



Ημερομηνία Παράδοσης: 18-12-2019

Π. Βασιλειάδης

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 **3 προσώπων**. 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.

Στον ιστότοπο Kaggle (<https://www.kaggle.com>), υπάρχουν δημοσιευμένα σύνολα δεδομένων από τον πραγματικό κόσμο, για να τα χρησιμοποιούν οι ερευνητές για τη διεξαγωγή μελετών. Ένα από αυτά είναι το Household Electric Power Consumption data set (<https://www.kaggle.com/uciml/electric-power-consumption-data-set>), το οποίο έχει την καταγραφή από την κατανάλωση ενέργειας από ένα σπίτι σε διάστημα περισσότερο των 3 ετών. Συγκεκριμένα, στο χρονικό διάστημα της καταγραφής, οι αισθητήρες που είχαν προσαρμοστεί στο σπίτι, δειγματοληπτούσαν την κατανάλωση ενέργειας κάθε δευτερόλεπτο (με αποτέλεσμα να έχουμε στη διάθεσή μας πάνω από 2 εκατομμύρια μετρήσεις, για το πώς το σπίτι αυτό κατανάλωνε ρεύμα). Στη σελίδα του data set θα βρείτε αναλυτικά το τι ακριβώς μετρήθηκε, αλλά εμάς μας ενδιαφέρουν τα εξής χαρακτηριστικά: (α) Ημέρα (με format dd/mm/yyyy), (β) ώρα (με format hh:mm:ss), (γ) Κατανάλωση κουζίνας (energy sub-metering No. 1, watt-hour of active energy), (δ) κατανάλωση δωματίου πλυντηρίων (energy sub-metering No. 2, watt-hour of active energy), (ε) κατανάλωση θέρμανσης (energy sub-metering No. 3, watt-hour of active energy).

Κεντρική ιδέα του συστήματος

Καλείσθε να κατασκευάσετε ένα σύστημα λογισμικού που θα μπορεί να χρησιμοποιήσει ένας αναλυτής για να μπορεί να εξάγει συμπεράσματα για την κατανάλωση ενέργειας σε σχέση με τη χρονική περίοδο. Ενδεικτικά, για να γίνει κατανοητό το τι ζητείται, ένας αναλυτής θα ήθελε να ξέρει τις απαντήσεις σε ερωτήσεις όπως (α) ποια περίοδο της ημέρας η κατανάλωση του δωματίου πλυντηρίων είναι μέγιστη? (β) ποια περίοδο του χρόνου / ποιο μήνα είναι η κατανάλωση θέρμανσης ελάχιστη / μέγιστη? (γ) η κατανάλωση ενέργειας στην κουζίνα, αλλάζει ανάλογα με τη μέρα της εβδομάδας?

Επειδή έχουμε μετρήσεις ανά δευτερόλεπτο (άρα πάρα πολλές), οι αναλυτές ενδιαφέρονται για στατιστικά στοιχεία για αυτές. Εμείς θα περιοριστούμε στο να αποτιμούμε (α) **άθροισμα** ή (β) **μέση τιμή** κατανάλωσης **ανά χρονική περίοδο**. Οπότε, για παράδειγμα, ο αναλυτής θα πρέπει να μπορεί να ζητήσει "για κάθε μήνα, μέση τιμή κατανάλωσης για κάθε μία από τις 3 μετρήσεις του σπιτιού". Ή, ως άλλο παράδειγμα, "για κάθε περίοδο της ημέρας (βλ. παρακάτω), το σύνολο της κατανάλωσης, για κάθε μία από τις 3 μετρήσεις του σπιτιού".

Οι χρονικές περίοδοι που μας ενδιαφέρουν είναι:

- Εποχή του χρόνου: WINTER, SPRING, SUMMER, AUTUMN
- Μήνας -- μπορείτε να χρησιμοποιήσετε είτε συμβολοσειρές JAN - DEC, είτε συμβολοσειρές "01" - "12" είτε μια αριθμητική κωδικοποίηση, π.χ., [1-12].
- Μέρα της εβδομάδας (MON-SUN) -- μπορείτε να χρησιμοποιήσετε μια αριθμητική κωδικοποίηση, π.χ., [0-6] ή [1-7]
- Περίοδος της μέρας: NIGHT [21.00 - 05.00), EARLY_MORNING [05.00 - 09.00), MORNING [09.00 - 13.00), AFTERNOON [13.00 - 17.00), EVENING [17.00 - 21.00)

