

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

Διάλεξη #13: Μεταβλητές/μέθοδοι κλάσης, αφηρημένες κλάσεις και διαπροσωπείες

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 1

Μεταβλητές /πεδία κλάσης [class variables]

Τα αντικείμενα "ανήκουν" σε κλάσεις...

The diagram shows an object (Anτικείμενο) on the left with instance variables: "Michael", "McKinnon", 3204, and "grey". An arrow points from the object to a class (Κλάση) on the right. The class contains a setName() method and three dots representing other methods. A text box says "Οι κλάσεις ορίζουν μεθόδους". Another text box points to the class and says "Τα αντικείμενα έχουν μεταβλητές /πεδία".

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 2

Μεταβλητές /πεδία κλάσης (2)

The diagram shows an object (Anτικείμενο) with instance variables: "Michael", "McKinnon", 3204, and "grey". An arrow points from the object to a class (Κλάση) with a setName() method. A text box points to the class and says "Οι κλάσεις μπορεί επίσης να περιέχουν μεταβλητές /πεδία!".

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 3

Μεταβλητές /πεδία κλάσης (3)

- Τα πεδία κλάσης τα κατέχει η («αποθηκεύονται στην») κλάση
- Τα πεδία κλάσης είναι προσπελάσιμα από αντικείμενα
- Υπάρχει μόνο ένα αντίγραφο των πεδίων κλάσης στο σύστημα (αποθηκεύονται «στην κλάση»)
- Όλα τα αντικείμενα μοιράζονται τα πεδία
- Τα πεδία κλάσης μπορεί να χρησιμοποιηθούν για την ανταλλαγή πληροφοριών μεταξύ αντικειμένων (μια όχη και τόσο καλή πρακτική)

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 4

Μεταβλητές /πεδία κλάσης (4)

The diagram shows four objects, each with an equals sign (=) in a circle. Arrows from all four objects point to a class (Κλάση) which has a postCodes variable. A text box says "Όλα τα αντικείμενα έχουν πρόσβαση στο πεδίο postCodes".

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 5

Μεταβλητές /πεδία κλάσης (συντακτικό)

```
class Person
{
    private static int numberOfPeople = 0;

    Person()
    {
        numberOfPeople++;
    }
    ...
}
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 6

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

Μέθοδοι κλάσης [class methods]

- Οι μέθοδοι μπορεί να ανήκουν σε μία κλάση (αντί σε ένα αντικείμενο)
- Καλούνται μέσω της χρήσης του ονόματος της κλάσης (και όχι του αντικειμένου)

Στην κλάση `Integer`:

```
public static int parseInt(String s)
```

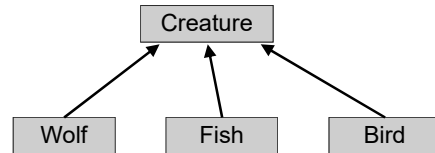
Χρήση:

```
String input;  
input = ... // get input from user  
int players = Integer.parseInt(input);
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 7

Κλάσεις αφηρημένου τύπου [abstract classes]

- Μερικές φορές δεν είναι δυνατόν να υλοποιηθούν όλες οι μέθοδοι!
- Παράδειγμα: κλάση `creature`, method `move`



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 8

Κλάσεις αφηρημένου τύπου (2)

- Στις κλάσεις αφηρημένου τύπου, δεν υλοποιούνται όλες οι μέθοδοι
- Μερικές μέθοδοι δηλώνονται ως «αφηρημένες» [abstract]
- Εάν μία κλάση περιέχει «αφηρημένες» μεθόδους, τότε πρέπει να δηλωθεί ως «abstract»
- Δεν είναι δυνατό να δημιουργήσουμε αντικείμενα με βάση μια κλάση αφηρημένου τύπου

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 9

Κλάσεις αφηρημένου τύπου (3)

