
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ
ΑΚΑΔΗΜΑΪΚΟ ΈΤΟΣ 2019-2020

ΟΜΑΔΑ 2766-2784-2821

ΜΠΑΡΜΠΑΛΙΑΣ ΕΥΑΓΓΕΛΟΣ, ΑΜ:2766

ΠΑΝΑΓΙΩΤΟΥ ΔΗΜΗΤΡΙΟΣ, ΑΜ:2784

ΣΤΑΒΑΡΑΚΗΣ ΠΑΝΑΓΙΩΤΗΣ, ΑΜ:2821

Νοέμβριος 2019

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
25/10/2019	v.01	Οργάνωση απαιτήσεων σε use cases	Μπαρμπαλιάς Ευάγγελος, Παναγιώτου Δημήτριος, Σταβαράκης Παναγιώτης
17/11/2019	v.02	Αρχική σχεδίαση κλάσεων και ελέγχων	Μπαρμπαλιάς Ευάγγελος, Παναγιώτου Δημήτριος, Σταβαράκης Παναγιώτης
21/11/2019	v.03	Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων	Μπαρμπαλιάς Ευάγγελος, Παναγιώτου Δημήτριος, Σταβαράκης Παναγιώτης
23/11/2019	v.04	Διορθώσεις στη σχεδίαση κλάσεων και γενικές διορθώσεις.	Μπαρμπαλιάς Ευάγγελος, Παναγιώτου Δημήτριος, Σταβαράκης Παναγιώτης
24/11/2019	v.05	Τελική έκδοση	Μπαρμπαλιάς Ευάγγελος, Παναγιώτου Δημήτριος, Σταβαράκης Παναγιώτης

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

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

LOADFILE

ID: UC 1

DESCRIPTION AND GOAL

Η use case “LoadFile” φορτώνει το αρχείο και τροφοδοτεί ένα collection εγγραφών από αυτό.

ACTORS (ESP. PRIMARY ACTOR)

Ο αναλυτής.

PRECONDITIONS

-

BASIC FLOW

1. Το use case “LoadFile” ξεκινάει όταν ο αναλυτής επιλέξει “Load File”.
2. Το σύστημα ζητάει από τον αναλυτή το path του αρχείου με τις μετρήσεις το οποίο θέλει να φορτώσει.
3. Ο αναλυτής δίνει το path του αρχείου μετρήσεων που θέλει.
4. Το σύστημα φορτώνει το αρχείο το οποίο του δόθηκε.
5. Το σύστημα τροφοδοτεί ένα collection εγγραφών.

EXTENSIONS / VARIATIONS

1. Στην περίπτωση κατά την οποία οι εγγραφές έχουν κενά πεδία, αυτές δεν θα φορτώνονται από το σύστημα.
2. Στην περίπτωση κατά την οποία οι εγγραφές παραβιάζουν το format των δεδομένων δεν θα φορτώνονται από το σύστημα.

POST CONDITIONS

ALTERNATIVE FLOW 1

1. Ανά πάσα στιγμή ο αναλυτής μπορεί να εξέλθει του συστήματος.

CREATESTATISTICS

ID: UC 2

DESCRIPTION AND GOAL

Η use case “CreateStatistics” δημιουργεί τα συναθροιστικά στατιστικά ανάλογα με τις απαιτήσεις του αναλυτή.

ACTORS (ESP. PRIMARY ACTOR)

Ο αναλυτής.

PRECONDITIONS

Πρέπει να έχουν φορτωθεί τα collections μετρήσεων.

BASIC FLOW

1. Το use case “CreateStatistics” ξεκινά όταν ο αναλυτής επιλέξει “Create Statistics”.
2. Το σύστημα ρωτάει τον αναλυτή :
 - 2.1. Τον τύπο της χρονικής περιόδου.
 - 2.2. Την αθροιστική συνάρτηση που θα χρησιμοποιηθεί.
3. Ο αναλυτής επιλέγει τον τύπο της χρονικής περιόδου και την αθροιστική συνάρτηση.
4. Το σύστημα ζητά από τον αναλυτή να δώσει ένα κείμενο.
5. Ο αναλυτής γράφει το κείμενο-περιγραφή.
6. Το σύστημα δημιουργεί :
 - 6.1. Αθροιστικά Στατιστικά εάν η επιλογή του αναλυτή ήταν Άθροισμα.
 - 6.2. Στατιστικά μέσου όρου εάν η επιλογή του αναλυτή ήταν Μέσος Όρος.

EXTENSIONS / VARIATIONS

1. Εάν ο αναλυτής επιλέξει/δώσει εσφαλμένη επιλογή/απάντηση, το σύστημα επαναλαμβάνει τη διαδικασία ή ζητά από τον αναλυτή να εξέλθει της διαδικασίας αυτής.

POST CONDITIONS

1. Ο κατάλληλος τύπος στατιστικών έχει δημιουργηθεί.

