

Αντικειμενοστρεφής Προγραμματισμός

Εργασία Ακαδημαϊκού Έτους 2022-2023 – Χειμερινό Εξάμηνο

1 Εισαγωγή

Στο πλαίσιο της εργασίας καλείστε να σχεδιάσετε και να υλοποιήσετε σε Java μια πλατφόρμα αξιολόγησης καταλυμάτων. Μέσα από την πλατφόρμα οι πάροχοι καταλυμάτων, όπως ξενοδοχοί και ιδιώτες, μπορούν να καταχωρούν τις επιχειρήσεις τους και τα καταλύματα που προσφέρουν. Οι χρήστες της πλατφόρμας μπορούν να τα αναζητούν και να τα αξιολογούν. Στόχος της εργασίας είναι να εξοικειωθείτε με τις διαδικασίες σχεδιασμού, υλοποίησης και ελέγχου αντικειμενοστρεφούς λογισμικού χρησιμοποιώντας τεχνικές απόκρυψης πληροφορίας, πολυμορφισμού και επαναχρησιμοποίησης κώδικα.

2 Γενικές Προδιαγραφές του Λογισμικού

Η (desktop) εφαρμογή που θα αναπτύξετε θα πρέπει να υποστηρίζει 2 γενικούς τύπους χρηστών: παρόχους καταλυμάτων και απλούς χρήστες. Οι πάροχοι (ξενοδοχοί, ιδιώτες) θα μπορούν να καταχωρούν καταλύματα, για παράδειγμα ξενοδοχεία ή μεμονωμένες κατοικίες (μεζονέτες κτλ.) μαζί με τις περιγραφές τους. Οι απλοί χρήστες θα μπορούν να αναζητούν τα καταχωρημένα καταλύματα με βάση διάφορα κριτήρια και να τα αξιολογούν γράφοντας σχόλια. Θα χρειαστεί να υλοποιήσετε κάποιον απλό μηχανισμό σύνδεσης (login/authentication) στην εφαρμογή ώστε να εμφανίζονται οι κατάλληλες επιλογές/καρτέλες ανάλογα με τον ρόλο του χρήστη (authorization). Η χρήση αρχείων είναι απαραίτητη για την αποθήκευση των ρόλων και των credentials των χρηστών, των πληροφοριών των καταλυμάτων, των αξιολογήσεων και γενικά όλων των δεδομένων και αλλαγών που γίνονται κατά τη διάρκεια χρήσης του προγράμματος, για να είναι διαθέσιμα και μετά τον τερματισμό του. Το πρόγραμμα θα πρέπει να αρχικοποιείται (μέσω αρχείων) με ικανοποιητικό αριθμό χρηστών και να έχει ικανοποιητικό αριθμό καταχωρημένων καταλυμάτων με τα απαραίτητα χαρακτηριστικά. Για κάθε κλάση που θα αναπτύξετε, εκτός από τις κλάσεις που αφορούν τη διασύνδεση με τον χρήστη, θα πρέπει να υλοποιήσετε και αντίστοιχη κλάση ελέγχου χρησιμοποιώντας το πλαίσιο ελέγχου JUnit.

Το λογισμικό σας θα πρέπει να είναι σχεδιασμένο βάσει των αρχών του αντικειμενοστρεφούς προγραμματισμού που παρουσιάζονται στο μάθημα, ώστε να είναι εύκολη η συντήρηση και η επέκτασή του. Για παράδειγμα στο μέλλον μπορεί να θέλουμε να προστεθούν τύποι καταλυμάτων (τροχόσπιτα, camping, κτλ.), χρηστών (μεσίτες), αξιολόγηση παρόχων και χρηστών κτλ. Οφείλετε να διαχωρίσετε την λογική της εφαρμογής από τη διασύνδεση του με τον χρήστη (user interface), έτσι ώστε αν κάποιος θελήσει να αλλάξει τη διασύνδεση, αυτό να μην απαιτεί αλλαγές στη λογική. Επομένως, θα πρέπει να αναλύσετε το λογισμικό σας σε έναν ικανό αριθμό κλάσεων, και όχι να βάλετε όλο τον κώδικα σε μια κλάση ή σε έναν μικρό αριθμό κλάσεων.