Παράδειγμα:

```
abstract class Creature  
{  
    private age;  
  
    public int getAge()  
    {  
        return age;  
    }  
  
    public abstract void move();  
}
```

Μερικές μέθοδοι μπορεί να έχουν υλοποιηθεί

Άλλες, όχι

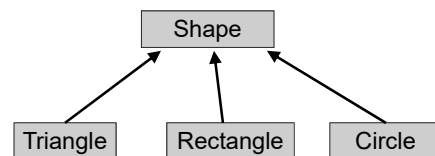
Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 10

«Απτές» κλάσεις [concrete classes]

- Concrete = απτή, χειροπιαστή
- Ως απτές [concrete] κλάσεις αναφέρονται όσες κλάσεις δεν είναι αφηρημένου τύπου
- Αντικείμενα μπορεί να δημιουργηθούν μόνο με βάση τις απτές κλάσεις

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 11

Παράδειγμα: κλάση shapes



```
abstract class Shape  
{  
    public abstract void circumference();  
}
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 12

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

Λόγος ύπαρξης των «αφηρημένων» κλάσεων

- Εάν δεν μπορούμε να δημιουργήσουμε αντικείμενα με βάση τις κλάσεις αφηρημένου τύπου, ποια η χρησιμότητά τους;

Απάντηση:

- Πολυμορφισμός!
- Μία «αφηρημένη» κλάση είναι ένας τύπος δεδομένων
- Μπορούμε να δηλώσουμε μεταβλητές αυτού του τύπου
- Υπο-τύποι μπορεί να δημιουργηθούν

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 13

Προσομοίωση διακριτών γεγονότων

[Discrete Event Simulation]

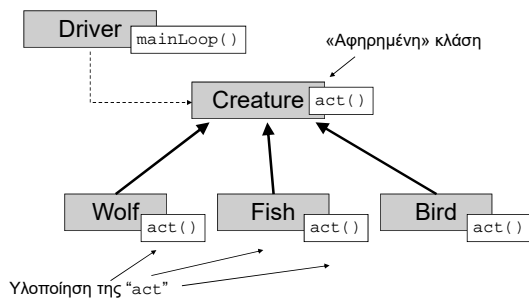
- προσομοίωση συστήματος
- το σύστημα αποτελείται από ένα σύνολο ενεργών στοιχείων [actors]
- Βασίζεται σε ένα επαναλαμβανόμενο σύνολο εντολών [main loop] το οποίο δίνει τη δυνατότητα στα ενεργά στοιχεία να δράσουν

Παραδείγματα:

- Προσομοίωση περιβαλλοντικών συστημάτων, αεροδρομίων, συγκοινωνιακών συστημάτων, πρόγνωση καιρού, πυρηνικών εκρήξεων, κλπ.

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 14

Προσομοίωση διακριτών γεγονότων



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 15

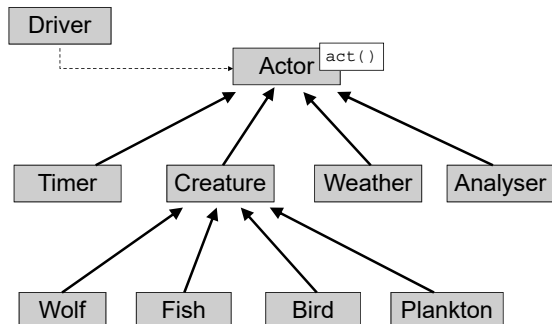
Προσομοίωση – βασικός βρόγχος

```
private void mainLoop()
{
    while(!endOfSimulation())
    {
        // let every actor do what they want to do
        for(int i=0; i < actors.size(); i++)
        {
            Actor actor = actors.get(i);
            actor.act();
        }
    }
}
```

- Υποθέτουμε ότι η μεταβλητή actors έχει δηλωθεί ως:
`ArrayList<Actor> actors`

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 16

Πιο περίπλοκο παράδειγμα



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 17

Βασικός βρόγχος

