

Αντικειμενοστρεφής Προγραμματισμός

Εργασία Ακαδημαϊκού Έτους 2020-2021

Εισαγωγή

Στο πλαίσιο της εργασίας καλείστε να σχεδιάσετε και να υλοποιήσετε σε Java ένα παιχνίδι ερωτήσεων, βασισμένο σε ένα βραβευμένο παιχνίδι στο PlayStation, το [Buzz! Quiz World](#). Στόχος της εργασίας είναι να εξοικειωθείτε με τις διαδικασίες σχεδιασμού, υλοποίησης και ελέγχου αντικειμενοστρεφούς λογισμικού χρησιμοποιώντας τεχνικές απόκρυψης πληροφορίας, πολυμορφισμού και επαναχρησιμοποίησης κώδικα.

Προδιαγραφές του Λογισμικού

Το παιχνίδι εκτυλίσσεται σε έναν αριθμό (π.χ. 6) γύρων. Ο κάθε γύρος περιλαμβάνει έναν αριθμό (π.χ. 5) ερωτήσεων πολλαπλής επιλογής με 4 απαντήσεις, εκ των οποίων μόνο η μία είναι σωστή. Υπάρχουν δύο είδη ερωτήσεων: α) ερωτήσεις που συνοδεύονται και από εικόνα (π.χ. ποιος είναι ο εικονιζόμενος ηθοποιός;), και β) ερωτήσεις που δεν συνοδεύονται από εικόνα. Επίσης θα πρέπει να υπάρχει ένας αριθμός (π.χ. 4) από κατηγορίες ερωτήσεων (π.χ. αθλητικά, ιστορία, κτλ). Το παιχνίδι μπορεί να παίζεται είτε από έναν παίκτη, είτε ανταγωνιστικά από 2 παίκτες. Υπάρχουν διάφοροι τύποι γύρων:

- *Σωστή απάντηση.* Κάθε παίκτης που απαντάει σωστά κερδίζει 1000 πόντους.
- *Σταμάτησε το χρονόμετρο.* Υπάρχει ένα χρονόμετρο που μετράει αντίστροφα 5 δευτερόλεπτα και ο κάθε παίκτης που απαντάει σωστά κερδίζει τόσους πόντους όσους τα millisecond που απέμεναν ακόμα όταν απάντησε επί 0,2.
- *Ποντάρισμα.* Αρχικά εμφανίζεται η κατηγορία της ερώτησης. Ο παίκτης μπορεί να ποντάρει 250, 500, 750 και 1000 πόντους. Στη συνέχεια εμφανίζεται η ερώτηση και αν απαντήσει σωστά κερδίζει τους πόντους που πόνταρε, αλλιώς τους χάνει.
- *Γρήγορη απάντηση.* Ο πρώτος που απαντάει σωστά κερδίζει 1000 πόντους και ο δεύτερος 500 πόντους (αυτός ο τύπος ερώτησης έχει νόημα μόνο όταν παίζουν 2 παίκτες).
- *Θερμόμετρο.* Ο πρώτος παίκτης που θα απαντήσει σωστά 5 ερωτήσεις κερδίζει 5000 πόντους (αυτός ο τύπος ερώτησης έχει νόημα μόνο όταν παίζουν 2 παίκτες).

Το λογισμικό σας θα πρέπει να κρατάει αρχείο με τις νίκες κάθε παίκτη στο παιχνίδι 2 παιχτών, καθώς και με το υψηλότερο του σκορ στο ατομικό παιχνίδι και να παρουσιάζει αυτά τα στατιστικά όποτε ζητηθούν. Θα πρέπει επίσης να περιλαμβάνει τυχαιότητα (τυχαία σειρά εμφάνισης ερωτήσεων, αλλά και απαντήσεων ανά ερώτηση), καθώς και αποφυγή εμφάνισης της ίδιας ερώτησης στο ίδιο παιχνίδι. Οι ερωτήσεις καθώς και τα σχετικά δεδομένα τους (απαντήσεις, σωστή απάντηση, κατηγορία ερώτησης) θα πρέπει να είναι αποθηκευμένες σε αρχεία κειμένου. Το λογισμικό που θα αναπτύξετε θα πρέπει να έχει graphical user interface. Μπορείτε να χρησιμοποιήσετε τυχαία δεδομένα για ερωτήσεις και απαντήσεις ή να χρησιμοποιήσετε το υλικό από κάποιο παρόμοιο παιχνίδι (π.χ. [Το Παιχνίδι των Γνώσεων 2020](#)).