Προσέξτε πως δεν αναφερόμαστε σε χρονικές περιόδους (π.χ., στο μήνα) μιας συγκεκριμένης χρονιάς (π.χ., JAN 2008) αλλά γενικά στην χρονική περίοδο (π.χ., στο μήνα JAN). Αυτό έχει επίπτωση αν στα δεδομένα έχουμε παραπάνω από μία χρονιές.

```
2007Aggregate_Output.md
Monthly avg aggregation over 2007

avg consumption (watt-hours) over (a) Kitchen, (b) Laundry, (c) A/C

## Kitchen

* 11 1.176512801518594
* 01 1.2642933954655262
* 12 1.659759403212322
* 02 1.1802172726822482
* 03 1.361343220054142
* 04 1.0658864655368077
* 05 1.6966173835125122
* 06 1.3826725676021008
* 07 0.9672650475185032
* 08 0.81247478596092
* 09 1.2232279272188316
* 10 0.9681885389130608

## Laundry

* 11 1.7053104310383322
* 01 1.7760103951966497
* 12 1.8578149152085401
* 02 1.60236122823561
* 03 2.3468715697036586
* 04 0.9731489221570293
* 05 1.6158602150539272
* 06 1.6205714020899822
* 07 1.2521737176752197
* 08 1.1141467569143342
* 09 1.7426038242509723
* 10 1.9694878802812905

## A/C

* 11 6.937589703227237
* 01 7.383681333452795
* 12 8.11897667958513
* 02 6.70355672404292
* 03 6.504648401622817
* 04 4.8003394381544995
* 05 5.139964157706087
* 06 4.375906573672921
* 07 3.478285290615267
* 08 5.062714151230473
* 09 5.062714151230473
* 10 5.062714151230473

Markdown Source Preview
Problems Declaration Search Progress Console Call Hierarchy
No consoles to display at this time.
```

Δείτε εδώ πώς για κάθε μήνα βγάζουμε την μέση τιμή κάθε κατηγορίας μέτρησης (τη θέρμανση τη συμβολίζουμε με AC στον κώδικα που σας έχει δοθεί).

Οι μήνες αντί για JAN – DEC κωδικοποιούνται με συμβολοσειρές 01-12

/* Πρακτικά, οι χρονικές περίοδοι λειτουργούν ως ομαδοποιήσεις για τις μετρήσεις: αν π.χ., θέλουμε τη μέση τιμή ανά μήνα, το τρόπο να το πετύχουμε είναι:

(0) παίρνουμε όλες τις μετρήσεις του συνόλου δεδομένων,

(1) μία-μία μέτρηση, την βάζουμε σε ένα υποσύνολο μετρήσεων που αντιστοιχεί στο μήνα της μέτρησης αυτής, και,

(2) για κάθε μήνα, βγάζουμε τη μέση τιμή από το υποσύνολο των μετρήσεων που του έχουμε αντιστοιχήσει, για κάθε μία από τις τρεις μετρούμενες κατηγορίες (κουζίνα, πλυντήριο, A/C) */

Λειτουργικότητα του συστήματος

ΕΞΟΔΟΣ. Ο αναλυτής επιλέγει την έξοδο και το πρόγραμμα τερματίζεται με αποχαιρετιστήριο μήνυμα.

ΦΟΡΤΩΣΗ ΑΡΧΕΙΟΥ. Ο αναλυτής θέλει να φορτώσει στο σύστημα ένα αρχείο μετρήσεων, με τη δομή που προδιαγράφεται στη σελίδα

<https://www.kaggle.com/uciml/electric-power-consumption-data-set> Ο αναλυτής ερωτάται ποιο αρχείο θέλει να φορτώσει και παρέχει το σχετικό path. Το σύστημα φορτώνει το αρχείο και τροφοδοτεί ένα collection εγγραφών από αυτό. Το σύστημα οφείλει να φορτώσει ΜΟΝΟ τις εγγραφές που δεν έχουν κενά πεδία ή πεδία που παραβιάζουν το format των δεδομένων.