Σημαντικό: Οι κλάσεις της διασύνδεσης (JFrames, κτλ.) θα πρέπει να είναι σε διαφορετικό Java package από τις κλάσεις που υλοποιούν την λογική. Χρησιμοποιήστε για ονόματα πακέτων τα “gui” και “ari” για τις κλάσεις διασύνδεσης και λογικής, αντίστοιχα (ο σκελετός του repository ήδη περιέχει αυτά τα root packages). Η ύπαρξη κώδικα λογικής σε κλάσεις διασύνδεσης θα μετρήσει αρνητικά στην αξιολόγηση (**δείτε το Παράρτημα Βασικών Λειτουργιών για περισσότερες λεπτομέρειες**).

3 Λογιστικά

Η εργασία θα πρέπει να παραδοθεί μέχρι τις **08/01/2023** και μετράει κατά 30% στον τελικό βαθμό του μαθήματος.

Η εργασία θα πρέπει να γίνει από ομάδες των δύο ατόμων, τα οποία θα δουλέψουν συνεργατικά σε ένα ιδιωτικό αποθετήριο κώδικα στο GitHub. Και τα δύο μέλη της ομάδας πρέπει να ακολουθήσουν το παρακάτω link για να δημιουργηθεί η ομάδα και να λάβουν πρόσβαση στο ιδιωτικό αποθετήριο, στο οποίο θα ανέβει η εργασία. Όταν κάνετε accept assignment, έχετε τις εξής δυνατότητες:

- A. Να δημιουργήσετε μια νέα ομάδα και να γίνεται μέλος της.
- B. Να ενταχθείτε σε μια ήδη υπάρχουσα ομάδα.

Συστήνεται τα δύο μέλη της ομάδας να συνεννοηθούν μεταξύ τους, ώστε το ένα μέλος να δημιουργήσει την ομάδα και το δεύτερο να την επιλέξει, έτσι ώστε να αποφευχθεί η δημιουργία άσκοπων ομάδων.

Προσοχή: Το όνομα της ομάδας που θα δημιουργηθεί να είναι της μορφής: **AEM1-AEM2**, π.χ. **1234-4321**

<https://classroom.github.com/a/OWqej9Df>

(αν δεν δουλεύει το link με κλικ, κάντε copy-paste το URL στον browser)

Όσοι δεν μπορείτε να βρείτε δεύτερο άτομο, θα το δηλώσετε σε forum που υπάρχει στο elearning για αυτόν τον σκοπό, έτσι ώστε να έρθετε σε επαφή μεταξύ σας και να δημιουργήσετε ομάδα. Αν και πάλι δεν καταφέρετε να βρείτε συνεργάτη, θα στείλετε σχετικό email στον υπεύθυνο της εργασίας Γεώργιο Μεδίτσκο (gmeditsk@csd.auth.gr) και θα σας οργανώσουμε τυχαία σε ομάδες. Η διαδικασία αυτή θα είναι στη διάθεση σας ως και τις **18/11/2022**. Μετά από αυτήν την ημερομηνία δεν θα γίνονται δεκτά άλλα τέτοια αιτήματα και οι ατομικές εργασίες που θα υποβληθούν χωρίς συνεννόηση, θα απορριφθούν χωρίς να εξεταστούν. Η καταληκτική ημερομηνία αποστολής της εργασίας είναι **08/01/2023 23:59:00**. Εκπρόθεσμη υποβολή μέχρι 2 ημέρες θα έχει ως αποτέλεσμα τη μείωση του βαθμού κατά 10% την ημέρα. Θα λάβουμε υπόψη την **τελευταία εμπρόθεσμη** έκδοση του κώδικα σας στο github. Προσοχή να μην κάνετε κατά λάθος κάποιο εκπρόθεσμο push.

Για οποιαδήποτε απορία, επικοινωνήστε μέσω του αντίστοιχου forum στο elearning.

Είναι σημαντικό να ορίσετε τις παραμέτρους στο git, όπως κάνουμε και στα εργαστήρια, ώστε να φαίνεται πιο μέλος της ομάδας κάνει commit. Επειδή χρησιμοποιείτε προσωπικούς υπολογιστές, ο ορισμός του username και του email σε global επίπεδο (command prompt) δεν θα έχει πρόβλημα:

- git config --global user.name "John Doe"
- git config --global user.email johndoe@example.com

Αν αντιμετωπίζετε πρόβλημα, μπορείτε να τα ορίσετε σε local επίπεδο, όπως κάνουμε στο εργαστήριο (terminal στο IntelliJ):