ALTERNATIVE FLOW 1

1. Ανά πάσα στιγμή ο αναλυτής μπορεί να εξέλθει του συστήματος.

ID: UC 3

DESCRIPTION AND GOAL

Η use case “GenerateReports” δημιουργεί αρχείο αναφοράς συγκεκριμένου τύπου και δομής.

ACTORS (ESP. PRIMARY ACTOR)

Ο αναλυτής.

PRECONDITIONS

1. Πρέπει να έχει δημιουργηθεί αποτέλεσμα στατιστικών.

BASIC FLOW

1. Το use case “GenerateReports” ξεκινά όταν επιλέξει ο αναλυτής “Generate Reports”.
2. Το σύστημα ζητάει από τον αναλυτή
 - 2.1. να δώσει το path στο οποίο θα αποθηκευτεί η αναφορά
 - 2.2. και τον τύπο του αρχείου.
3. Ο αναλυτής δίνει το path και τον τύπο του αρχείου.
4. Το σύστημα δημιουργεί το αρχείο με την αναφορά
 - 4.1. με συγκεκριμένο τύπο αρχείου και
 - 4.2. συγκεκριμένη δομή

EXTENSIONS / VARIATIONS

1. Σε περίπτωση που δεν έχουν δημιουργηθεί στατιστικά, το σύστημα ζητά από τον αναλυτή :
 - 1.1. να επαναλάβει τη διαδικασία
 - 1.2. είτε να εξέλθει από τη λειτουργία αυτή.

POST CONDITIONS

1. Η αναφορά έχει δημιουργηθεί στο path που έχει δοθεί

ALTERNATIVE FLOW 1

1. Ανά πάσα στιγμή ο αναλυτής μπορεί να εξέλθει του συστήματος.

VIEWHISTORYREPORTS

ID: UC 4

DESCRIPTION AND GOAL

Η use case “ViewHistoryReports” κρατάει αντίγραφα από τα αιτήματα των αναφορών και τα εμφανίζει στον αναλυτή.

ACTORS (ESP. PRIMARY ACTOR)

Ο αναλυτής

PRECONDITIONS

1. Να έχει ζητηθεί τουλάχιστον μία αναφορά.

BASIC FLOW

1. Η use case “ViewHistoryReports” ξεκινάει όταν ο αναλυτής ζητήσει μία αναφορά.
2. Το σύστημα κρατάει αντίγραφο από την μεταπληροφορία των αναφορών.
3. Ο αναλυτής ζητάει την εμφάνιση του ιστορικού αιτημάτων.
4. Το σύστημα εμφανίζει σχετική λίστα με την μεταπληροφορία των αναφορών.

EXTENSIONS / VARIATIONS

-

POST CONDITIONS

-

ALTERNATIVE FLOW 1

1. Ανά πάσα στιγμή ο αναλυτής μπορεί να εξέλθει του συστήματος.

2 ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάστηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω.

2.1 ΕΛΕΓΧΟΣ USE CASES VIA SYSTEM TESTS

2.1.1 TRACEABILITY MATRIX

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

UC 1	LoadFile
UC 2	CreateStatistics
UC 3	GenerateReports
UC 4	ViewHistoryReports

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
T1_V0_01	X			
T1_V0_02	X			
T1_V1_01	X			
T1_V1_02	X			
T1_V2_01	X			
T1_V2_02	X			
T2_V0		X		
T2_V1		X		
T3_V0_01			X	
T3_V0_02			X	
T3_V1_01			X	
T3_V1_02			X	
T4_V0				X
T4_V1				X

Πίνακας 2 Traceability matrix between use cases and tests

2.1.2 USE CASE UC 1: LOADFILE

Involved methods

`DataLoader.load()` , `MainEngine.loadData()`

Test cases

ID	T1_V0_01	HappyDayScenario for file loaders
Description	ON	any context
	RECEIVING	A request to load a file with consumption measurements
	ENSURE	That the System
	OUTPUTS	The appropriate number
	SUCH THAT	#records of the arraylist loaded from the file

Pre-cond.		No specific precondition constructed
Input		An input file with records ,without violating the specific structure and format
Output		The #rows of the arraylist with the same #records as the contexts of the input and no offending values
Post-cond.		No state properties tested
Method To test		DataLoader.load()

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

ID	T1_V1_01	Missing input file Scenario
Description	ON	Any context
	RECEIVING	A missing input file
	ENSURE	That the system
	OUTPUTS	Displays an error message
	SUCH THAT	The system requesting for valid file
Pre-cond.		No specific precondition constructed
Input		A missing input file or a wrong path to this file
Output		An error message informing the analyst of this
Post-cond.		No state properties tested
Methods to test		Loader.load()

ID	T1_V1_02	Missing file for MainEngine.loadData()
		...identical setup with T1_V1_01
Method to test		MainEngine.loadData()