ΚΑΤΑΣΚΕΥΗ ΣΥΝΑΘΡΟΙΣΤΙΚΩΝ ΣΤΑΤΙΣΤΙΚΩΝ. Με βάση το τελευταίο από φορτωθέντα collections μετρήσεων, ο αναλυτής προσδιορίζει τι είδους στατιστικά θέλει να παραχθούν. Ο αναλυτής ερωτάται -και απαντά- με βάση ποιον τύπο χρονικής περιόδου θέλει να ομαδοποιήσει τα δεδομένα (π.χ., μήνα, εποχή, κλπ) και με βάση ποια αθροιστική συνάρτηση επί των μετρήσεων (στη δική σας υλοποίηση: sum or average). Επίσης, ο αναλυτής δίνεται και ένα μικρό κείμενο που θα συνοδεύει το παραχθέν αποτέλεσμα ως περιγραφή και θα εμφανίζεται αργότερα και στα report. Αν κάποιο από τα στοιχεία δεν δοθεί σωστά, ζητείται από τον χρήστη να επαναλάβει την καταχώρηση σωστά ή να εξέλθει της λειτουργίας αυτής. /* Εσωτερικά, το αποτέλεσμα που παράγεται είναι ένα IResult. */

ΠΑΡΑΓΩΓΗ ΑΝΑΦΟΡΩΝ Το σύστημα ρωτά τον αναλυτή (α) το path για το αρχείο που θέλει να κατασκευαστεί, και (β) τον τύπο της αναφοράς (text, html, markdown) και όταν αυτά τα στοιχεία δοθούν, χρησιμοποιεί το τελευταίο IResult που παρήχθη, για να ετοιμάσει την αναφορά. Αν κάποιο από τα στοιχεία δεν δοθεί σωστά, ή δεν υπάρχει IResult, ζητείται από τον χρήστη να επιδιορθώσει την κατάσταση ή να εξέλθει της λειτουργίας αυτής.

Υπόδειγμα της αναφοράς φαίνεται στο Σχήμα 1.

- Στην πρώτη γραμμή είναι η περιγραφή που δόθηκε για το IResult, φαρμαρισμένο ως heading 1
- Στη δεύτερη γραμμή είναι ο τύπος για την αθροιστική συνάρτηση (sum / avg) και ένα σταθερό κείμενο
- Μετά, υπάρχει από ένα κείμενο για κάθε μία από τις 3 μετρούμενες ποσότητες, ως heading 2 (Kitchen / Laundry / AC)
- Ακολουθούν ζεύγη <time unit> <aggregate measurement> για την αντίστοιχη μετρούμενη ποσότητα.

Στο φάκελο ReportTemplates υπάρχουν υποδείγματα για το πώς μπορείτε να οργανώσετε την αναφορά σας.

ΕΠΙΔΕΙΞΗ ΙΣΤΟΡΙΚΟΥ ΑΝΑΦΟΡΩΝ Το σύστημα, κάθε φορά που ζητείται μια αναφορά, κρατά τα στοιχεία που ζητήθηκαν από τον αναλυτή σε ένα ιστορικό αιτημάτων αναφορών. Ο χρήστης μπορεί να αιτηθεί την οπτική παρουσίαση του ιστορικού αιτημάτων και το σύστημα να του παρουσιάσει τη σχετική λίστα με την μεταπληροφορία των αναφορών (όχι δλδ., τα περιεχόμενα, αλλά την περιγραφή, το path του output αρχείου κλπ.).

/ Αυτή η λειτουργικότητα είναι αμιγώς client-side, δλδ., δεν περνά από το back-end server via the IMainEngine interface */*

Προγραμματιστικές συστάσεις

Markdown

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

Heading

Sub-heading

Paragraphs are separated
by a blank line.

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>

Java LocalDateTime

Χρησιμοποιήστε το project ως ευκαιρία να εξασκηθείτε στο χειρισμό χρόνου μέσω java. Οι κρίσιμες κλάσεις είναι οι LocalDate (για ημ/νιες), LocalTime (για ώρες/λεπτά/...) και ο συνδυασμός τους, LocalDateTime, στο java.time. Δοκιμάστε σε κάθε μέτρηση να κατασκευάσετε και ένα LocalDateTime αντικείμενο (έχει λίγο δουλειά μέχρι να φέρετε το input date string στη δομή που χρειάζεται).

Προσοχή: ό,τι βρείτε στο internet, να αφορά τουλάχιστο java 8. Πριν την έκδοση 8, η διαχείριση χρόνου ήταν διαφορετική.

<https://www.baeldung.com/java-8-date-time-intro>

<https://www.mkymong.com/tutorials/java-date-time-tutorials/>

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>

Apache commons math statistics

Ένα από τα χαρακτηριστικά του μοντέρνου προγραμματισμού είναι να επαναχρησιμοποιήσουμε έτοιμους κώδικες. Το Apache commons δίνει έτοιμα κομμάτια λογισμικού που έχουν χρησιμοποιηθεί ευρέως (άρα έχουν αποσφαλματωθεί στη διάρκεια των χρόνων) και παρέχουν πολλά utilities φιλικά προς τον developer. Η κλάση DescriptiveStatistics σας δίνει έτοιμα απλά στατιστικά για ένα σύνολο μετρήσεων.

