

Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

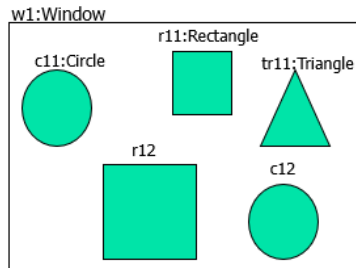
(CEID_NNY106)

Java as an OOP Language Activity 5.9 WindowsApp

Java

High-level programming
language

Κύρια Πηγή



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

Kleanthis Thramboulidis

Prof. of Software and System Engineering

University of Patras

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

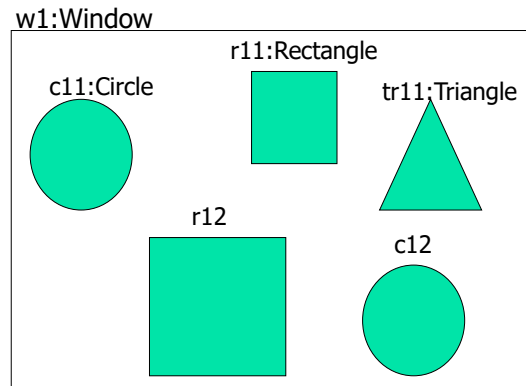
Η πηγή τους σεναρίου

- Σε εφαρμογές που ο χρήστης έχει τη δυνατότητα να δημιουργεί παράθυρα και το κάθε παράθυρο έχει το δικό του περιεχόμενο συναντάμε το **παρακάτω σενάριο**.
- Ο χρήστης κάνει τις παρακάτω ενέργειες:
 - **1.** δημιουργεί ένα παράθυρο w1
 - **2.** δημιουργεί μέσα σε αυτό δύο τετράγωνα (r11, r12), δύο κύκλους (c11,c12), και ένα τρίγωνο (tr11)
 - **3.** Στη συνέχεια δημιουργεί ένα άλλο παράθυρο w2 που επικαλύπτει το w1.
 - **4.** Δημιουργεί μέσα σε αυτό δύο τετράγωνα (r21,r22), δύο κύκλους (c21,c22) και δυο τρίγωνα (tr21,tr22),
 - **5.** Στη συνέχεια επιλέγει το w1 και με δεξί κλικ επιλέγει **bringToFront**. Η εφαρμογή φέρνει το παράθυρο w1 μπροστά από το w2.

Το σενάριο 1/3

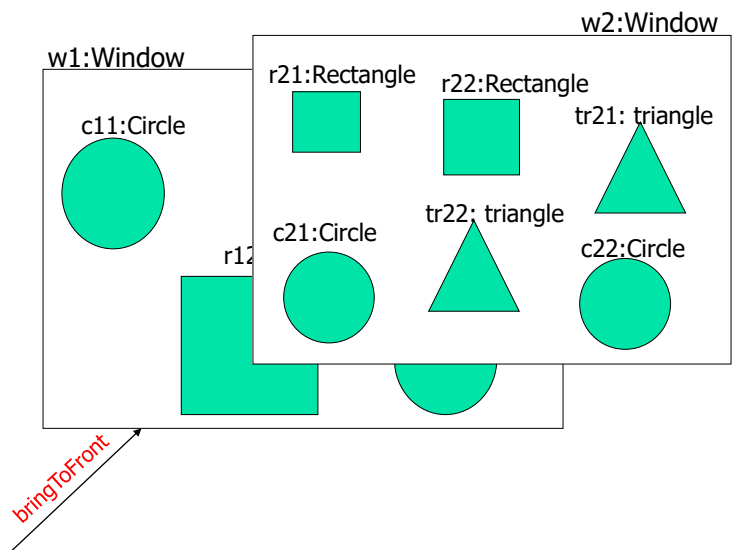
Ο χρήστης κάνει τις παρακάτω ενέργειες:

- 1. δημιουργεί ένα παράθυρο w1
- 2. δημιουργεί μέσα σε αυτό δύο τετράγωνα (r11, r12), δύο κύκλους (c11,c12), και ένα τρίγωνο (tr11)



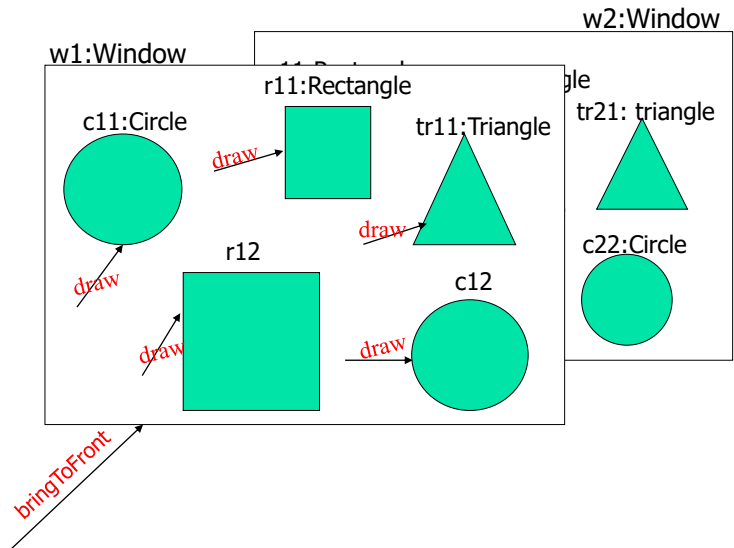
Το σενάριο 2/3

- 3. Στη συνέχεια δημιουργεί ένα άλλο παράθυρο w2 που επικαλύπτει το w1.
- 4. Δημιουργεί μέσα σε αυτό δύο τετράγωνα (r21,r22), δύο κύκλους (c21,c22) και δυο τρίγωνα (tr21,tr22),
- 5. Στη συνέχεια επιλέγει το w1 και με δεξί κλικ επιλέγει **BringToFront**.