ID	T1_V2_01	Invalid file's records format
Description	ON	Any context
	RECEIVING	A file with invalid records' format
	ENSURE	That the system
	OUTPUTS	A specific arraylist
	SUCH THAT	The properly arraylist with valid only records and a message of #invalid records
Pre-cond.		No specific precondition constructed
Input		A file with measurements that have blank fields or are

		violating the specific format
Output		The #rows of the arraylist with valid only measurements and a warning message of invalid records
Post-cond.		No state properties tested
Methods to test		Loader.load()

ID	T1_V2_02	Invalid file's records format
		...identical setup with T1_V2_01
Method to test		MainEngine.loadData()

2.1.3 USE CASE UC 2: CREATESTATISTICS

Involved methods

```
MainEngine.aggregateByTimeUnit()  
IResult --> MainEngine.aggregateByTimeUnit(ArrayList<MeasurementRecord>  
    inputMeasurements,String aggregatorType,String aggFunction,String  
    description)
```

Test cases

ID	T2_V0	HappyDayScenario for MainEngine.aggregateByTimeUnit
Description	ON	A collection of records
	RECEIVING	Request to create statistics for a specific time period and method
	ENSURE	That the System
	OUTPUTS	The appropriate statistics of the measurements
	SUCH THAT	state is intact
Pre-cond.		The correct collection of measurements has been created
Input		the abovementioned input file, the time period and the aggregation function(Summary or Average)
Output		The appropriate Consumption Result
Post-cond.		No state properties tested
Method To test		MainEngine.aggregateByTimeUnit(ArrayList<MeasurementRecord> inputMeasurements, String aggregatorType, String aggFunction, String description)

ID	T2_V1	Invalid time period
Description	ON	A collection of records
	RECEIVING	Request to create statistics for an invalid time period
	ENSURE	That the system
	OUTPUTS	An error message

	SUCH THAT	The process is being repeated
Pre-cond.		The correct collection of measurements has been created
Input		An invalid period of time or an unfeasible date
Output		An error message and a request for repeating the process
Post-cond.		Repeats process if request is been accepted. Exits otherwise.
Method To test		MainEngine.aggregateByTimeUnit(ArrayList<Measurement Record> inputMeasurements, String aggregatorType, String aggFunction, String description)

2.1.4 USE CASE UC 3: GENERATEREPORTS

Involved methods

MainEngine.reportResultInFile() ,
Reporter.reportResultInFile()

Test cases

ID	T3_V0_01	HappyDayScenario for MainEngine.reportResultInFile()
Description	ON	the specific statistics that have been created
	RECEIVING	Request to generate a report into a file
	ENSURE	That the System
	OUTPUTS	An appropriate file
	SUCH THAT	The file is created in the given path and contains the aggregated statistics
Pre-cond.		The specific statistics are created
Input		A path in which the file will be saved, the file type ("html","txt","md")
Output		0 for successful process
Post-cond.		The appropriate file has been created
Method To test		MainEngine.reportResultInFile(IResult result, String reportType, String filename)

ID	T3_V0_02	HappyDayScenario for Reporter.reportResultInFile()
		...identical setup with T3_V0_01
Method to test		Reporter.reportResultInFile()

ID	T3_V1_01	Invalid given path for MainEngine.reportResultInFile()
Description	ON	the specific statistics that have been created
	RECEIVING	Request to generate a report into a file with invalid path

	ENSURE	That the System
	OUTPUTS	An error message
	SUCH THAT	The file cannot be created in the given path
Pre-cond.		The specific statistics are created
Input		An invalid path in which the file will be saved, the file type ("html","txt","md")
Output		A negative number for unsuccessful process
Post-cond.		The process of giving path is been repeated
Method To test		MainEngine.reportResultInFile(IResult result, String reportType, String filename)

ID	T3_V1_02	HappyDayScenario for Reporter.reportResultInFile()
		...identical setup with T3_V1_01
Method to test		Reporter.reportResultInFile()

2.1.5 USE CASE UC 4 : VIEWHISTORYREPORTS

Involved methods

HistoryReporter.viewReportRecords()

Test cases

ID	T4_V0	HappyDayScenario for HistoryReporter.viewReportRecords()
Description	ON	One or more reports have been generated/stored in file
	RECEIVING	Request to display report records
	ENSURE	That the System
	OUTPUTS	An appropriate list
	SUCH THAT	The list is displaying the metadata of the reports
Pre-cond.		One or more generated reports that are stored in files
Input		Request to view history of reports
Output		A list of reports' metadata such that path and description
Post-cond.		No state properties tested
Method To test		HistoryReporter.viewReportRecords()

ID	T4_V1	None report record for HistoryReporter.viewReportDetails()
Description	ON	None previous generated report
	RECEIVING	Request to display report records

	ENSURE	That the system
	OUTPUTS	An error message
	SUCH THAT	The system must abort the process
Pre-cond.		The previous generated reports are zero
Input		Request to display report records
Output		An error message of invalid choice
Post-cond.		Aborts the process
Method To test		HistoryReporter.viewReportDetails()

2.2 UNIT TESTS

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

Class DataLoaderFactory

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

Class ResultFactory

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

Class TypeReportFactory

ON any context, RECEIVING a request for a new report, ENSURE THAT THE SYSTEM OUTPUTS a not null IReporter of the correct type SUCH THAT the appropriate report is generated.

