

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Ομαδοποίηση αντικειμένων

Εισαγωγή στις συλλογές
Γενικές κλάσεις
Iterators

6.0

Οι βασικές έννοιες που θα καλύψουμε

- Συλλογές (ειδικά την `ArrayList`)
- Συνεχίζουμε την αναφορά στο θέμα της *αφαίρεσης (abstraction)* από το προηγούμενο κεφάλαιο.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 2

Γιατί χρειάζεται η ομαδοποίηση αντικειμένων

- Πολλές εφαρμογές περιλαμβάνουν συλλογές από αντικείμενα, π.χ.:
 - Προσωπικά σημειωματάρια (ατζέντες).
 - Κατάλογοι βιβλιοθηκών.
 - Συστήματα εγγραφής σπουδαστών.
- Το πλήθος των στοιχείων που πρέπει να αποθηκεύονται διαφοροποιείται.
 - Προστίθενται στοιχεία.
 - Διαγράφονται στοιχεία.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 3

Παράδειγμα: Ένας κατάλογος μουσικών αρχείων

- Θα επιτρέπει την προσθήκη μουσικών κομματιών στη συλλογή.
- Δεν έχει περιορισμούς στον αριθμό των αρχείων/κομματιών που μπορούν να αποθηκευτούν.
- Θα αναφέρει τον τρέχοντα αριθμό των κομματιών στη συλλογή.
- Θα τυπώνει μια λίστα με όλα τα κομμάτια.
- Μελετήστε το έργο *music-organizer-v1*.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 4

Βιβλιοθήκες κλάσεων

- Συλλογές από χρήσιμες κλάσεις.
- Δεν γράφουμε ολόκληρο τον κώδικα εξαρχής.
- Η Java ονομάζει τις βιβλιοθήκες της *πακέτα (packages)*.
- Η ομαδοποίηση αντικειμένων είναι γενική απαίτηση.
 - Το πακέτο `java.util` περιέχει πολλές κλάσεις που επιτρέπουν την ομαδοποίηση.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 5

```
import java.util.ArrayList;

/**
 * ...
 */
public class MusicOrganizer
{
    // Μια ArrayList για την αποθήκευση ονομάτων
    // μουσικών αρχείων.
    private ArrayList<String> files;

    /**
     * Αρχικοποίηση του
     * καταλόγου αρχείων.
     */
    public MusicOrganizer()
    {
        files = new ArrayList<>();
    }

    ...
}
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 6

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Συλλογές

- Καθορίζουμε:
 - τον τύπο της συλλογής: `ArrayList`
 - τον τύπο των αντικειμένων που θα περιέχει:
`<String> private ArrayList<String> files;`
- Λέμε ότι έχουμε μια "ArrayList of String", μια πινακοποιημένη λίστα αλφαριθμητικών.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

7

Γενικές κλάσεις

- Οι συλλογές είναι γνωστές ως *παραμετροποιημένοι (parameterized) ή γενικοί (generic) τύποι*.
- Η `ArrayList` υλοποιεί τη λειτουργία μιας λίστας:
 - `add`, `get`, `size`, κ.λπ.
- Η παράμετρος του τύπου δηλώνει τι είδους αντικείμενα θέλουμε να περιλαμβάνει η λίστα:
 - `ArrayList<Person>`
 - `ArrayList<TicketMachine>`
 - κ.λπ.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

8

Δημιουργία αντικειμένου ArrayList

- Στις εκδόσεις της Java που προηγούνταν της έκδοσης 7:
 - `files = new ArrayList<String>();`
- Η Java 7 εισήγαγε τη *σημειογραφία διαμαντιού (diamond notation)*
 - `files = new ArrayList<>();`
- Ο τύπος των στοιχείων της συλλογής εξάγεται αυτόματα από τον τύπο της μεταβλητής.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

9

Δομές αντικειμένων με συλλογές

```
classDiagram
    class myMusic_MusicOrganizer {
        files ArrayList<String>
    }
    class ArrayList_String
    class String_MorningBlues_mp3["MorningBlues.mp3"]
    class String_DontGo_mp3["DontGo.mp3"]
    myMusic_MusicOrganizer --> ArrayList_String
    ArrayList_String --> String_MorningBlues_mp3
    ArrayList_String --> String_DontGo_mp3
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

