagrams



ActionListener - ActionEvent

java.awt.event

Interface ActionListener

Method Summary

void	actionPerformed(ActionEvent e) Invoked when an action occurs.
------	--

java.awt.event

Class ActionEvent

[java.lang.Object](#)

└ [java.util.EventObject](#)

└ [java.awt.AWTEvent](#)

└ [java.awt.event.ActionEvent](#)

This high-level event is generated by a component (such as a Button) when the component-specific action occurs (such as being pressed).

Method Summary

getActionCommand()

[String](#) [getActionCommand\(\)](#)

Returns the command string associated with this action.

Methods inherited from class [java.util.EventObject](#)

public [Object](#) [getSource\(\)](#)

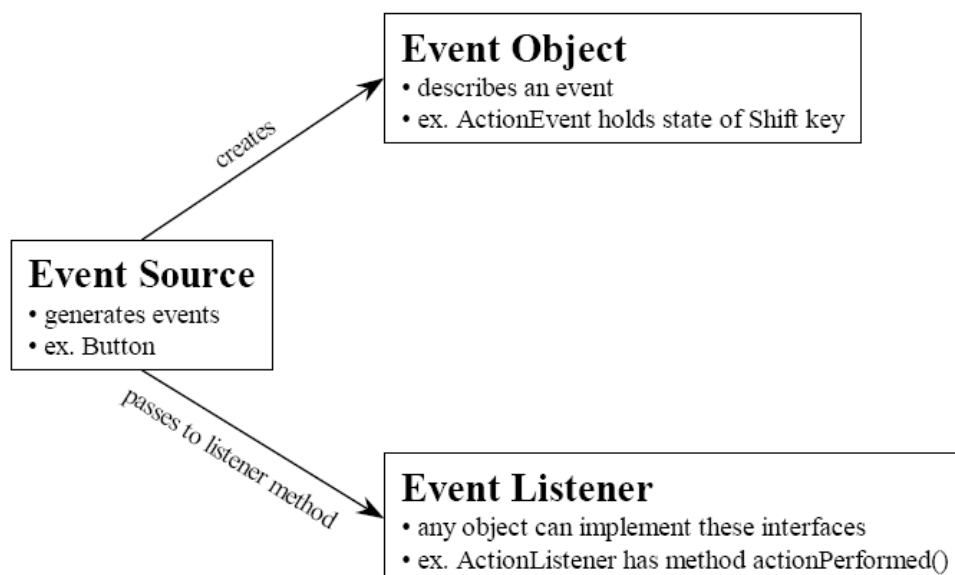
Returns: The object on which the Event initially occurred.

© 2023 Κλεάνθης Θραμπουλιδης

Event Handling

Διαφάνεια 13

Event sources, Listeners and Objects



Source: Object Computing, Inc

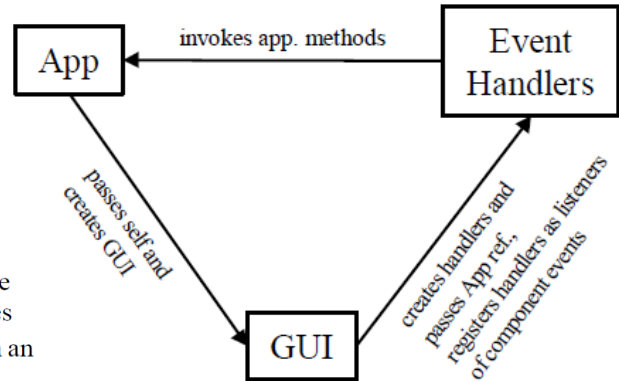
© 2023 Κλεάνθης Θραμπουλιδης

Event Handling

Διαφάνεια 15

Design For Flexibility and Maintainability

- Can separate
 - application code
 - GUI code
 - event handling code
- Steps to achieve this separation
 - create a single class whose constructor creates the entire GUI, possibly using other GUI-only classes
 - create the GUI by invoking this constructor from an application class
 - create classes whose only function is to be notified of GUI events and invoke application methods
 - their constructors should accept references to application objects whose methods they will invoke
 - create event handling objects in a GUI class and register them with the components whose events they will handle



Source: Object Computing, Inc

Event Handling Revisited

Button event handling
vs.
Goody's event handling:
A comparison



1. Ορίζουμε την κλάση Button0Handler

```
class Button0Handler implements ActionListener {
    public void actionPerformed(ActionEvent pushingButton0) {
        System.out.println("button0 was pressed");
    }
}
```

2. Δημιουργούμε ένα στιγμιότυπο της κλάσης Button0Handler

3. Το κάνουμε register στο πλήκτρο 0 χρησιμοποιώντας την μέθοδο addActionListener της Button.

```
button0.addActionListener(new Button0Handler());
```


Event Handling στο Goody's

Ο Νικολάου μόλις έδωσε την παραγγελία του στην Μαίρη. Για να μην περιμένει όρθιος μέχρι αυτή να ετοιμαστεί ζητά από την Μαίρη να τον ειδοποιήσει.
Απαραίτητη προϋπόθεση είναι:

- η γνωστοποίηση του ονόματος του (της αναφοράς του) στην Μαίρη και
- η γνώση που θα πρέπει να έχει ο Νικολάου (event Listener) για το πως θα αντιδράσει όταν η Μαίρη του αποστείλει το μήνυμα «Νικολάου έτοιμος».

