



Η προγραμματιστική άσκηση για το μάθημα είναι **υποχρεωτική** και αφορά τη σχεδίαση, υλοποίηση και ρύθμιση ενός συστήματος λογισμικού. Η εργαστηριακή άσκηση προσφέρει **3 μονάδες** στον τελικό βαθμό του μαθήματος και εκπονείται σε ομάδες των **1 - 3 προσώπων**.

**ΕΙΔΙΚΑ ΓΙΑ ΤΟ 2020-2021 ΕΠΙΤΡΕΠΕΤΑΙ ΤΟ PROJECT ΝΑ ΕΚΠΟΝΗΘΕΙ ΚΑΙ ΑΤΟΜΙΚΑ!**

Φυσικά, πρέπει να πιάσετε τουλάχιστον τη βάση στην εργασία, όπως και στο διαγώνισμα. Σε περιπτώσεις εξαιρετικών εργασιών, η επίδοση επιβραβεύεται με bonus στον τελικό βαθμό.

**Το σύστημα πρέπει να υλοποιηθεί σε όλα τα επί μέρους στάδια.**

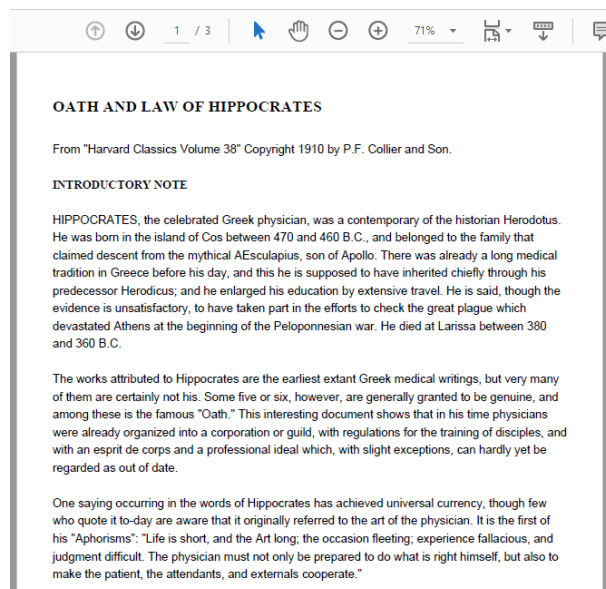
Το φετινό project απαιτεί την κατασκευή ενός προγράμματος μορφοποίησης κειμένων τα οποία βρίσκονται σε απλή πηγαία μορφή, με τρόπο που να παράγουμε μια μορφοποίησή τους που να είναι πιο εμφανίσιμη και φιλική στον αναγνώστη.

## Κεντρική ιδέα του συστήματος

Ο βασικός χρήστης του προγράμματός μας είναι ένας άνθρωπος που καταχρηστικά θα αποκαλούμε *στοιχειοθέτη κειμένων*, ένας άνθρωπος δηλαδή που παίρνει κείμενα σε ακατέργαστη ηλεκτρονική μορφή και τα ετοιμάζει προς παραγωγή με τρόπο πιο όμορφο και εύληπτο από τους αναγνώστες. Για να διευκολύνουμε την εργασία του ανθρώπου αυτού, θα κατασκευάσουμε ένα πρόγραμμα που αυτοματοποιεί σε μεγάλο βαθμό τη δουλειά του.

