

Αντικειμενοστρεφής Προγραμματισμός

Εργασία Ακαδημαϊκού Έτους 2014-2015

Εισαγωγή

Στο πλαίσιο της εργασίας καλείστε να σχεδιάσετε και να υλοποιήσετε σε Java ένα σύστημα κράτησης θέσεων για έναν πολυχώρο κινηματογραφικών προβολών και θεατρικών παραστάσεων. 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

Ο πολυχώρος μπορεί να περιλαμβάνει δύο βασικές κατηγορίες αιθουσών: α) προβολής ταινιών, β) θεατρικών παραστάσεων. Οι αίθουσες έχουν όνομα και μια συγκεκριμένη χωρητικότητα θεατών. Περιέχουν θέσεις οργανωμένες σε σειρές. Κάθε σειρά μπορεί να έχει διαφορετικό αριθμό καθισμάτων από τις άλλες. Οι αίθουσες θεατρικών παραστάσεων επιπλέον μπορεί να έχουν έναν ή παραπάνω εξώστες, στους οποίους οι θέσεις είναι και πάλι οργανωμένες σε σειρές. Οι θέσεις μπορεί να είναι απλές ή πολυτελείας, οι οποίες περιλαμβάνουν τραπεζάκι. Οι αίθουσες θεατρικών παραστάσεων περιλαμβάνουν επίσης έναν αριθμό από καμαρίνια, ενώ οι αίθουσες προβολής ταινιών περιέχουν οθόνες, οι οποίες χαρακτηρίζονται από μέγεθος. Κάποιες οθόνες υποστηρίζουν προβολή τριών διαστάσεων. Επιπλέον οι αίθουσες προβολής ταινιών χαρακτηρίζονται και από κάποιο σύστημα ήχου (Dolby Digital, Dolby Digital Surround, Dolby Digital Live, Dolby Digital Plus, Dolby TrueHD). Στις αίθουσες παρουσιάζονται ταινίες ή θεατρικές παραστάσεις, οι οποίες έχουν όνομα, σύντομη περιγραφή, σκηνοθέτη και λίστα βασικών ηθοποιών. Οι ταινίες χαρακτηρίζονται και από χρονική διάρκεια. Οι θεατρικές παραστάσεις δίνονται σε κάποια από τις αίθουσες θεατρικών παραστάσεων συγκεκριμένες ημέρες και ώρες. Οι ταινίες μπορεί να παίζονται και σε παραπάνω από μια αίθουσα προβολής ταινιών και πάλι συγκεκριμένες ημέρες και ώρες. Τα εισιτήρια έχουν διαφορετική τιμή ανάλογα με τη θεατρική παράσταση, ενώ για τις ταινίες οι τιμές παίζουν ανάλογα με την αίθουσα και την ποιότητα των θέσεων που αυτή περιλαμβάνει. Το λογισμικό σας θα χρησιμοποιείται από δύο είδη χρηστών: α) διαχειριστής, β) απλός χρήστης. Ο διαχειριστής θα μπορεί να προσθέτει, να τροποποιεί και να διαγράφει αίθουσες, θεατρικές παραστάσεις/ταινίες και το πρόγραμμα των παραστάσεων/προβολών. Όλες αυτές οι πληροφορίες θα πρέπει να μπορούν να αποθηκεύονται στον δίσκο. Ο απλός χρήστης θα μπορεί να κάνει κρατήσεις συγκεκριμένων θέσεων για συγκεκριμένες προβολές θεατρικών παραστάσεων ή ταινιών. Θα πρέπει να μπορεί να επιλέγει έναν αριθμό συγκεκριμένων θέσεων οι οποίες να μπορούν να είναι αν θέλει διπλανές στην ίδια σειρά. Οι πληροφορίες κρατήσεων συμπεριλαμβανομένου του ονόματος και του τηλεφώνου του πελάτη θα πρέπει να μπορούν να αποθηκεύονται στον δίσκο. Θα πρέπει να αναπτυχθούν δύο γραφικές διεπαφές (μία για κάθε είδος χρήστη). Η διεπαφή του απλού χρήστη θα πρέπει να απεικονίζει γραφικά στην οθόνη την διάταξη των θέσεων της αίθουσας και το ποιες είναι διαθέσιμες, έτσι ώστε ο χρήστης να επιλέγει θέσεις με κλικ/touch επάνω τους. Πέρα από την κράτηση θέσεων, θα πρέπει να αναπτυχθούν λειτουργίες για υποστήριξη συνηθισμένων σεναρίων χρήστη, όπως ακύρωση κράτησης, αλλαγή κράτησης (προσθήκη/αφαίρεση θέσεων, αλλαγή αίθουσας, κτλ).

Απαιτήσεις και Υποδείξεις Σχεδίασης

Το λογισμικό σας θα πρέπει να είναι σχεδιασμένο βάσει των αρχών του αντικειμενοστρεφούς προγραμματισμού που παρουσιάστηκαν στο μάθημα, ώστε να είναι εύκολη η συντήρη και η επέκτασή του. Για παράδειγμα θα πρέπει να είναι εύκολη η υλοποίηση περαιτέρω παραλλαγών (άλλα είδη αιθουσών, παραστάσεων, θέσεων, κτλ), είτε από εσάς τους ίδιους είτε από άλλους προγραμματιστές. Για το σκοπό αυτό θα πρέπει να το αναλύσετε σε έναν ικανό αριθμό κλάσεων, και όχι να βάλετε όλο τον κώδικα σε μια κλάση. Μπορείτε για παράδειγμα να έχετε ξεχωριστή κλάση για κάθε είδος αίθουσας, οι οποίες θα είναι βασισμένες σε κάποια πρόγονο κλάση. Ομοίως για τις ταινίες και τις θεατρικές παραστάσεις. Οφείλετε να έχετε για κάθε είδος χρήστη ξεχωριστή κλάση με την λογική της εφαρμογής και ξεχωριστή κλάση με το graphical user interface της εφαρμογής, έτσι ώστε αν κάποιος θελήσει να αλλάξει το interface, αυτό να μην απαιτεί αλλαγές στη λογική.

