

Εργασία Εξαμήνου «Ανάλυση Πληροφοριακών Συστημάτων»

2023-2024

Ονοματεπώνυμο: Γεώργιος Δάλλας

AEM: 4116

Εργαλεία που χρησιμοποιήθηκαν:

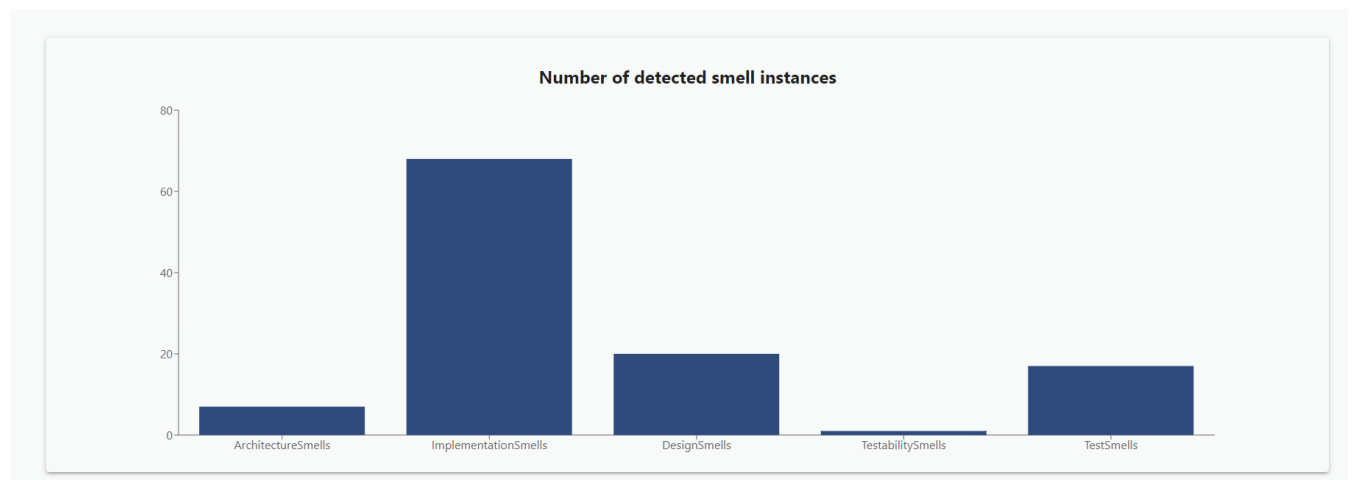
- DesigniteJava (<https://github.com/tushartushar/DesigniteJava>)
Εργαλείο για την εύρεση κακοσμιών του κώδικα και την εξαγωγή τους σε αρχεία csv.
- DJ-viz (<https://github.com/DesigniteTools/DJ-viz>)
Εργαλείο για την οπτικοποίηση των csv αρχείων με τις αναφορές σε κακοσμίες κώδικα του DesigniteJava και την παροχή χρήσιμων γράφων για αυτές.

Έργο που επιλέχθηκε:

- JsonList(<https://github.com/SlaVcE14/JsonList>)

Εφαρμογή για κινητά Android που παρέχει ένα φιλικό προς τον χρήστη περιβάλλον για προβολή αρχείων json.

Οπτικοποίηση των αποτελεσμάτων έπεται από την χρήση του DJ-viz:



Smells

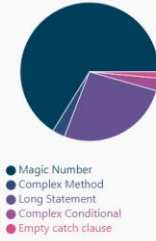
ARCHITECTURESMELLS

IMPLEMENTATIONSMELLS

DESIGNSMELLS

TESTABILITYSMELL:

TESTSMELLS



Project Name	Package Name	Type Name	Method Name
JsonList-master	com.sjapps.jsonlist	FileSystem	LoadDataFromFile
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	onCreate
JsonList-master	com.sjapps.jsonlist	MainActivity	LoadData
JsonList-master	com.sjapps.jsonlist	MainActivity	open
JsonList-master	com.sjapps.jsonlist	MainActivity	main