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
hippocratesOath.txt
1 Internet Wiretap Edition of CRLE
2 CRLE
3 OATH AND LAW OF HIPPOCRATES CRLE
4 CRLE
5 From "Harvard Classics Volume 38" CRLE
6 Copyright 1910 by P.F. Collier and Son. CRLE
7 CRLE
8 This text is placed in the Public Domain, June 1993. CRLE
9 CRLE
10 CRLE
11 INTRODUCTORY NOTE CRLE
12 CRLE
13 HIPPOCRATES, the celebrated Greek physician, was a contemporary CRLE
14 of the historian Herodotus. He was born in the island of Cos between CRLE
15 470 and 460 B.C., and belonged to the family that claimed descent CRLE
16 from the mythical Aesculapius, son of Apollo. There was already a CRLE
17 long medical tradition in Greece before his day, and this he is CRLE
18 supposed to have inherited chiefly through his predecessor Herodotus. CRLE
19 and he enlarged his education by extensive travel. He is said, CRLE
20 though the evidence is unsatisfactory, to have taken part in the CRLE
21 efforts to check the great plague which devastated Athens at the CRLE
22 beginning of the Peloponnesian war. He died at Larissa between 380 CRLE
23 and 360 B.C. CRLE
24 CRLE
25 The works attributed to Hippocrates are the earliest extant CRLE
26 Greek medical writings, but very many of them are certainly not his. CRLE
27 Some five or six, however, are generally granted to be genuine, CRLE
28 and among these is the famous "Oath." This interesting document CRLE
29 shows that in his time physicians were already organized into a CRLE
30 corporation or guild, with regulations for the training of disciples, CRLE
31 and with an esprit de corps and a professional ideal which, with CRLE
32 slight exceptions, can hardly yet be regarded as out of date. CRLE
33 CRLE
34 One saying occurring in the words of Hippocrates has achieved CRLE
35 universal currency, though few who quote it to-day are aware that CRLE
36 it originally referred to the art of the physician. It is the first CRLE
37 of his "Aphorisms": "Life is short, and the Art long; the occasion CRLE
38 fleeting; experience fallacious, and judgment difficult. The CRLE
39 physician must not only be prepared to do what is right himself, but CRLE
40 also to make the patient, the attendants, and externals cooperate." CRLE
41 CRLE
```

Normal text file length: 6.644 lines: 124 Ln: 1 Col: 1 Sel



Σχήμα 1. Το πρωτογενές κείμενο για τον όρκο του Ιπποκράτη όπως φαίνεται στο Notepad++, ώστε να φαίνονται όλοι οι whitespaces χαρακτήρες (Αριστερά) και το pdf που παρήχθη (Δεξιά), εξαιρώντας την 0-κή και την 3<sup>η</sup> παράγραφο, και έχοντας ως τίτλο τη φράση που αρχίζει από "OATH AND" και με bold όσες παραγράφους έχουν μόνο κεφαλαία γράμματα.

Τα κείμενα που διαχειριζόμαστε είναι απλά αρχεία κειμένου, για τα οποία πρέπει (α) να αποτιμήσουμε διάφορα στατιστικά, όπως αρ. λέξεων, γραμμών κλπ., και (β) να παράξουμε μια μορφοποίηση που θα είναι πιο όμορφη και ευανάγνωστη από την αρχική. Η παρουσίαση των επεξεργασθέντων κειμένων θα γίνει μέσω (α) επισημείωσης σε Markdown και (β) εξαγωγής σε pdf. Τα αρχεία που θα χρησιμοποιήσουμε είναι δύο ειδών: (α) πρωτογενή, ή (β) προ-επισημειωμένα. Τα πρωτογενή αρχεία είναι αρχεία απλού κειμένου, με κείμενο γεμάτο \n σε πολύ στενές γραμμές, που δεν έχουν καμία επέμβαση. Τα προ-επισημειωμένα κείμενα είναι κείμενα στα οποία ο στοιχειοθέτης έχει πάει και έχει βάλει από πριν μικρά και σύντομα σημάδια στο κείμενο (π.χ., html ή markdown annotations) ώστε το πρόγραμμα να παράγει μετατροπές στη θέση τους.

## Λειτουργικότητα του συστήματος

**Καταγραφή κανόνων.** Ο στοιχειοθέτης καταγράφει στο πρόγραμμα γενικούς κανόνες για ένα κείμενο. Οι κανόνες σκοπό έχουν να καθορίσουν πώς θα εμφανίσουμε τις παραγράφους του κειμένου εισόδου και είναι δύο ειδών:

(α) κανόνες στυλ, για το αν μια παράγραφος είναι είτε κρυφή και παραλείπεται, είτε heading 1, είτε heading 2, είτε κανονική (normal).

(β) κανόνες γραμματοσειράς, για το αν μια παράγραφος θα εμφανιστεί είτε bold, είτε italics, είτε απλή (regular).

