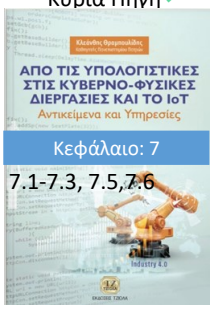
Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

Κληρονομικότητα- Πολυμορφισμός (Inheritance-Polymorphism)

Κύρια Πηγή



Inheritance:
the act of inheriting property

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
https://sites.google.com/site/thramboulidiskleanthis/

Java
High-level programming language

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

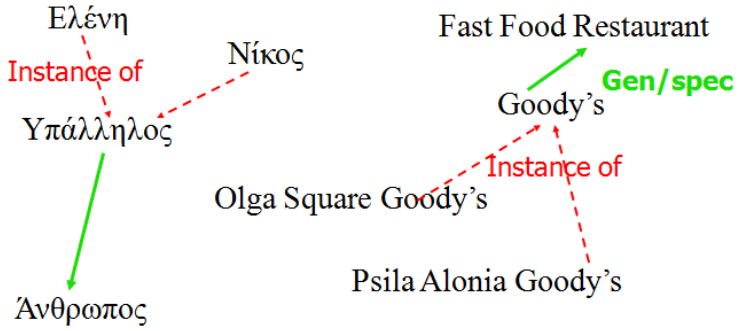
First appeared: May 23, 1995; 27 years ago

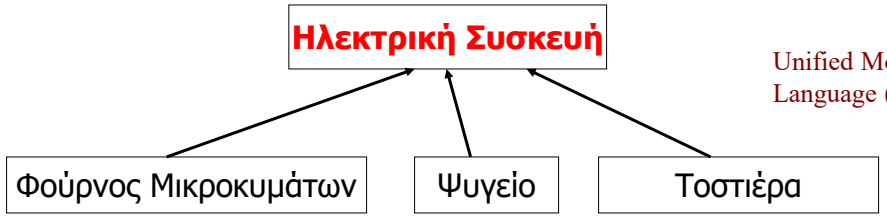
Paradigm: Multi-paradigm: generic, object-oriented (class-based), functional, imperative, reflective, concurrent

Οργάνωση Διάλεξης

- **Εισαγωγή στην Κληρονομικότητα**
  - **Βασικές έννοιες**
  - Κληρονομικότητα
    - στην WindowsApp
    - στην RPN Calculator
  - Abstract μέθοδοι και Abstract κλάσεις
  - Shadowed variables-Overriding methods
- Πολλαπλή κληρονομικότητα (Multiple inheritance)
- Πολυμορφισμός (Polymorphism)
  - Πολυμορφισμός στην WindowsApp
- Building the inheritance tree
  - Alternatives in writing a sub class
  - Access level modifiers

Σχέση Γενίκευσης-Εξειδίκευσης





Unified Modeling Language (UML)

Η κλάση GraphicCircle

```
public class GraphicCircle {
    public double x,y;
    public double r;
    Color outline, fill;
    public double circumference(){
            return 2*3,14*r;}
    public double area() {
            return 3.14*r*r;}
    public void draw(...) { ..............}
    }
```

Η κλάση Circle

```
public class Circle {
    public double x,y;
    public double r;
    public double circumference(){
            return 2*3,14*r;}
    public double area() {
            return 3.14*r*r;}
    }
```

Ορισμός ως απόγονης κλάσης

```
public class GraphicCircle extends Circle {
    Color outline, fill;
    public void draw(...) {  .....................}
    }

GraphicCircle gc = new GraphicCircle();
double area = gc.area();
```

Circle **c** = gc;

κάθε στιγμιότυπο της υποκλάσης μπορεί να θεωρηθεί και στιγμιότυπο της πρόγονης κλάσης

Python

```
class GraphicCircle(Circle):
    pass
```


Subclass constructor - Super

```
public GraphicCircle(double x, double y, double r, Color
   outline, Color fill) {
   this.x = x; this.y = y; this.r = r;
   this.outilne = outline; this.fill =
   fill;
      }
```

