

## 8 Κληρονομικότητα

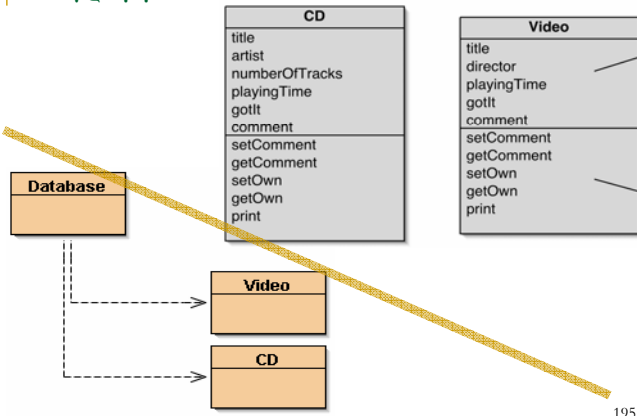
Πώς δημιουργούμε ιεραρχίες κλάσεων που μοιράζονται κοινή συμπεριφορά

### Συλλογή από CD και Βίντεο

- Λειτουργικότητα:
  - Να εισάγουμε πληροφορίες για CD και βίντεο
  - Να αποθηκεύει τις πληροφορίες για να μη χάνονται με τον τερματισμό του προγράμματος
  - Να υποστηρίζει αναζήτηση με διάφορα κριτήρια
  - Να εκτυπώνουμε καταστάσεις με τις συλλογές
  - Να διαγράφουμε πληροφορίες
- Ιδιότητες (πεδία) για CD
  - τίτλος, καλλιτέχνης, πλήθος τραγουδιών, διάρκεια, το έχω αγοράσει;, σχόλιο
- Ιδιότητες (πεδία) για βίντεο
  - τίτλος, καλλιτέχνης, διάρκεια, το έχω αγοράσει;, σχόλιο
- Μέθοδοι για ιδιότητες που μπορεί να αλλάξουν με το χρόνο
  - το έχω αγοράσει;, σχόλιο

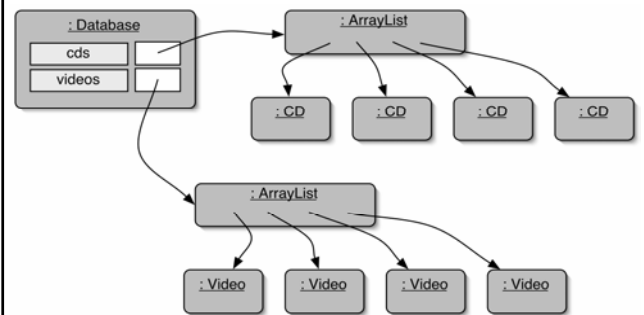
194

### Διάγραμμα Κλάσεων



195

### Διάγραμμα Αντικειμένων



196

### Η Πρώτη Απόπειρα: ο Πυρήνας

- Ο κώδικας των κλάσεων CD και Video είναι απλός: ορίζει ιδιωτικές μεταβλητές στιγμιότυπου για τα πεδία, έναν κατασκευαστή και τις βασικές μεθόδους.
- Όχι πλήρης:
  - δεν αποθηκεύει
  - δεν αναζητά
  - δεν έχει διεπαφή χρήστη.
- Παρατηρήστε ότι τμήμα του κώδικα επαναλαμβάνεται