Για να μπορέσουμε να πούμε στο σύστημα πότε μια παράγραφος είναι υποψήφια για να πληροί οποιονδήποτε από τους παραπάνω κανόνες, (π.χ., για να πούμε ότι θα πρέπει να εμφανιστεί ως bold ή ως h1) έχουμε τα εξής εργαλεία.

- Μια λίστα από αύξοντες αριθμούς των παραγράφων
- Όλα τα γράμματα της παραγράφου να είναι κεφαλαία
- Η παράγραφος θα πρέπει να ξεκινά με ένα συγκεκριμένο prefix

Για παράδειγμα, αν θέλουμε να πούμε ότι θα πρέπει να παραληφθούν οι παράγραφοι 4,5,7 πρέπει να καθορίσουμε ένα κανόνα "OMIT" "POSITIONS" "4,5,7". Αν θέλουμε να πούμε ότι θα εμφανίσουμε ως heading 1 μια παράγραφο που ξεκινά με "OATH AND" θα πρέπει να προσδιορίσουμε την τριάδα "H1" "STARTS\_WITH" "OATH AND".

Η προτεραιότητα είναι με τη σειρά που αναφέρονται οι κανόνες πιο πριν στην παρούσα εκφώνηση (δλδ., αν μια παράγραφος π.χ. είναι υποψήφια και για να παραλειφθεί και για heading 2, θα παραληφθεί. Ομοίως, αν είναι υποψήφια και για italics και για bold, θα επισημειωθεί ως bold). Οι κανόνες για bold / italics εφαρμόζονται μόνο σε κανονικό κείμενο.

Το πρόβλημα που καλείσθε να λύσετε είναι πώς ακριβώς θα πείτε στο πρόγραμμά σας τους κανόνες μετατροπής. Είναι βολικό να χρησιμοποιήσετε ένα rule set που περιέχει μια λίστα από τέτοιους κανόνες.

**Φόρτωση και επεξεργασία αρχείου.** Ο στοιχειοθέτης πρέπει να μπορεί να φορτώνει ένα κείμενο (πρωτογενές ή επισημειωμένο) και το σύστημα να το αναπαριστά σωστά ως μια λίστα παραγράφων. Για την φόρτωση του αρχείου πρέπει να δίνονται ως πληροφορίες το path στο οποίο βρίσκεται, ο τύπος (πρωτογενές ή επισημειωμένο), και ένα φιλικό όνομα (alias) για να μπορούμε στη συνέχεια να αναφερόμαστε σ' αυτό.

1. Σε κάθε περίπτωση, θεωρούμε καθολικά ότι μια κενή γραμμή περιέχει αποκλειστικά ένα "\n" χωρίς κάποιον άλλο χαρακτήρα. Επίσης θεωρούμε καθολικά ότι έχουμε νέα παράγραφο κάθε φορά που η προηγούμενη γραμμή είναι κενή. Ο κανόνας αυτός ισχύει σε όλα τα κείμενα και όλες τις μετατροπές και δε χρειάζεται να ενσωματώνεται σε κάποιο σύνολο κανόνων.
2. Το αν στην εσωτερική αναπαράσταση που θα διαλέξετε θα κρατάτε και τις κενές γραμμές ως παραγράφους είναι στην διακριτική σας ευχέρεια. Λάβετε υπόψιν π.χ., ότι σε ένα κείμενο markdown οι κενές γραμμές είναι απαραίτητες για να ξεχωρίζουν οι παράγραφοι μεταξύ τους, ενώ σε ένα κείμενο που παράγεται σε pdf όχι. Αλλά αυτό μπορεί να γίνει και από τις μεθόδους εξαγωγής στο αντίστοιχο format και όχι απαραίτητα επειδή το έλυσε η εσωτερική αναπαράστασή σας.

Ένα κείμενο που έχει φορτωθεί ήδη, πρέπει να μπορεί το σύστημα να το επεξεργαστεί, χαρακτηρίζοντας τις παραγράφους του, μία προς μία, ανάλογα με τους κανόνες που έχουν προ-καταχωρηθεί σε προηγούμενο βήμα. Αν δεν έχουν ορισθεί κανόνες, θα πρέπει όλες οι παράγραφοι να χαρακτηρίζονται ως normal και regular.

