



Προγραμματισμός II (Java)

3. Διαχείριση πολλαπλών αντικειμένων

Πίνακες

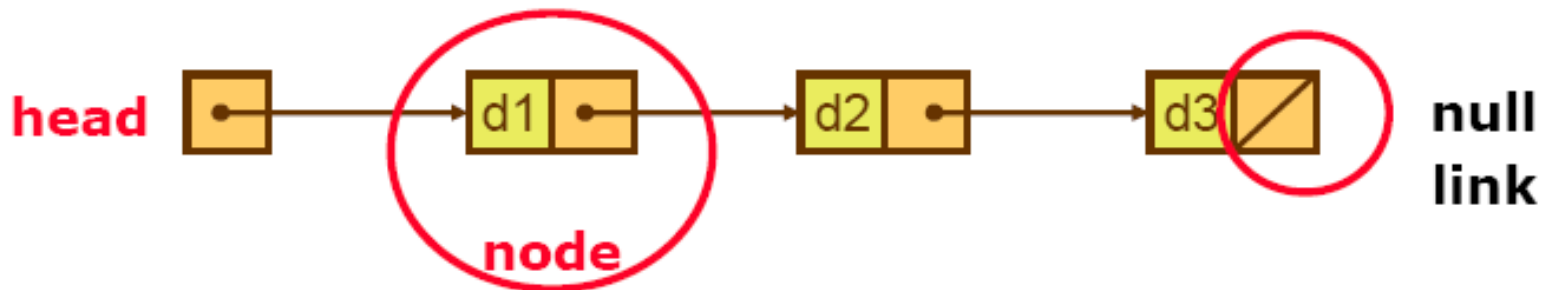
- Προκαθορισμένο μέγεθος
- Συνεχείς θέσεις
- Πίνακες τιμών
 - Αρχικοποιούνται σε 0 ή false
 - `int[] a=new int[10];`
 - `boolean[] b=new boolean[10];`
- Πίνακες αντικειμένων
 - Αρχικοποιούνται σε null
 - `String[] c= new String[10]`

Περιορισμοί των πινάκων

- Οι εισαγωγές και οι διαγραφές σε κάποια θέση «κοστίζουν»
 - Πρέπει να μετακινηθούν όλα τα υπόλοιπα στοιχεία για να καλύψουν το κενό ή να δημιουργήσουν χώρο.
 - Αν γεμίσουν οι θέσεις θα πρέπει να αντιγράψουμε όλα τα στοιχεία σε νέο, μεγαλύτερο πίνακα.
- Η λογική διαδοχή των θέσεων του πίνακα ταυτίζεται με την πραγματική (ο πίνακας αποθηκεύεται σε συνεχόμενες θέσεις μνήμης)
- Θέλουμε να αποσυνδέσουμε τη λογική από την πραγματική διαδοχή θέσεων.
 - Έτσι θα έχουμε διαγραφές και εισαγωγές χωρίς μετακίνηση

Δυναμικές δομές δεδομένων

- Περιέχουν αντικείμενα (οποιοδήποτε τύπου)
 - Επιτρέπουν την εισαγωγή αντικειμένων με περισσότερες ελευθερίες από τους πίνακες
- Λίστες
 - Κάθε στοιχείο (κόμβος) αποτελείται από το αντικείμενο και από ένα σύνδεσμο στο «επόμενο» στοιχείο
 - Το τελευταίο στοιχείο δείχνει σε null



Υλοποιήσεις

■ Η κλάση `Vector`

- Επιτρέπει να φτιάξουμε δομές όπως οι πίνακες που το μέγεθός τους αυξομειώνεται δυναμικά.