10

Προσθήκη τρίτου αρχείου

```
classDiagram
    class myMusic_MusicOrganizer {
        files ArrayList<String>
    }
    class ArrayList_String
    class String_MorningBlues_mp3["MorningBlues.mp3"]
    class String_DontGo_mp3["DontGo.mp3"]
    class String_MatchBoxBlues_mp3["MatchBoxBlues.mp3"]
    myMusic_MusicOrganizer --> ArrayList_String
    ArrayList_String --> String_MorningBlues_mp3
    ArrayList_String --> String_DontGo_mp3
    ArrayList_String --> String_MatchBoxBlues_mp3
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

11

Χαρακτηριστικά της συλλογής

- Αυξάνει την εσωτερική χωρητικότητα της ανάλογα με τις απαιτήσεις.
- Διατηρεί έναν δικό της μετρητή:
 - τη μέθοδο πρόσβασης `size()`.
- Διατηρεί τη σειρά των στοιχείων που αποθηκεύονται.
- Οι λεπτομέρειες για το πώς γίνονται αυτά είναι κρυφές.
 - Έχει σημασία; Το ότι δεν τις γνωρίζουμε μας εμποδίζει να χρησιμοποιούμε αυτές τις δυνατότητες;

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

12

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Γενικές κλάσεις

- Μπορούμε να χρησιμοποιήσουμε την **ArrayList** με **οποιοδήποτε** τύπο κλάσης:
 - `ArrayList<TicketMachine>`
 - `ArrayList<ClockDisplay>`
 - `ArrayList<Track>`
 - `ArrayList<Person>`
- Σε καθεμία αποθηκεύονται πολλά αντικείμενα συγκεκριμένου τύπου.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 13

Χρήση της συλλογής

```
public class MusicOrganizer
{
    private ArrayList<String> files;

    ...

    public void addFile(String filename)
    {
        files.add(filename);
    }

    public int getNumberOfFiles()
    {
        return files.size();
    }

    ...
}
```

Προσθήκη νέου αρχείου

Επιστροφή του αριθμού των αρχείων (εκκώρυξη)

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 14

Αριθμοδείκτες

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 15

Ανάκτηση στοιχείων από τη συλλογή

```
public void listFile(int index)
{
    if(index >= 0 && index < files.size()) {
        String filename = files.get(index);
        System.out.println(filename);
    }
    else {
        // This is not a valid index.
    }
}
```

Έλεγχος εγκυρότητας αριθμοδείκτη

Ανάκτηση και εκτύπωση του ονόματος αρχείου

Είναι απαραίτητο; (Μήνυμα σφάλματος;)

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 16

Η αφαίρεση στοιχείων επηρεάζει την αριθμοδεικτοδότηση

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 17

Ανασκόπηση

- Οι συλλογές επιτρέπουν την αποθήκευση ενός οσοδήποτε μεγάλου αριθμού αντικειμένων.
- Οι βιβλιοθήκες κλάσεων περιέχουν κλάσεις συλλογών οι οποίες είναι δοκιμασμένες και λειτουργούν σωστά.
- Οι βιβλιοθήκες κλάσεων της Java ονομάζονται *πακέτα* (*packages*).
- Χρησιμοποιήσαμε την κλάση **ArrayList** από το πακέτο **java.util**.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 18

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Ανασκόπηση

- Υπάρχει δυνατότητα για προσθήκη και διαγραφή στοιχείων.
- Σε κάθε στοιχείο αντιστοιχεί ένας αριθμοδείκτης.
- Οι τιμές των αριθμοδεικτών ενδέχεται να αλλάξουν σε περίπτωση διαγραφής (ή προσθήκης περαιτέρω) στοιχείων.
- Οι κύριες μέθοδοι της `ArrayList` είναι οι `add`, `get`, `remove` και `size`.
- Η `ArrayList` είναι *παραμετροποιημένη* (parameterized) ή *γενική* (generic).

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 19

Ομαδοποίηση αντικειμένων

Συλλογές και ο βρόχος *for-each*

«Επανάληψη» σε συλλογές

- Στις συλλογές, συχνά θέλουμε να επαναλάβουμε κάποιες ενέργειες: *ακριβώς μία φορά για κάθε αντικείμενο που περιέχεται στη συλλογή.*

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 21

Ψευδοκώδικας Βρόχου *for-each*

Γενική μορφή του βρόχου *for-each*

for (Λέξη-κλειδί του βρόχου) η επικεφαλίδα του βρόχου

for (ΤύποςΣτοιχείου στοιχείο : συλλογή) {

 σώμα βρόχου

Ενέργειες που πρέπει να επαναληφθούν

Ψευδοκώδικας για τη λειτουργία του βρόχου *for-each*

για κάθε στοιχείο μέσα στη συλλογή κάνε: {

 σώμα βρόχου

}

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 22

Ένα παράδειγμα στην Java