Class MainEngineFactory

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

Class AggregatorFactory

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

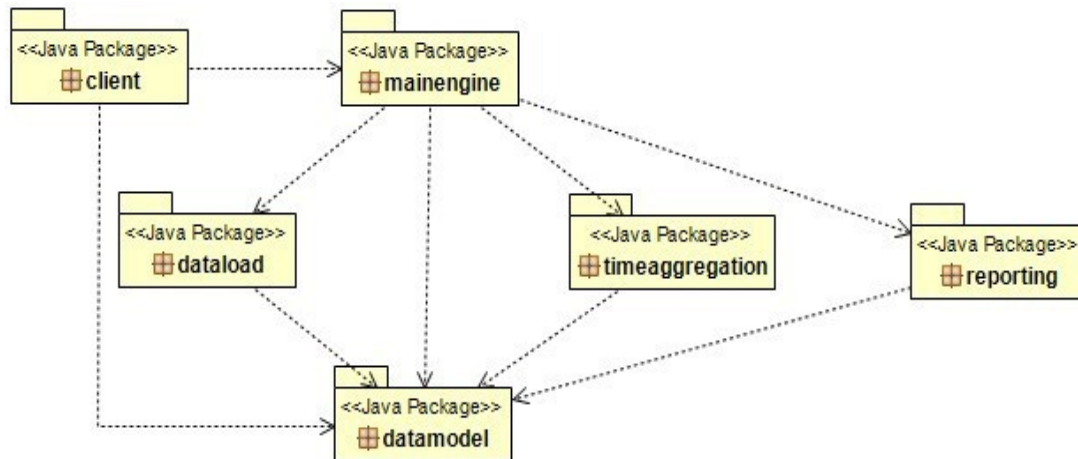
2.3 ΕΚΚΡΕΜΟΤΗΤΕΣ (TODO)

Εκκρεμούν μη υλοποιημένοι έλεγχοι ως ακολούθως (αν υπάρχουν εκκρεμότητες, παραθέστε την TODO λίστα ελέγχων που πρέπει να ετοιμαστούν)

3. ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

A. ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

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



Σχήμα 1. Διάγραμμα πακέτων

Ακολουθεί μια συνοπτική περιγραφή των πακέτων του συστήματος.

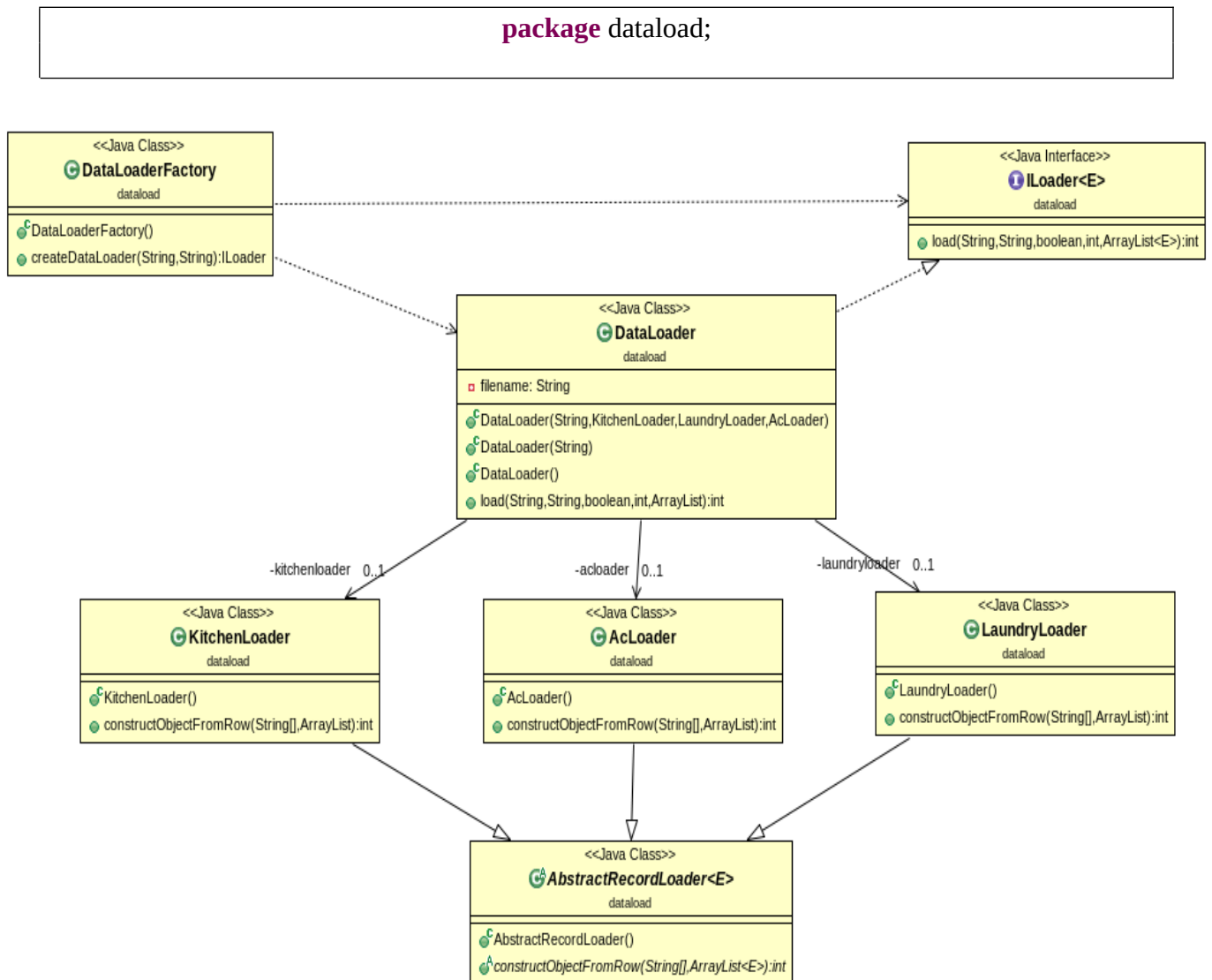
ΠΑΚΕΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

