



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Προγραμματισμός Διαδικτύου

Ενότητα 3 : Κατηγορίες και κληρονομικότητα

Ιωάννης Τσούλος



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Προγραμματισμός Διαδικτύου

Ενότητα 3 : Κατηγορίες και κληρονομικότητα

Ιωάννης Τσούλος

Επίκουρος Καθηγητής

Άρτα, 2015



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

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 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.
- Το έργο «**Ανοιχτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



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



Ορισμός

- Στον πραγματικό κόσμο τα πάντα είναι δομημένα σε κατηγορίες (οχήματα) και αντικείμενα (λιμουζίνες, φορτηγά, ποδήλατα).
- Κάθε κατηγορία περιέχει κάποια συστατικά στοιχεία που σε προγραμματιστικό περιβάλλον υλοποιούνται από μεταβλητές και θα τα ονομάζουμε πεδία
- Για παράδειγμα στην κατηγορία ΟΧΗΜΑ πεδία μπορεί να είναι: η μάρκα του οχήματος ο αριθμός των θέσεων το χρώμα κτλ.
- Η πρόσβαση στα πεδία επιτυγχάνεται μέσω συναρτήσεων που θα τις ονομάζουμε μεθόδους.



Πεδία_{1/3}

- Τα πεδία είναι μεταβλητές που συνήθως είτε είναι εντελώς κρυφά από τον έξω κόσμο είτε δεν μπορούν να τροποποιηθούν με άμεσο τρόπο.
- Η δημιουργία αντικειμένων δεν έχει καμία σχέση με δυναμική κατανομή μνήμης και για αυτόν τον λόγο δεν υπάρχει και αντίστοιχος τελεστής delete.
- Η αναφορά στα πεδία της κατηγορίας γίνεται με τον τελεστή . να παρεμβάλλεται ανάμεσα στο όνομα του αντικειμένου (όχι της κατηγορίας) και το όνομα του πεδίου.
- Η παραπάνω χρήση της κατηγορίας αν και αποτελεσματική αντιβαίνει σε έναν από τους βασικότερους κανόνες του αντικειμενοστραφούς προγραμματισμού που είναι η ενθυλάκωση.



Πεδία _{2/3}

- Σύμφωνα με την ενθυλάκωση δεν πρέπει να έχουμε άμεση πρόσβαση στα πεδία της κατηγορίας όπως γίνεται στο παράδειγμα.
- Για αυτόν τον λόγο και δεν πρέπει να δηλώσουμε τα πεδία σαν public αλλά σαν private.



Πεδία 3/3

Algorithm 1. Η απλή κατηγορία σημείου

```
class point_simple
{
    public double x , y ;
}
```

Algorithm 2. Μια main() που χρησιμοποιεί τα απλά σημεία.

```
public class MainPointSimple
{
    public static void main (String argf[])
    {
        point_simple mypoint=new point_simple ();
        mypoint.x = 100; mypoint.y = 200;

        System.out. println ("point wxw="+point.x );
        System.out . println ("point wyw="+point.y );
    }
}
```




Μέθοδοι_{1/2}

■ Κατηγορίες μεθόδων

- **Αρχικοποίησης.** Εκτελούνται με την δημιουργία ενός αντικειμένου και σκοπός τους είναι να αρχικοποιήσουν κάποια ή όλα τα πεδία της κατηγορίας.
 - Εχουν σαν όνομά τους το όνομα της κατηγορίας.
 - Είναι σχεδόν πάντα public
 - Δεν επιστρέφουν τίποτα
- **Μέθοδοι get.** Εχουν σαν στόχο να επιστρέψουν την τιμή ενός πεδίου της κατηγορίας. Ονομάζονται έτσι γιατί το όνομά τους είναι συνήθως `getFieldNameQ` όπου `fieldName` το όνομα του πεδίου που θέλουμε να μάθουμε.
 - Επιστρέφουν πάντα τιμή (δεν είναι σχεδόν ποτέ void)
 - Είναι σχεδόν πάντα public (αλλιώς δεν θα είχαν και λόγω ύπαρξης)
 - Σχεδόν ποτέ δεν δέχονται ορίσματα



Μέθοδοι_{2/2}

- **Μέθοδοι set.** Είναι οι αντίθετες σαν λογική από τις μεθόδους. Βασικός τους σκοπός είναι να αλλάξουν την τιμή ενός πεδίου μιας κατηγορίας.
 - Δεν επιστρέφουν σχεδόν ποτέ τιμή (είναι void), εκτός και αν επιστρέψουν κάποιον κωδικό με τον οποίο διαπιστώνει ο προγραμματιστής αν όλα πήγαν καλά κατά την αλλαγή του πεδίου.
 - Είναι σχεδόν πάντα public.
 - Πάντα έχουν τουλάχιστον ένα όρισμα.
- **Μέθοδοι υπολογισμού.** Είναι όλες οι υπόλοιπες μέθοδοι που δεν περιλαμβάνονται σε κάποια από τις παραπάνω κατηγορίες. Σκοπός του είναι είτε να υπολογίσουν κάτι είτε να κάνουν κάποια άλλη λειτουργία πχ να ανοίξουν ένα αρχείο και να αποθηκεύσουν δεδομένα.