```
/**
 * Εμφάνιση της λίστας όλων των αρχείων στη
 * συλλογή.
 */
public void listAllFiles()
{
    for(String filename : files) {
        System.out.println(filename);
    }
}
```

Για κάθε όνομα αρχείου που περιέχεται στη συλλογή *files* διαδοχικά, τύπωσε το όνομα αρχείου

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 23

Ανασκόπηση

- Ο βρόχος *for-each* επιτρέπει την πραγματοποίηση επαναλήψεων σε ολόκληρη τη συλλογή.
- Στον βρόχο *for-each*, *κάθε* αντικείμενο είναι στη διάθεση των εντολών που περιέχονται στο σώμα του βρόχου *ακριβώς μία φορά.*

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 24

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Επιλεκτική επεξεργασία

- Οι εντολές μπορούν να είναι ένθετες, κάτι που επιτρέπει μεγαλύτερη δυνατότητα επιλογής ως προς τις ενέργειες που θα εκτελεστούν:

```
public void findFiles(String searchString)
{
    for(String filename : files) {
        if(filename.contains(searchString)) {
            System.out.println(filename);
        }
    }
}
```

Η `contains` δίνει μερική ταύτιση με το όνομα αρχείου - χρησιμοποιήστε την `equals` για απόλυτη ταύτιση

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 25

Χρήση “break” και “continue”

- Ισοδύναμος κώδικας:

```
public void findFiles(String searchString)
{
    for(String filename : files) {
        if(!filename.contains(searchString))
            continue;
        System.out.println(filename);
    }
}
```

Η `contains` δίνει μερική ταύτιση με το όνομα αρχείου - χρησιμοποιήστε την `equals` για απόλυτη ταύτιση

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 26

Ο ισοδύναμος “while”-Βρόχος

```
/**
 * Εμφάνιση της λίστας όλων των αρχείων στη
 * συλλογή.
 */
public void listAllFiles()
{
    int index = 0;
    while(index < files.size()) {
        String filename = files.get(index);
        System.out.println(filename);
        index++;
    }
}
```

Προσαύξησε τον δείκτη κατά 1

ενόσω η τιμή του δείκτη είναι μικρότερη από το μέγεθος της συλλογής, ανάκτησε και τύπωσε το επόμενο όνομα αρχείου, και κατόπιν προσαύξησε τον δείκτη

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 27

Ο ισοδύναμος “for”-Βρόχος

```
/**
 * Εμφάνιση της λίστας όλων των αρχείων στη
 * συλλογή.
 */
public void listAllFiles()
{
    for (int index=0; index < files.size(); index++)
    {
        String filename = files.get(index);
        System.out.println(filename);
    }
}
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 28

Αναζήτηση σε συλλογή (με “while”)

```
int index = 0;
boolean found = false;
while(index < files.size() && !found) {
    String file = files.get(index);
    if(file.equals(searchString)) {
        // Δεν χρειάζεται να συνεχίσουμε την
        // αναζήτηση.
        found = true;
    }
    else {
        index++;
    }
}
// Είτε βρήκαμε αυτό που αναζητούσαμε στην τιμή
// index, είτε εξαντλήσαμε τη συλλογή.
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 29

Αναζήτηση σε συλλογή (με “for each”)

```
boolean found = false;
for (String f: files) {
    if(f.equals(searchString)) {
        // Δεν χρειάζεται να συνεχίσουμε την
        // αναζήτηση.
        found = true;
        break;
    }
}
// Εάν βρήκαμε αυτό που αναζητούσαμε, τότε η
// μεταβλητή found λαμβάνει τιμή true.
// Δεν έχουμε ένδειξη σε ποια θέση βρέθηκε!
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 30

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Ομαδοποίηση αντικειμένων

Αντικείμενα Iterator

Iterator και iterator()