■ Η κλάση `ArrayList`

- Υλοποιεί τη δομή της λίστας στη Java
- Χρησιμοποιεί πίνακες τους οποίους διπλασιάζει σε μέγεθος δυναμικά, όποτε απαιτείται

Η κλάση ArrayList

- Προτείνεται από τους δημιουργούς της Java
- **ArrayList list = new ArrayList();**
- ή
- **ArrayList<type> list = new ArrayList<>();**
- **add(Object searchkey), get(int position)**

```
Iterator it = list.iterator();
```

```
StringBuffer buf = new StringBuffer();
```

```
while (it.hasNext())
```

```
    buf.append( it.next() ).append( " " );
```

```
System.out.println(buf.toString());
```

Άλλες μέθοδοι της ArrayList

void **clear**(); //αδειάζει τη λίστα

Object **clone**(); //δημιουργεί αντίγραφο

boolean **addAll**(Collection c); //προσθέτει όλα τα
//αντικείμενα της c στη λίστα

boolean **contains**(Object elem); // αναζητά το elem

int **indexOf**(Object elem); //βρίσκει την θέση 1^{ης}
//εμφάνισης

Object **remove**(int index); //διαγράφει στοιχείο

Object[] **toArray**(); // επιστρέφει τα στοιχεία της
//λίστας σε πίνακα αντικειμένων

Μειονέκτημα των containers

(Vector, ArrayList κλπ).

- Αν δε χρησιμοποιηθεί ο περιορισμός τύπου:
 - Όλα τα στοιχεία αποθηκεύονται ως Object
 - Πρέπει να μετατραπούν (cast) για να χρησιμοποιηθούν οι μέθοδοί τους.
- Τα containers μπορεί να περιέχουν αντικείμενα διαφόρων κλάσεων

```
list.add(new Employee());
```

```
list.add(new Address());
```

```
list.add(new Manager());
```

```
((Employee)list.get(0)).getName();
```

```
((Address)list.get(1)).getCity();
```

```
((Address)list.get(2)).getCity(); //ΛΑΘΟΣ casting
```

Όπως πάντα... το μειονέκτημα ...

έγινε τελικά πλεονέκτημα



Χρήσιμες μέθοδοι

Εισαγωγή με έλεγχο διπλοτύπων

- Η εισαγωγή με έλεγχο διπλοτύπων θα γίνει σε δύο φάσεις:
 - Αναζήτηση για το αν υπάρχει ή όχι το αντικείμενο στον πίνακα ή τη λίστα
 - Εισαγωγή – στο τέλος ή σε κενή θέση - σε περίπτωση που δεν υπάρχει
- Υπάρχουν πιο γρήγορες λύσεις;
- Η ArrayList διαθέτει μεθόδους:
 - `int indexOf(Object elem)`: επιστρέφει τη θέση πρώτης εμφάνισης του `elem` στη λίστα
 - `int lastIndexOf(Object elem)`
 - `boolean contains(Object elem)` : επιστρέφει `true` αν η λίστα περιέχει το `elem`
 - `Object remove(int index)` : διαγράφει το στοιχείο στη θέση `index` της λίστας και μας το επιστρέφει

Σύγκριση: Η μέθοδος equals

- Για να δουλέψουν οι προηγούμενοι μέθοδοι για λίστες με αντικείμενα δικών μας κλάσεων πρέπει στις κλάσεις μας να έχουμε μια μέθοδο equals π.χ.

```
public boolean equals(Object o){
    Department d=(Department)o; // πιθανό να παράγει
                                   //ClassCastException
    if (this.id==d.getId() && this.name.equals(d.getName()) &&
        this.numStudents==d.getNumStudents())
        return true;
    else
        return false;
}
```

Ταξινόμηση

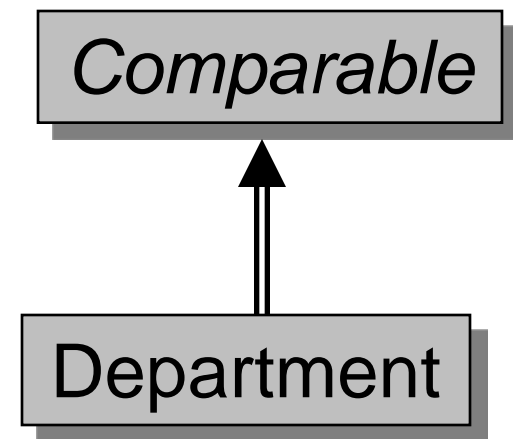
- Με ποιο τρόπο μπορώ να ταξινομήσω τα στοιχεία ενός πίνακα ή μιας λίστας; BubbleSort:

```
public void bubbleSort(int[] unsortedArray, int length) {  
    int temp, counter, index;  
    for(counter=0; counter<length-1; counter++) {  
        for(index=0; index<length-1-counter; index++) {  
            if(unsortedArray[index] > unsortedArray[index+1]) {  
                temp = unsortedArray[index];  
                unsortedArray[index] = unsortedArray[index+1];  
                unsortedArray[index+1] = temp;  
            }  
        }  
    }  
}
```

