

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12

Διάλεξη #12:
Υπο-τύποι και πολυμορφισμός
[sub-typing and polymorphism]

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

Διάγραμμα κλάσεων [Class diagram]

```
classDiagram
    Database ..> Item
    Item <|-- MusicCD
    Item <|-- Video
```

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

Διάγραμμα αντικειμένων [Object diagram]

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

Πηγαίος κώδικας

```
void list()
{
    for (int i = 0; i < myItems.length; i++)
    {
        myItems[i].print();
    }
}
```

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

Σύνοψη: μέθοδος `print`

```
classDiagram
    Item <|-- MusicCD
    Item <|-- Video
    class Item {
        print()
    }
    class MusicCD {
        print()
    }
    class Video {
        print()
    }
```

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

Υπενθύμιση: αντικείμενα και κλάσεις

class

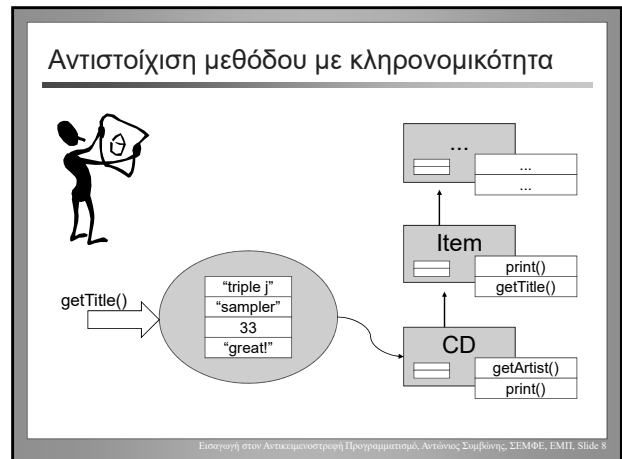
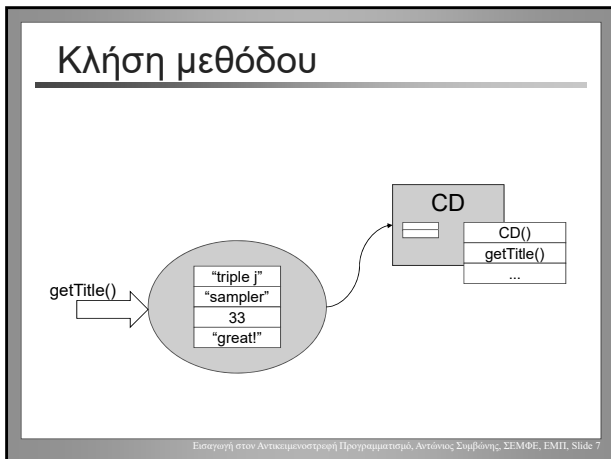
```
class CD {
    CD()
    getTitle()
    ...
}
```

object

```
{
    "triple j"
    "sampler"
    33
    "great!"
}
```

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

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12



“super”

- Μία μέθοδος της υπερκλάσης μπορεί να κληθεί από μια μέθοδο της κλάσης μέσω της ειδικής μεταβλητής **super**.

```
public void print()
{
    super.print();
    System.out.println(...);
}
```

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

Χρήση της “super” σε κατασκευαστές

- Οι κατασκευαστές των υποκλάσεων πρέπει να έχουν μια κλήση προς τον “super” (τον κατασκευαστή της υπερκλάσης) στην πρώτη γραμμή του κώδικά τους.
- Εάν η κλήση έχει παραληφθεί, η εντολή `super()`; εισάγεται αυτόματα από τον μεταφραστή.
- Προσοχή:** Ο κατασκευαστής που δεν έχει παραμέτρους είναι αυτός που επιλέγεται να εισαχθεί αυτόματα!
- Σύσταση:** Πάντα να εισάγετε μία κλήση προς τον κατασκευαστή `super` – μην εξαρτάστε από την αυτόματη εισαγωγή του

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

Παραδείγματα κατασκευαστών

```
MusicCD (String title)
{
    super();
    ...
}

MusicCD (String title)
{
    ...
}

MusicCD (String title)
{
    super(title);
    ...
}
```

- Άμεση κλήση στον **super** χωρίς παραμέτρους
- Έμμεση κλήση στον **super** χωρίς παραμέτρους (δουλεύει μόνο όταν υπάρχει στην υπερκλάση κατασκευαστής χωρίς παραμέτρους)
- Άμεση κλήση στον **super** με παραμέτρους