**ΚΑΚΟ!**

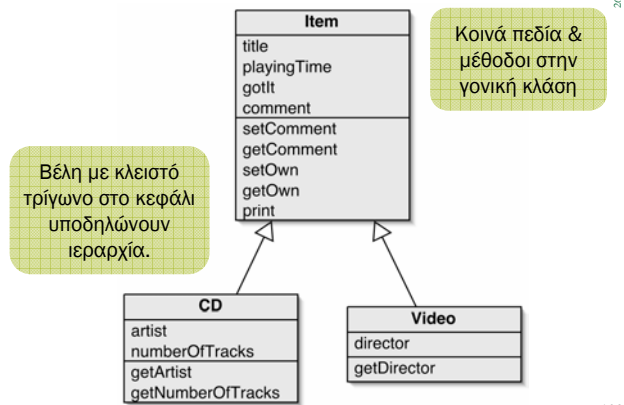
- Ίδια πεδία στις κλάσεις Video & CD
- Παρόμοιος κώδικας στη Database
- Αν χρειαστεί να προσθέσουμε κι άλλο, εκτός από βίντεο & CD;



Project: dome-v1: επισκόπηση κώδικα, απαριθμητής σε ΕΟΤ (Database.java)

197

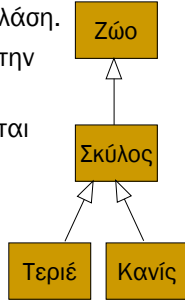
### Κληρονομικότητα



198

## Ορολογία Κληρονομικότητας

- Η κλάση 'Ζώο' ονομάζεται γονική κλάση.
- Η κλάση 'Σκύλος' κληρονομεί από την 'Ζώο' (υποκλάση).
- Στο διάγραμμα κλάσεων συμβολίζεται με ένα βέλος με κλειστή αιχμή.
- Η κληρονομικότητα ονομάζεται και **σχέση είναι-ένα** (is-a).
  - Ο σκύλος είναι-ένα ζώο.
- Η κληρονομικότητα είναι μια μέθοδος αφαίρεσης που μας επιτρέπει να κατηγοριοποιήσουμε τα αντικείμενα.



199

## Κληρονομικότητα στη Java

Γονική κλάση

```
public class Item
{
    private String title;
    private int playingTime;
    private boolean gotIt;
    private String comment;
    ...
}
```

Τα κοινά πεδία στη γονική κλάση και οι υποκλάσεις προσθέτουν ό,τι ιδιαίτερο χρειάζονται.

```
public class CD extends Item
{
    private String artist;
    private int numberOfTracks;
    ...
}
```

Υποκλάσεις

```
public class Video extends Item
{
    private String director;
    ...
}
```

Project: dome-v2: επισκόπηση κώδικα, **extends**, **super** κατασκευαστής

200

## Αναφορά στην Γονική Κλάση

```
public class CD extends Item {
    private String artist;
    private int numberOfTracks;

    /**
     * Κατασκευαστής αντικειμένων της κλάσης CD
     */
    public CD (String theTitle, String theArtist,
               int tracks, int time)
    {
        super(theTitle, time);
        artist = theArtist;
        numberOfTracks = tracks;
    }
    ...
}
```

Κλήση του κατασκευαστή της γονικής κλάσης (πρέπει να είναι η πρώτη εντολή)

201

## Συμβατότητα Υποκλάσεων

```
public void addItem(Item theItem)
{
    items.add(theItem);
}

/**
 * Τυπώνει λίστα με όλα τα στοιχεία της συλλογής.
 */
public void list()
{
    for (Iterator iter = items.iterator(); iter.hasNext(); ) {
        Item item = (Item) iter.next();
        item.print();
        System.out.println(); // κενή γραμμή ανάμεσα σε item
    }
}
```

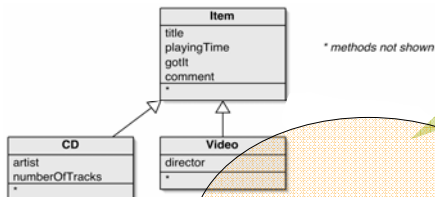
Καλείται με αντικείμενα CD ή Video επειδή είναι-ένα Item

Μπορεί να είναι αντικείμενο CD ή Video

202

## Επαναχρησιμοποίηση

- Αποφύγαμε την επανάληψη κώδικα στις κλάσεις CD και Video. Προσθέστε κλάσεις για επιτραπέζιο παιχνίδι και ηλεκτρονικό παιχνίδι.