**Παραγωγή report με στατιστικά.** Αφού έχει φορτωθεί ένα κείμενο, θα πρέπει το σύστημα να επιτρέπει στον στοιχειοθέτη να ζητήσει ένα report με στατιστικά στοιχεία για το κείμενο. Το σύστημα θα πρέπει να επιστρέφει ένα List<String> με το 0-κό στοιχείο να περιέχει μια περιγραφή για τον αριθμό των παραγράφων του κειμένου, όπως φορτώθηκαν και έχουν αναπαρασταθεί στην κύρια μνήμη (ασχέτως αν στις εξαγωγές θα εμφανιστούν ή όχι), το 1<sup>ο</sup> στοιχείο να έχει τον συνολικό αριθμό λέξεων, και μετά, για κάθε παράγραφο, πόσες γραμμές και πόσες λέξεις έχει. Π.χ., για το κείμενο με τον όρκο του Ιπποκράτη, ένα report θα πρέπει να ξεκινά περίπου με:

```
Total number of paragraphs: 17,  
Total number of words: 1145,  
Lines: 1           Words: 4,  
Lines: 1           Words: 5,  
Lines: 2           Words: 12,  
...
```

**Εξαγωγή κειμένου σε άλλο format.** Ο στοιχειοθέτης πρέπει να μπορεί να ζητά από το σύστημα να μετατρέψει ένα κείμενο σε markdown, το οποίο πρέπει να έχει ήδη χαρακτηριστεί με τον προαναφερθέντα τρόπο.

Αντίστοιχα με την εξαγωγή σε markdown, ο στοιχειοθέτης κειμένων πρέπει να μπορεί να ζητά από το σύστημα να μετατρέψει ένα κείμενο σε PDF, παράγραφο προς παράγραφο.

## Προγραμματιστικές συστάσεις

### Διαχείριση Markdown

Η Markdown είναι μια απλή γλώσσα επισημείωσης κειμένου με δομικά στοιχεία. Από Wikipedia:

```
# Heading  
## Sub-heading  
### Sub-sub-heading
```

Paragraphs are separated  
by a blank line.

```
**bold text**  
_italicized text_
```

Horizontal rule  
---

Bullet list:

```
* apples  
* oranges  
* pears
```

Links: <https://guides.github.com/features/mastering-markdown/>, <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>, <https://en.wikipedia.org/wiki/Markdown>

### Διαχείριση αρχείων Pdf προγραμματιστικά

Θα χρησιμοποιήσετε τη βιβλιοθήκη **itext5** που διατίθεται ως free/open source software (F/OSS) κάτω από την AGPL license (<https://itextpdf.com/Products/itext-core>) και την οποία θα βρείτε στο <https://github.com/itext/itextpdf> μέσω του <https://developers.itextpdf.com/itext-java>.

Ο σύνδεσμος <https://github.com/itext/itextpdf/releases/latest> έχει ίσως την πιο ολοκληρωμένη εκδοχή για να κατεβάσετε.

**Το βασικό jar που θα πρέπει να προσθέσετε στο build path της εφαρμογής σας είναι το itext5-itextpdf-5.5.12.jar.**

Για να μπορέσετε να μπειτε στο πνεύμα, από το βιβλίο <https://www.manning.com/books/itext-in-action-second-edition> μπορείτε να χρησιμοποιήσετε τα κεφάλαια 1 και 6 και κυρίως τα παραδείγματα που δίδονται στο

<https://manning-content.s3.amazonaws.com/download/b/e71cc48-85e7-4530-ace6-e8bd1a92e08c/SourceCodeIText.zip>

καθώς και στο <https://developers.itextpdf.com/examples-itext5>.