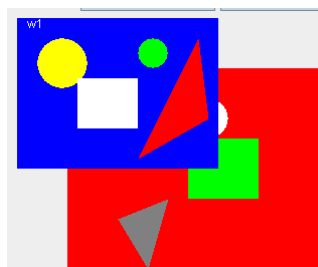
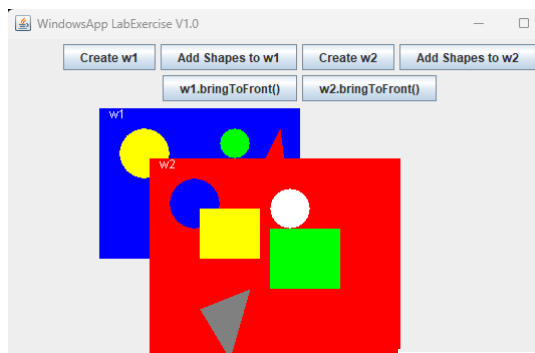
Ο Νικολάου κάθεται σε ένα τραπέζι και αναμένει την ειδοποίηση της Μαίρης. Όταν η παραγγελία του Νικολάου ετοιμασθεί (**συμβεί το event για το οποίο έκανε register ο Νικολάου**) η Μαίρη δημιουργεί το μήνυμα «παραγγελία έτοιμη» και το αποστέλλει στον Νικολάου.

Ο Νικολάου, που **πρέπει να γνωρίζει πως αντιδρά στο event** «παραγγελία έτοιμη», εκτελεί τις ενέργειες που η υλοποίηση της αντίστοιχης μεθόδου ορίζει.

Οργάνωση Διάλεξης

- Βασικές έννοιες
 - Delegation Event Model
 - Event Handling στο Goody's
- Event Handling in Java Gui
 - ActionListener Interface
 - Design For Flexibility and Maintainability
- **WindowsApp με Graphics**
- RpnCalculator εναλλακτικές υλοποιήσεις
 - Εσωτερική (inner) κλάση
 - Ανώνυμη (anonymous) κλάση
 - έκφραση Lambda (lambda expression)

WindowsApp με γραφική διεπαφή V1



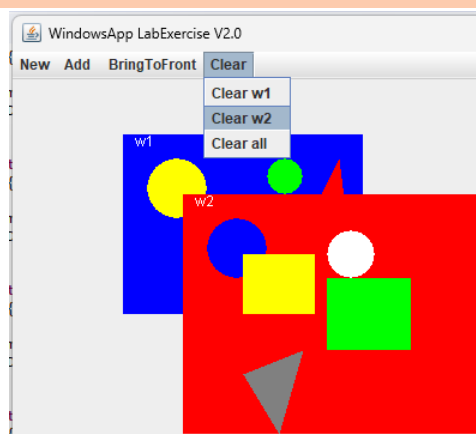
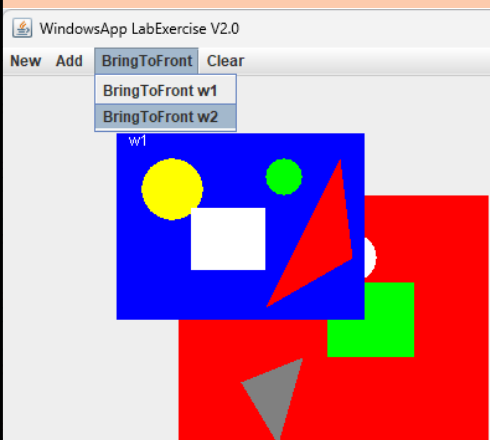
java.awt

Class Graphics

`drawOval(int x, int y, int width, int height)`
Draws the outline of an oval.

`fillOval(int x, int y, int width, int height)`
Fills an oval bounded by the specified rectangle with the current color.

WindowsApp με γραφική διεπαφή V2



```
JMenuBar menuBar;  
JMenu menu;  
JMenuItem menuItem;
```

```
menu = new JMenu("New");  
menuItem = new JMenuItem("New w1");  
menuItem.addActionListener(new ActionListener() {  
    menu.add(menuItem);
```