Αφηρημένη (abstract) κλάση: δεν επιτρέπεται η κατασκευή στιγμιότυπων της

203

## Πλεονεκτήματα Κληρονομικότητας

- Μειώνει την επανάληψη κώδικα.
- Ευνοεί την επαναχρησιμοποίηση κώδικα.
- Ευκολότερη συντήρηση
  - Αλλαγές στην γονική κλάση αυτόματα ανακλώνται στις υποκλάσεις της.
- Επεκτασιμότητα
  - Προσθέτοντας νέες κλάσεις στην ιεραρχία.

204

## Ασκήσεις Κληρονομικότητας

- Τοποθετήστε τα παρακάτω σε μια σχέση είναι-ένα: μήλο, παγωτό, ψωμί, φαγώσιμο, δημητριακά, πορτοκάλι, επιδόρπιο, ζελέ, χωριάτικο
- Σε ποια ιεραρχία θα μπορούσαν να ενταχθούν ένα ποντίκι και μια πινακίδα αφής; Επεκτείνετε την ιεραρχία ώστε να περιλαμβάνει τα δομικά στοιχεία ενός υπολογιστή (επεξεργαστής, μονάδα αποθ., ...).
- Ποια σχέση ιεραρχίας έχει ένα 'Τετράγωνο' με ένα 'Ορθογώνιο'; Εξηγήστε την πρότασή σας. Επεκτείνετε την ιεραρχία με άλλα γεωμετρικά σχήματα.

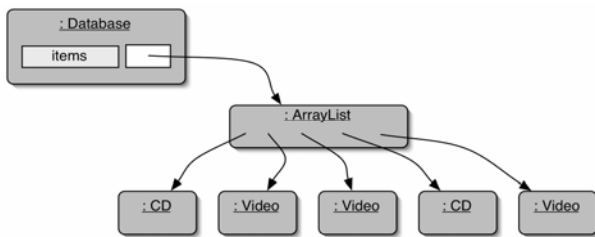
205

## Η Κλάση Object

- Όλες οι κλάσεις που δεν έχουν γονική κλάση, επεκτείνουν μια ειδική κλάση που ονομάζεται Object.
- Η βιβλιοθήκη της Java παρέχει την Object για να την κληρονομούν όλα τα αντικείμενα.
- Λόγοι:
  - Η Object ορίζει κάποιες μεθόδους που τις κληρονομούν όλα τα αντικείμενα.
  - Μπορούμε να ορίσουμε μεταβλητές τύπου Object οι οποίες μπορούν να αποθηκεύουν αντικείμενα οποιασδήποτε κλάσης.

206

## Συλλογές με Ετερογενή Περιεχόμενα



207

## Ετερογενείς Συλλογές

- Οι συλλογές της Java μπορούν να αποθηκεύουν αντικείμενα διαφορετικών τύπων (κλάσεων). Πχ
 

```
ArrayList a = new ArrayList();
a.add(aVideo); // προσθέτει ένα βίντεο
a.add(aCD); // προσθέτει ένα CD
```
- Η υπογραφή της add είναι:
 

```
public void add (Object element)
```
- Το κακό είναι ότι όταν ανασύρουμε ένα στοιχείο, πχ με `a.get(0)` δεν γνωρίζουμε τον ακριβή τύπο του. Η υπογραφή της get είναι:
 

```
public Object get (int index)
```

208

## Κλάσεις Περιτυλίγματος

- Ερώτημα: Πώς μπορούμε να εισάγουμε σε μια συλλογή έναν ακέραιο;
- Απάντηση: Δεν γίνεται, διότι ο ακέραιος δεν είναι αντικείμενο και άρα δεν είναι υποκλάση του Object.
- Λύση: Για όλους τους βασικούς τύπους πχ **int**, **double**, κλπ υπάρχουν αντίστοιχες **κλάσεις περιτυλίγματος** (wrapper classes).

