
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ
ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2018-2019

ΟΜΑΔΑ 4026-4042-4268

ΔΗΜΗΤΡΑ ΔΕΣΠΟΙΝΑ ΑΝΤΩΝΙΟΥ, 4026

ΑΓΓΕΛΙΚΗ ΓΚΑΒΑΡΔΙΝΑ, 4042

ΙΩΑΝΝΑ ΚΟΚΚΑΛΗ, 4268

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2018

ΙΣΤΟΡΙΚΟ ΕΚΔΟΣΕΩΝ ΤΗΣ ΠΑΡΟΥΣΑΣ ΑΝΑΦΟΡΑΣ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφείς
2018/11/04	v.01	Οργάνωση απαιτήσεων σε use cases	4026,4042,4268
2018/11/25	v.02	Αρχική σχεδίαση κλάσεων και ελέγχων	4026,4042,4268
2018/12/15	v.03	Διορθώσεις στις uses cases, επεκτάσεις στη σχεδίαση κλάσεων και ελέγχων	4026,4042,4268
2018/12/15	v.03	ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ	4026,4042,4268

1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ – USE CASES

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

QUITAPP

ID: UC 1

DESCRIPTION AND GOAL

Αυτό το use case κλείνει όλα τα επιμέρους παράθυρα της εφαρμογής.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να επιλέξει την επιλογή έξοδος ο χρήστης.

BASIC FLOW

1. Το σύστημα εμφανίζει παράθυρο με την επιλογή έξοδος.
2. Ο χρήστης πατάει έξοδο.
3. Το σύστημα κλείνει τα επιμέρους παράθυρα.

EXTENSIONS / VARIATIONS

1. Αν ο χρήστης δεν επιλέξει την έξοδο τα παράθυρα της εφαρμογής μένουν ανοιχτά.

POST CONDITIONS

Κλείνει η εφαρμογή.

SAVEFILE

ID: UC 2

DESCRIPTION AND GOAL

Το use case «SaveFile» φορτώνει στο σύστημα ένα αρχείο, το καταγράφει ως σύνολο δεδομένων και ονομάζει το σύνολο αυτό.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να υπάρχουν διαθέσιμα αποθηκευμένα αρχεία στον δίσκο.

BASIC FLOW

1. Το use case ξεκινάει όταν ο χρήστης επιλέξει από το file chooser το αρχείο που επιθυμεί.
2. Το σύστημα καταγράφει τα δεδομένα του αρχείου ως ένα σύνολο δεδομένων έχοντας μία λίστα από εγγραφές, όλες με τον ίδιο αριθμό πεδίων, ενώ, η πρώτη γραμμή του αρχείου είναι μια λίστα με τα ονόματα των πεδίων.
3. Το σύστημα ανοίγει ένα παράθυρο και ζητά από το χρήστη να ονομάσει αυτό το σύνολο δεδομένων.

EXTENSIONS / VARIATIONS

1. Στην περίπτωση κατά την οποία δεν υπάρχουν δεδομένα στο αρχείο δεν εμφανίζεται τίποτα.
2. Σε περίπτωση που ο χρήστης δώσει λάθος όνομα, του εμφανίζεται διαγνωστικό παράθυρο με το σχετικό μήνυμα.

POST CONDITIONS

Υπάρχει αποθηκευμένο στο σύστημα το σύνολο δεδομένων.

LOADFILE

ID: UC 3

DESCRIPTION AND GOAL

To use case «LoadFile» φορτώνει στο σύστημα ένα σύνολο δεδομένων.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να υπάρχουν διαθέσιμα σύνολα δεδομένων στο σύστημα.

BASIC FLOW

1. Το σύστημα ρωτάει το χρήστη ποιο σύνολο δεδομένων τον ενδιαφέρει.
2. Ο χρήστης δίνει το όνομα του συνόλου.
3. Το σύστημα εμφανίζει το σύνολο αυτό.

EXTENSIONS / VARIATIONS

1. Στην περίπτωση κατά την οποία δεν υπάρχουν δεδομένα στο σύνολο δεδομένων δεν εμφανίζεται τίποτα.
2. Σε περίπτωση που ο χρήστης δώσει λάθος όνομα, του εμφανίζεται διαγνωστικό παράθυρο με το σχετικό μήνυμα.

POST CONDITIONS

--

FILTERDATASET

ID: UC 4

DESCRIPTION AND GOAL

Γίνεται αναζήτηση με βάση κάποιο φίλτρο και αποθηκεύεται το αποτέλεσμα της αναζήτησης ως ένα νέο σύνολο δεδομένων.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να υπάρχει σύνολο δεδομένων.