Το σενάριο 3/3

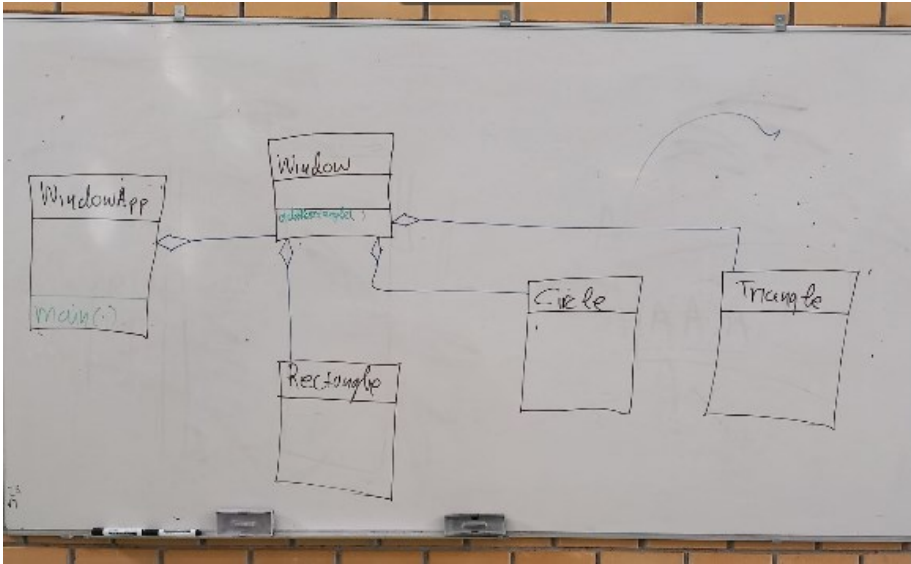
- 5. Στη συνέχεια επιλέγει το w1 και με δεξί κλικ επιλέγει BringToFront. Η εφαρμογή φέρνει το παράθυρο w1 μπροστά από το w2.



EA5 - Development process

- Αναγνώριση Αντικειμένων που θα απαρτίζουν την εφαρμογή
Identify Objects (**Class**),
- Ανάθεση αρμοδιοτήτων στα αντικείμενα της εφαρμογής
Assign Responsibilities (**Responsibility**)
- Ορισμός της συνεργασίας των αντικειμένων
Define object collaborations (**Collaboration**)

EA5-WindowsApp (Identify Objects)