Smells

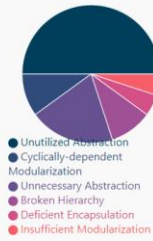
ARCHITECTURESMELL

IMPLEMENTATIONSMELLS

DESIGNSMELLS

TESTABILITYSMELLS

TESTSMELLS



Project Name	Package Name	Type Name	Design Smell	Category
JsonList-master	com.sjapps.jsonlist	App	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.jsonlist	MainActivity	Cyclically-dependent Modularization	Threat
JsonList-master	com.sjapps.about	AboutActivity	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.about	AboutListAdapter ViewHolder	Unnecessary Abstraction	Threat
JsonList-master	com.sjapps.about	AboutListAdapter ViewHolder	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.about	AboutListitem	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.about	ImageItemClick	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.about	LibraryList	Broken Hierarchy	Threat
JsonList-master	com.sjapps.about	ListGenerator	Broken Hierarchy	Threat
JsonList-master	com.sjapps.adapters	ListAdapter	Deficient Encapsulation	Threat
JsonList-master	com.sjapps.adapters	ListAdapter	Cyclically-dependent Modularization	Threat
JsonList-master	com.sjapps.adapters	ListAdapter ViewHolderShort	Unnecessary Abstraction	Threat
JsonList-master	com.sjapps.adapters	ListAdapter ViewHolderShort	Unutilized Abstraction	Threat
JsonList-master	com.sjapps.adapters	ListAdapter ViewHolderShort	Unnecessary Abstraction	Threat



Implementation Smells:

1. Magic Numbers (45 εμφανίσεις)

Ένας από τους πιο συχνούς τύπους "implementation smells" που εντοπίστηκαν στο έργο είναι οι "Magic Numbers". Αυτοί είναι αριθμοί που χρησιμοποιούνται στον κώδικα χωρίς κάποιο προφανές νόημα ή εξήγηση, καθιστώντας τον κώδικα δυσκολότερο στην κατανόηση και συντήρηση. Ακολουθούν τα σημαντικότερα παραδείγματα:

- **MainActivity - onCreate:** Πολλαπλά "magic numbers" όπως `150`, `.8f`, `.5f`

- **JsonData - getPathFormat:** Οι αριθμοί `3` εμφανίζονται πολλαπλές φορές στη μέθοδο, γεγονός που υποδηλώνει την ανάγκη για σταθερά ονόματα για να διευκρινιστεί η χρήση τους.

- **GetListFromPathTest** -

testGetListFromPathWithNestedStructureArrayOfObjects: Οι

αριθμοί `4`, `2`, `3` εμφανίζονται αρκετές φορές, γεγονός που δυσκολεύει την κατανόηση της λογικής της δοκιμής.

2. Long Statements (18 εμφανίσεις)

Υπάρχουν πολλές περιπτώσεις μακροσκελών δηλώσεων που κάνουν τον κώδικα δύσκολο να διαβαστεί και να συντηρηθεί.

- **MainActivity - onCreate:** Υπάρχουν πολλές δηλώσεις που υπερβαίνουν τους `120` χαρακτήρες, γεγονός που υποδηλώνει την ανάγκη για διάσπαση σε μικρότερες, πιο κατανοητές δηλώσεις.

- **ListItem - toString:** Η μέθοδος περιέχει μία δήλωση με μήκος `442` χαρακτήρες, η οποία είναι ιδιαίτερα δύσκολη να αναγνωσθεί και να συντηρηθεί.

- **LogActivity - share:** Η δήλωση έχει μήκος `210` χαρακτήρες, κάτι που την καθιστά πολύπλοκη και δύσκολη στην κατανόηση.

3. Complex Method (2 εμφανίσεις)

Μία άλλη σημαντική κατηγορία "implementation smells" είναι οι πολύπλοκες μέθοδοι με υψηλή κυκλωματική πολυπλοκότητα.