- Οι συλλογές έχουν μέθοδο `iterator()`.
- Αυτή επιστρέφει αντικείμενο `Iterator`.
- Η `Iterator<E>` έχει τις μεθόδους:
 - `boolean hasNext()`
 - `E next()`
 - `void remove()`

Χρήση ενός αντικειμένου Iterator

`java.util.Iterator` επιστρέφει ένα αντικείμενο `Iterator`

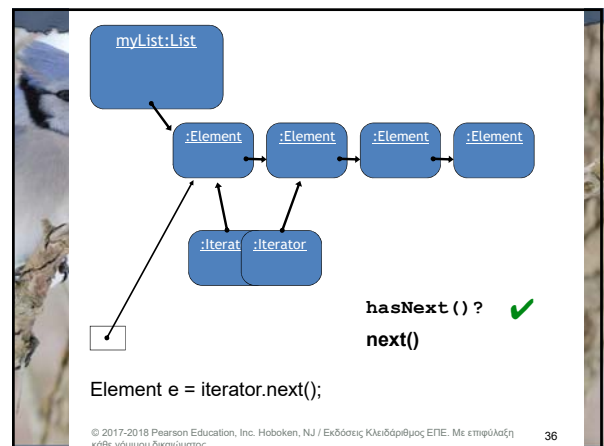
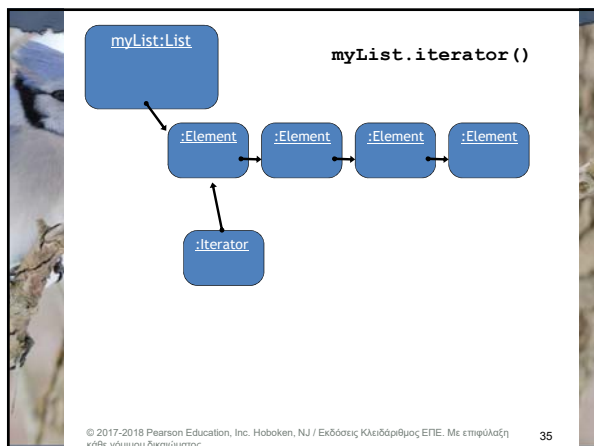
```
Iterator<ElementType> it = myCollection.iterator();
while(it.hasNext()) {
    κλήση της it.next() για την ανάκτηση του επόμενου στοιχείου
    εκτέλεση ενέργειας με αυτό το στοιχείο
}

public void listAllFiles()
{
    Iterator<Track> it = files.iterator();
    while(it.hasNext()) {
        Track tk = it.next();
        System.out.println(tk.getDetails());
    }
}
```