BASIC FLOW

1. Το σύστημα ζητάει από τον χρήστη φίλτρο αναζήτησης.
2. Ο χρήστης τοποθετεί φίλτρο αναζήτησης.
3. Το σύστημα φτιάχνει ένα νέο σύνολο με βάση το πηγαίο σύνολο και το φίλτρο του χρήστη.
4. Το σύστημα ζητάει όνομα για το νέο σύνολο.
5. Ο χρήστης τοποθετεί όνομα.
6. Το σύστημα αποθηκεύει το νέο σύνολο.

EXTENSIONS / VARIATIONS

1. Σε περίπτωση που ο χρήστης δώσει λάθος όνομα, του εμφανίζεται διαγνωστικό παράθυρο με το σχετικό μήνυμα.

POST CONDITIONS

Το αποτέλεσμα του φιλτραρίσματος καταγράφεται ως νέο σύνολο δεδομένων.

PRINT HISTORY

ID: UC 5

DESCRIPTION AND GOAL

Υπάρχουν δύο σύνολα τα πηγαία και τα παραγόμενα. Το use case στα πηγαία κρατά την πληροφορία του αρχείου από το οποίο προήλθαν και στα παραγόμενα το σύνολο δεδομένων από όπου προήλθαν και το φίλτρο που επεβλήθη. Το use case αυτό τυπώνει από που προήλθε ένα σύνολο δεδομένων.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει να υπάρχει σύνολο δεδομένων.

BASIC FLOW

1. Αν το σύνολο είναι πηγαίο το σύστημα κρατά το όνομα αρχείου από το οποίο προήλθε ενώ αν το σύνολο είναι παραγόμενο το σύστημα κρατά το όνομα αρχείου το οποίο προήλθε και το φίλτρο του.
2. Το σύστημα εμφανίζει από που προήλθε το σύνολο δεδομένων.

POST CONDITIONS

Αποθηκεύονται οι πληροφορίες για την ιστορία παραγωγής του κάθε συνόλου στο σύστημα.

PLOTGRAPH

ID: UC 6

DESCRIPTION AND GOAL

Ορίζεται ποιο πεδίο θα είναι το X και ποιο το Y και μετά εμφανίζει τη γραφική αναπαράσταση από το front-end.

ACTORS (ESP. PRIMARY ACTOR)

Ο χρήστης.

PRECONDITIONS

Πρέπει ο άξονας X να είναι μόνο ένα μέγεθος με ταξινόμηση και ισομορφισμό στους αριθμούς.

BASIC FLOW

1. Ζητείται από το χρήστη να δώσει το όνομα του συνόλου δεδομένο για το οποίο ενδιαφέρεται.
2. Ζητείται από το χρήστη να ορίσει τα X και Y και αν θέλει να δώσει τίτλους στα πεδία αυτά για ωραιότερη εμφάνιση στη γραφική παράσταση.
3. Προβάλλεται η γραφική παράσταση.

EXTENSIONS / VARIATIONS

1. Σε περίπτωση που ο χρήστης δώσει λάθος όνομα, του επιδεικνύεται διαγνωστικό παράθυρο με το σχετικό μήνυμα.

POST CONDITIONS

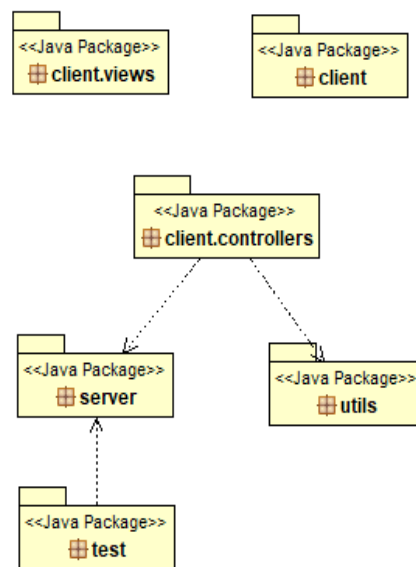
-

2 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 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

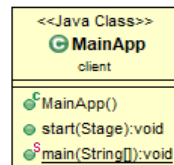
client.views	Περιέχει ένα fxmI και καμία κλάση
client	Περιέχει την κλάση που διαχειρίζεται την εφαρμογή
client.controllers	Είναι το front-end του συστήματος
server	Είναι το back-end του συστήματος
utils	Περιέχει την κλάση που διαβάζει CSV αρχεία
test	Περιέχει τα tests