- **MainActivity - onCreate:** Η κυκλωματική πολυπλοκότητα της μεθόδου είναι `18`, γεγονός που υποδηλώνει ότι η μέθοδος περιέχει πολλούς κλάδους ή καταστάσεις και μπορεί να είναι δύσκολη στην κατανόηση και συντήρηση. Ενδείκνυται να γίνει ανασχεδιασμός και να σπάσει η μέθοδος σε μικρότερες, πιο κατανοητές μεθόδους.

- **JsonFunctions - getListFromPath:** Η κυκλωματική πολυπλοκότητα της μεθόδου είναι `8`, που σημαίνει ότι περιλαμβάνει πολλαπλές συνθήκες και λογική, κάτι που θα μπορούσε να απλοποιηθεί με τη διάσπαση της μεθόδου σε μικρότερες.

4. Complex Conditional (2 εμφανήσεις)

- **JsonData - getName** και **JsonFunctions - getListFromPath**:

Περιλαμβάνουν πολύπλοκες συνθήκες που είναι δύσκολο να κατανοηθούν και συντηρηθούν. Προτείνεται η αναδιάρθρωση αυτών των συνθηκών σε πιο κατανοητές μορφές ή η χρήση βοηθητικών μεθόδων.

DesignSmells

1. Unutilized Abstraction (10 εμφανίσεις)

Οι παρακάτω κλάσεις ενδέχεται να είναι αχρησιμοποίητες:

- **jsonlist/App**
- **about/AboutActivity**
- **about/AboutListAdapter-ViewHolder**
- **about/AboutListItem**
- **/about/ImageItemClick**
- **/adapters/ListAdapter-ViewHolderShort**
- **/adapters/ListAdapter-ViewHolderLong**
- **/adapters/ListAdapter-ViewHolderSpace**
- **/logs/CustomExceptionHandler**
- **/logs/LogActivity**
-

2. Cyclically-dependent Modularization (2 εμφανίσεις)

-**MainActivity**: Η κλάση συμμετέχει σε κυκλική εξάρτηση με την κλάση **ListAdapter**.

3. Unnecessary Abstraction (4 εμφανίσεις)

Οι παρακάτω κλάσεις περιέχουν μόνο λίγα μέλη δεδομένων χωρίς υλοποίηση μεθόδων

- **/about/AboutListAdapter-ViewHolder**
- **adapters/ListAdapter-ViewHolderShort**
- **/adapters/ListAdapter-ViewHolderLong**
- **/adapters/ListAdapter-ViewHolderSpace**

4. Broken Hierarchy (2 εμφανίσεις)

- **JsonList-master/about/LibraryList**: Η κλάση δεν υλοποιεί ή δεν παρακάμπτει καμία μέθοδο από τις υπερκλάσεις του (ListGenerator).
- **JsonList-master/about/ListGenerator**: Η κλάση δεν υλοποιεί ή δεν παρακάμπτει καμία μέθοδο από τις υπερκλάσεις του (ListInit).

5. Deficient Encapsulation (1 εμφάνιση)

-adapters/ListAdapter: Η κλάση εκθέτει πεδία με δημόσια προσβασιμότητα. Τα πεδία περιλαμβάνουν: list, context, activity, path, selectedItem, highlightedItem, title, value.

6. Insufficient Modularization (1 εμφάνιση)

-java/ListItem: Η κλάση έχει διογκωμένο interface με μεγάλο αριθμό δημόσιων μεθόδων (21 δημόσιες μέθοδοι).

Test Smells

1.Assertion Roulette (17 εμφανίσεις)

Οι μέθοδοι τεστ περιέχουν πολλές assertions, γεγονός που μπορεί να οδηγήσει σε δυσκολία εντοπισμού αποτυχιών και συντήρησης των τεστ.