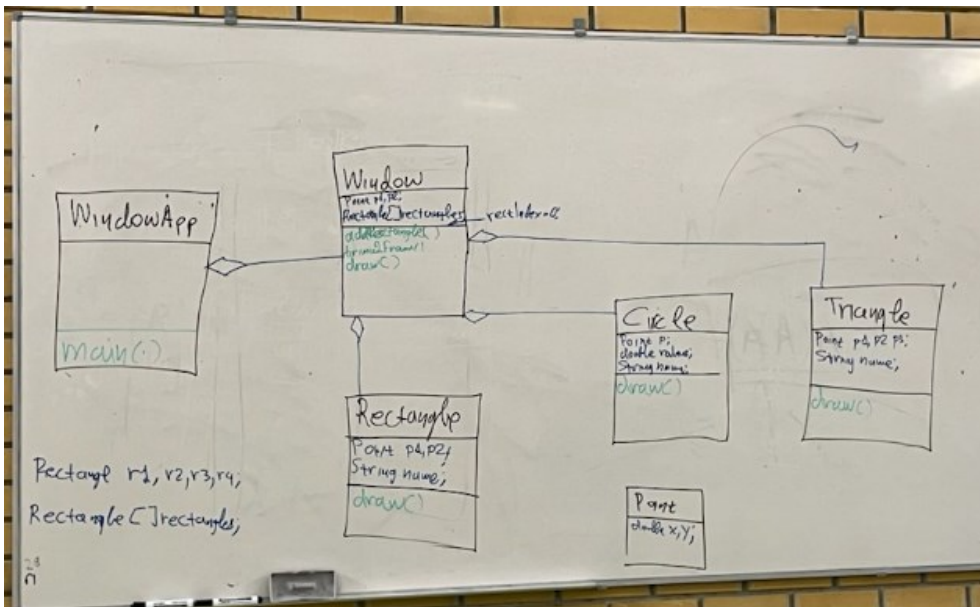


© 2023 Κλεάνθης Θραμπουλίδης

Activity 5.9 WindowsApp

Διαφάνεια 7

Assign Responsibilities

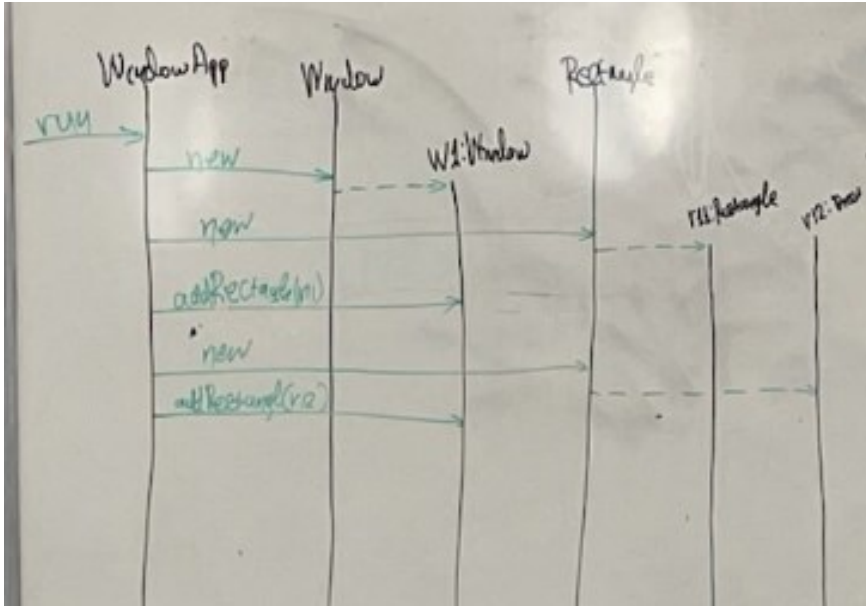


© 2023 Κλεάνθης Θραμπουλίδης

Activity 5.9 WindowsApp

Διαφάνεια 8

Define object collaborations

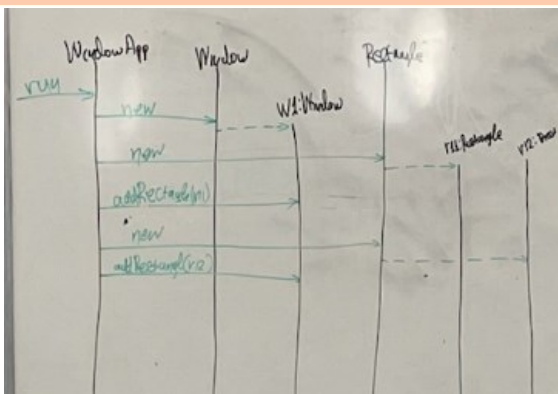


© 2023 Κλεάνθης Θραμπουλίδης

Activity 5.9 WindowsApp

Διαφάνεια 9

From Sequence Diagram to Source Code



```

WindowApp {
  main() {
    var w1 = new Window();
    var r1 = new Rectangle(1);
    w1.addRectangle(r1);
    var r2 = new Rectangle(2);
    w1.addRectangle(r2);
  }
}

void addRectangle(Rectangle r) {
  rectangles[rectangles.length] = r;
}

```

© 2023 Κλεάνθης Θραμπουλίδης

Activity 5.9 WindowsApp

Διαφάνεια 10

addRectangle

```

class Window {
private
    Rectangle[] rectangles;
    int rectIndex=0;
public void addRectangle(Rectangle r) {
    rectangles[rectIndex++] = r;
}
public void bringToFront() {
}
}
    
```

Identify Generalization/Specialization

Assign Responsibilities

```

classDiagram
    class WindowApp {
        main()
    }
    class Window {
        Point p1, p2
        Rectangle[] rectangles
        addRectangle(Rectangle r)
        bringToFront()
        draw()
    }
    class Shape {
        <<abstract>>
        String name
        <<abstract method>>
        draw()
    }
    class Rectangle {
        Point p1, p2
        String name
        draw()
    }
    class Circle {
        Point p1, p2
        double radius
        String name
        draw()
    }
    class Triangle {
        Point p1, p2, p3
        String name
        draw()
    }
    class Point {
        double x, y
    }
    WindowApp o-- Window
    Window o-- Rectangle
    Window o-- Shape
    Shape <|-- Rectangle
    Shape <|-- Circle
    Shape <|-- Triangle
    
```

Shape: String name, abstract void draw();
 Rectangle: Point p1, p2, String name, draw()
 Circle: Point p1, p2, double radius, String name, draw()
 Triangle: Point p1, p2, p3, String name, draw()
 Point: double x, y;

Implementation of Generalization/Specialization

```
var w1 = new Window("w1");
var w2 = new Window("w2");
```

```
var r = new Rectangle("r1");
```

```
w1.add(Rectangle());
w1.add(new Circle("c1"));
w1.buttonPressed();
```

```
abstract class Shape {
    String name;
    abstract void draw();
}
```

```
class Rectangle extends Shape {
```

```
    void draw() {
        System.out.println("draw executed on " + name);
    }
}
```