

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ
ΑΚΑΔΗΜΑΪΚΟ ΈΤΟΣ 2019 -2020

ΟΜΑΔΑ 4382 -4398 -4417

ΚΑΣΑΠΗΣ ΣΤΕΦΑΝΟΣ, ΑΜ :4382

ΚΟΥΦΟΠΟΥΛΟΣ ΜΥΡΩΝΑΣ, ΑΜ: 4398

ΜΑΡΚΑΚΗΣ ΦΩΤΙΟΣ, ΑΜ: 4417

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2019

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2019/11/02	v.01	Οργάνωση απαιτήσεων σε use cases	4382,4398,4417
2019/11/24	v.02	Αρχική σχεδίαση κλάσεων και ελέγχων	4382,4398,4417

• ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

Στην παρούσα ενότητα, παρατίθενται οι περιγραφές των use cases με βάση τις καταγεγραμμένες απαιτήσεις.

ΦορτωσεΑρχαιο

ID: UC 1

Description and Goal

Η use case «ΦορτωσεΑρχαιο» φορτώνει στο σύστημα ένα αρχείο μετρήσεων.

Actors (esp. primary actor)

Ο χρήστης.

Preconditions

Το σύστημα φορτώνει όλα τα αρχεία.

Basic Flow

- Το use case ξεκινάει όταν ο χρήστης παρέχει το σχετικό path.
- Το σύστημα φορτώνει το αρχείο που θα του ζητηθεί.
- Το σύστημα τροφοδοτεί ένα collection εγγράφων από το σχετικό path.

Extensions / Variations

Το σύστημα φορτώνει μόνο τις έγγραφες που δεν έχουν κενά πεδία ή πεδία που παραβιάζουν το format των δεδομένων.

Post conditions

ΚατασκευασεΣτατιστικά

ID: UC 2

Description and Goal

Η use case «ΚατασκευάσεΣυναθροιστικάΣτατιστικά» παράγει τα στατιστικά που θα ζητηθούν.

Actors (esp. primary actor)

Ο χρήστης.

Preconditions

Το σύστημα φορτώνει τις μετρήσεις των collections.

Basic Flow

- Το σύστημα ρωτάει τον χρήστη με ποιο τύπο χρονικής περιόδου θέλει να ομαδοποιήσει τα δεδομένα (π.χ. μήνα, εποχή κτλ).
- Ο χρήστης απάντα και φορτώνει η ζητούμενη χρονική περίοδος.
- Το σύστημα ρωτάει τον χρήστη με βάση ποια αθροιστική συνάρτηση θέλει να ομαδοποιήσει τις μετρήσεις (άθροισμα ή μέσο όρο).
- Ο χρήστης δίνει την αθροιστική συνάρτηση.
- Το αποτέλεσμα συνοδεύεται από ένα κείμενο (περιγραφή) του χρήστη.
- Το σύστημα αποθηκεύει το κείμενο το οποίο παρουσιάζεται αργότερα στα reports.

Extensions / Variations

Αν κάποιο από τα στοιχεία δεν δοθεί σωστά το σύστημα ζητάει από τον χρήστη να επαναλάβει την καταχώρηση ή να εξέλθει της λειτουργίας αυτής.

ΔημιουργησεΑναφορες

ID: UC 3

Description and Goal

Η use case «ΔημιούργησεΑναφορά» παράγει την τελική αναφορά.

Actors (esp. primary actor)

Ο χρήστης.

Preconditions

Το σύστημα φορτώνει τις μετρήσεις των collections.

Basic Flow

- Το σύστημα ρωτάει τον χρήστη το path που θέλει να κατασκευαστεί .
- Ο χρήστης γράφει το path .
- Το σύστημα ρωτάει τον τύπο της αναφοράς.
- Ο χρήστης απαντάει τον τύπο της αναφοράς.
- Το σύστημα χρησιμοποιεί το IResult που παρήχθη και ετοιμάζει την αναφορά.