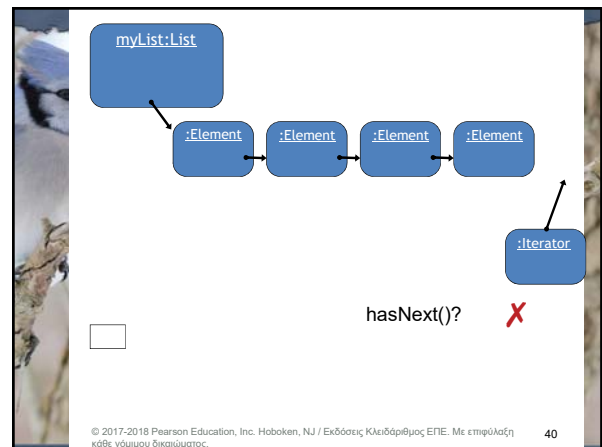
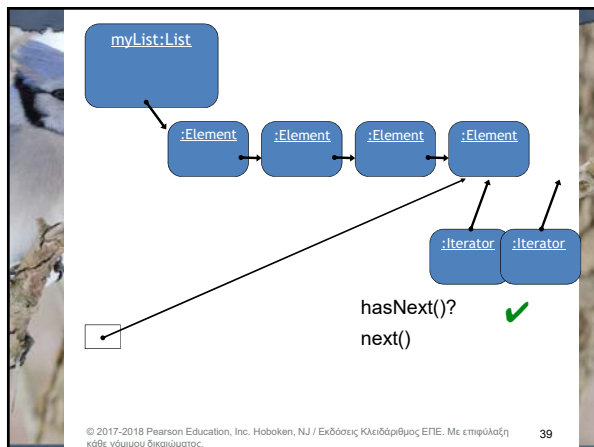
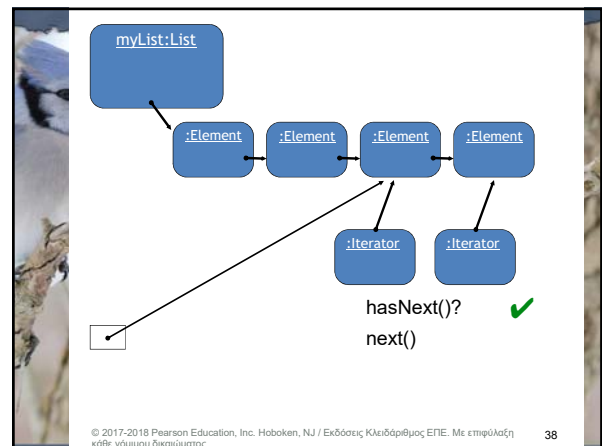
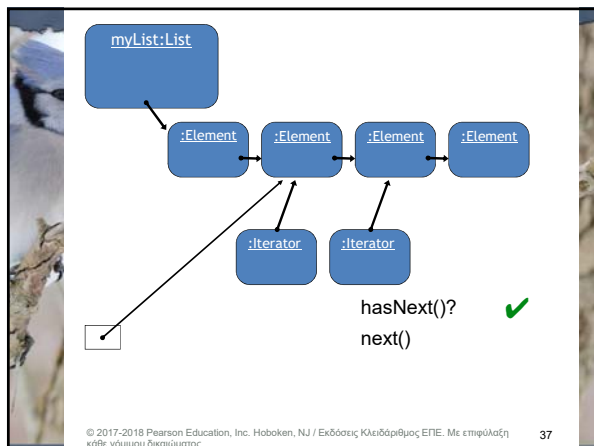
© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 33

Λειτουργία των Iterator

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 34



Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15



Αριθμοδείκτης ή Iterator

- Τρόποι για εκτέλεση επαναλήψεων σε μια συλλογή:
 - Βρόχος *for-each*.
 - Τον χρησιμοποιούμε αν θέλουμε να επεξεργαστούμε όλα τα στοιχεία της συλλογής.
 - Βρόχος *for* ή *while*.
 - Τον χρησιμοποιούμε αν ενδέχεται να θελήσουμε να σταματήσουμε σε κάποιο σημείο.
 - Τον χρησιμοποιούμε για επανάληψη εκτός συλλογών.
 - Αντικείμενο **Iterator**.
 - Τον χρησιμοποιούμε αν ενδέχεται να θελήσουμε να σταματήσουμε σε κάποιο σημείο.
 - Συχνά χρησιμοποιείται σε συλλογές στις οποίες η προσπέλαση με αριθμοδείκτη δεν είναι αποδοτική ή δεν είναι καν δυνατή.
 - Το χρησιμοποιούμε για να διαγράψουμε από συλλογές.
- Η επανάληψη είναι σημαντικό μοτίβο που χρησιμοποιούμε στον προγραμματισμό.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 41

Διαγραφή στοιχείων από συλλογή

```

Iterator<Track> it = tracks.iterator();
while(it.hasNext()) {
    Track t = it.next();
    String artist = t.getArtist();
    if(artist.equals(artistToRemove)) {
        it.remove();
    }
}
    
```

Με τη μέθοδο **remove** του **Iterator**.

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 42

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #15

Διαγραφή στοιχείων από συλλογή - λάθος τρόπος!

```
int index = 0;
while(index < tracks.size()) {
    Track t = tracks.get(index);
    String artist = t.getArtist();
    if(artist.equals(artistToRemove)) {
        tracks.remove(index);
    }
    index++;
}
```

Μπορείτε να βρείτε το λάθος;

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 43

Το έργο *auction*

```
classDiagram
    class Auction {
        bid() Bid
    }
    class Bid {
        bidder() Person
    }
    class List {
        bid() Bid
    }
    class Person {
        bid() Bid
    }
    Auction --> Bid
    Bid --> Person
    List --> Bid
    Person --> Bid
```

© 2017-2018 Pearson Education, Inc. Hoboken, NJ / Εκδόσεις Κλειδάριθμος ΕΠΕ. Με επιφύλαξη κάθε νόμιμου δικαιώματος. 44