```
private void mainLoop()
{
    while(!endOfSimulation())
    {
        // let every actor do what they want to do
        for(int i=0; i < actors.size(); i++)
        {
            Actor actor = actors.get(i);
            actor.act();
        }
    }
}
```

Τι αλλάζει;



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 18

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

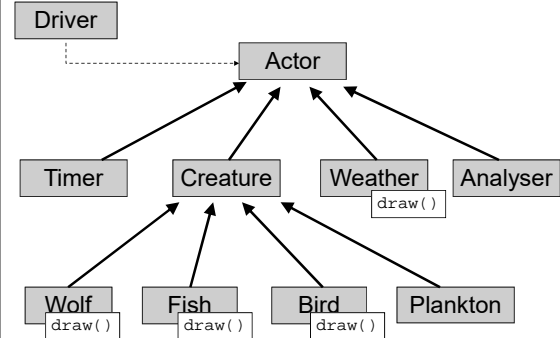
Μία ακόμα προσθήκη...

- Να προστεθεί κώδικας που εμφανίζει την προσομοίωση στην οθόνη

```
private void mainLoop()
{
    while(!endOfSimulation())
    {
        // let every actor do what they want to do
        for(int i=0; i < actors.size(); i++)
        {
            Actor actor = actors.get(i);
            actor.act();
        }
        ... // draw all visible actors on screen
    }
}
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 19

Ένα πιο ενδιαφέρον παράδειγμα



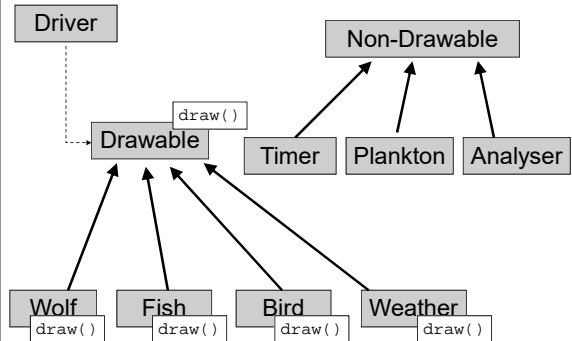
Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 20

Θα επιθυμούσαμε να...

```
private void mainLoop()
{
    while(!endOfSimulation())
    {
        // let every actor do what they want to do
        for(int i=0; i < actors.size(); i++)
        {
            Actor actor = actors.get(i);
            actor.act();
        }
        for(int i=0; i < drawables.size(); i++)
        {
            Drawable drawable = drawables.get(i);
            drawable.draw();
        }
    }
}
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 21

Μια διαφορετική ιεραρχία



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 22

Το πρόβλημα

```
private void mainLoop()
{
    while(!endOfSimulation())
    {
        // let every actor do what they want to do
        for(int i=0; i < actors.size(); i++)
        {
            Actor actor = actors.get(i);
            actor.act();
        }
        for(int i=0; i < drawables.size(); i++)
        {
            Drawable drawable = drawables.get(i);
            drawable.draw();
        }
    }
}
```

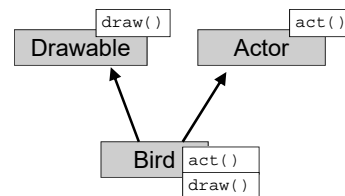
Έπαψε να είναι σωστός κώδικας.



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 23

Πολλαπλή κληρονομικότητα

[Multiple Inheritance]



```
Bird myBird = new Bird();
actors.add(myBird);
drawables.add(myBird);
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 24

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

Το επόμενο πρόβλημα...