Ο κώδικας θα πρέπει να είναι τεκμηριωμένος με χρήση JavaDoc.

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.

Απαιτήσεις και Υποδείξεις Σχεδίασης

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 έχουμε και ερωτήσεις με ήχο, καθώς και ερωτήσεις με βίντεο. Μπορεί να θέλουμε να προστεθούν και άλλες κατηγορίες ερωτήσεων. Μπορεί επίσης να θέλουμε να προστεθεί η δυνατότητα να παίζουν ως 4 παίκτες. Οφείλετε να διαχωρίσετε την λογική του παιχνιδιού 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.

Λογιστικά

Η εργασία χωρίζεται σε δύο επαναληπτικά μέρη ανάπτυξης της εργασίας, διάρκειας ενός μήνα περίπου το κάθε ένα, σε αρμονία με τις αρχές της ευέλικτης ανάπτυξης λογισμικού (agile software development) που έχει υιοθετήσει η βιομηχανία ανάπτυξης λογισμικού στις μέρες μας. Στο πρώτο μέρος θα πρέπει να παραδώσετε μια πρώτη ελάχιστη έκδοση του λογισμικού σας, η οποία θα έχει command line interface, θα αφορά μόνο ατομικό παιχνίδι, θα υπάρχουν ερωτήσεις χωρίς εικόνα, θα υπάρχουν μόνο δύο τύποι γύρων (σωστή απάντηση, ποντάρισμα) και δεν θα διατηρεί αρχείο με νίκες και σκορ. Ένας περιορισμένος αριθμός από ερωτήσεις (καθώς και τα σχετικά δεδομένα τους) θα είναι αποθηκευμένες στη μνήμη (στον κώδικα του προγράμματος). Το παραδοτέο αυτό θα μετρήσει κατά 10% στον τελικό βαθμό του μαθήματος. Στο δεύτερο μέρος θα πρέπει να παραδώσετε την πλήρη έκδοση του λογισμικού σας. Το παραδοτέο αυτό θα μετρήσει κατά 20% 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. Δημιουργήστε τις ομάδες σας και λάβετε πρόσβαση στο ιδιωτικό αποθετήριο, στο οποίο θα ανεβάσετε την εργασία σας. Όσοι δεν μπορείτε να βρείτε δεύτερο άτομο, θα το δηλώσετε σε forum που υπάρχει στο e-learning 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 του εργαστηρίου Α. Λαγόπουλο (lathanag@csd.auth.gr) και θα σας οργανώσουμε τυχαία σε ομάδες. Η διαδικασία αυτή θα είναι στη διάθεση σας ως και τις 20/11. Μετά από αυτήν την ημερομηνία δεν θα γίνονται δεκτά άλλα τέτοια αιτήματα και οι ατομικές εργασίες που θα υποβληθούν, θα απορριφθούν χωρίς να εξεταστούν. Η καταληκτική ημερομηνία αποστολής κάθε παραδοτέου φαίνεται στο χρονοδιάγραμμα του μαθήματος. Εκπρόθεσμη υποβολή μέχρι 2 ημέρες θα έχει ως αποτέλεσμα τη μείωση του βαθμού κατά 10% την ημέρα. Για την υποβολή αρκεί ένα push του repository σας. Θα λάβουμε υπόψη την τελευταία εμπρόθεσμη έκδοση του κώδικα σας. Προσοχή να μην κάνετε κατά λάθος κάποιο εκπρόθεσμο push.

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

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

Προϊόντα αντιγραφής θα μηδενίζονται και όλοι οι εμπλεκόμενοι φοιτητές θα παραπέμπονται στη γενική συνέλευση του Τμήματος. Οι ποινές που προβλέπονται περιλαμβάνουν τον αποκλεισμό από μελλοντικές εξεταστικές περιόδους.