## ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

MYY301/ΠΛΥ308ΑΝΑΠΤΥΞΗΛΟΓΙΣΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΑΣΚΗΣΗ

ΣΕΠΤΕΜΒΡΙΟΣ2019

ΗμερομηνίαΠαράδοσης: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.

ΣτονιστότοποKaggle(https://[www.kaggle.com](http://www.kaggle.com/)),υπάρχουνδημοσιευμένασύνολαδεδομένωναπότονπραγματικόκόσμο,γιαναταχρησιμοποιούνοιερευνητέςγιατηδιεξαγωγήμελετών.Ένα από αυτά είναι το Household Electric Power Consumption data set (https://[www.kaggle.com/uciml/electric-power-consumption-data-set](http://www.kaggle.com/uciml/electric-power-consumption-data-set)),τοοποίοέχειτηνκαταγραφήαπότηνκατανάλωσηενέργειαςαπόένασπίτισεδιάστημαπερισσότεροτων3ετών. Συγκεκριμένα,στοχρονικόδιάστηματηςκαταγραφής,οιαισθητήρεςπουείχανπροσαρμοστεί στοσπίτι,δειγματοληπτούσαντηνκατανάλωσηενέργειαςκάθεδευτερόλεπτο(μεαποτέλεσμανα έχουμε στη διάθεσή μας πάνω από 2 εκατομμύρια μετρήσεις, για το πώς το σπίτι αυτό κατανάλωνερεύμα).Στησελίδατουdatasetθαβρείτεαναλυτικάτοτιακριβώςμετρήθηκε,αλλάεμάςμαςενδιαφέρουνταεξήςχαρακτηριστικά:(α)Ημέρα(μεformatdd/mm/yyyy),(β)ώρα(μεformat hh:mm:ss), (γ) Κατανάλωση κουζίνας (energy sub-metering No. 1, watt-hour of active energy),(δ)κατανάλωσηδωματίουπλυντηρίων(energysub-meteringNo.2,watt-hourofactive energy),(ε)κατανάλωσηθέρμανσης(energysub-meteringNo.3,watt-hourofactiveenergy).

# 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?

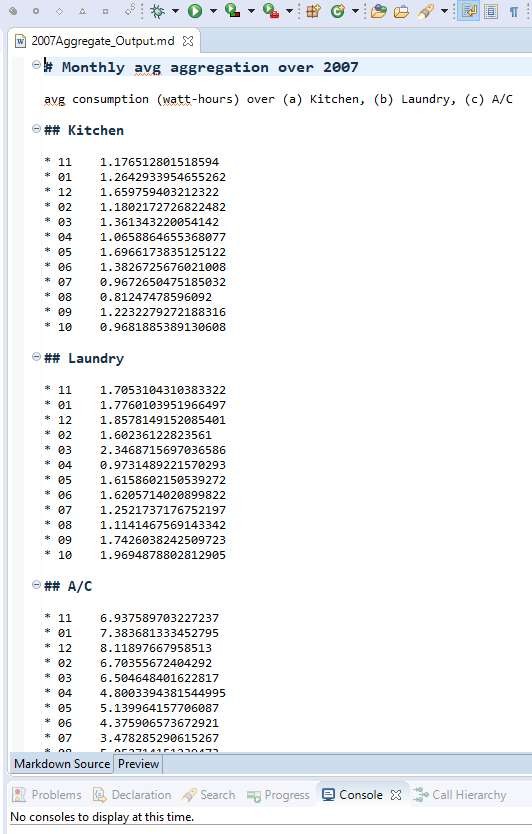
Επειδή έχουμε μετρήσεις ανά δευτερόλεπτο (άρα πάρα πολλές),οι αναλυτές ενδιαφέρονται για στατιστικά στοιχεία για αυτές. Εμείς θα περιοριστούμε στο να αποτιμούμε (α) άθροισμα ή (β) μέση τιμή κατανάλωσης ανά χρονική περίοδο. Οπότε, για παράδειγμα, ο αναλυτής θα πρέπει να μπορεί να ζητήσει “για κάθε μήνα,μέση 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:

* Εποχήτουχρόνου:WINTER,SPRING,SUMMER,AUTUMN
* Μήνας -- μπορείτε να χρησιμοποιήσετε είτε συμβολοσειρές JAN – DEC, είτε συμβολοσειρές"01"–"12"είτεμιααριθμητικήκωδικοποίηση,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 συγκεκριμένηςχρονιάς (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.