- Η Java δεν επιτρέπει να κληρονομήσουμε από περισσότερες της μίας κλάσης!
- Η πολλαπλή κληρονομικότητα είναι χρήσιμη αλλά προκαλεί προβλήματα (κάνει πιο πολύπλοκο) στον ορισμό της γλώσσας
- Για το λόγο αυτό, μερικές γλώσσες δεν επιτρέπουν την πολλαπλή κληρονομικότητα (πχ. Java)
- Πως αντιμετωπίζουμε το πρόβλημα;



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 25

Η λύση: διαπροσωπείες* [interfaces]

- Μια διαπροσωπεία μοιάζει με μία πλήρως «αφηρημένη» κλάση
- Οι διαπροσωπείες [interfaces]:
 - περιέχουν μόνο δηλώσεις μεθόδων [method signatures]
 - περιέχουν μόνο ορισμούς σταθερών [constants]
 - δεν περιέχουν πεδία
 - δεν περιέχουν υλοποιήσεις μεθόδων



*ή «διασυνδέσεις»

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 26

Διαπροσωπείες και κληρονομικότητα

- Μία κλάση μπορεί να κληρονομήσει μόνο από μία άλλη κλάση, αλλά...
- ...μπορεί να «κληρονομήσει» από περισσότερες από μία διαπροσωπείες!

Ορολογία (για να ξεχωρίζουμε τις δυο αυτές μορφές κληρονομικότητας):

- Μία κλάση **επεκτείνει** [extends] μια υπερ-κλάση.
- Μία κλάση **υλοποιεί** [implements] μία διαπροσωπεία.

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 27

Παράδειγμα

```
interface Drawable
{
    /**
     * Draw this entity on screen.
     */
    void draw();
}
```

Οι μέθοδοι των διαπροσωπειών είναι εξ' ορισμού "public" – δεν χρειάζεται μετατροπέας πρόσβασης

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 28

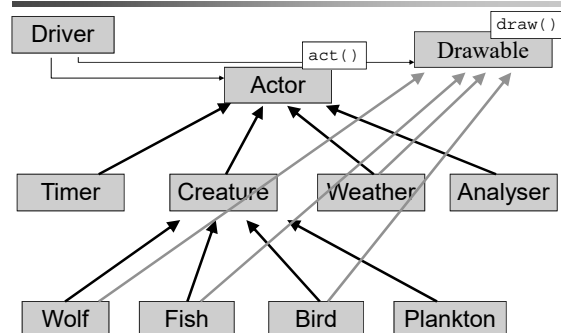
Η υπο-κλάση

```
class Bird extends Creature
    implements Drawable
{
    ...
    /**
     * Act. Here that means fly around and search
     * for bird food.
     */
    public void act()
    { ... }

    /**
     * Draw this bird on screen.
     */
    public void draw()
    { ... }
}
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 29

Η λύση



Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 30

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #13

Πολλαπλές διαπροσωπείες

```
class Bird extends Actor  
  implements Drawable, Storable  
{  
  ...  
}
```

- Το πολύ μία υπερ-κλάση
- Απεριόριστος αριθμός διαπροσωπειών

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 31

Πολλαπλές διαπροσωπείες (2)

```
class Bird  
  implements Actor, Drawable, Storable  
{  
  ...  
}
```

- Η κλάση Actor θα μπορούσε να ήταν μία διαπροσωπεία.

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 32

Οι διαπροσωπείες ως τύποι δεδομένων

- Οι διαπροσωπείες ορίζουν τύπους δεδομένων (όπως και οι κλάσεις).
- Όπως ισχύει και για τις κλάσεις αφηρημένου τύπου, δεν μπορούμε να δημιουργήσουμε ένα στιγμιότυπο **[instance]** από μία διαπροσωπεία, αλλά μπορούμε να ορίζουμε μεταβλητές
- Στιγμιότυπα των κλάσεων που υλοποιούν τις διαπροσωπείες μπορεί να καταχωρηθούν σε μεταβλητές τύπου διαπροσωπειών



```
Drawable nextToDraw;  
Bird myBird = new Bird();  
nextToDraw = myBird;
```

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 33