- git config --local user.name "John Doe"
- git config --local user.email johndoe@example.com

4 Αξιολόγηση

Για την αξιολόγηση των εργασιών θα συνεκτιμηθεί:

- Η ποιότητα του λογισμικού (ευκολία συντήρησης και επέκτασης). Θα γίνει χρήση μετρικών της [σύζευξης, της συνοχής, της πολυπλοκότητας και του μεγέθους των κλάσεων](#), μέσα από το εργαλείο [CodeMR](#) που προσφέρεται ως plugin για το IntelliJ. Οι έννοιες αυτές θα εξηγηθούν σε προσεχή διάλεξη. Μπορείτε να εγκαταστήσετε το CodeMR για να έχετε μια εικόνα της ποιότητας του κώδικα σας καθώς αναπτύσσεται.
- Η συνεισφορά του κάθε μέλους της ομάδας βάσει των commit στο GitHub.
- Η υλοποίηση του λογισμικού (πληρότητα υλοποίησης λειτουργιών, ποιότητα υλοποίησης).
- Ο έλεγχος του λογισμικού (πληρότητα και ποιότητα κλάσεων ελέγχου).
- Η τεκμηρίωση εντός του κώδικα.
- Η επίδοση κάθε φοιτητή/φοιτήτριας στην προφορική εξέταση της εργασίας. Η συμμετοχή στην προφορική εξέταση είναι υποχρεωτική για την καταχώρηση βαθμολογίας στην εργασία.

5 Παράρτημα Βασικών Λειτουργιών

Σε αυτήν την ενότητα παρουσιάζονται λεπτομερώς βασικά χαρακτηριστικά και λειτουργίες που θα πρέπει να υποστηρίζονται από το πρόγραμμα. Ο αριθμός και η ορθότητα των λειτουργιών που θα υλοποιηθούν συνεκτιμάται στην αξιολόγηση (όπως αναφέρεται στην ενότητα Αξιολόγηση). Οποιαδήποτε επιπρόσθετη λειτουργία είναι προαιρετική και θα μετρήσει θετικά.

A. Καταχώρηση καταλύματος

[A1] Όνομα: το όνομα του ξενοδοχείου ή κάποια σύντομη περιγραφή που θα ορίζει ο πάροχος ως όνομα σε περίπτωση που είναι κάτι άλλο π.χ. «διώροφο στο Παγκράτι» ή κάτι σχετικό.

[A2] Τύπος: ξενοδοχείο, διαμέρισμα, μεζονέτα.

[A3] Τοποθεσία: διεύθυνση, πόλη, ταχυδρομικός κώδικας (ξεχωριστά πεδία).

[A4] Περιγραφή: ελεύθερο κείμενο που παρουσιάζονται κάποια στοιχεία από τον ιδιοκτήτη για το κατάλυμα/ξενοδοχείο (πότε χτίστηκε, κάποια λόγια για την τοποθεσία, κοντινά αξιοθέατα, κτλ.).

[A5] Παροχές καταλύματος: Συσχέτιση των καταλυμάτων με τους παρακάτω προκαθορισμένους τύπους παροχών:

1. **Θέα:** Θέα σε πισίνα, Θέα σε παραλία, Θέα στη θάλασσα, Θέα στο λιμάνι, Θέα στο βουνό, Θέα στον δρόμο
2. **Μπάνιο:** Πιστολάκι μαλλιών
3. **Πλύσιμο ρούχων:** Πλυντήριο ρούχων, Στεγνωτήριο
4. **Ψυχαγωγία:** Τηλεόραση
5. **Θέρμανση και κλιματισμός:** Εσωτερικό τζάκι, κλιματισμός, κεντρική θέρμανση
6. **Διαδίκτυο:** wifi, ethernet
7. **Κουζίνα και τραπεζαρία:** Κουζίνα, Ψυγείο, Φούρνος μικροκυμάτων, Μαγειρικά είδη, Πιάτα και μαχαιροπίρουνα, Πλυντήριο πιάτων, Καφετιέρα
8. **Εξωτερικός χώρος:** Μπαλκόνι, αυλή
9. **Χώρος στάθμευσης:** Δωρεάν χώρος στάθμευσης στην ιδιοκτησία, Δωρεάν πάρκινγκ στο δρόμο

Σημειώσεις:

- Αν ένα κατάλυμα δεν αναφέρει κάποια παροχή, σημαίνει ότι δεν την προσφέρει.
- Όλα τα χαρακτηριστικά είναι υποχρεωτικά (* required) εκτός από το **[A5]** για το οποίο κάποιος μπορεί να μην ορίσει τίποτα.
- Θα πρέπει να υπάρχουν κατάλληλα μηνύματα στο GUI σε περίπτωση που υπάρχει κάποιο σφάλμα στην εισαγωγή, π.χ. κάποια υποχρεωτικά πεδία δεν έχουν τιμή ή γίνεται κάποιος άλλος έλεγχος που αποτυγχάνει και δεν μπορεί να γίνει η καταχώρηση.

