

ΕΠΛ 133: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΕΡΓΑΣΤΗΡΙΟ 2

Javadoc Tutorial

Introduction

Το Javadoc είναι ένα εργαλείο που παράγει αρχεία html (παρόμοιο με τις σελίδες στη διεύθυνση http://docs.oracle.com/javase/8/docs/api/index.html) από τα σχόλια του κώδικα. Σε αυτό το εργαστήριο θα δούμε πως γράφουμε Javadoc σχόλια στο κώδικα και πώς να χρησιμοποιήσουμε τις δυνατότητες που μας παρέχει η Javadoc για να κάνουμε τεκμηρίωση του κώδικα παράγοντας αρχεία html. Σε όλες τις εργασίες του μαθήματος ΕΠΛ133, θα χρησιμοποιήσετε τον τρόπο που περιγράφεται σε αυτό το εργαστήριο για να κάνετε την τεκμηρίωση του κώδικα.

Javadoc Comments

Η Javadoc αναγνωρίζει ειδικά σχόλια /** */ , που επισημαίνονται με μπλε χρώμα στο Eclipse (τα κανονικά σχόλια // και /*...*/ επισημαίνονται με πράσινο χρώμα). Η Javadoc επιτρέπει να βάλουμε σχόλια στις κλάσεις, τους κατασκευαστές, τα παιδία, τις διεπαφές και τις μεθόδους.

Javadoc Tags

Για να μετατρέψει τα σχόλια σε μορφή αρχείων html, η Javadoc χρησιμοποίει ετικέτες (tags). Οι ετικέτες είναι λέξεις κλειδιά που αναγνωρίζονται από τη Javadoc και χρησιμοποιούνται ανάλογα με τον τύπο της πληροφορίας που ακολουθεί.

Οι πιο χρησιμοποιημένες ετικέτες είναι:

- @author [author name] identifies author(s) of a class or interface.
- @version [version] version info of a class or interface.
- @param [argument name] [argument description] describes an argument of method or constructor.
- @return [description of return] describes data returned by method (unnecessary for constructors and void methods).
- @exception [exception thrown] [exception description] describes exception thrown by method.
- @throws [exception thrown] [exception description] same as @exception.

Javadoc in Eclipse

Η Eclipse μπορεί να παράγει σχόλια Javadoc για κλάσεις και μεθόδους. Για να γίνει αυτό:

- 1. Τοποθετήστε το δείκτη στη κλάση ή τη μέθοδο
- 2. Δεξί click Source -> Generate Element Comment
- 3. Παράγονται σχόλια Javadoc, αλλά πρέπει να γράψετε τις περιγραφές

ΕΠΛ 133: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Για να παράγει η Eclipse τα αρχεία html:

- 1. Διαλέγετε το Project στη λίστα των project αριστερά
- 2. Μετά, στο menu του Eclipse επιλέξτε: Project->Generate Javadoc
- 3. Αν είναι η πρώτη φορά που κάνετε αυτήν την διαδικασία, το Eclipse θα σας ζητήσει να επιλέξετε την εντολή. Οπότε στο παράθυρο που ανοίγει, πρέπει να πατήσετε στο κουμπί Configure και να βρείτε το πρόγραμμα javadoc το οποίο είναι στο κατάλογο που έχετε εγκαταστήσει τη Java JDK.
- 4. Finish

Types of Javadoc comments

Υπάρχουν 2 είδη σχολίων Javadoc: σχόλια σχετικά με τις κλάσεις (class-level comments) και σχόλια σχετικά με τα μέλη της κλάσης (member-level comments). Τα σχόλια κλάσεων περιγράφουν τις κλάσεις και τα σχόλια μελών περιγράφουν τα χαρακτηριστικά και τις μεθόδους των κλάσεων.

Class-level Comments

Τα σχόλια κλάσεων παρέχουν μια περιγραφή της κλάσης, και τοποθετούνται ακριβώς πάνω από τον κώδικα που δηλώνει την κλάση. Αυτά τα σχόλια γενικά περιέχουν τις ετικέτες @author, και μια περιγραφή της κλάσης. Ένα παράδειγμα σχόλιων κλάσεων είναι το παρακάτω:

```
* Brief summary of this class, ending with a period.
 * It is common to leave a blank line between the summary and further details.
 * The summary (everything before the first period) is used in the class or package
 * overview section.
 * The following inline tags can be used (not an exhaustive list):
 * {@link some.other.class.Documentation} for linking to other docs or symbols
 * \{ @ code code goes here \} for formatting as code
 * {@literal <>[]()foo} for interpreting literal text without converting to HTML
markup
 * or other tags.
 * Optionally, the following tags may be used at the end of class documentation
  (not an exhaustive list):
 * @author John Doe
 * @version 1.0
 * @since 5/10/15
 * @see some.other.class.Documentation
 * @deprecated This class has been replaced by some.other.package.BetterFileReader
public class FileReader {
```

Με τον ίδιο τρόπο μπορούμε να σχολιάσουμε τις διεπαφές (interface) και enums.