209

## Χρήση Περιτυλιγμάτων

```

int i = 18;
Integer iwrap = new Integer(i);
myCollection.add(iwrap);
...
Integer element = (Integer) myCollection.get(0);
int value = element.intValue()
  
```

Απλός τύπος	Περιτύλιγμα
<b>int</b>	Integer
<b>float</b>	Float
<b>char</b>	Character
...	...

«τύλιξε» την τιμή του ακέραιου

πρόσθεσε στη συλλογή

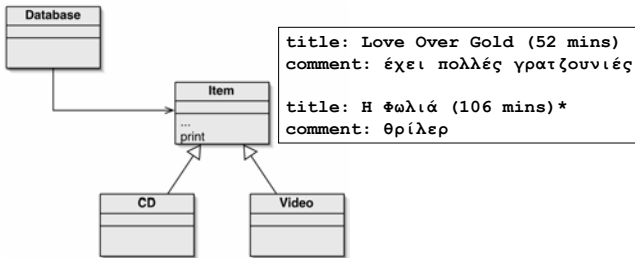
«ξετύλιξε» το

ανάσυρε το αντικείμενο

210

## Πρόβλημα στην print

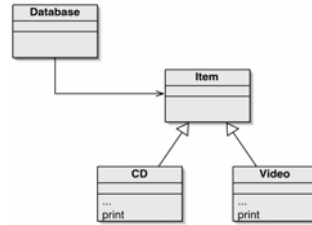
- Η μέθοδος `print` της κλάσης `Item` δεν μπορεί να δει τα πεδία των υποκλάσεών της.
- Η κληρονομικότητα δεν είναι συμμετρική σχέση.



211

## print στις Υποκλάσεις

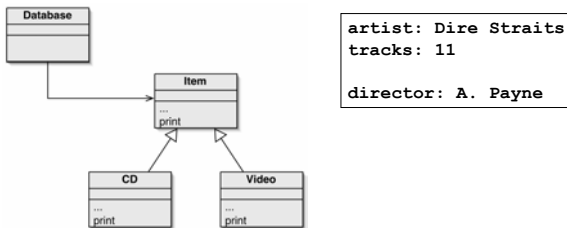
- Αν τη μεταφέρουμε στις υποκλάσεις, τότε δεν μπορεί να κληθεί ( βγάζει compile error).
- Η εκτελούμενη μέθοδος καθορίζεται από τον τύπο του αντικείμενου κατά την εκτέλεση.



212

## print στις Υποκλάσεις και στο Γονέα

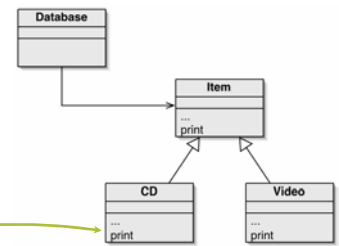
- Αν την αφήσουμε και στην `Item` και στις `CD`, `Video`, καλείται μόνο η `print` της υποκλάσης.



213

## Αναφορά σε Μέθοδο του Γονέα

- Ανάλογα με τον τύπο του αντικείμενου θα κληθεί αρχικά η `print` της `CD` ή της `Video`.



```
public void print ()
{
    super.print ();
    System.out.println("artist: " + artist);
    System.out.println("tracks: " + numberOfTracks);
}
```

214

## Πολυμορφισμός

- Μια μεταβλητή είναι **πολυμορφική** όταν μπορεί να αποθηκεύει αντικείμενα διαφόρων τύπων.
- Κάθε μεταβλητή αντικείμενου είναι πολυμορφική.
- Μια κλήση μεθόδου είναι πολυμορφική όταν εκτελεί διαφορετικές μεθόδους κατά την εκτέλεση.
- Παράδειγμα: κλήση στην `print` θα εκτελεί άλλες φορές την `print` της `CD` και άλλες φορές την `print` της `Video`.
- Ανάλογα με τον τύπο του αντικείμενου πάνω στο οποίο καλείται, πχ `aCD.print()`