Extensions / Variations

-

ΔειξεΤελικεςΑναφορες

ID: UC 4

Description and Goal

Επιδεικνύει των αναλυτικών πληροφοριών του ιστορικού των αναφορών.

Actors (esp. primary actor)

Ο χρήστης.

Preconditions

Το σύστημα φορτώνει τις μετρήσεις των collections.

Basic Flow

- Ο χρήστης ζητάει μια αναφορά από το σύστημα.
- Το σύστημα αποθηκεύει τα στοιχεία σε ένα ιστορικό αιτημάτων αναφορών .
- Δίνεται η δυνατότητα στον χρήστη να αιτηθεί οπτική παρουσίαση του ιστορικού αιτημάτων αναφορών

3.1. Το σύστημα παρουσιάζει την σχετική λίστα (την περιγραφή, όχι τα περιεχόμενα)

Extensions / Variations

-

Post conditions

:

- ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάσθηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω. Εδώ, ως υπόδειγμα: το project με την διάσπαση χρονοσειράς σε φάσεις.

- ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

- TRACEABILITY MATRIX

Η αντιστοίχιση use cases σε id's φαίνεται στον Πίνακα 1:

UC1	Load File
UC2	Construct Statistics
UC3	Create Reports
UC4	Show Final Reports
UC5	ExitProgram

Πίνακας 1 Σύνοψη use cases και των id's τους

Ο Πίνακας 2 είναι ο traceability matrix για τους ελέγχους μας. Στη συνέχεια, οι έλεγχοι επεξηγούνται πιο αναλυτικά.

	UC1	UC2	UC3	UC4	UC5
T1_V0_01	X				
T1_V0_02	X				
T2_V0_01		X			
T3_V0_01			X		
T3_V0_02			X		
T3_V0_03			X		
T4_V0_01				X	
T4_V0_02				X	
T5_V0					X

Πίνακας 2 Traceability matrix between use cases and tests

- USE CASE UC1: LOAD DATA

Involved methods

Test cases

ID	T1_V0_01	HappyDayScenario for DataLoad.loadFile()
Description	ON	All files have to be loaded
	RECEIVING	Request to load a specific measurement file
	ENSURE	That the System
	OUTPUTS	An updated collection of a measurement file
	SUCH THAT	It has the right structure
Pre-cond.		All files have to be loaded
Input		Give the right path
Output		The measurement file will be saved in a collection
Post-cond.		The system loads only the right files
Method To test		DataLoad.load()

ID	T1_V0_02	HappyDayScenario for DataLoad.collectionSave()
		...identical setup with T1_V0_01
Method To test		DataLoad.loadData()

Not designed yet: subscriptions with no empty fields, fields that don't pair the format

- USE CASE UC2: CONSTRUCT STATISTICS

Involved methods

Test cases

ID	T2_V0_01	HappyDayScenario for TimeAggregation.constructStatistics()
Description	ON	The last collection that have been loaded
	RECEIVING	Request to specify the time and the cumulative function
	ENSURE	That the System
	OUTPUTS	The statistics that the user asked
	SUCH THAT	The month or the day, the sum or the average
Pre-cond.		The system has loaded all the collections
Input		A text that later will be on the report
Output		The user asked the right things(if not repeat) or that he will exit the program
Post-cond.		The user gives valid data or exits
Method To test		TimeAggregation.aggregagteByTimeUnit()

Not designed yet: if user give wrong input the system have to ask the user to repeat or exit

- USE CASE UC3: CREATE REPORTS

Involved methods

Test cases

ID	T3_V0_01	HappyDayScenario for Reporting.createReports()
Description	ON	The last IResult that was produced
	RECEIVING	Request the user to give a path and the type of the report
	ENSURE	That the System
	OUTPUTS	A report
	SUCH THAT	If the data form the user are wrong they are being fixed or the user exits the operation
Pre-cond.		The last IResult that was produced
Input		The user gave the right path and a report(text, html, markdown)
Output		A final report
Post-cond.		If the IResult doesn't exist it is being fixed or the user exits the operation
Method To test		Reporting.reportResultInFile()