## Software Architecture & Specifications

Η αρχιτεκτονική του λογισμικού σε πακέτα, σας δίδεται σχετικά προκαθορισμένα. Το υπό κατασκευή σύστημα είναι ένα **σύστημα 2 επιπέδων**, ενός **client** κομματιού που είναι υπεύθυνο για την διάδραση με τον τελικό χρήστη (που είναι τελείως στην δική σας σχεδιαστική επιλογή) και ενός **server** κομματιού που είναι υπεύθυνο για την διεκπεραίωση των use cases που προκύπτουν από την αρχική περιγραφή της λειτουργικότητας του συστήματος.

Στην back-end πλευρά του server υπάρχουν διάφορα packages (πακέτα) που το καθένα έχει τη δική του λειτουργικότητα. Τα πακέτα είναι:

**engine:** the package exporting the system's functionality via an interface (to be materialized in a concrete class by you), and its implementation, along with any other needed classes. For each file to process, we will launch a dedicated Engine object – it is the responsibility of the client (if deemed necessary) to handle multiple such engine instances. At the server-side, each engine instance handles a single input file.

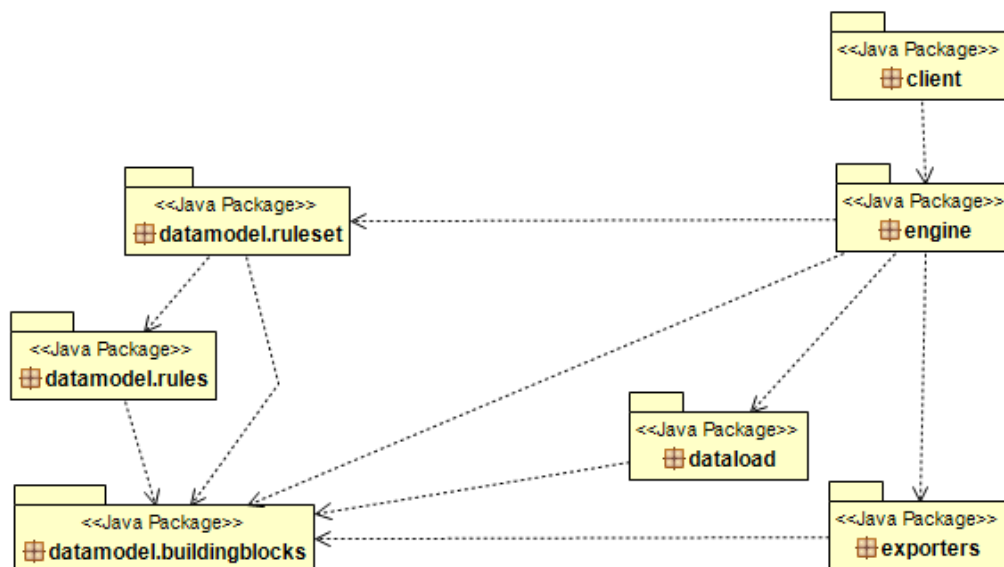
**dataload:** the package responsible for reading the input file and loading it in the appropriate collection(s) in memory.

**exporters:** the package responsible for creating output files with exporter classes for PDF and Markdown. Given to you empty.

**datamodel:** the package keeping the domain classes of the system. The internal architecture is depicted on the left-hand side of Fig. 1. See the code given to you for details.

Στη front-end πλευρά έχει προβλεφθεί ένα πακέτο για να στεγάσει τη λειτουργικότητα

**client:** package to be filled by you with the client-side functionality (given to you empty; you must complete it)



Σχήμα 1 Αρχιτεκτονική των πακέτων του συστήματος

Έχετε δικαίωμα να υλοποιήσετε το client package με όποιο τρόπο θέλετε εσείς: με ένα απλό σύστημα διαπροσωπείας από το τερματικό, με μια γραφική διεπαφή, ... -- είναι απολύτως στη διακριτική σας ευχέρεια. **Είναι υποχρεωτικό και απαράβατο, όμως, ΝΑ ΜΗΝ ΑΛΛΑΞΕΤΕ τα interfaces του back-end που σας δίνονται (ένα και μόνο για φέτος), αλλά να τα υλοποιήσετε επακριβώς.** Το ποιες κλάσεις θα εντάξετε μέσα στο κάθε πακέτο είναι δικό σας θέμα και αντικείμενο της σχεδίασης, υλοποίησης και ελέγχου που θα κάνετε, καθώς και της αξιολόγησής τους.