Διάταξη

- Η διάταξη στους ακεραίους είναι δεδομένη
- Τι γίνεται όμως με τις δικές μας κλάσεις;
- Πώς μπορούμε να ορίσουμε διάταξη στα αντικείμενά τους;

```
public interface Comparable
{
    int compareTo(Object o);
}
```





```
public class Department implements Comparator{
```

```
...
```

```
    public int compareTo(Object o){
```

```
        Department d=(Department)o; // πιθανό να παράγει
```

```
                                     //ClassCastException
```

```
        if (this.numStudents>d.getNumStudents())
```

```
            return 1;
```

```
        else if (this.numStudents<d.getNumStudents())
```

```
            return -1;
```

```
        else
```

```
            return 0;
```

```
    }
```

```
}
```

Ταξινόμηση

```
Collections.sort(allDeps);
```

Ταξινομεί τα τμήματα με βάση τον αριθμό σπουδαστών που έχουν

Χρησιμοποιεί την QuickSort



Χρήση κώδικα τρίτων

http://jsoup.org/

- Έτοιμη βιβλιοθήκη για το χειρισμό του περιεχομένου ιστοσελίδων

[jsoup](#) [News](#) [Bugs](#) [Discussion](#) [Download](#) [API Reference](#) [Cookbook](#) [Try jsoup](#)

[jsoup](#) » [Download jsoup](#)

Download jsoup

[jsoup](#) is available as a downloadable `.jar` java library. The current release version is **1.8.1**.

- [jsoup-1.8.1.jar](#) core library
- [jsoup-1.8.1-sources.jar](#) optional sources jar
- [jsoup-1.8.1-javadoc.jar](#) optional javadoc jar

Cookbook contents

Introduction

1. Parsing and traversing a Document

Input

2. Parse a document from a String
3. Parsing a body fragment
4. Load a Document from a URL
5. Load a Document from a File

Extracting data

6. Use DOM methods to navigate a document
7. Use selector-syntax to find elements
8. Extract attributes, text, and HTML from elements
9. Working with URLs
10. Example program: list links

Modifying data

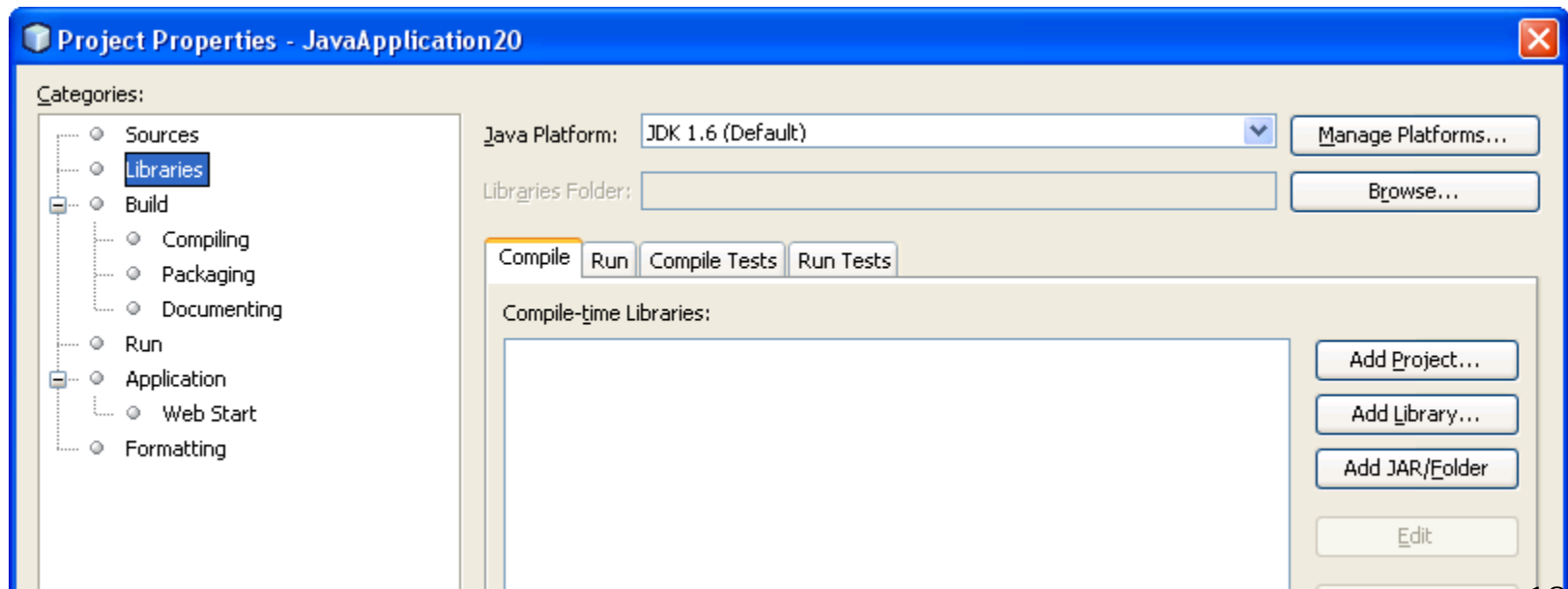
11. Set attribute values
12. Set the HTML of an element
13. Setting the text content of elements