Not designed yet: If input from user is wrong or IResult doesn't exist the system have to ask the user to repeat or exist

USE CASE UC4: SHOW FINAL REPORTS

Involved methods

Test cases

ID	T4_V0_01	HappyDayScenario for Client.showFinalReports()
Description	ON	The system has loaded all the collections
	RECEIVING	Request the user if he wants a visual presentation of the history of the reports
	ENSURE	That the System
	OUTPUTS	The list that the user asked
	SUCH THAT	The list contains the right information
Pre-cond.		The system can store the data
Input		The user asks for a report
Output		The system stores the requests from the reports
Post-cond.		The reports can be shown to the user anytime
Method To test		Client.main()

- **USE CASE UC5: EXIT PROGRAM**

Involved methods

Test cases

ID	T4_V0	HappyDayScenario for Client.exitProgram()
Description	ON	The program has no errors
	RECEIVING	Request the user to exit the program
	ENSURE	That the System
	OUTPUTS	The system prints a farewell message
	SUCH THAT	The user can't use the program
Pre-cond.		The program has executed all the instructions the user asked
Input		The user ask to exit the program
Output		The system prints a farewell message
Post-cond.		The user can't use the program
Method To test		Client.exitProgram()

- **UNIT TESTS**

Στην παρούσα υποενότητα, παραθέτουμε πολύ συνοπτικά τα πιο σημαντικά unit test που σχεδιάστηκαν, οργανωμένα ανά κλάση. Δίνουμε μόνο το OREOS text.

Class ParserFactory

ON any context, RECEIVING a request for a new parser, ENSURE THAT THE SYSTEM OUTPUTS a not null parser SUCH THAT state remains intact.

Class AnalyzerFactory

ON any context, RECEIVING a request for a new analyzer, ENSURE THAT THE SYSTEM OUTPUTS a not null analyzer of the correct kind SUCH THAT state remains intact.

- **EKKPEMOTHTEΣ (TODO)**

Εκκρεμούν μη υλοποιημένοι έλεγχοι ως ακολούθως

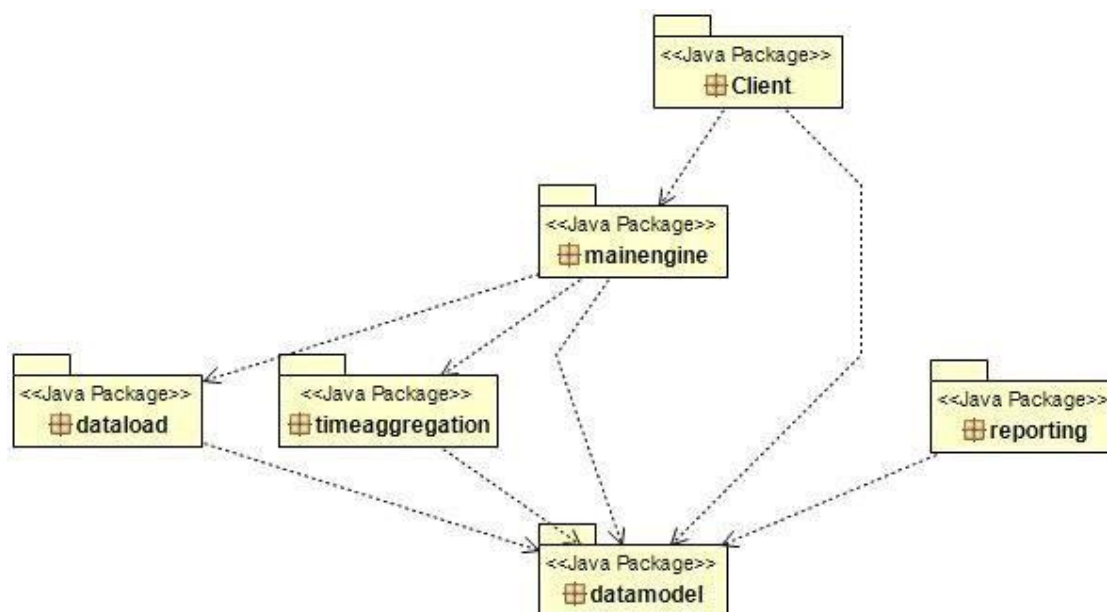
Unit tests are missing for several classes, both at the model and at the business logic leve

- ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

- ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

Η ανάλυση του κώδικα σε υποσυστήματα και πακέτα έχει νόημα μόνο όταν το μέγεθος και η πολυπλοκότητα του κώδικα επιτάσσουν την εν λόγω διαίρεση.