B. Προβολή Καταχώρησης

Για κάθε κατάλυμα θα πρέπει να υπάρχει η δυνατότητα προβολής των χαρακτηριστικών του, καθώς και να υπάρχουν διάφορες επιλογές ανάλογα με το ποιος τύπος χρήστης «βλέπει» το κατάλυμα:

- [B1]** Θα πρέπει να εμφανίζονται όλες οι πληροφορίες/χαρακτηριστικά του καταλύματος (ενότητα **A. Καταχώρηση καταλύματος**), μαζί με τον αριθμό των αξιολογήσεων που έχει δεχθεί, τον μέσο όρο των αξιολογήσεων της συγκεκριμένης καταχώρησης και να φαίνονται όλες οι αξιολογήσεις του (π.χ. σε μια λίστα), όπως κείμενο, βαθμολογία, ημερομηνία, μικρό όνομα του χρήστη που έκανε την κάθε αξιολόγηση.
- [B2]** Αν ο χρήστης είναι ο πάροχος που έχει καταχωρήσει το κατάλυμα, θα πρέπει να υπάρχουν επιλογές διαγραφής και επεξεργασίας της καταχώρησης.
- [B3]** Αν ο χρήστης δεν είναι ο πάροχος, θα πρέπει να υπάρχει επιλογή καταχώρησης αξιολόγησης.

Γ. Καταχώρηση αξιολόγησης

Αφού επιλέξει ένας χρήστης ένα κατάλυμα, θα μπορεί να το αξιολογήσει, καταχωρώντας:

- [Γ1]** Το κείμενο της αξιολόγησης (ελεύθερο κείμενο μέχρι 500 χαρακτήρες με κενά).
- [Γ2]** Βαθμολογία (1 μέχρι 5).

Σημειώσεις

- Όλα τα πεδία είναι υποχρεωτικά.
- Θα πρέπει να υπάρχουν κατάλληλα μηνύματα στο GUI σε περίπτωση που υπάρχει κάποιο σφάλμα, π.χ. κάποια υποχρεωτικά πεδία δεν έχουν τιμή.

Δ. Λειτουργίες παρόχων

- [Δ1]** Προσθήκη καταχώρησης
- [Δ2]** Επεξεργασία καταχώρησης
- [Δ3]** Διαγραφή καταχώρησης
- [Δ4]** Καρτέλα (dashboard) που να εμφανίζει
1. Τον συνολικό αριθμό των αξιολογήσεων που έχουν δεχθεί τα καταλύματά τους.
 2. Τον μέσο όρο όλων των αξιολογήσεων των καταλυμάτων τους.
 3. Όλες τις καταχωρήσεις καταλυμάτων που έχουν κάνει, εμφανίζοντας (π.χ. σε μια λίστα) το όνομα, τον τύπο, την τοποθεσία και τον μέσο όρο των αξιολογήσεων της κάθε καταχώρησης. Επιλέγοντας μια καταχώρηση, θα πρέπει να τους εμφανίζει την καρτέλα **B. Προβολή Καταχώρησης**.

Σημειώσεις

- Κατά την εισαγωγή/επεξεργασία καταλύματος θα πρέπει να υπάρχει η δυνατότητα χρήσης των χαρακτηριστικών που παρουσιάστηκαν στην ενότητα **A. Καταχώρηση καταλύματος**.

- Οι πάροχοι πρέπει να μπορούν να επεξεργαστούν / διαγράψουν μόνο καταλύματα που έχουν καταχωρήσει οι ίδιοι, καθώς επίσης δεν μπορούν να επεξεργαστούν τις αξιολογήσεις.
- Κατά την διαγραφή μιας καταχώρησης, θα πρέπει να «χάνονται» και όσες αξιολογήσεις έχουν γίνει για αυτό το κατάλυμα, δηλαδή οι χρήστες δεν θα μπορούν να δουν τις αξιολογήσεις που τυχόν έχουν κάνει για το κατάλυμα που έχει διαγραφεί.
- Θα πρέπει να υπάρχουν κατάλληλα μηνύματα στο GUI σε περίπτωση που υπάρχει κάποιο σφάλμα.