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

Κληρονομικότητα και υπο-τύποι

Η κληρονομικότητα δημιουργεί μία σχέση **υπο-τύπων [subtypes]** μεταξύ κλάσεων

Υπενθύμιση: Η Java είναι μία γλώσσα με **στατικό σύστημα τύπων δεδομένων [typed language]**. Μόνο τιμές με συμβατούς τύπους δεδομένων (που είναι γνωστοί κατά την μετάφραση) μπορεί να καταχωρηθούν σε μεταβλητές και παραμέτρους

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

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12

Χρήση υπο-τύπων

```
Person aPerson;  
Student aStudent;    aStudent = new Student (...);  
Staff aStaffMember;  aStaffMember = new Staff (...);
```

```
aPerson = aStaffMember;  
aPerson = aStudent; } Σωστή χρήση  
aPerson.changeAddress (...);  
aStudent = aPerson;  
aPerson.graduate(); } Λάθος χρήση
```

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

Υπο-τύποι και συμφωνία τύπων [conformance]

- Αντικείμενα υπο-τύπων μπορεί να υποκαθιστούν αντικείμενα των υπερ-τύπων τους (μπορεί να χρησιμοποιηθούν σε περιπτώσεις όπου αναμένονταν ένα αντικείμενο του υπερ-τύπου).
- «Ένα αντικείμενο υπο-τύπου είναι-ένα [is-a] αντικείμενο υπερ-τύπου»
- Ένας υπο-τύπος **συμφωνεί** [conforms to] με τον υπερ-τύπο.

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

Στατικός – δυναμικοί τύποι δεδομένων

```
Person aPerson;
```



Οι μεταβλητές μπορεί να αναφέρονται σε αντικείμενα του συγκεκριμένου τύπου δήλωσης τους ή σε αντικείμενα που ανήκουν σε οποιονδήποτε υπο-τύπο του.

Student

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

Στατικός – δυναμικοί τύποι δεδομένων

Ο **στατικός τύπος** [static type] είναι ο τύπος δήλωσης μιας μεταβλητής ή παραμέτρου όπως αυτός εμφανίζεται στον πηγαίο κώδικα.

Ο **δυναμικός τύπος** [dynamic type] είναι ο τύπος του αντικειμένου (κατά το χρόνο εκτέλεσης).



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

Υπο-τύποι / πολυμορφισμός



Οι υπο-τύποι παρέχουν τις ίδιες μεθόδους αλλά οι **υλοποιήσεις** [implementations] των μεθόδων μπορεί να διαφέρουν!

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

Δυναμικός καθορισμός τύπου [Dynamic dispatch]

Κώδικας:

```
anItem = aCD;  
...  
anItem.print();
```

Αποτέλεσμα:

```
CD: Triple J Hottest 100 (79 min)  
artist: sampler, 33 tracks  
double CD - great!
```

Ίδιος κώδικας,
Διαφορετικό αποτέλεσμα!

```
anItem = aVideo;  
...  
anItem.print();
```

```
Video: The South Park Movie (102 min)  
director: Fred Smith  
(not seen yet)
```

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

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12

Προσδιορισμός μεθόδου [method lookup, binding]

anItem.print();

Item
getTitle()
setComment()
print()

Video
getDirector()
print()

anItem

Video

"instance of"

Σημείωση: Ο δυναμικός τύπος των δεδομένων προσδιορίζει το σημείο εκκίνησης της διαδικασίας προσδιορισμού της κατάλληλης μεθόδου!

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

Επεκτασιμότητα [Extendability]

Ο δυναμικός προσδιορισμός του τύπου δεδομένων υποστηρίζει την **επεκτασιμότητα** – νέες υποκλάσεις μπορεί να προστεθούν αργότερα χωρίς να είναι απαραίτητη η τροποποίηση του κώδικα που χρησιμοποιεί τις κλάσεις βάσης.

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

Η κλάση "Object"

- (Σχεδόν) κάθε κλάση έχει μία υπερ-κλάση.
- Εάν η υπερ-κλάση δεν έχει άμεσα δηλωθεί, τότε ως υπερκλάση θεωρείται η κλάση **Object**
- Οι μέθοδοι της κλάσης **Object** είναι διαθέσιμες σε κάθε κλάση

Object

Person

Game

Staff

Student

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

"toString"

- Η κλάση **Object** παρέχει τη μέθοδο **toString**, η οποία μετατρέπει ένα αντικείμενο σε ένα **String**.
- **Παράδειγμα:**
`String s = person.toString();`
- Η μέθοδος **toString** καλείται έμμεσα κατά την εκτέλεση της συνένωσης συμβολοσειρών (+)
- **Παράδειγμα:**
`System.out.println("Details: " + person);`

Συνέπεια: όλα τα αντικείμενα μπορεί να λάβουν μέρος σε συνένωση συμβολοσειρών – κατάλληλη λειτουργία για παρουσίαση αποτελεσμάτων!