/\*Πρακτικά,οιχρονικέςπερίοδοι λειτουργούν ως ομαδοποιοί για τις μετρήσεις: αν π.χ., θέλουμε τη μέση τιμή ανά μήνα, το τρόπος να το πετύχουμεείναι:



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

* 1. παίρνουμε όλες τις μετρήσεις του συνόλου δεδομένων,
  2. μία-μία μέτρηση, την βάζουμε σε ένα υποσύνολο μετρήσεων που αντιστοιχεί στο μήνα της μέτρησης αυτής,και,
  3. για κάθε μήνα,βγάζουμε τη μέση τιμή από το υποσύνολο των μετρήσεων που του έχουμε αντιστοιχίσει, για κάθε μία από τις τρεις μετρούμενες κατηγορίες (κουζίνα,πλυντήρια,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](https://www.kaggle.com/uciml/electric-power-consumption-data-setΟαναλυτής)αναλυτής ερωτάται ποιο αρχείο θέλει να φορτώσει και παρέχει το σχετικό path.To 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 δίνει 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, φορμαρισμένο ως heading1
* Στηδεύτερηγραμμήείναιοτύποςγιατηναθροιστικήσυνάρτηση(sum/avg)καιένα σταθερόκείμενο
* Μετά,υπάρχειαπόένακείμενογιακάθεμίααπότις3μετρούμενεςποσότητες,ωςheading 2(Kitchen/Laundry/AC)
* Ακολουθούνζεύγη<timeunit> <aggregatemeasurement> για την αντίστοιχημετρούμενη 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., ταπεριεχόμενα, 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

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

#Heading

##Sub-heading

Paragraphs are separated by a blankline.

Bulletlist:

* 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. ΟικρίσιμεςκλάσειςείναιοιLocalDate(γιαημ/νιες),LocalTime(γιαώρες/λεπτά/…)καιοσυνδυασμός 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 πουχρειάζεται).

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](http://www.baeldung.com/java-8-date-time-intro)https://[www.mkyong.com/tutorials/java-date-time-tutorials/](http://www.mkyong.com/tutorials/java-date-time-tutorials/)https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html

# Apachecommonsmathstatistics

Ένα από τα χαρακτηριστικά του μοντέρνου προγραμματισμού είναι να επαναχρησμιποίηση έτοιμου κώδικα. To 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 σύνολομετρήσεων.

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.

ΕΙΝΑΙΜΙΑΕΞΑΙΡΕΤΙΚΗΕΥΚΑΙΡΙΑΝΑΜΕΛΕΤΗΣΕΤΕΕΤΟΙΜΟΚΩΔΙΚΑΚΑΙΝΑΣΚΕΦΤΕΙΤΕ ΚΡΙΤΙΚΑ ΠΩΣΔΟΥΛΕΥΕΙ.

# Software Architecture &Specifications

Η αρχιτεκτονική του λογισμικού σε πακέτα, σας δίδεται σχετικά προκαθορισμένα. Το υπό κατασκευήσύστημαείναιένασύστημα2επιπέδων,ενόςclientκομματιούπουείναιυπεύθυνο για την διάδραση με τον τελικό χρήστη και ενόςserverκομματιού που είναι υπεύθυνο για την διεκπεραίωση (α) του φορτώματος των αρχικών δεδομένων, (β) της ομαδοποίησης και υπολογισμούστατιστικώνανάλογαμετοτιζητάοαναλυτής,και,(γ)τηνπαραγωγήαναφορών, σε περισσότερα του ενός format, με το αποτέλεσμα του υπολογισμού. Στην πλευρά του server υπάρχουνδιάφοραpackages(πακέτα)πουτοκαθέναέχειτηδικήτουλειτουργικότητα.Σεκάθε πακέτο περιέχεται ένα interface, το οποίο εσείς πρέπει να υλοποιήσετε για να μπορείτε να διεκπεραιώσετετηλειτουργικότητατουκάθεπακέτου.