client	Περιέχει τις boundary classes που είναι υπεύθυνες για την αλληλεπίδραση με το χρήστη
mainengine	Κεντρική business logic engine, με το απαραίτητο interface για την υλοποίηση του
dataload	Υποσύστημα αλληλεπίδρασης με τα αρχεία δεδομένων, για την ανάκτησή τους από το σύστημα
reporting	Υποσύστημα παραγωγής αναφορών
datamodel	Domain classes του συστήματος
timeaggregation	Υποσύστημα για την οργάνωση των μετρήσεων ανα χρονική περίοδο

Πίνακας 3. Συνοπτική περιγραφή πακέτων συστήματος

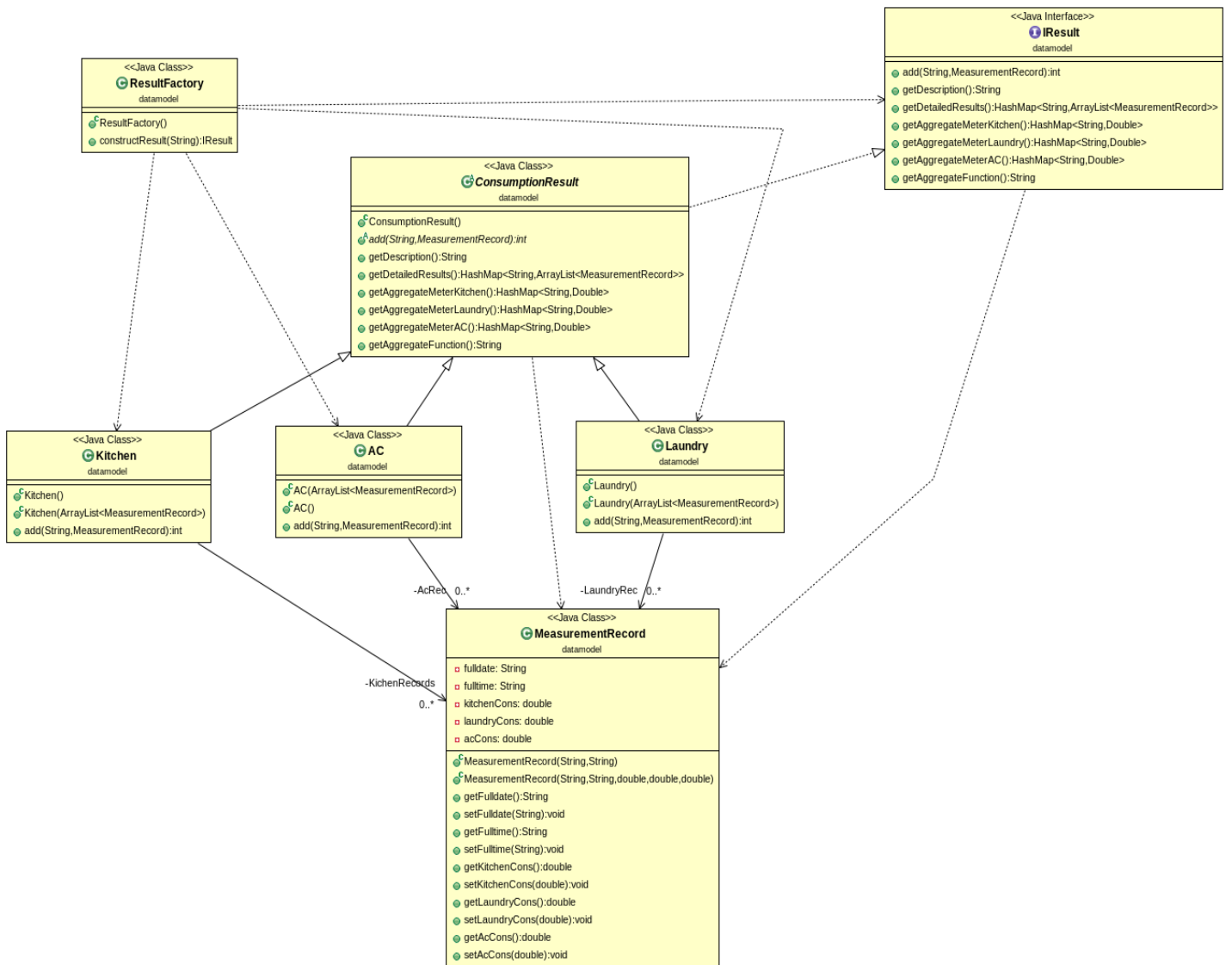
B. ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων.