Η μέθοδος `toString()`

- Στο προηγούμενο παράδειγμα κατηγορίας για να εμφανίσουμε τα στοιχεία της κατηγορίας λάβαμε με μεθόδους `get()` τα πεδία και τα εμφανίσαμε.
- Ωστόσο η Java παρέχει σαν διευκόλυνση έναν καλύτερο τρόπο για την εμφάνιση πληροφοριών σχετικά με μια κατηγορία και αυτό είναι η μέθοδος `toString()` η οποία υπάρχει σε όλα τα αντικείμενα που παράγουμε.
- Για να μπορέσουμε να την χρησιμοποιήσουμε θα πρέπει να την επαναπρορίσουμε μέσα σε μια κατηγορία.
- Στο παράδειγμα του σημείου αρκεί να προσθέσουμε την επόμενη μέθοδο μέσα στον κώδικα της κατηγορίας:

```
public String toString()  
{  
    return "x:_" + x + "_y:_" + y;  
}
```



Η μέθοδος toString()

- Για να μπορέσουμε να χρησιμοποιήσουμε την παραπάνω μέθοδο μπορούμε για παράδειγμα στην συνάρτηση main() να γράψουμε:

```
System.out.println (" point1_="+pt1 );
```

```
System.out.println (" point2_="+pt2 );
```

- Με αυτόν τον τρόπο η Java καταλαβαίνει πως όταν απαιτείται να συνενώσει αλφαριθμητικό με κάποιο αντικείμενο θα καλέσει την μέθοδο toString() που υπάρχει ήδη στην κατηγορία.



Κληρονομικότητα

```
class People
{
    protected String name, lastname ;
    protected int age;
    public People ()
    {
        name=" " ; lastname=""; age = 0;
    }
    public People ( String n, String l , int a)
    {
        name=n ; lastname=l ; age=a ;
    }
    public String getName() {return name;}
    public String getLastName() {return lastname;}
    public int getAge() {return age;}
    public void advanceAge() {age++;}
    public String toString()
    {
        return name+" : "+lastname+" : "+age ;
    }
}
```



Κληρονομικότητα

Class Student extends People

```
{
    private int semester ;
    public Student ()
    {
        super();
        semester=0;
    }
    public Student ( String n, String l , int a, int s)
    {
        super (n ,l , a);
        semesier s ;
    }
    Public int getSemester () {return semester;}
    Public String toString ()
    {
        return super. toString ()+" : "+semester ;
    }
}
```



Κληρονομικότητα

Class Teacher extends People

```
{
    private String schoolname;
    public Teacher ()
    {
        super ();
        schoolname=" " ;}
    public Teacher ( String n,String l , int a, String s)
    {
        super (n ,l , a);
        schoolname=s ;
    }
    public String getSchoolName () {return schoolname;}
    public String toString ()
    {
        return super. toString ()+" : "+semester ;
    }
}
```



Κληρονομικότητα

```
Public class UsePeople
{
    public static void main( String arg[])
    {
        People John=new People (" JohnTsoulos " ,35);
        Student Dimitra=new Student (" Dimitra", "Xronopoulou" ,30 ,8);
        Teacher Maria new Teacher (" Maria" ," Ioannou " , 40 , "KOZANH" );
        System. out . println (" John data "+John);
        System. out . println (" Dimitra data "+Dimitra);
        System. out. println (" Maria data "+Maria ):
    }
}
```




Κληρονομικότητα

- **Η λέξη final**
 - Η λέξη final χρησιμοποιείται όταν θέλουμε να ορίσουμε πως μια μέθοδος ή πεδίο θα είναι σταθερό δηλαδή δεν μπορεί να μεταβληθεί από κληρονομιά ή από κάποια ανάθεση.
 - Συνήθως χρησιμοποιείται μπροστά από πεδία προκειμένου να καθορίσουμε πως αυτό το πεδίο θα χρησιμοποιηθεί σαν σταθερά μέσα σε ένα πρόγραμμα.
 - Για παράδειγμα αν θέλουμε να ορίσουμε πως ο φορολογικός συντελεστής είναι σταθερός στο 23% μπορούμε να γράψουμε:
final double taxrate=0.23;



Κληρονομικότητα

■ Η λέξη `static`

- Σκοπός της είναι να ορίσει πως ένα πεδίο ή μέθοδος ανήκει στην κατηγορία και όχι στο αντικείμενο
- Για παράδειγμα αν ορίσουμε στην κατηγορία `People` την μεταβλητή `public static int counter=0`; για να έχουμε πρόσβαση σε αυτήν την μεταβλητή από οπουδήποτε θα πρέπει να γράφουμε `People.counter`
- Το ίδιο ισχύει και για μεθόδους που ορίζονται ως `static`.
- Ο μόνος πρακτικός λόγος χρήσης `static` πεδίων είναι να χρησιμοποιηθούν ως μετρητές αντικειμένων
- Επιπλέον οι `static` μέθοδοι μπορούν να καλέσουν μόνον `static` μεθόδους από την κατηγορία τους.