Πίνακας 1. Συνοπτική περιγραφή πακέτων συστήματος

3 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

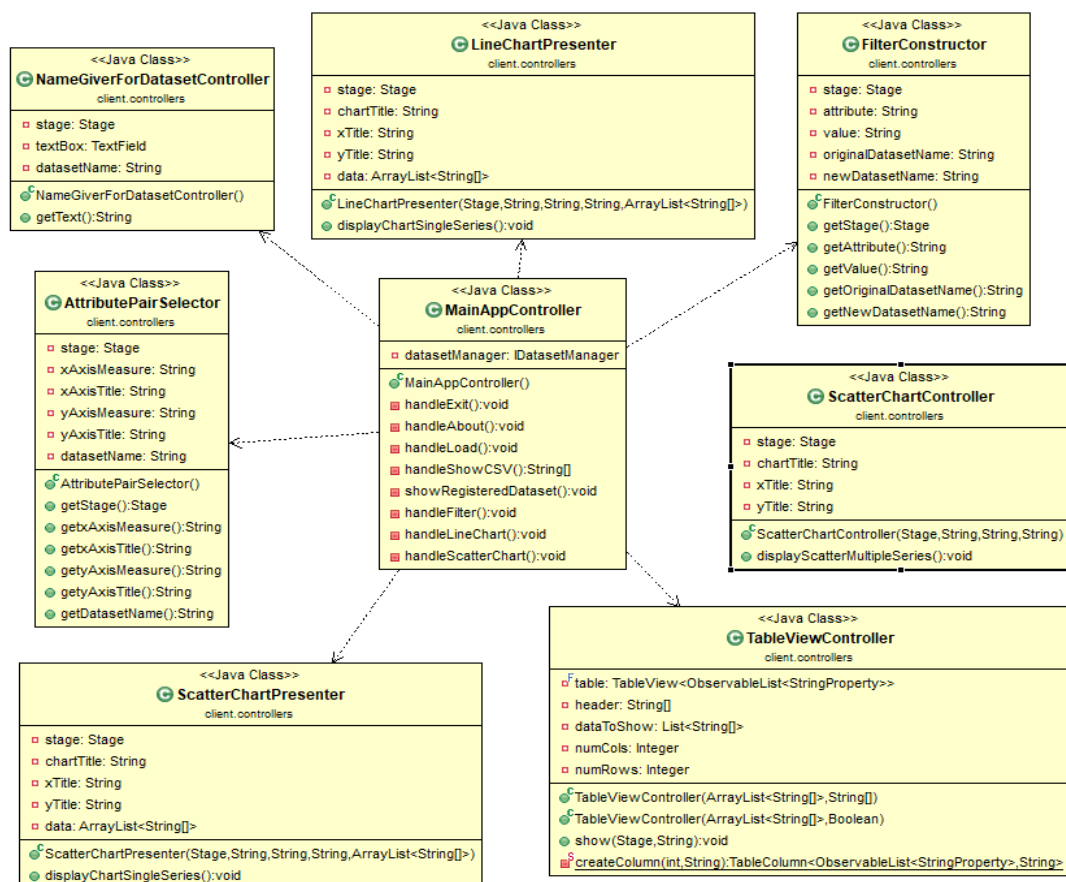
Στην παρούσα υποενότητα, παρατίθενται τα διαγράμματα κλάσεων και ακολουθιών.

package client;