Σας δίνουμε το jar από το apache commons math στα supporting material στο website του μαθήματος. Αν θέλετε να το χρησιμοποιήσετε (προαιρετικά) βάλτε το σε ένα φάκελο lib μέσα στο project σας και ενημερώστε το build path.

<https://commons.apache.org/proper/commons-math/javadocs/api-3.6/index.html?org/apache/commons/math3/stat/descriptive/DescriptiveStatistics.html>

Φόρτωση δεδομένων

Το interface για τη διαχείριση της φόρτωσης δεδομένων είναι εσκεμμένα επηρεασμένο από λογισμικό για data loaders που έχει δοθεί ήδη στα λυμένα παραδείγματα του μαθήματος. Αναζητήστε το και μοντάρετέ το στον κώδικά σας με κριτική σκέψη.

ΕΙΝΑΙ ΜΙΑ ΕΞΑΙΡΕΤΙΚΗ ΕΥΚΑΙΡΙΑ ΝΑ ΜΕΛΕΤΗΣΕΤΕ ΕΤΟΙΜΟ ΚΩΔΙΚΑ ΚΑΙ ΝΑ ΣΚΕΦΤΕΙΤΕ ΚΡΙΤΙΚΑ ΠΩΣ ΔΟΥΛΕΥΕΙ.

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 και ενός **server** κομματιού που είναι υπεύθυνο για την διεκπεραίωση (α) του φορτώματος των αρχικών δεδομένων, (β) της ομαδοποίησης και υπολογισμού στατιστικών ανάλογα με το τι ζητά ο αναλυτής, και, (γ) την παραγωγή αναφορών, σε περισσότερα του ενός format, με το αποτέλεσμα του υπολογισμού. Στην πλευρά του server υπάρχουν διάφορα packages (πακέτα) που το καθένα έχει τη δική του λειτουργικότητα. Σε κάθε πακέτο περιέχεται ένα interface, το οποίο εσείς πρέπει να υλοποιήσετε για να μπορείτε να διεκπεραιώσετε τη λειτουργικότητα του κάθε πακέτου.

Τα πακέτα είναι:

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

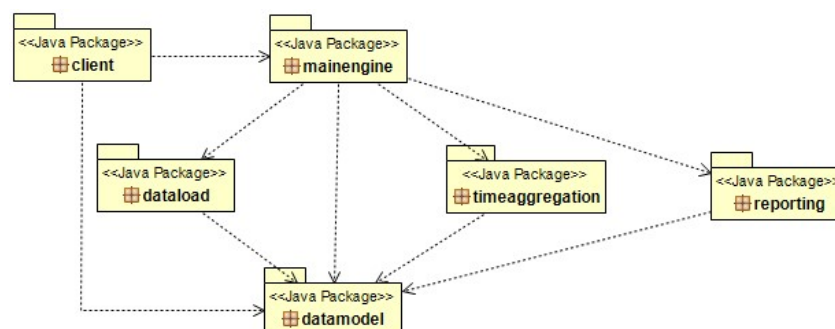
mainengine: main package including the IMainEngine interface that exports the fundamental functionality of the back-end, and its implementation.

dataload: the package responsible for reading the measurement data from the input files and loading them in the appropriate collection(s) in memory. The main functionality of this subsystem is summarized by the interface ILoader.

timeaggregation: the package responsible for grouping the loaded measurements into groups, one per time period, and for extracting the respective statistics. The main functionality of this subsystem is summarized by the interface IAggregator.

reporting: the package responsible for creating output reports with the statistics for the data sets. Reports come in different formats, e.g., text, html, markdown (md). The main functionality of this subsystem is summarized by the interface IResultReporter.

datamodel: the package keeping the domain classes of the system. Notable modules are the IResult interface, that prescribes the functionality of the aggregation result, and, the MeasurementRecord class that is the representative of each measurement = line in the input files.



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, 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. Αλλά το πώς “κουμπώνουν” τα υποσυστήματα μεταξύ τους, είναι ήδη προδιαγεγραμμένο και δεν μπορείτε να επεμβείτε σε αυτό.