- κλήση του constructor της πρόγονης κλάσης

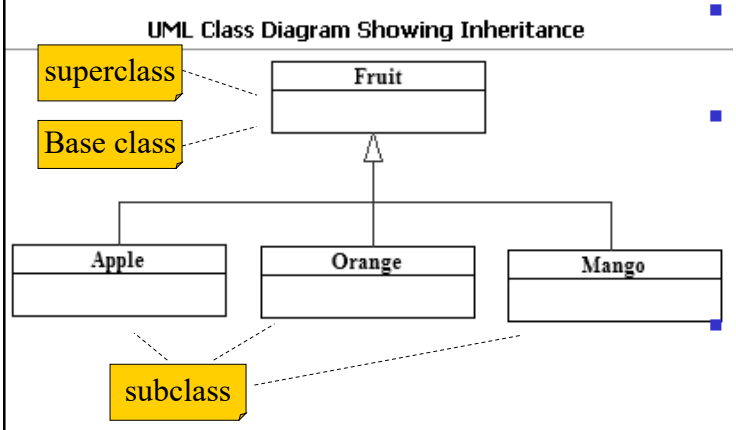
```
public GraphicCircle(double x, double y, double r, Color outline,
   Color fill) {
      super(x,y,r);
      this.outilne = outline; this.fill = fill;
      }
```

Super

- δεσμευμένη λέξη
- μέσα σε ένα constructor χρησιμοποιείται μόνο για να καλέσει τον constructor του προγόνου
- πρέπει
  - να είναι η πρώτη πρόταση του constructor
  - **να προηγείται ακόμη και από τις δηλώσεις μεταβλητών**

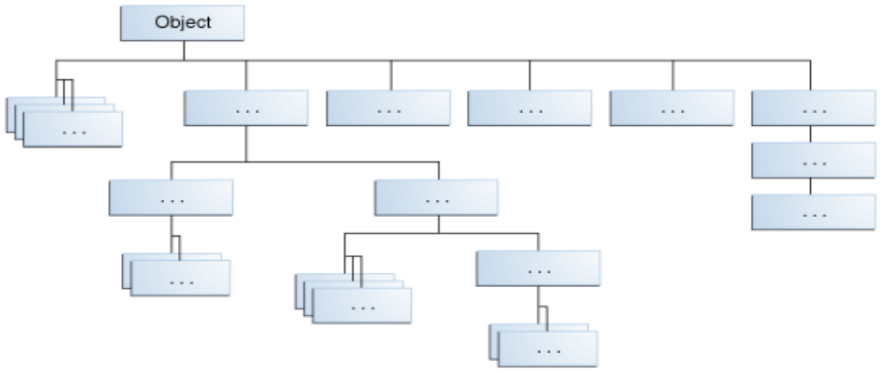
Αναπαράσταση στο Διάγραμμα Κλάσεων

inheritance tree



- **subclass**
  - A class that is derived from another class (**derived** class, **extended** class, or **child** class).
- **superclass**
  - The class from which the subclass is derived (***base*** *class* or ***parent*** *class*).
- **A subclass inherits**
  - all the *members* (fields, methods, and nested classes) from its superclass.
  - **Constructors** are not members, so they **are not inherited by subclasses**, but the constructor of the superclass can be invoked from the subclass.
- **Inheritance**
  - is a way to reuse code of existing objects, or to establish a subtype from an existing object, or both, depending upon programming language support

Java Platform Class Hierarchy



`public class Object`

Class `Object` is the root of the class hierarchy. Every class has `Object` as a superclass. All objects, including arrays, implement the methods of this class.

- In the Java platform, many classes derive directly from Object, other classes derive from some of those classes, and so on, forming a hierarchy of classes.
- The **Object** class (java.lang package) defines and implements behavior common to all Java classes—including the ones that you write.