## Τι σας δίνεται έτοιμο

Στο φάκελο

[http://www.cs.uoi.gr/~pvassil/courses/sw\\_dev/exercises/supportingMaterial/2020-2021/](http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/supportingMaterial/2020-2021/)

Σας δίνονται:

A. ένα αρχείο .zip με φακέλους για (α) τη δομή του src και τα σχετικά interfaces, (β) δεδομένα (υπο-φάκελος Resources) για να κάνετε ελέγχους και για να τρέξετε το σύστημα που θα φτιάξετε, (γ) ένας φάκελος libs με τα σχετικά jar που σας χρειάζονται για να κάνετε εύκολη τη ζωή σας.

Εσείς πρέπει:

- Να φτιάξετε ένα νέο Eclipse project με όνομα <AM1>\_<AM2>\_<AM3>\_TextProcessor με τα AM των μελών της ομάδας σας (ώστε αφού τα κάνετε turnin να ξέρουμε ποιανού είναι κάθε project), και προφανώς, αν δεν είστε τρεις, με τα AM όσων είναι μέλη της ομάδας.
- Να κάνετε paste μέσα του τα περιεχόμενα του .zip file
- Να συμπληρώσετε και τις υπόλοιπες κλάσεις του συστήματος.
- Να προσθέσετε ένα αρχείο κειμένου Readme.txt που θα λέει (α) ονόματα και AM, (β) αν τυχόν χρειάζεται, τι απαιτείται για να τρέξει το πρόγραμμά σας χωρίς προβλήματα στον έλεγχο από τους βοηθούς (π.χ., κάποιες ειδικές βιβλιοθήκες που τυχόν χρησιμοποιήσατε)

Σας δίνουμε έτοιμο το unit test TestEngine για την κεντρική μηχανή, με ελέγχους για μερικές από τις βασικές λειτουργίες του συστήματος, στο πακέτο test. **Αυτό θα πρέπει να τρέχει με επιτυχία στο τελικό σας παραδοτέο!** Φυσικά θα πρέπει να φτιάξετε και επί μέρους unit tests και εσείς. Χρησιμοποιήστε Junit 4 και όχι 5 (έχει πολύ περισσότερο υλικό στο διαδίκτυο).

B. Υλικό για την παραγωγή pdf από τα σχετικά jar όπως περιγράφηκε και πιο πριν.

## Οδηγίες και υποχρεώσεις

**ΕΙΝΑΙ ΑΠΑΡΑΙΤΗΤΟ ΤΟ PROJECT ΣΑΣ ΝΑ ΦΤΑΣΕΙ ΜΕΧΡΙ ΚΑΙ ΤΗΝ ΥΛΟΠΟΗΣΗ ΚΑΙ Ο ΚΩΔΙΚΑΣ ΣΑΣ ΝΑ ΤΡΕΧΕΙ! (Δεν είναι προβιβάσιμα projects που ο κώδικάς τους δεν τρέχει)**

**ΕΙΝΑΙ ΑΠΑΡΑΙΤΗΤΟ Ο ΚΩΔΙΚΑΣ ΣΑΣ ΝΑ ΣΥΝΟΔΕΥΕΤΑΙ ΑΠΟ ΤΑ ΑΝΤΙΣΤΟΙΧΑ TESTS!**

**Είναι απαράβατο είναι ΝΑ ΥΛΟΠΟΙΗΣΕΤΕ ΕΠΑΚΡΙΒΩΣ ΤΟ INTERFACE που σας δίδεται. Αυτή είναι και η ιδέα, να δείτε στην πράξη, τι στην ευχή τα θέλουμε αυτά τα interfaces: μόλις σας δόθηκε ένα «συμβόλαιο» λειτουργικότητας του back-end server με τον client, το οποίο πρέπει να υλοποιήσετε επακριβώς.**

Επιβάλλεται να ακολουθήσετε τις βασικές αρχές ενθυλάκωσης (υποχρεωτικά), χαμηλής σύζευξης, DIP, OCP, abstract coupling, factories κλπ (όσο αυτό είναι εφικτό και εύλογο).