Ε. Λειτουργίες Χρηστών

[E1] Αναζήτηση καταλυμάτων με κριτήρια:

1. Όνομα
2. Τύπος
3. Τοποθεσία
4. Παροχές

Στα αποτελέσματα της αναζήτησης θα πρέπει να εμφανίζονται, π.χ. σε μια λίστα, το όνομα, ο τύπος, η τοποθεσία και ο μέσος όρος των αξιολογήσεων των καταλυμάτων που ταιριάζουν στα κριτήρια αναζήτησης. Επιλέγοντας μια καταχώρηση, θα πρέπει να τους εμφανίζει την καρτέλα **Β. Προβολή Καταχώρησης**.

[E2] Εισαγωγή αξιολόγησης για κατάλυμα.

[E3] Επεξεργασία αξιολόγησης για κατάλυμα.

[E4] Διαγραφή αξιολόγησης για κατάλυμα.

[E5] Καρτέλα (dashboard) που να εμφανίζει:

1. Όλα τα καταλύματα που έχουν αξιολογήσει, π.χ. σε μια λίστα, εμφανίζοντας το όνομα, τον τύπο, την τοποθεσία και τον βαθμό αξιολόγησης που έχουν δώσει. Επιλέγοντας μια αξιολόγηση, θα πρέπει να τους εμφανίζει την καρτέλα **Β. Προβολή Καταχώρησης**.
2. Τον μέσο όρο της βαθμολογίας που έχουν δώσει συνολικά στα καταλύματα που έχουν αξιολογήσει.

Σημειώσεις

- Ως κριτήρια αναζήτησης θα μπορούν να χρησιμοποιηθούν και πάνω από ένα, π.χ. και τύπος και τοποθεσία ταυτόχρονα, οπότε τα αποτελέσματα θα πρέπει να τα ικανοποιούν όλα τα κριτήρια (λογικό AND).
- Θα πρέπει να υπάρχουν κατάλληλα μηνύματα στο GUI σε περίπτωση που υπάρχει κάποιο σφάλμα.

ΣΤ. Εγγραφή χρηστών

Φόρμα εγγραφής παρόχων και χρηστών στην πλατφόρμα (registration). Θα πρέπει να υπάρχει επιλογή στην φόρμα για τον τύπο του χρήστη για τον οποίον γίνεται εγγραφή (πάροχος, χρήστης):

[ΣΤ1] Όνομα

[ΣΤ2] Επίθετο

[ΣΤ3] Username

[ΣΤ4] Password

[ΣΤ5] Τύπος χρήστη

Σημειώσεις

- Δεν θα πρέπει να επιτρέπεται η εγγραφή κάποιου χρήστη με username το οποίο ήδη υπάρχει.
- Όλα τα πεδία είναι υποχρεωτικά.
- Θα πρέπει να υπάρχουν κατάλληλα μηνύματα στο GUI σε περίπτωση που υπάρχει κάποιο σφάλμα, π.χ. όταν υπάρχει ήδη το username.

Z. Σύνδεση / αποσύνδεση χρηστών

- [Z1] Φόρμα σύνδεσης παρόχων και χρηστών στην πλατφόρμα (login) με username και password. **Δεν πρέπει να υπάρχει επιλογή** στην φόρμα του τύπου του χρήστη που συνδέεται. Ανάλογα με τα credentials, το πρόγραμμα θα πρέπει να καταλαβαίνει ποιος συνδέεται.
- [Z2] Επιλογή αποσύνδεσης χρήστη.

H. Αρχικοποίηση προγράμματος

- [H1] Τουλάχιστον 2 πάροχοι με credentials:
- Username: provider1, password: password1
 - Username: provider2, password: password2
- [H2] Τουλάχιστον 2 χρήστες με credentials:
- Username: user1, password: password1
 - Username: user2, password: password2
- [H3] ≥ 10 καταλύματα με πληροφορία σε όλα τα χαρακτηριστικά (οι τιμές να μην είναι ίδιες στα χαρακτηριστικά για να έχει νόημα η αναζήτηση) που θα μοιραστούν στους 2 παραπάνω παρόχους.
- [H4] ≥ 2 αξιολογήσεις σε κάθε κατάλυμα που θα μοιραστούν στους 2 παραπάνω χρήστες.