Cleaning HTML

14. Sanitize untrusted HTML (to prevent XSS)

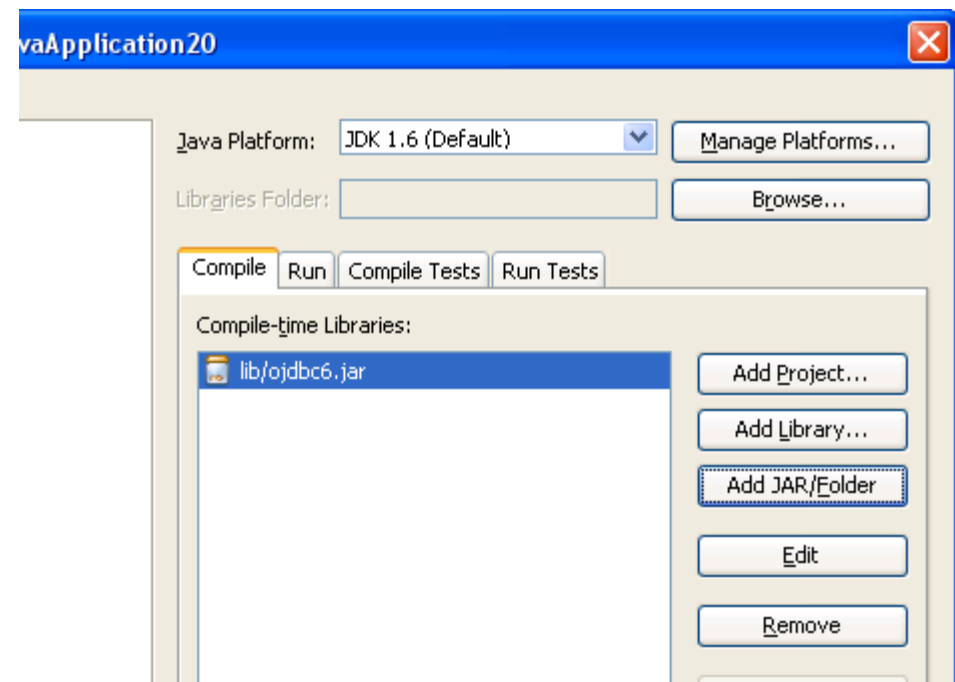
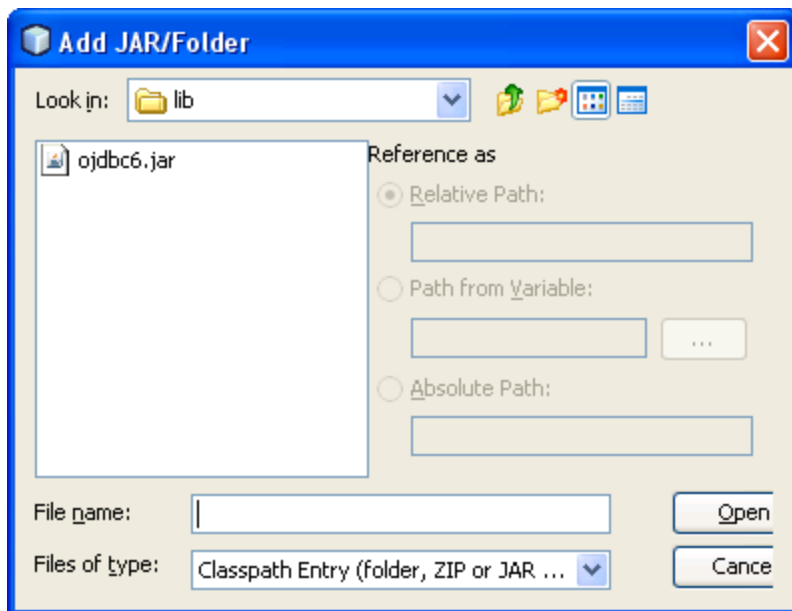
Προσθήκη package στο Netbeans