/ Ναι, κάποια πράγματα στα interfaces που σας δίνονται θα μπορούσαν να είναι πιο απλά. Αλλά και στη ζωή, ποτέ δεν θα έχετε να αντιμετωπίσετε specifications που να είναι βέλτιστα ως προς την απλότητά τους. Ασχέτως αν θα μπορούσαν να είναι απλούστερα τα interfaces, είναι αυτά που είναι, και δεν μπορείτε να τα αλλάξετε... */*

Τι σας δίνεται έτοιμο

Στο φάκελο

http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/supportingMaterial/2019-2020/

σας δίνεται ένα αρχείο .zip στο οποίο περιλαμβάνονται δύο φάκελοι (α) με τη δομή του src και τα σχετικά interfaces, και (β) ένας φάκελος Resources με δεδομένα (i) σε μικρά μεγέθη για να κάνετε ελέγχους και (ii) σε κανονικά μεγέθη για να τρέξετε το σύστημα που θα φτιάξετε. Επειδή τα δεδομένα από το Kaggle έχουν ατέλειες και ελλείψεις εμείς σας δίνουμε: (α) ένα υποσύνολό τους με το 2007, και (β) την τριετία 2007-09, επίσης χωρίς ατελείς εγγραφές. Επίσης, για να κάνετε ελέγχους, σας δίδονται και μικρά σύνολα δεδομένων, για τα οποία μπορείτε να επιβεβαιώσετε την ορθότητα των υπολογισμών σας

Εσείς πρέπει

- Να φτιάξετε ένα νέο Eclipse project με όνομα 2019-2020_<AM1>_<AM2>_<AM3> με τα AM των μελών της ομάδας σας (ώστε αφού τα κάνετε turnin να ξέρουμε ποιανού είναι κάθε project).
- Να κάνετε paste μέσα του τους δύο υποφακέλους που βρίσκονται στο 2019-2020_projectMaterial.zip που σας δίνεται στο παραπάνω link.
- Να μην πειράξετε τίποτα από τα έτοιμα interfaces, αλλά μόνο τις κλάσεις MeasurementRecord και MainEngineFactory, τις οποίες θα πρέπει να συμπληρώσετε. Φυσικά θα πρέπει να συμπληρώσετε και τις υπόλοιπες κλάσεις του συστήματος.

Σας δίνουμε έτοιμο το **unit test MainEngineTest** 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.

Οδηγίες και υποχρεώσεις

Είναι απαράβατο είναι ΝΑ ΥΛΟΠΟΙΗΣΕΤΕ ΕΠΑΚΡΙΒΩΣ ΤΑ INTERFACES που σας δίδονται. Άλλωστε 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, αλλά και μεταξύ των υποσυστημάτων του back-end server.

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

Χρησιμοποιήστε κατά προτίμηση Junit 4 και όχι 5.

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.

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/ 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)
- **Οι λεπτομέρειες της εξέτασης θα εξαρτηθούν από τον αριθμό των ομάδων από τη μία, και τον αριθμό των βοηθών από την άλλη. Άρα: μέσα στο εξάμηνο θα διευκρινιστεί το πώς θα εξεταστείτε ως ομάδα και ατομικά.**

Χρονοδιάγραμμα

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

[30/09]	Εκφώνηση	
3 weeks		<i>Setup of Infrastructure</i>
[20/10:: 23.59]	Εγκατάσταση Java (ideally 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	
2 weeks		<i>Καταγραφή των Use Cases</i>
[03/11:: 23.59]	Turnin: DLV1.1: First version of the report with all the use cases (as pdf file)	
3 weeks		<i>Design of classes && first implementations</i>
[24/11:: 23.59]	Turnin: DLV2.1: Second version of the report (pdf) with (a) the section for the test cases + (b) class diagram(s) with the design of the system (προαιρετικά: any other diagrams) DLV2.2: the current state of your src Print (OBLIGATORILY!!): DLV2.1 to be handed over on the next day of the deadline	
3.5 weeks		<i>Complete implementation</i> <i>[ΕΝΔΕΧΟΜΕΝΩΣ ΕΞΕΤΑΣΗ ΕΝΔΙΑΜΕΣΩΣ]</i>
[18/12:: 13.00] ΑΝΕΛΑΣΤΙΚΑ	Turnin : DLV 3.1: tar file with the code for all classes DLV 3.2: FINAL version of the report (pdf) with all the design and the documentation of the project Print (OBLIGATORILY!!): DLV3.2 to be handed over BEFORE/AT the deadline	
@ ALL turnin's	Exclude the folder Resources from your turnin (which cannot be larger than 10MB). If you want to use your own test files, include just them..	

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!