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

Μετατροπείς πρόσβασης [Access Modifiers]

- Οι μετατροπείς πρόσβασης καθορίζουν την ορατότητα [visibility] ενός πεδίου ή μεθόδου.

Παράδειγμα:

```
private int number;
protected String name;

public void changeAddress(Address newAddress)
{ ... }
private int calculateResult(int parameter)
{ ... }
```

Μετατροπείς πρόσβασης: private, protected, public.

νέο!

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

"private"

- Ορατά/προσπελάσιμα μόνο από το εσωτερικό της κλάσης

client 1

client 2

subclass 1

subclass 2

Ένα "private" πεδίο

Περιοχή ορατότητας

a class

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

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12

“public”

- Ορατά/προσπελάσιμα από το εσωτερικό της κλάσης και από κάθε άλλη κλάση

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

“protected”

- Ορατά/προσπελάσιμα από το εσωτερικό της κλάσης και από κάθε υποκλάση της

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

Οδηγίες χρήσης μετατροπέων πρόσβασης

- Χρησιμοποιείτε πάντοτε ένα μετατροπέα πρόσβασης.
- Περιορίστε την πρόσβαση **όσο το δυνατό** περισσότερο.
- Μη χρησιμοποιείτε **public** πεδία.

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

Η δεσμευμένη λέξη “final”

- Δηλώστε μία μέθοδο ως **final** για να αποτρέψετε τον εκ' νέου ορισμό της [prevent redefinition, overriding]
- Δηλώστε μία κλάση ως **final** για να είναι όλες οι μέθοδοί της “final”.

Παραδείγματα:

```
final public String getPassword()
{ ... }
```

```
final class SecurityManager
{ ...
}
```

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

Το πρόβλημα του «αντίστροφου πολυμορφισμού»

```
class A {
  ...
}
```

```
class B extends A {
  ...
  void doSomething(int n)
  { ... }
}
```

```
// variables
A a;
B b;
...
b = new B();
a = b;
a.doSomething(42);

b = a;
b.doSomething(42);
```

Πως εκτελούμε μία B-μέθοδο μετά από καταχώρηση στο αντικείμενο a (τύπου A);

«λανθασμένα» στις περισσότερες γλώσσες

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

Ανομοιογενείς συλλογές αντικειμένων

- Μία ανομοιογενής συλλογή [heterogenous collection] είναι μία συλλογή από αντικείμενα διαφορετικών τύπων (πολυμορφική συλλογή [polymorphic collection])
- Ανομοιογενείς συλλογές δημιουργούνται δηλώνοντας τα στοιχεία τους ως μέλη μιας υπερ-κλάσης τους.
- Η γενικότερη περίπτωση: Τα στοιχεία είναι τύπου “object” – η συλλογή μπορεί να περιέχει οποιοδήποτε αντικείμενο (πχ. ListArray)

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

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό – Διάλεξη #12

Απώλεια τύπου [type loss]

- Το πρόβλημα: η απώλεια τύπου

```
String myNote = "consider this!";  
ArrayList notes = new ArrayList();  
notes.add(myNote);  
...  
String note;  
note = notes.get(0);
```

Σωστό -- η παράμετρος είναι τύπου Object

Λάθος -- Καταχώρηση Object σε String!

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

Μετατροπή τύπου [casting]

- Η λύση: μετατροπή τύπου [casting]

```
String myNote = "consider this!";  
ArrayList notes = new ArrayList();  
notes.add(myNote);  
...  
String note;  
note = (String)notes.get(0);
```

Μετατροπή σε String!

Μία μετατροπή τύπου μετατρέπει τον στατικό τύπο ενός αντικειμένου σε έναν άλλο τύπο (συνήθως υπο-τύπο). Είναι σωστή μόνο εάν ο δυναμικός τύπος του αντικειμένου είναι σύμφωνος (conforms) με τον τύπο της μεταβλητής στην οποία καταχωρείται.

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

Τμήμα κώδικα

```
String myNote = "consider this!";  
ArrayList notes = new ArrayList();  
notes.add(myNote);  
...  
System.out.println(notes.get(0));
```

- `println` χωρίς μετατροπή τύπου.
- Δουλεύει; γιατί; / γιατί όχι;



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

Γενικές κλάσεις [Generics]

- Συλλογές που περιέχουν αντικείμενα ενός συγκεκριμένου τύπου ή υποτύπων του.
- Παραμετροποιημένοι [parameterized] ή γενικοί [generic] τύποι.
 - `ArrayList<Person>`
 - `ArrayList<EntertainmentItem>`

Περισσότερα
ΠΡΟΣΕΧΩΣ...

Εισαγωγή στον Αντικειμενοστρέφη Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 34