- Κατεβάζουμε ένα μεταγλωττισμένο package για σύνδεση με την Oracle
- Βάζουμε το αρχείο σε ένα φάκελο lib εντός του project
- Ανοίγουμε τα properties του project και πηγαίνουμε στην επιλογή Libraries



Προσθήκη package στο Netbeans

- Επιλέγουμε Add JAR/Folder και εντοπίζουμε το αρχείο που κατεβάσαμε





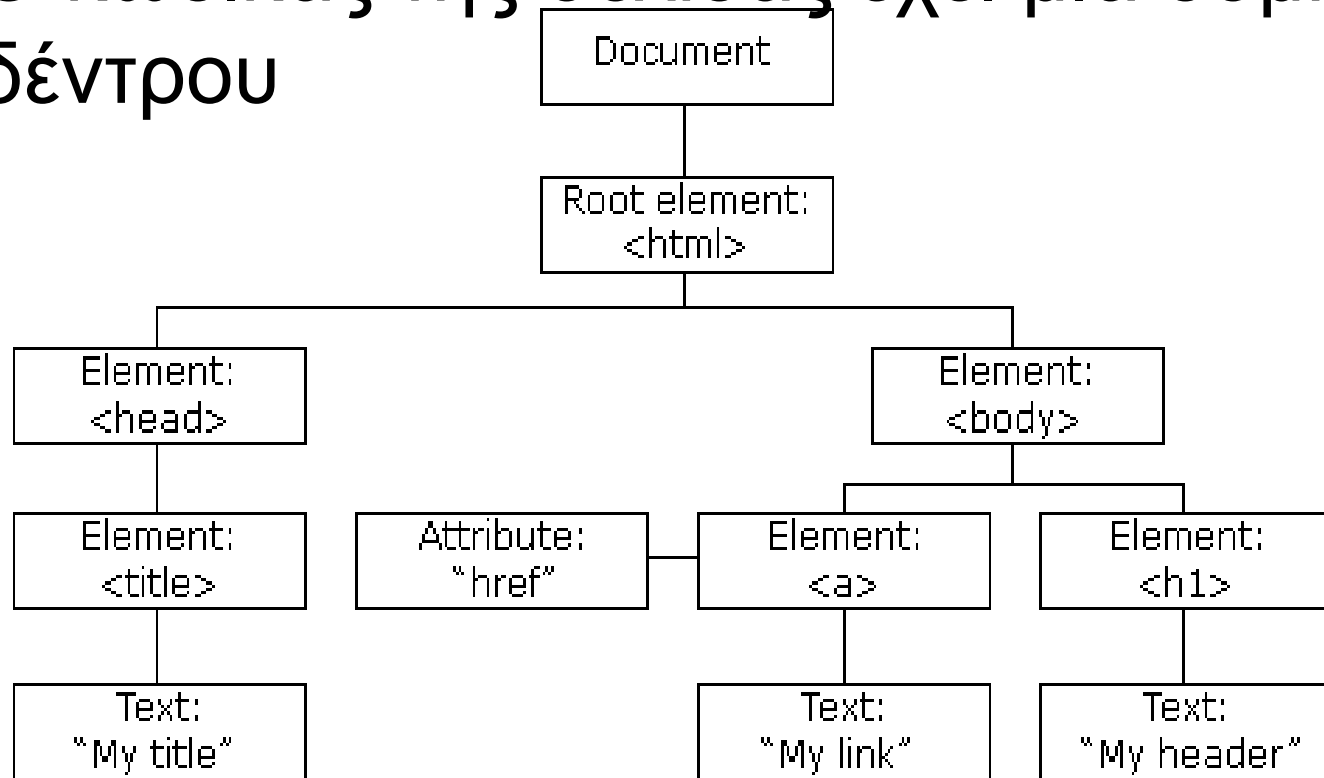
Βασικά στοιχεία HTML

Παράδειγμα HTML

```
<html>
  <head>
    <title>My Title</title>
  </head>
  <body>
    <h1>My header</h1>
    <a href="http://www.hua.gr">My link</a>
  </body>
</html>
```

HTML DOM

- Ο κώδικας της σελίδας έχει μια δομή δέντρου



DOM Nodes

- Τα πάντα στη σελίδα είναι κόμβοι DOM
- Το ίδιο το έγγραφο (document node)
- Οι ετικέτες (tags) της σελίδας (element node)
- Το περιεχόμενο των ετικετών (text nodes)
- Οι παράμετροι και οι τιμές τους (attribute node)
- Τα σχόλια (comment node)

Σημαντικοί κόμβοι (elements)

- Root (Document)
- Γείτονες ενός Node
 - Parents
 - Children
 - Siblings
- Leaf nodes
 - Κόμβοι χωρίς παιδιά
- Div, Span: ομαδοποιούν περιοχές με κοινό περιεχόμενο

Σημαντικά attributes

- **id**

- Μοναδικό σε όλο το έγγραφο

- **class**

- Επαναλαμβάνεται σε κείμενα που έχουν κοινή μορφοποίηση



Εργασία

Χρήσιμοι σύνδεσμοι

- <https://transport.opendata.ch/>
- <http://jsonviewer.stack.hu/>
- <https://code.google.com/archive/p/json-simple/>
- http://www.tutorialspoint.com/json/json_java_example.htm
- <http://www.mkyong.com/java/json-simple-example-read-and-write-json/>
- <http://www.mkyong.com/java/how-to-send-http-request-getpost-in-java/>

JSON example

- “JSON” stands for “JavaScript Object Notation”