Τα πακέτα είναι:

client:packagetobefilledbyyouwiththeclient-sidefunctionality(giventoyouempty;youmust complete it)

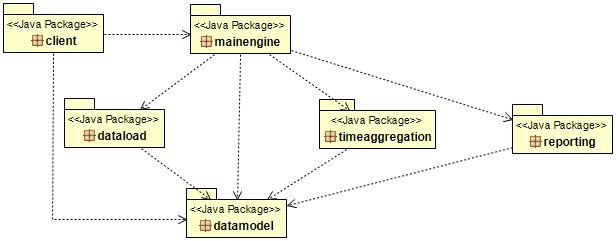
mainegine:mainpackageincludingtheIMainEngineinterfacethatexportsthefundamentalfunctionality of the back-end, and its implementation.

dataload:thepackageresponsibleforreadingthemeasurementdatafromtheinputfilesandloadingthemintheappropriatecollection(s)inmemory.Themainfunctionalityofthis subsystemissummarizedbytheinterfaceILoader.

timeaggregation:thepackageresponsibleforgroupingtheloadedmeasurementsintogroups,onepertimeperiod,andforextractingtherespectivestatistics.ThemainfunctionalityofthissubsystemissummarizedbytheinterfaceIAggregator.

reporting:thepackageresponsibleforcreatingoutputreportswiththestatisticsforthedatasets.Reports come in different formats, e.g., text, html, markdown (md). The main functionality of this subsystem is summarized by the interfaceIResultReporter.

datamodel:thepackagekeepingthedomainclassesofthesystem.NotablemodulesaretheIResultinterface, that prescribes the functionality of the aggregation result, and, the MeasurementRecordclassthatistherepresentativeofeachmeasurement=lineinthe inputfiles.



Σχήμα 1Αρχιτεκτονική 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 θαεντάξετεμέσαστοκάθεπακέτοείναιδικόσαςθέμακαιαντικείμενοτηςσχεδίασης,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 επέμβετε σε αυτό.

/\*Ναι,κάποιαπράγματασταinterfacesπουσαςδίνονταιθαμπορούσανναείναιπιοαπλά.Αλλάκαιστηζωή,ποτέδενθαέχετενααντιμετωπίσετεspecificationsπουναείναιβέλτισταωςπροςτην απλότητάτους.Ασχέτωςανθαμπορούσανναείναιαπλούστεραταinterfaces,είναιαυτάπουείναι,καιδενμπορείτενατααλλάξετε…\*/

# Τι σας δίνεται έτοιμο

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

σαςδίνεταιένααρχείο.zipστοοποίοπεριλαμβάνονταιδύοφάκελοι(α)μετηδομήτουsrcκαιτα σχετικάinterfaces,και(β)έναςφάκελοςResourcesμεδεδομένα(i)σεμικράμεγέθηγιανακάνετε ελέγχους και (ii) σε κανονικά μεγέθη για να τρέξετε το σύστημα που θα φτιάξετε. Επειδή τα δεδομένααπότοKaggleέχουνατέλειεςκαιελλείψειςεμείςσαςδίνουμε:(α)έναυποσύνολότους μετο2007,και(β)τηντριετία2007-09,επίσηςχωρίςατελείςεγγραφές.Επίσης,γιανακάνετε ελέγχους,σαςδίδονταικαιμικράσύνολαδεδομένων,γιαταοποίαμπορείτεναεπιβεβαιώσετετην ορθότητα των υπολογισμώνσας

Εσείς πρέπει

* ΝαφτιάξετεένανέοEclipseprojectμεόνομα2019-2020\_<AM1>\_<AM2>\_<AM3>μετα ΑΜ των μελών της ομάδας σας (ώστε αφού τα κάνετε turnin να ξέρουμε ποιανού είναι κάθεproject).
* Να κάνετε paste μέσα του τους δύο υποφακέλους που βρίσκονται στο 2019- 2020\_projectMaterial.zipπουσαςδίνεταιστοπαραπάνωlink.
* Να μην πειράξετε τίποτα από τα έτοιμα interfaces, αλλά μόνο τις κλάσεις MeasurementRecord και MainEngineFactory, τις οποίες θα πρέπει να συμπληρώσετε. Φυσικάθαπρέπεινασυμπληρώσετεκαιτιςυπόλοιπεςκλάσειςτουσυστήματος.