Test Methods με Assertion Roulette:

GetArrayListTest/testGetArrayList

- Αριθμός Assertions: 9

GetArrayListTest/testGetArrayListEmpty

- Αριθμός Assertions: 2

GetJsonArrayRootTest/testGetJsonArrayRootWithEmptyArray

- Αριθμός Assertions: 5

GetJsonObjectTest/testGetJsonObjectWithNestedObject

- Αριθμός Assertions: 7

- JsonObjectTest/testJsonObjectWithNestedArray
 - Αριθμός Assertions: 6
- JsonObjectTest/testJsonObjectFromString
 - Αριθμός Assertions: 6
- JsonObjectTest/getJsonObjectFromString
 - Αριθμός Assertions: 7
- JsonObjectTest/getJsonObjectFromStringObject
 - Αριθμός Assertions: 8
- JsonObjectTest/getJsonObjectFromStringArray
 - Αριθμός Assertions: 7
- GetListFromPathTest/testGetListFromPathWithNestedStructureObject
 - Αριθμός Assertions: 9
- GetListFromPathTest/testGetListFromPathWithNestedStructureArrayOfObjects
 - Αριθμός Assertions: 12
- GetListFromPathTest/testGetListFromPathGoBack
 - Αριθμός Assertions: 6
- IsArrayOfObjectsTest/testIsArrayOfObjects_true4
 - Αριθμός Assertions: 2
- IsArrayOfObjectsTest/testIsArrayOfObjects_false3
 - Αριθμός Assertions: 2
- JsonFunctionsTest/testGetJsonArrayRoot
 - Αριθμός Assertions: 5
- JsonFunctionsTest/testGetJsonArray
 - Αριθμός Assertions: 2
- JsonFunctionsTest/testGetJsonObject
 - Αριθμός Assertions: 2

Συμπεράσματα-Προτάσεις

Τα "implementation smells" δείχνουν διάφορα σημεία βελτίωσης του κώδικα. Οι μαγικοί αριθμοί θα πρέπει να αντικατασταθούν με σταθερές για καλύτερη αναγνωσιμότητα, ενώ οι πολύπλοκες και μακροσκελείς δηλώσεις πρέπει να αναδιαμορφωθούν σε πιο κατανοητές και εύχρηστες μορφές. Με την αντιμετώπιση αυτών των ζητημάτων, η ποιότητα του

κώδικα μπορεί να βελτιωθεί σημαντικά, διευκολύνοντας τη συντήρηση και την ανάπτυξη νέων λειτουργιών.

Η ανάλυση των design smells δείχνει ότι υπάρχουν κλάσεις που δεν χρησιμοποιούνται (unutilized abstraction) και κυκλικές εξαρτήσεις (cyclically-dependent modularization) που μπορούν να προκαλέσουν σύγχυση και δυσκολία στη συντήρηση. Υπάρχουν κλάσεις με περιττή αφαίρεση (unnecessary abstraction) και ελλιπή ενθυλάκωση (deficient encapsulation), που δείχνουν την ανάγκη για καλύτερη δομή και προστασία του κώδικα. Συνιστάται η αφαίρεση αχρησιμοποίητων κλάσεων, η αποφυγή κυκλικών εξαρτήσεων, η βελτίωση της ενθυλάκωσης και η αναδιάρθρωση των κλάσεων για καλύτερη διαχείριση των ευθυνών της καθεμίας.

Για την αντιμετώπιση των "test smells", ιδιαίτερα της τεχνικής "Assertion roulette", είναι απαραίτητο να ακολουθηθούν βέλτιστες πρακτικές σχεδίασης τεστ που ενισχύουν τη σαφήνεια και την αποτελεσματικότητα. Ο διαχωρισμός των test methods που περιέχουν πολλές assertions σε μικρότερες, πιο εστιασμένα test, τα οποία ελέγχουν μία μόνο λογική ενότητα ή συμπεριφορά, αποτελεί κρίσιμο βήμα. Η χρήση descriptive assertions διευκολύνει την κατανόηση των αποτυχιών, ενώ η εφαρμογή helper methods για την οργάνωση κοινών assertions συνεισφέρει στην καθαρότητα των τεστ. Έτσι η αναγνωσιμότητα και συντήρηση των τεστ θα επιτευχθεί καλύτερα, διευκολύνοντας την ανίχνευση και επίλυση προβλημάτων.