Το διάγραμμα των πακέτων του συστήματος ακολουθεί στο Σχ. 1.



Σχήμα 1. Διάγραμμα πακέτων (εδώ: από την αξιολόγηση εστιατορίου)

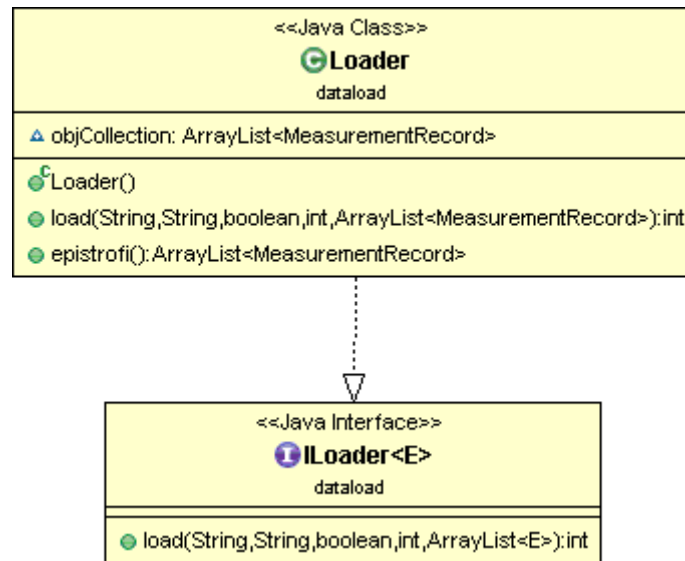
Ακολουθεί μια συνοπτική περιγραφή των πακέτων του συστήματος.

ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

client	Περιέχει τις boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη
mainengine	Κεντρική business logic engine, along with the necessary interface to export to the boundary classes
dataLoad	Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα
reporting	Υποσύστημα παραγωγής αναφορών
dataModel	Domain classes of the system
dataload	Φόρτωση αρχείου και του path που θα δώσει ο χρήστης