**Υπόδειγμα αναφοράς:** για τα επιμέρους στάδια, μπορείτε να συμπληρώνετε / αναθεωρείτε σταδιακά την αναφορά σας. Για διευκόλυνσή σας, υπάρχει ένα υπόδειγμα στο [http://www.cs.uoi.gr/~pvassil/courses/sw\\_dev/exercises/TemplateFinalReport.zip](http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/TemplateFinalReport.zip).  
**ΥΠΟΧΡΕΩΤΙΚΑ ΠΡΕΠΕΙ ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΤΟ ΥΠΟΔΕΙΓΜΑ ΠΟΥ ΣΑΣ ΔΙΝΕΤΑΙ!**

**Υλικό.** Όλο το υποστηρικτικό υλικό για το project θα βρίσκεται στο URL

[http://www.cs.uoi.gr/~pvassil/courses/sw\\_dev/exercises/supportingMaterial/](http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/supportingMaterial/)  
στον υποφάκελο της φετινής χρονιάς

### Στη διάρκεια του εξαμήνου:

- Σίγουρα θα δοθούν περαιτέρω εξηγήσεις στην εκφώνηση & ενδεχομένως να αλλάξει/εμπλουτισθεί κάποιο μέρος της εκφώνησης! (άρα το δημοσιευθέν αρχείο της εκφώνησης μπορεί να αλλάξει)
- Αργότερα στο εξάμηνο, θα σας ζητηθεί να εγγράψετε την ομάδα σας σε μία φόρμα εγγραφής. Ομάδες που δεν εγγραφούν κινδυνεύουν να μην βαθμολογηθούν. Θα δοθούν οδηγίες αφού έχουμε μια πιο καθαρή εικόνα του τι γίνεται με το εξάμηνο.

## Χρονοδιάγραμμα

Στη συνέχεια παρατίθενται στάδια της ανάπτυξης, ενδιάμεσες προθεσμίες (milestones) και καταληκτικές ημερομηνίες ολοκλήρωσης (deadlines).

<b>[01/10]</b>	Εκφώνηση	
3 weeks		<i>Setup of Infrastructure</i>
<b>[25/10:: 23.59]</b>	Εγκατάσταση Java (version >=1.12), Eclipse (last v.) στους Η/Υ σας Για όσους έχουν δυσκολία με την Java: ασκήσεις επανάληψης Εξοικείωση και πειραματισμός με το υλικό που σας δίδεται Εκκίνηση εργασιών στα Use Cases της εκφώνησης Δεν υπάρχει κάτι να παραδώσετε	
4 weeks		<i>Design of classes &amp;&amp; first implementations</i>
<b>[22/11:: 23.59]</b>	<b>Turnin:</b> DLV1.1: First version of the report (pdf) with (a) use cases + (b) the OREOS part for the test cases + (c) a first version of the class diagram(s) with the design of the system (προαιρετικά: any other diagrams)	
4 weeks		<i>Complete implementation</i>
<b>[20/12:: 23.59]</b> <b>ΑΝΕΛΑΣΤΙΚΑ</b>	<b>Turnin :</b> DLV 2.1: tar file with the code for all classes DLV 2.2: FINAL version of the report (pdf) with all the design and the documentation of the project	
<b>@ ALL turnin's</b>	<b>Exclude the folder Resources from your turnin (which cannot be larger than 10MB).</b>	

Δεν θα ξεπεράσουμε το όριο των Χριστουγέννων. Η πράξη έχει αποδείξει ότι στις γιορτές οι ομάδες αποσυντονίζονται σε πολύ μεγάλο βαθμό. Έτσι, **Η ΠΡΟΘΕΣΜΙΑ ΤΗΣ ΠΑΡΑΔΟΣΗΣ ΕΙΝΑΙ ΑΝΕΛΑΣΤΙΚΗ!** Θα πιεστείτε περισσότερο πριν τις γιορτές, αλλά θα φύγετε για τις γιορτές χωρίς το φορτίο του project.

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**