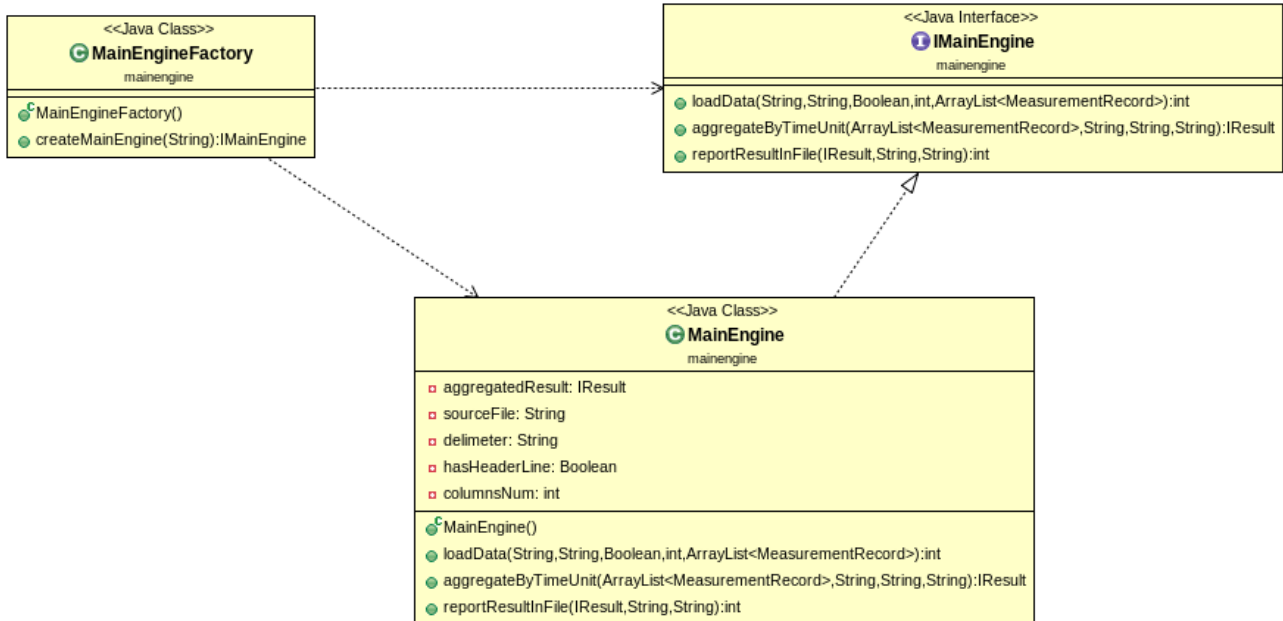
Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο dataLoad

package datamodel;