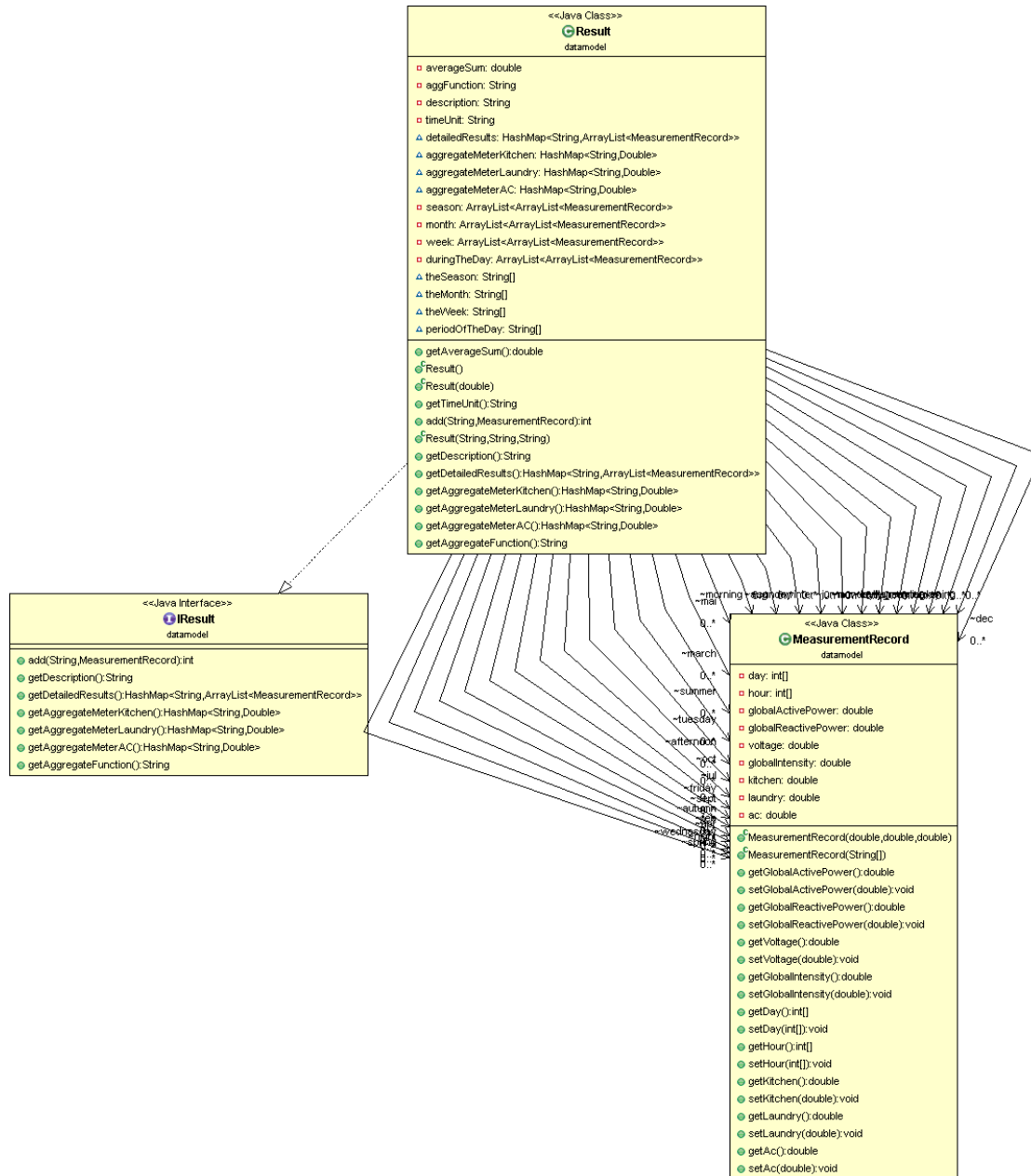
Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος (εδώ: από την αξιολόγηση εστιατορίου)

package dataload;



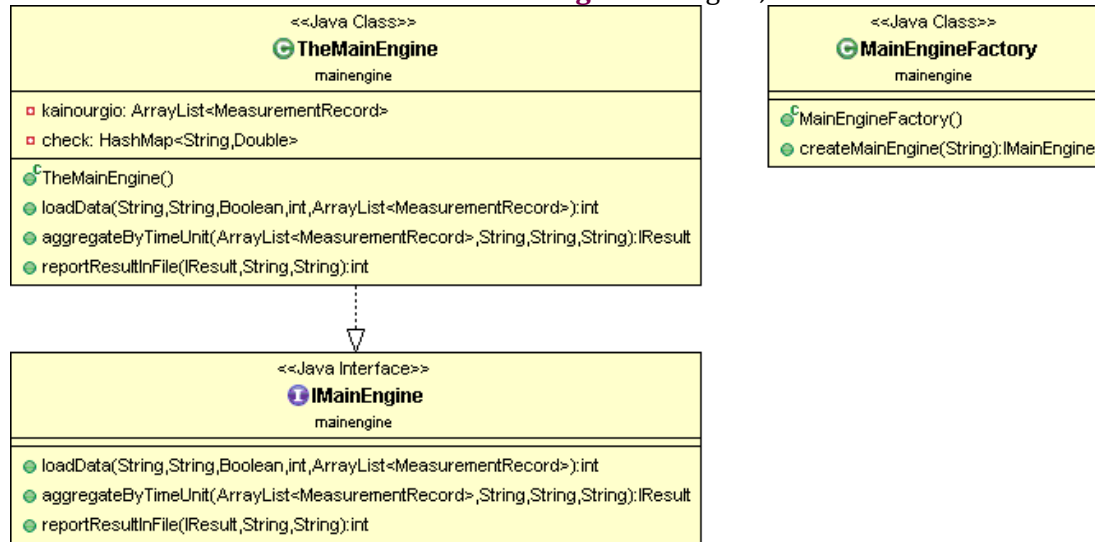
Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο dataload