Σημειώσεις

- Η αρχικοποίηση θα πρέπει να γίνεται μέσα από αρχεία.
- Αν κατά την διάρκεια εκτέλεσης του προγράμματος γίνουν αλλαγές, π.χ. προστεθεί ένα κατάλυμα, αυτές οι αλλαγές πρέπει να αποθηκευτούν, έτσι ώστε όταν κλείσει και ξανά ανοίξει το πρόγραμμα, να είναι διαθέσιμες.

Θ. Μη Λειτουργικές Απαιτήσεις / Διευκρινήσεις

- [Θ1] Το πρόγραμμα πρέπει να υλοποιηθεί στο github repository / IntelliJ project που θα δημιουργηθεί για την ομάδα όταν γίνει accept το assignment (όπως κάνουμε και στα εργαστήρια).
- [Θ2] Είναι απαραίτητο να ελέγξετε ότι το πρόγραμμα είναι λειτουργικό (να κάνει compile και να τρέχει) μετά από ένα «καθαρό» clone. Προτείνεται να κάνετε ένα clone το τελικό repository στο IntelliJ σε έναν υπολογιστή που δεν έχει χρησιμοποιηθεί στην υλοποίηση (π.χ. σε κάποιον υπολογιστή στο PC Lab), ώστε να γίνουν οι απαραίτητοι έλεγχοι. Προγράμματα τα οποία δεν κάνουν compile δεν θα βαθμολογηθούν.
- [Θ3] Είναι απαραίτητο να φαίνεται το ιστορικό της υλοποίησης (commits) και από τους 2 προγραμματιστές. Οπότε και οι δύο πρέπει να δουλεύουν (συνεργατικά) στο ίδιο github repository [Θ1]. Συνεργατικά δεν σημαίνει ότι πρέπει και οι δύο να δουλεύουν μαζί στον ίδιο χώρο ή στον ίδιο υπολογιστή. Μπορεί να γίνει αρχικά ένας διαμοιρασμός των απαιτήσεων και της υλοποίησης, και ο καθένας / καθεμία να δουλεύει ανεξάρτητα.
- [Θ4] Πρέπει να υπάρχουν tests για όλες τις κλάσεις/μεθόδους, εκτός αυτών που ασχολούνται με την υλοποίηση του GUI. Για να έχετε μια εικόνα του ποσοστού του κώδικα που ελέγχεται μέσω tests, μπορείτε να τρέξετε τα tests από την επιλογή "Run All Tests with Coverage" (η επιλογή υπάρχει με δεξί κλικ στον φάκελο με τα tests). Θα εμφανιστούν ποσοστά για τις

κλάσεις και τις μεθόδους (δεν χρειάζεται για γραμμές κώδικα). Οι κλάσεις για GUI (πακέτο gui) πρέπει να εξαιρεθούν για μεγαλύτερη ακρίβεια (εδώ βοηθάει το ότι θα πρέπει να υπάρχουν 2 root packages για την λογική και την διασύνδεση, είναι μέρος του αρχικού σκελετού του project που δημιουργείται. Μπορείτε να δημιουργήσετε όσα πακέτα θέλετε κάτω από τα ar και gui για να οργανώσετε περαιτέρω τον κώδικά σας).

- [05]** Ο κώδικας θα πρέπει να είναι επαρκώς τεκμηριωμένος με χρήση Javadoc. Δεν χρειάζεται Javadoc σε κλάσεις που υλοποιούν το GUI.
- [06]** CodeMR: επικεντρωθείτε μόνο στις προβληματικές κλάσεις (όχι σε όλες τις μετρικές) όλου του project, εξαιρώντας βέβαια τις κλάσεις ελέγχου.
- [07]** Είναι απαραίτητο να υπάρχουν τα usernames και τα passwords για τους χρήστες όπως αυτά περιγράφονται στα **[H1]** και **[H2]**.
- [08]** Είναι απαραίτητο το πρόγραμμά σας να αρχικοποιείται με επαρκή δεδομένα απαραίτητα για τον έλεγχο όλων των λειτουργιών (**[H3]** και **[H4]**).