ΣαςδίνουμεέτοιμοτοunittestMainEngineTestγιατηνκεντρικήμηχανή,μεελέγχουςγια μερικές από τις βασικές λειτουργίες του συστήματος,στο πακέτο test. Αυτό θα πρέπει να τρέχειμεεπιτυχίαστοτελικόσαςπαραδοτέο!Φυσικάθαπρέπειναφτιάξετεκαιεπίμέρους unit tests καιεσείς.

# Οδηγίες και υποχρεώσεις

ΕίναιαπαράβατοείναιΝΑΥΛΟΠΟΙΗΣΕΤΕΕΠΑΚΡΙΒΩΣΤAINTERFACESπουσαςδίδονται.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-endserver.

EINAI 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> .ΥΠΟΧΡΕΩΤΙΚΑΠΡΕΠΕΙΝΑΑΚΟΛΟΥΘΗΣΕΤΕΤΟΥΠΟΔΕΙΓΜΑΠΟΥΣΑΣΔΙΝΕΤΑΙ!

Υλικό. Όλο το υποστηρικτικό υλικό για το project θα βρίσκεται στοURL<http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/supportingMaterial/> στονυποφάκελο της φετινήςχρονιάς

Στη διάρκεια του εξαμήνου:

* Σίγουρα θα δοθούν περαιτέρω εξηγήσεις στην εκφώνηση & ενδεχομένωςνααλλάξει/εμπλουτισθεί κάποιο μέρος της εκφώνησης! (άρα το δημοσιευθέν αρχείο της εκφώνησης μπορεί νααλλάζει)
* Οιλεπτομέρειεςτηςεξέτασηςθαεξαρτηθούναπότοναριθμότωνομάδωναπότημία, και τον αριθμό των βοηθών από την άλλη. Άρα: μέσα στο εξάμηνο θα διευκρινιστείτοπώςθαεξεταστείτεωςομάδακαιατομικά.

# Χρονοδιάγραμμα

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

[30/09] Εκφώνηση

3weeks Setup ofInfrastructure

[20/10::23.59] ΕγκατάστασηJava(ideally1.12),Eclipse(lastv.)στουςΗ/ΥσαςΓια όσους έχουν δυσκολία με την Java: ασκήσεις επανάληψης Εξοικείωση και πειραματισμός με το υλικό που σας δίδεται Εκκίνηση εργασιών στα Use Cases τηςεκφώνησης

Δεν υπάρχει κάτι να παραδώσετε

1. weeks Καταγραφή των UseCases

[03/11::23.59] Turnin:

DLV1.1: First version of the report with all the use cases (as pdf file)

1. weeks Design of classes && firstimplementations

[24/11::23.59] Turnin:

DLV2.1:Secondversionofthereport(pdf)with(a)thesectionforthetestcases+

(b) class diagram(s) with the design of the system (προαιρετικά: any other diagrams)

DLV2.2: the current state of your srcPrint(OBLIGATORILY!!):DLV2.1tobehandedoveronthenextdayofthedeadline

3.5weeks Completeimplementation

[ΕΝΔΕΧΟΜΕΝΩΣ ΕΞΕΤΑΣΗΕΝΔΙΑΜΕΣΩΣ]

[18/12:: 13.00]

ΑΝΕΛΑΣΤΙΚΑ

Turnin:

DLV 3.1: tar file with the code for all classes

DLV3.2:FINALversionofthereport(pdf)withallthedesignandthe documentation of theproject

Print(OBLIGATORILY!!):DLV3.2tobehandedoverBEFORE/ATthedeadline

@ALLturnin’s Exclude the folder Resources from your turnin (which cannot belarger than

10MB).Ifyouwanttouseyourowntestfiles,includejustthem..

ΔενθαξεπεράσουμετοόριοτωνΧριστουγέννων.Ηπράξηέχειαποδείξειότιστιςγιορτέςοιομάδεςαποσυντονίζονται σε πολύ μεγάλο βαθμό. Έτσι,Η ΠΡΟΘΕΣΜΙΑ ΤΗΣ ΠΑΡΑΔΟΣΗΣ ΕΙΝΑΙ ΑΝΕΛΑΣΤΙΚΗ!Θαπιεστείτεπερισσότεροπριντιςγιορτές,αλλάθαφύγετεγιατιςγιορτέςχωρίς το φορτίο τουproject.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!