Package datamodel;



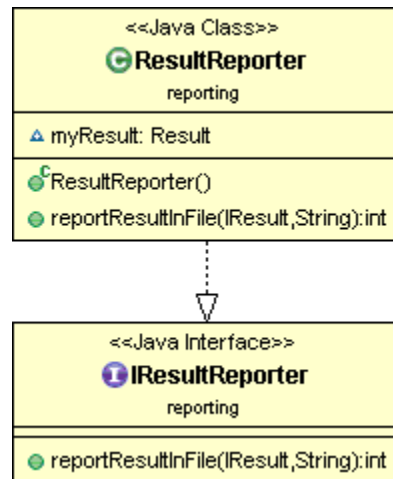
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο datamodel

Package mainengine;



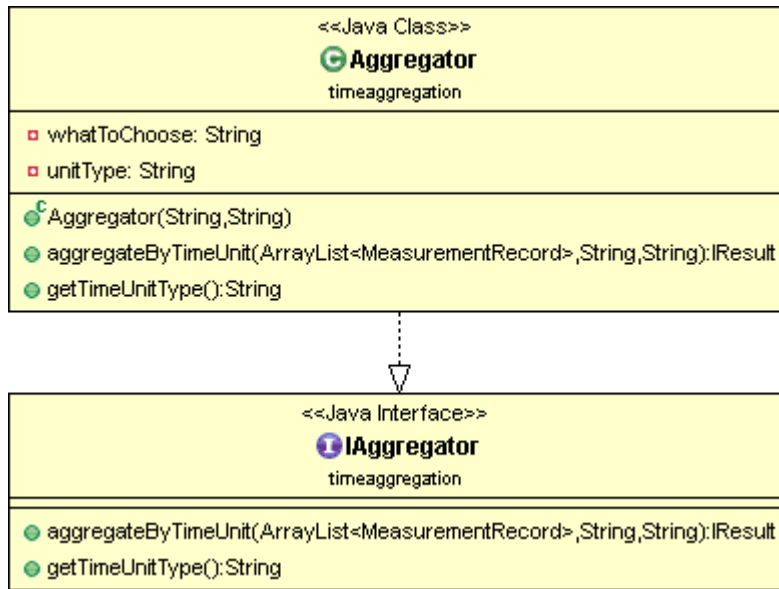
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο mainengine

Package resultreporter;