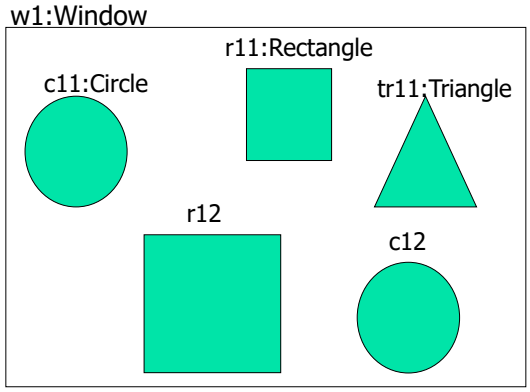
Κληρονομικότητα

- **είναι** ο μηχανισμός του περιβάλλοντος υλοποίησης που μας επιτρέπει
  - να υλοποιήσουμε την σχέση γενίκευσης/εξειδίκευσης μεταξύ των εννοιών της περιοχής του προβλήματος (**problem space**),
  - να αναπτύξουμε νέες κλάσεις της περιοχής λύσης του προβλήματος (**solution space**) οι οποίες θα κληρονομούν χαρακτηριστικά από άλλες ήδη υπάρχουσες κλάσεις.
- **Η κληρονομικότητα επιτρέπει**
  - σε νέες κλάσεις να κληρονομούν την δομή ή/και την συμπεριφορά που ορίζεται σε μία (απλή κληρονομικότητα-**single inheritance**) ή περισσότερες (πολλαπλή κληρονομικότητα-**multiple inheritance**) άλλες κλάσεις.

Οργάνωση Διάλεξης

- **Εισαγωγή στην Κληρονομικότητα**
  - Βασικές έννοιες
  - **Κληρονομικότητα**
    - **στην WindowsApp**
    - **στην RPN Calculator**
  - **Abstract μέθοδοι και Abstract κλάσεις**
  - Shadowed variables-Overriding methods
- Πολλαπλή κληρονομικότητα (Multiple inheritance)
- Πολυμορφισμός (Polymorphism)
  - Πολυμορφισμός στην WindowsApp
- Building the inheritance tree
  - Alternatives in writing a sub class
  - Access level modifiers

WindowsApp – Identify Inheritance



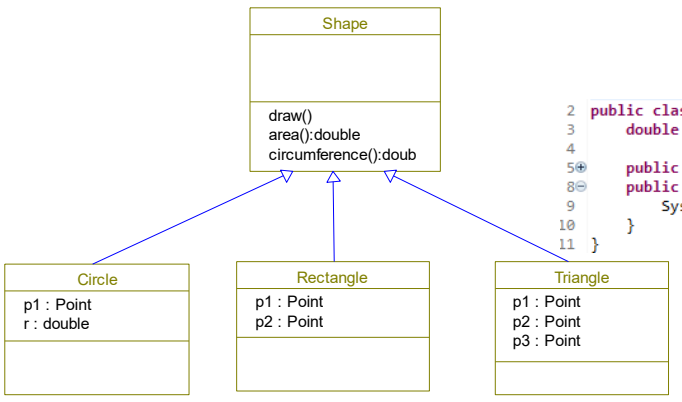

```
Circle [] circles = new Circle[100];
circles[0] = new Circle(…);
circles[1] = new Circle(…);

Rectangle [] rectangles = new Rectangle[100];
rectangles[0] = new Rectangle(…);
rectangles[1] = new Rectangle(…);
```

```
for(int i=0;i<circles.length && circles[i]!=null; i++)
        circles[i].draw();
```

Προβλήματα;

WindowsApp – Inheritance Tree




```
2  public abstract class Shape {
3      String name;
4
5      public Shape(String s) {
8      abstract void draw();
9  }
```

```
2  public class Circle extends Shape{
3      double x,y,r;
4
5      public Circle (String st) {
8      public void draw (){
9          System.out.println(this.getClass().getName()+" draw() "+this.name);
10     }
11 }
```

```
Window.draw() w1
Circle draw() c11
Rectangle draw() r11
Circle draw() c12
```