Κληρονομικότητα

■ Αφηρημένες μέθοδοι

- Οι αφηρημένες μέθοδοι είναι ειδικές κατηγορίες μεθόδων οι οποίες δεν έχουν κώδικα απλά έχουν όνομα και καλούνται από την κατηγορία.
- Σκοπός τους είναι να παρέχουν ένα γενικό interface χωρίς όμως να είναι ακόμα καθορισμένο τι ακριβώς θα κάνουν.
- Αφηρημένες κατηγορίες έχουμε μόνον όταν θα κληρονομηθεί η κατηγορία που τις περιέχει από κάποια άλλη.
- Για αυτόν τον λόγο και οι κατηγορίες που περιέχουν αφηρημένες μεθόδους δεν μπορούν να χρησιμοποιηθούν άμεσα παρά μόνον σε κληρονομικότητα.



Κληρονομικότητα

```
abstract class Shape
{
    public abstract double getArea();
    public abstract double getPerimeter();
    public String toString()
    {
        return " Area:_" + getArea() + "_Perimeter:_" + getPerimeter();
    }
    public boolean same Area ( Shape other)
    {
        boolean ret ;
        if (getArea() == other. getArea()) ret true ;
        else ret false ;
        return ret ;
    }
}
```



Κληρονομικότητα

```
class Square extends Shape
{
    private double a;
    public Square (double mya)
    {
        a= mya;
    }
    public double get Area ()
    {
        return a*a;
    }
    public double getPerimeter ()
    {
        return 4*a;
    }
}
```

```
class Circle extends Shape
{
    private double x0,y0,R;
    public Circle (double mx, double my, double
    {
        x0=mx;
        y0=my;
        R=mr;
    }
    public double get Area ()
    {
        return Math.PI*R*R;
    }
    public double getPerimeter ()
    {
        return 2.0 * Math.PI*R;
    }
}
```



Πολυμορφισμός

- Με τον όρο πολυμορφισμό στον αντικειμενοστραφή προγραμματισμό εννοούμε πως μπορούμε να χρησιμοποιούμε αντικείμενα με έναν γενικό τρόπο και ανάλογα με το περιβάλλον στο οποίο εμφανίζονται να έχουν διαφορετικό αποτέλεσμα.

```
public class TestShape
{
    public static void main (String arg[])
    {
        Shape s1=new Square (10);
        Shape s2=new Circle (5,5,2);
        System . out. println ("Shape1_=_"+s1);
        System . out. println ("Shape2_=_"+s2);
        if ( s1. same Area ( s2 ))
            System . out . println ( "Same_areas" );
        else
            System . out . println ( "Not_same_areas " );
    }
}
```



Βιβλιογραφία

- Πλήρες Εγχειρίδιο της Java 2 Platform, Laura Lemay & Rogers Cadenhead, Ελληνική έκδοση Γκιούρδα
- Java Προγραμματισμός, 6η έκδοση, Harvey M. Deitel, Paul J. Deitel . Εκδόσεις Μ. Γκιούρδας, Αθήνα 2005.
- Εισαγωγή στη γλώσσα προγραμματισμού Java, Εμμ. Σκορδαλάκης, Εργαστήριο Λογισμικού Ομάδα Τεχνολογίας Λογισμικού ΕΜΠ
- Java με UML, Else Lervik και Vegard B. Havdal, Εκδόσεις Κλειδάριθμος, 2005.
- Προγραμματισμός με Java, Γιάννη Κάβουρα, Εκδόσεις Κλειθάρθμος, Αθήνα 2003.
- Java in a nutshell, Deluxe Edition και Java Examples in a Nutshell, David Flanagan, Εκδόσεις O'Reilly
- Java Collections An Introduction to ADTs, Data Structures and Algorithms, David Watt, Deryck Brown, John Wiley and Sons
- Thinking in Java (3rd edition), B. Eckel, Pearson Prentice Hall, 2003.
- Introduction to Java Programming (5th edition), Y. Daniel Liang, Pearson Prentice Hall, 2005.



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Ιωάννης Τσούλος.
Προγραμματισμός Διαδικτύου.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP114/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



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



Τέλος Ενότητας

Κατηγορίες και κληρονομικότητα



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