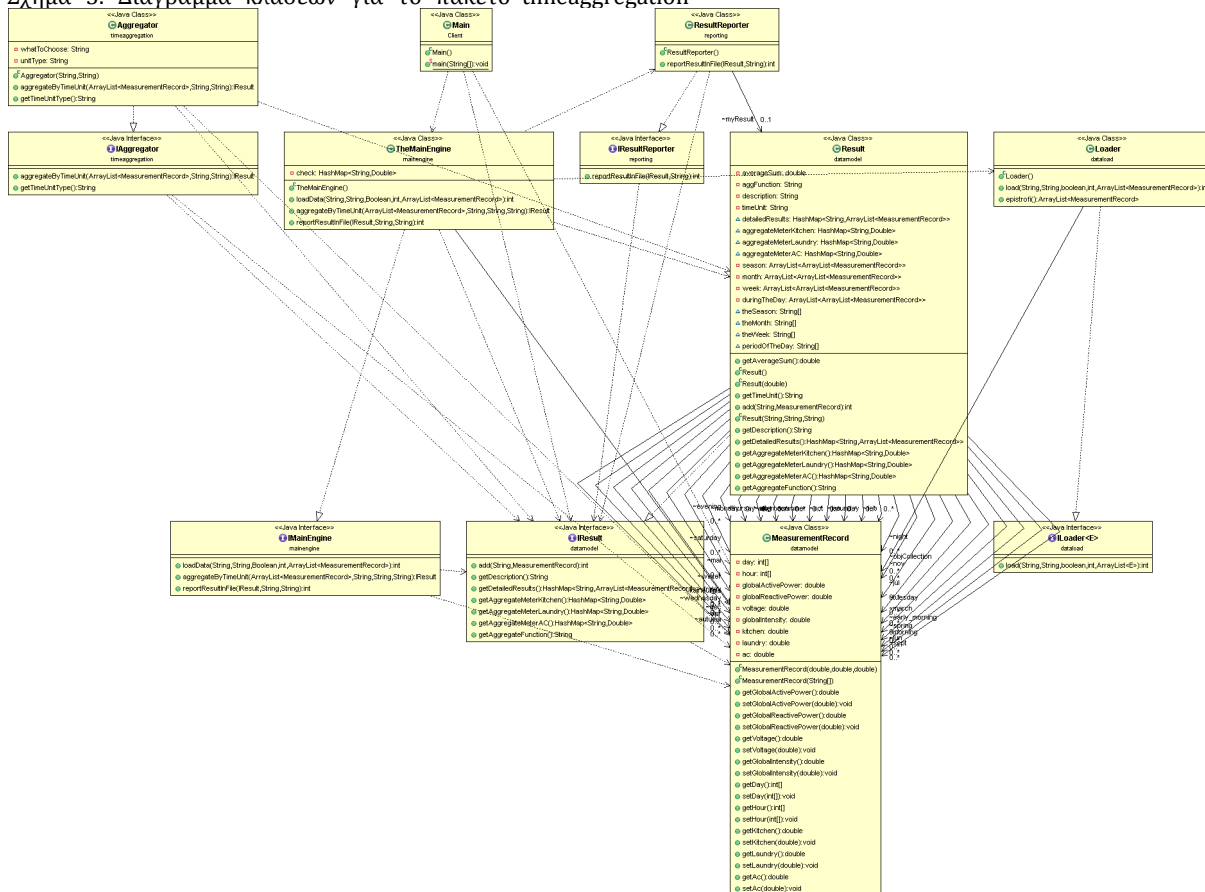


Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο resultreporter

Package timeaggregation;



Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο timeaggregation



- ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ ΚΑΙ ΣΥΝΕΠΕΙΑ ΠΡΟΣ ΤΙΣ ΑΠΑΙΤΗΣΕΙΣ
- DOMAIN CLASSES

Package dataLoad	MeasurementRecord
---------------------	-------------------

- BUSINESS LOGIC CLASSES

Package dataLoad	Loader η οποία είναι υπευθυνη για την φορτωση των αρχειων , ILoader
Package dataModel	Result η οποία είναι υπεθυνη για την σωστη τοποθετηση των αποτελεσματος , IResult , MeasurementRecord
Package Reporting	ResultReporter , IResultReporter
Package timeAggregat ion	Aggregator η οποία είναι υπευθυνη για τις πραξεις

- **BOUNDARY CLASSES**

Package client	Main, μια client class για την αλληλεπίδραση με το χρήστη μέσω κονσόλας.
Package mainEngine	TheMainEngine

- **ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ**

Υπάρχουν 4 use cases: ΦορτώσεΑρχείο, ΚατασκεύασεΣτατιστικά, ΔημιούργησεΑναφορές, ΔείξεΤελικέςΑναφορές.

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods
ΦορτώσεΑρχείο	Int load()	Int loadData()
ΚατασκεύασεΣτατιστικά	IAggregator createAggregator()	IResult aggregateByTimeUnit()
ΔημιούργησεΑναφορές	ReportResultInFile()	ReportResultInFile()

Πίνακας 4 Επαλήθευση απεικόνισης use cases σε μεθόδους