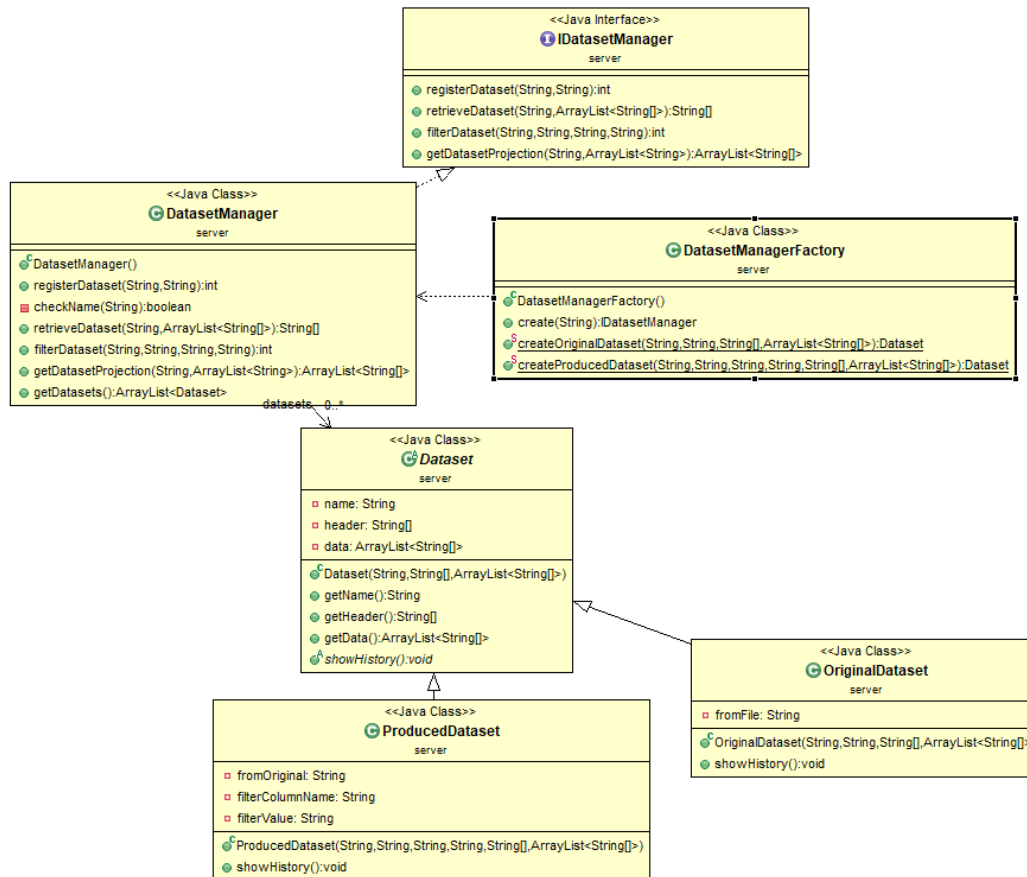
Σχήμα 2. Διάγραμμα κλάσεων για το πακέτο client

package cleint.controllers;



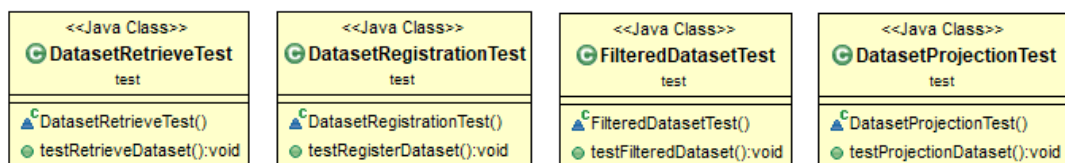
Σχήμα 3. Διάγραμμα κλάσεων για το πακέτο cleint.controllers

package server;



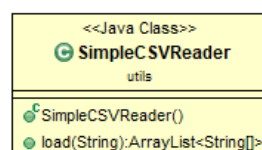
Σχήμα 4. Διάγραμμα κλάσεων για το πακέτο server

package test;



Σχήμα 5. Διάγραμμα κλάσεων για το πακέτο test

package utils;



Σχήμα 5. Διάγραμμα κλάσεων για το πακέτο utils

4 ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ ΚΑΙ ΣΥΝΕΠΕΙΑ ΠΡΟΣ ΤΙΣ ΑΠΑΙΤΗΣΕΙΣ

Στην παρούσα ενότητα παραθέτουμε μια ανάλυση των κλάσεων και μια τεκμηρίωση της κάλυψης των βασικών use cases του συστήματος.

4.1.1 DOMAIN CLASSES

Package server	Dataset, OriginalDataset, ProducedDataset
----------------	---

4.1.2 BUSINESS LOGIC CLASSES

Package server	DatasetManager, DatasetManagerFactory, IDatasetManager
----------------	--

4.1.3 BOUNDARY CLASSES

Package client.controllers	Εδώ βρίσκεται το front-end του συστήματος από το οποίο οι κλάσεις χρησιμοποιούνται για τη διεπαφή με το χρήστη
----------------------------	--

5 ΑΠΕΙΚΟΝΙΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΜΕΘΟΔΟΥΣ

Υπάρχουν 5 uses cases .

ΑΠΕΙΚΟΝΙΣΗ USE CASES ΣΕ ΜΕΘΟΔΟΥΣ

Use case	Back-end methods	Front-end methods
QuitApp	--	MainAppController.handleExit()
SaveFile	DatasetManager.registerDataset()	MainAppController.handleLoad()
LoadFile	DatasetManager.retrieveDataset()	MainAppController.showRegisteredDataset()
FilterData	DatasetManager.filterDataset()	MainAppController.handleFilter()
PrintHistory	OriginalDataset.showHistory() ProducsdDataset.showHistory()	--
PlotGraph	DatasetManager.getDatasetProjection()	MainAppController.handleLineChart(), MainAppController.handleScatterChart()

Πίνακας 2 Επαλήθευση απεικόνισης use cases σε μεθόδους

6 ΣΧΕΔΙΑΣΗ ΕΛΕΓΧΩΝ

Οι έλεγχοι που σχεδιάσθηκαν και εντάχθηκαν στην υλοποίηση περιγράφονται παρακάτω.