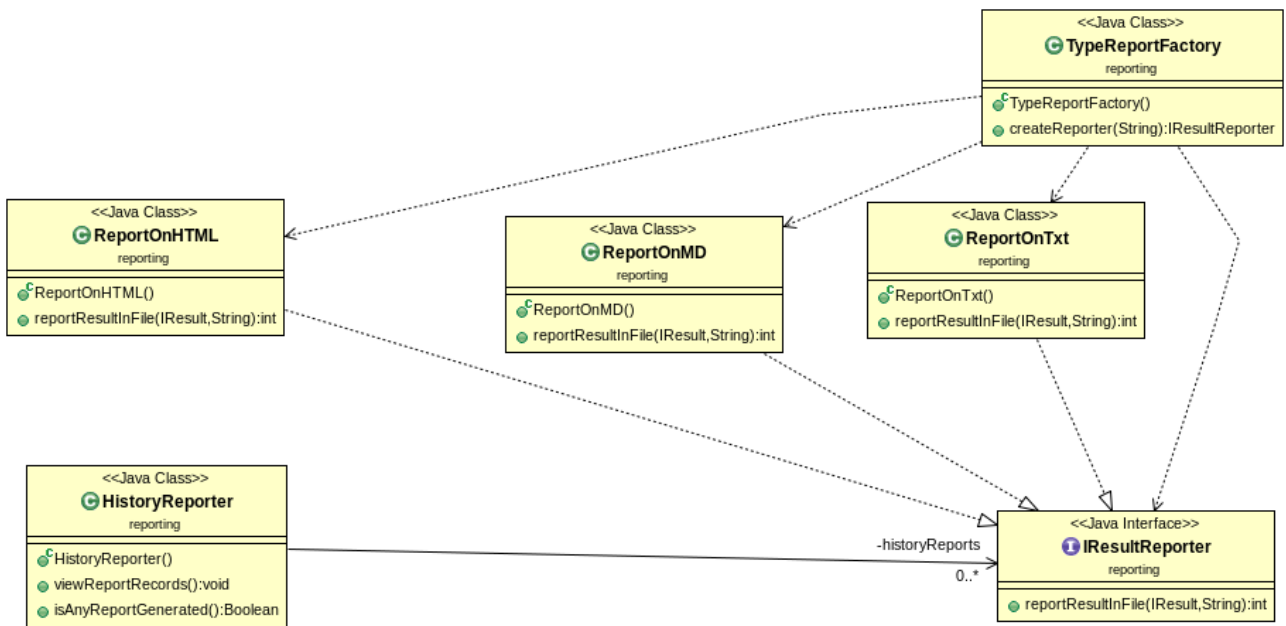
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο datamodel

package mainengine;



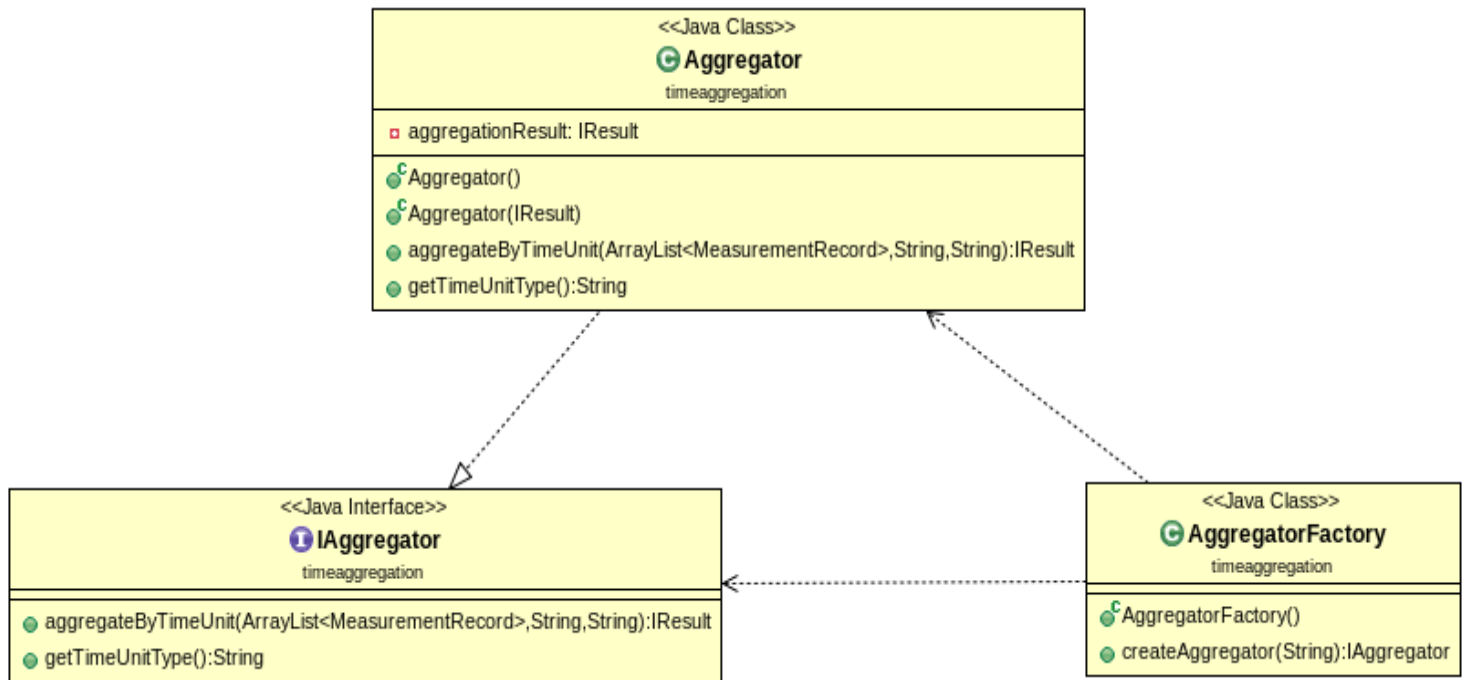
Σχήμα 4. Διάγραμμα κλάσεων επεξήγησης του πακέτου `mainengine`

package reporting;