Έλεγχος του Λογισμικού

Για κάθε κλάση που θα αναπτύξετε, εκτός από τις κλάσεις που αφορούν τη γραφική διεπαφή, θα πρέπει να υλοποιήσετε και αντίστοιχη κλάση ελέγχου. Αυτό μπορεί να γίνει είτε αναπτύσσοντας δικές σας κλάσεις ελέγχου είτε χρησιμοποιώντας το πλαίσιο ελέγχου JUnit.

Λογιστικά

Η εργασία χωρίζεται σε δύο μέρη. Στο πρώτο μέρος θα πρέπει να παραδώσετε την σχεδίαση του λογισμικού σας, δηλαδή να παρουσιάσετε ένα διάγραμμα κλάσεων του λογισμικού σας και για κάθε κλάση να παρουσιάσετε τις δημόσιες μεθόδους που περιλαμβάνει. Τόσο οι κλάσεις όσο και οι δημόσιες μέθοδοι θα πρέπει να είναι τεκμηριωμένες. Όσο περισσότερο χρόνο αφιερώσετε στην σχεδίαση, τόσο λιγότερο χρόνο θα χρειαστείτε μετά στην υλοποίηση. Το παραδοτέο αυτό θα μετρήσει κατά 10% στον τελικό βαθμό του μαθήματος.

Στο δεύτερο μέρος θα πρέπει να παραδώσετε την πλήρη υλοποίηση του λογισμικού σας μαζί με την ενδεχομένως ενημερωμένη έκδοση της σχεδίασης. Επιτρέπεται δηλαδή να αποκλίνετε από τον αρχικό σχεδιασμό σας, χωρίς αυτό να έχει κάποια αρνητική επίπτωση στη βαθμολογία. Το παραδοτέο αυτό θα μετρήσει κατά 20% στον τελικό βαθμό του μαθήματος.

Η εργασία μπορεί να γίνει ατομικά ή από ομάδες των δύο ατόμων. 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. Θα πρέπει να σταλεί μέσω του συστήματος ΠΗΛΕΑΣ, ένα αρχείο zip ή rar με όνομα **AEM1-AEM2.zip** ή **AEM1-AEM2.rar** ή **AEM.zip** ή **AEM.rar**, το οποίο θα περιέχει:

Μέρος 1°

1. Ένα αρχείο PDF με το διάγραμμα κλάσεων του λογισμικού σας
2. Ένα αρχείο java για κάθε κλάση, το οποίο θα περιέχει μόνο τις υπογραφές των δημοσίων μεθόδων της κλάσης (χωρίς δηλαδή πεδία και χωρίς υλοποίηση του κώδικα των μεθόδων). Οι κλάσεις και οι δημόσιες μέθοδοι θα πρέπει να είναι τεκμηριωμένες με σχόλια τύπου Javadoc. Οι κλάσεις θα πρέπει να είναι οργανωμένες σε ένα η παραπάνω πακέτα.

Μέρος 2°

1. Ένα αρχείο PDF με το διάγραμμα κλάσεων ενδεχομένως ενημερωμένο.
2. Τον πηγαίο κώδικα που αναπτύξατε για την εφαρμογή καθώς και για τον έλεγχό της. Ο κώδικας θα πρέπει να είναι τεκμηριωμένος με σχόλια τύπου Javadoc.
3. Ένα jar αρχείο με το bytecode της εφαρμογής σας.
4. Ένα txt αρχείο, το οποίο θα αναφέρει τα ονόματα των δύο εκτελέσιμων κλάσεων του λογισμικού σας, μία για το graphical user interface του διαχειριστή και μία για του χρήστη.

Προσοχή: στο σύστημα ΠΗΛΕΑΣ μπορεί να ανεβεί **ένα** μόνο αρχείο ανά χρήστη. Αν ανεβεί δεύτερο, **αντικαθίσταται το πρώτο**. Έτσι, αν ξεχάσατε κάποιο αρχείο και θέλετε να το συμπληρώσετε, θα πρέπει να ξαναστείλετε το .zip αρχείο σας – όχι μόνο αυτό που έλειπε. Οι ομαδικές εργασίες θα πρέπει να υποβάλλονται από έναν εκπρόσωπο της ομάδας και μόνο.

Αξιολόγηση

Για την αξιολόγηση των εργασιών θα συνεκτιμηθεί:

1. Η σχεδίαση του λογισμικού (ευκολία συντήρησης και επέκτασης)
2. Η υλοποίηση του λογισμικού (πληρότητα υλοποίησης λειτουργιών, ποιότητα υλοποίησης)
3. Ο έλεγχος του λογισμικού (πληρότητα και ποιότητα κλάσεων ελέγχου)
4. Η τεκμηρίωση εντός του κώδικα και εντός του εγγράφου τεκμηρίωσης
5. 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.