215

## Μια Ενδιάμεση Εμβέλεια

- Μέχρι τώρα έχουμε δει εμβέλεια:
  - **public** : σε βλέπουν όλοι
  - **private** : σε βλέπουν μόνο από το εσωτερικό της κλάσης σου
- Η εμβέλεια **protected** είναι μια ενδιάμεση εμβέλεια: σημαίνει ότι σε βλέπουν από το εσωτερικό της κλάσης σου και από τις υποκλάσεις που κληρονομούν την κλάση σου
- ... και από πουθενά αλλού
- Ισχύει για ονόματα στιγμιότυπου και μεθόδους.

216

## Δυναμικός Προσδιορισμός Τύπου

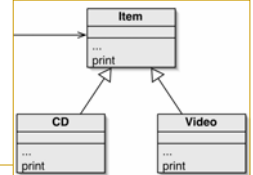
- Κάθε αντικείμενο είναι στιγμιότυπο μιας κλάσης.
- Σε διάφορες περιστάσεις μπορεί να μην γνωρίζουμε τον τύπο ενός αντικειμένου. Πχ
  - λόγω κληρονομικότητας
  - λόγω προσθήκης σε μια συλλογή
- Με τον τελεστή  
αντικείμενο **instanceof** Κλάση  
ελέγχουμε αν το αντικείμενο είναι στιγμιότυπο της κλάσης.
- Συνήθως ακολουθεί μετατροπή του αντικειμένου με εκμαγείο.

217

## Παράδειγμα Προσδιορισμού Τύπου

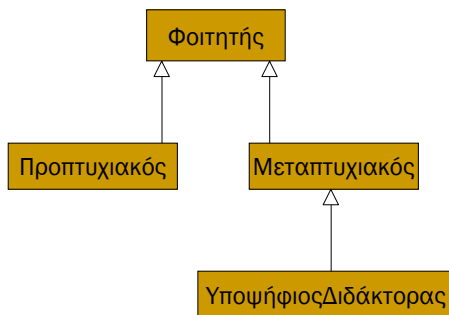
```
CD aCD = new CD("Styx", "Paradise");
ArrayList ar = new ArrayList();
ar.add(aCD);

Object x = ar.get(0); //δεν ξέρω αν είναι CD ή Video
if (x instanceof CD) {
    System.out.println("CD");
    CD someCD = (CD) x; //
}
if (x instanceof Video) {
    System.out.println("Video");
    Video someVideo = (Video) x;
}
if (x instanceof Object)
    //πάντα αληθές
;
if (x instanceof ArrayList)
    System.out.println("συλλογή");
```



218

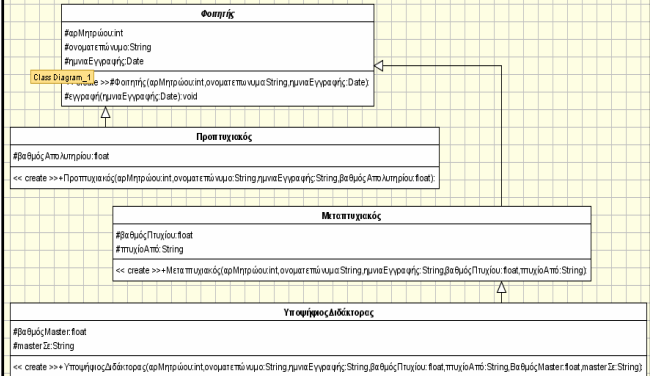
## Μελέτη Περίπτωσης: Φοιτητές



poseidon for uml Αρχείο: Φοιτητές.uml, generate code

219

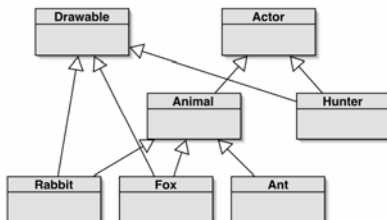
## Διάγραμμα Κλάσεων για Φοιτητές



220

## Πολλαπλή Κληρονομικότητα

- Όταν μια κλάση κληρονομεί από πολλές προγονικές κλάσεις.