ΕΠΛ 133: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Member-level Comments

Τα σχόλια μελών, περιγράφουν τα πεδία, τις μεθόδους, και κατασκευαστές. Τα σχόλια πριν τις μεθόδους και τους κατασκευαστές μπορεί να περιέχουν ετικέτες που περιγράφουν τις παραμέτρους της μεθόδου. Το σχόλιο μεθόδου μπορεί επίσης να περιέχει ετικέτες επιστροφής @return. Ένα παράδειγμα είναι οι παρακάτω:

```
* Brief summary of method, ending with a period.
 * Further description of method and what it does, including as much detail as is
 * appropriate. Inline tags such as
 * {@code code here}, {@link some.other.Docs}, and {@literal text here} can be used.
 * If a method overrides a superclass method, {@inheritDoc} can be used to copy the
 * documentation
 * from the superclass method
 * @param stream Describe this parameter. Include as much detail as is appropriate
                 Parameter docs are commonly aligned as here, but this is optional.
                 As with other docs, the documentation before the first period is
                 used as a summary.
  @return Describe the return values. Include as much detail as is appropriate
           Return type docs are commonly aligned as here, but this is optional.
           As with other docs, the documentation before the first period is used as a
           summary.
  @throws IOException Describe when and why this exception can be thrown.
                       Exception docs are commonly aligned as here, but this is
                       optional.
                       As with other docs, the documentation before the first period
                       is used as a summary.
                       Instead of @throws, @exception can also be used.
 * @since 2.1.0
 * @see some.other.class.Documentation
 * @deprecated Describe why this method is outdated. A replacement can also be
specified.
public String[] read(InputStream stream) throws IOException {
   return null;
```

Package Documentation

Από την έκδοση 1.5 της Java, είναι δυνατό να κάνουμε τεκμηρίωση του πακέτου (package) χρησιμοποιώντας ένα αρχείο που ονομάζεται package-info.java. Αυτό το αρχείο πρέπει να διαμορφωθεί όπως στο παρακάτω παράδειγμα:

```
/**
 * Package documentation goes here; any documentation before the first period will be
 * used as a summary.
 *
 * It is common practice to leave a blank line between the summary and the rest of
 * the documentation; use this space to describe the package in as much detail as is
 * appropriate.
```

ΕΠΛ 133: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

```
* Inline tags such as {@code code here}, {@link reference.to.other.Documentation},
  * and {@literal text here} can be used in this documentation.
  */
package com.package.name;
// Remainder of file must be empty
```

Field Documentation

Παράδειγμα σχολίων πεδίων των κλάσεων:

```
/**
 * Fields can be documented as well.

* As with other javadocs, the documentation before the first period is used as a
 * summary, and is usually separated from the rest of the documentation by a blank
 * line.

* Documentation for fields can use inline tags, such as:
 * {@code code here}
 * {@literal text here}
 * {@link other.docs.Here}
 *
 * Field documentation can also make use of the following tags:
 *
 * @since 2.1.0
 * @see some.other.class.Documentation
 * @deprecated Describe why this field is outdated
 */
public static final String CONSTANT_STRING = "foo";
```

Linking to Other Javadocs

Η σύνδεση με άλλα σχόλια Javadoc γίνεται με την ετικέτα @link:

```
/**
  * You can link to the javadoc of an already imported class using {@link ClassName}.
  *
  * You can also use the fully-qualified name, if the class is not already imported:
  * {@link some.other.ClassName}
  *
  * You can link to members (fields or methods) of a class like so:
  * {@link ClassName#someMethod()}
  * {@link ClassName#someMethodWithParameters(int, String)}
  * {@link ClassName#someField}
  * {@link #someMethodInThisClass()} - used to link to members in the current class
  *
  * You can add a label to a linked javadoc like so:
  * {@link ClassName#someMethod() link text}
  */
  */
```

ΕΠΛ 133: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ασκήσεις

- 1. Επιλέξτε ένα πρόγραμμα από τα προηγούμενα εργαστήρια (π.χ. RationalNumber.java), προσθέστε σχόλια σε αυτό και δημιουργήστε τα html αρχεία (javadoc).
- 2. Σχεδιάστε μια κλάση, Quadratic Equation, για μια εξίσωση δευτέρου βαθμού: $ax^2 + bx + c = 0$. Προσθέστε τις απαραίτητες μεθόδους που βρίσκουν τις ρίζες τις εξίσωσης. Στην περίπτωση που δεν υπάρχουν να εκτυπωθεί κατάλληλο μήνυμα.
- 3. Σχεδιάστε μια κλάση, LinearEquation, για ένα σύστημα γραμμικών εξισώσεων 2 x 2:

$$ax + by = e$$

$$cx + dy = f$$

Οι λύσεις τις εξίσωσης είναι:

$$x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc}$$