 - Despite the name, JSON is a (mostly) language-independent way of specifying objects as name-value pairs
- Example (http://secretgeek.net/json_3mins.asp):
 - ```
{ "skillz": {
 "web": [
 { "name": "html",
 "years": 5
 },
 { "name": "css",
 "years": 3
 }
]
 "database": [
 { "name": "sql",
 "years": 7
 }
]
}}
```

# JSON syntax, I

- An *object* is an unordered set of name/value pairs
  - The pairs are enclosed within braces, { }
  - There is a colon between the name and the value
  - Pairs are separated by commas
  - Example: { "name": "html", "years": 5 }
- An *array* is an ordered collection of values
  - The values are enclosed within brackets, [ ]
  - Values are separated by commas
  - Example: [ "html", "xml", "css" ]

# JSON syntax, II

- A *value* can be: A string, a number, true, false, null, an object, or an array
  - Values can be nested
- *Strings* are enclosed in double quotes, and can contain the usual assortment of escaped characters
- *Numbers* have the usual C/C++/Java syntax, including exponential (E) notation
  - All numbers are decimal--no octal or hexadecimal
- *Whitespace* can be used between any pair of tokens

# Java project alternatives

- Δύο εναλλακτικές για τα project σας σε Netbeans (και αλλού):
  - Ant: Εσείς κατεβάζετε τα third party libraries και τα αντιγράφετε στο project
  - Maven: Αναζητάτε τα third party libraries (dependencies) σε maven repositories.

# Βήματα

- Δημιουργούμε ένα νέο Maven Project
- Δημιουργούμε το package basics με τις βασικές κλάσεις
- Προσθέτουμε τα dependencies για jsoup και json-simple
- Κάνουμε το request με java ή και με jsoup
- Επεξεργαζόμαστε την απάντηση με json-simple.