6.1 ΕΛΕΓΧΟΣ USE CASES

Ο Πίνακας 3 είναι ο traceability matrix για τους ελέγχους μας. Στη συνέχεια, οι έλεγχοι επεξηγούνται πιο αναλυτικά.

	UC1	UC2	UC3	UC4	UC5	UC6
DatasetProjectionTest						X
DatasetRegistrationTest		X				
DatasetRetrieveTest			X			
FilteredDataSetTest				X		

Πίνακας 3 Traceability matrix between use cases and tests

Use case UC2: Save File

Fixture: new a DatasetManager, new a DatasetManagerFactory, new an ArrayList<Dataset>
 Description: Δημιουργώ ένα νέο Dataset.
 Method to test: DatasetManager.registerDataset()
 Assertion: Οι τιμές του header(1ης) και της 2ης στήλης του πίνακα

Use case UC3: Load File

Fixture: new a DatasetManager, new a DatasetManagerFactory, new an ArrayList<Dataset>
 Description: Ανοίγω ένα υπάρχων Dataset.
 Method to test: DatasetManager.retrieveDataset()
 Assertion: Οι τιμές του header(1ης) και της 2ης στήλης του πίνακα

Use case UC4: Filter Data

Fixture: new a DatasetManager, new a DatasetManagerFactory, new an ArrayList<Dataset> and an ArrayList<String[]>
 Description: Δημιουργώ ένα παραγόμενο Dataset με φιλτράρισμα.
 Method to test: DatasetManager.filterDataset()
 Assertion: Η στήλη του φίλτρου, όλων των στοιχείων του πίνακα που επιστρέφεται.

Use case UC6: Plot Graph

Fixture: new a DatasetManager, new a DatasetManagerFactory, new an ArrayList<String[]>

Description: Ζητώ έναν πίνακα με τα στοιχεία που χρειάζονται για τη γραφική αναπαράσταση.
Method to test: `DatasetManager.getDatasetProjection()`
Assertion: Οι τιμές των πρώτων 3 στηλών του πίνακα.

6.2 UNIT TESTS

Στην παρούσα υποενότητα, παραθέτουμε πολύ συνοπτικά τα πιο σημαντικά unit test που σχεδιάστηκαν, οργανωμένα ανά κλάση.

ΚΛΑΣΗ `DatasetProjectionTest`

Αρχικά: κατασκευή ενός `DatasetManager`.

Κλήση της μεθόδου `registerDataset()` με συγκεκριμένο αρχείο στο δίσκο.

Έλεγχος τιμής – assert ότι η συνάρτηση `getDatasetProjection()` επιστρέφει σωστά τις 3 πρώτες εγγραφές του πίνακα.

ΚΛΑΣΗ `DatasetRegistrationTest`

Αρχικά: κατασκευή ενός `DatasetManager`.

Κλήση της μεθόδου `registerDataset()` με συγκεκριμένο αρχείο στο δίσκο.

Έλεγχος τιμής – assert ότι η συνάρτηση `registerDataset()` αποθηκεύει σωστά τις τιμές των header καθώς και τις τιμές της πρώτης γραμμής.

ΚΛΑΣΗ `DatasetRetrieveTest`

Αρχικά: κατασκευή ενός `DatasetManager`.

Κλήση της μεθόδου `registerDataset()` με συγκεκριμένο αρχείο στο δίσκο.

Έλεγχος τιμής – assert ότι η συνάρτηση `retrieveDataset()` επιστρέφει σωστά τις τιμές των header καθώς και τις τιμές της πρώτης γραμμής.

ΚΛΑΣΗ `FilteredDatasetTest`

Αρχικά: κατασκευή ενός `DatasetManager`.

Κλήση της μεθόδου `registerDataset()` με συγκεκριμένο αρχείο στο δίσκο.

Ζητείται συγκεκριμένο φίλτρο.

Έλεγχος τιμής – assert ότι η συνάρτηση `filterDataset()` επιστρέφει ως τιμή στη στήλη που χρησιμοποιήθηκε στο φίλτρο, την τιμή που ζητείται.