- Η Java δεν υποστηρίζει πολλαπλή κληρονομικότητα
  - Μόνον απλή κληρονομικότητα (η κλάση κληρονομεί από έναν πρόγονο)

221

## Ορισμός Διεπαφής (interface)

- Ως πλησιέστερο υποκατάστατο στην πολλαπλή κληρονομικότητα, η Java χρησιμοποιεί τις διεπαφές (interface).
- Στον ορισμό μιας διεπαφής προδιαγράφεται ποιες μεθόδους πρέπει να υλοποιεί κάθε κλάση που 'υπακούει' σε αυτή τη διεπαφή.

```
public interface Actor {
    void add(Object anObject);
    Object get(int index);
}
```

Δεν έχουν σώμα (υλοποίηση) – μόνον επικεφαλίδα.

222

## Χρήση Διεπαφής (interface)

- Μια κλάση υλοποιεί μια ή περισσότερες διεπαφές δηλώνοντας το στην επικεφαλίδα της με τη δεσμευμένη λέξη **implements**.

Απλή κληρονομικότητα

Υλοποιεί διεπαφή

```
public class Fox extends Animals implements Actor {
    void add(Object anObject) {
        ... //κώδικας που υλοποιεί τη μέθοδο
    }
    Object get(int index) ...
}
```

223

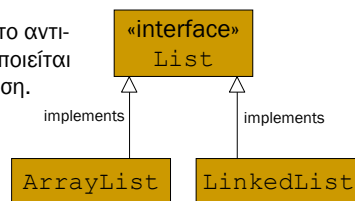
## Η Χρησιμότητα των Διεπαφών

- Παρότι δεν μπορούν να χρησιμοποιηθούν για επαναχρησιμοποίηση κοινού κώδικα,
  - ... αφού περιέχουν μόνον επικεφαλίδες μεθόδων ...
- Επιτρέπουν ενιαίο χειρισμό αντικειμένων των υποκλάσεων (αποφεύγουμε **instanceof**)
- Επιτρέπουν πλήρη διαχωρισμό του ορισμού μιας κλάσης από την υλοποίησή της.
  - Για έναν ορισμό (προδιαγραφή), μπορούμε να έχουμε πολλές υλοποιήσεις.
    - Παράδειγμα: σημεία στο χώρο. Υλοποίηση με boolean πίνακα τριών διαστάσεων ή με συλλογή τριάδων (x,y,z).

224

## Παράδειγμα Διεπαφής

- Η διεπαφή **List** ορίζει την προδιαγραφή που πρέπει να 'σέβονται' όλες οι υλοποιήσεις για λίστες.
  - Μέθοδοι `add`, `clear`, `get`, `size`, `iterator`, `remove`
  - Οι υλοποιήσεις υλοποιούν τη διεπαφή με διαφορετικούς τρόπους.
  - Αρκεί να αλλάξουμε το αντικείμενο, και χρησιμοποιείται διαφορετική υλοποίηση.



225

## Όροι Ενότητας 8

- Κληρονομικότητα
- Σχέση είναι-ένα
- Γονική κλάση
- Ιεραρχία κλάσεων
- Δυναμικά προσδιοριζόμενος τύπος
- Πολλαπλή κληρονομικότητα
- Διεπαφές

- Επαναχρησιμοποίηση

- Ετερογενείς συλλογές
- Κλάσεις περιτυλίγματος
- Πολυμορφισμός

### Τεχνήρια Ενότητας

- **extends**
- **super**
- **protected**
- **instanceof**
- **interface**
- **implements**

226