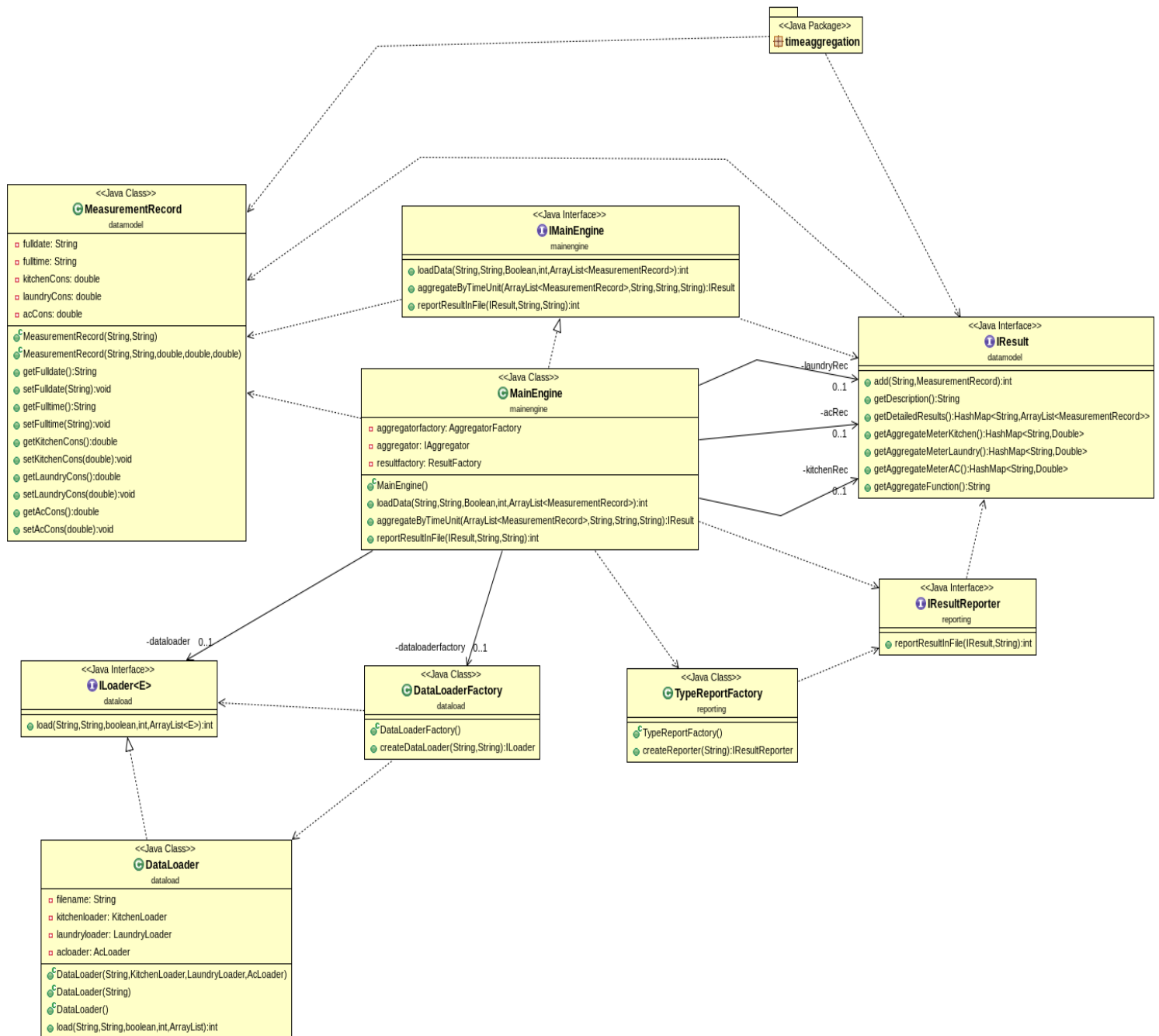
Σχήμα 5. Διάγραμμα κλάσεων επεξήγησης του πακέτου reporting

package timeaggregation;



Σχήμα 5. Διάγραμμα κλάσεων επεξήγησης του πακέτου `timeaggregation`

Παρακάτω παρατίθεται η σχεδίαση των κλάσεων και interfaces όλων των πακέτων του συστήματος :



Σχήμα 6. Διάγραμμα κλάσεων επεξήγησης βασικών κλάσεων και interfaces όλων των πακέτων του συστήματος (εκτός του client).