**ΣΑΚΕΛΛΑΡΙΟΥ ΓΕΩΡΓΙΟΣ 174935**

**ΕΡΓΑΣΙΑ 2019 ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΤΗΣ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΑΝΘΡΩΠΟΥ ΜΗΧΑΝΗΣ**

**ΠΡΟΛΟΓΟΣ**

Αρχικά η εργασία μου στηρίζεται πάνω στο 6ο εργαστηριακό μάθημα της Αλληλεπίδρασης Ανθρώπου μηχανής. Επέλεξα το συγκεκριμένο θέμα μιας και ήταν κάτι που μου είχε μείνει από το εργαστήριο.

**ΕΡΓΑΛΕΙΑ**

Για αυτή την εργασία έκανα χρήση διάφορων εργαλείων. Αρχικά ως IDE έκανα χρήση του IntelliJ Idea μιας και είναι το IDE που με έχει βολέψει εξαιτίας των εύκολων και χρήσιμων shortcuts που μου παρέχει, ενώ ταυτόχρονα υποστηρίζει όλα τα νέα Version της Java, σε αντίθεση με άλλα IDE (όπως NetBeans), ωστόσο η χρήση JavaFx με το IntelliJ με δυσκόλεψε περισσότερο απ’ότι θα ήθελα.

Επιπλέον χρησημοποίησα το SceneBuilder για να σχεδιάσω το UI της εφαρμογής, το οποίο οποίο δημιουργεί ένα αρχείο FXML, το οποίο μπορώ μέσω Java να το τροποποιήσω με όποιον τρόπο θέλω εάν χρειαστεί.

**ΕΦΑΡΜΟΓΗ**

Το src της εφαρμογής μου αποτελείται από το module-info.java το οποίο παρέχει μονάχα ένα module το javafxtest το οποίο λέει στην εφαρμογή ότι απαιτεί javafx.fxml και javafx.controls για να λειτουργήσει, καθώ και ότι ανοίγει το αρχειο sample.fxml.

Επιπλέον στην εφαρμογή έχουμε το sample.fxml, που δημιουργήθηκε μέσω του SceneBuilder, και αποτελείται από διάφορα Elements. Αρχικά υπάρχουν όλα τα απαραίτητα imports, όπως για παράδειγμα το javafx Button ή το javafx TextField. Έπειτα ακολουθεί ένα GridPane που είναι ουσιαστικά το μέρος στο οποίο χτίζεται όλη η εφαρμογή, επίσης το GridPane έχει κάποια στοιχεία, όπως alignment=”center” το οποίο στοιχίζει τα κείμενα της εφαρμογής στη μέση. Έπειτα ανοίγει ένα Pane, το οποίο έχει στο εσωτερικό του Τα 2 Tabs, ένα για την Φόρμα Παραγγελίας, και ένα για την Συνολική Παραγγελία. Το καθένα από αυτά τα Tabs, έχει στο εσωτερικό του άλλα αντίστοιχα στοιχεία, όπως διάφορα Label τα οποία έιναι τα διάφορα Text που βρίσκονται στην εφαρμογή, ένα ListView που μας παρέχει τις επιλογές των διάφορων φαγητών και ένα ComboBox που μας παρέχει τις τα διάφορα Αναψυκτικά-Ποτά. Επιπλέον έχουμε 2 Spinner, ώστε ο χρήστης να επιλέγει την ποσότητα των επιλεγμένων φαγητών ή ποτών, και ένα Button το οποίο προσθέτει τα επιλεγμένα τρόφιμα στο 2ο Tab. Το 2ο Tab μας παρέχει την συνολική παραγγελία σε ένα Table, που χωρίζεται σε 3 κατηγορίες, Προϊόν, Ποσότητα και Κόστος ανά Μονάδα. Τέλος υπάρχει ένα TextField που δείχνει το συνολικό κόστος της παραγγελίας.

Επίσης υπάρχει το αρχείο Main.java, που δίνει στο αρχείο που αρχικά δίνει το Path του sample.fxml στο root, δίνει τον τίτλο «Καντίνα ο Μπάμπης» στο Application, και δίνει το Height & Width στο root (που στην δική μου περίπτωση είναι 500px x 500px) ενώ στο τέλος εμφανίζει το application.

Ταυτόχρονα υπάρχει η κλάση Paraggelia, την οποία απαρτίζουν διάφορες μεταβλητές (eidos, posotita, kostosTemaxiou, kostosTemaxiwn), και κάθε προϊόν που προστίθεται στο καλάθι, έχει αυτές τις τιμές. Το eidos αντικατοπτρίζει το είδος του φαγητού (πχ γύρος, σουβλάκι κτλ), το posotita τον αριθμό των τεμαχίων που έχει επιλεχθεί για το συγκεκριμενο προϊόν (πχ, ποσότητα για Σουβλάκια = 5, 3, 100), το kostosTemaxiou που μας δίνει το κόστος του προϊόντος για κάθε μία μερίδα (πχ, κόστος ανά μερίδα Γεμιστών = 5), καθώς και το kostosTemaxiwn που προκύπτει από τον πολλαπλασιασμό του κόστους τεμαχίου, με την ποσότητα των τεμαχίων.   
Την ίδια στιγμή υπάρχει ο δομητής της Paraggelias, ο οποίος παίρνει 3 τιμές (για τα eidos, posotita, kostosTemaxiou), και υπολογίζει την 4η (kostosTemaxiwn), μέσω της μεθόδου ypologismosKostousTemaxiwn που κάνει τον πολλαπλασιασμό που προαναφέρθηκε προηγουμένως, ενώ ακολουθούν getters για όλες τις μεταβλητές μιας και είναι απαραίτητοι για την προσθήκη μεταβλητών στο Table.

To Controller, είναι η κλάση στην οποία συμβαίνει το μεγαλύτερο ποσοστό του κώδικα. Αρχικά υπάρχουν τα imports, τα οποία γίνονται με κάθε μία νέα προσθήκη ενός νέου Resource ή Action της javafx. Πχ για την προσθήκη ενός event όπως το onclick, σε ένα κουμπί, πρέπει να γίνει το αντίστοιχο import που στην συγκεκριμένη περίπτωση είναι το javafx.scene.input.MouseEvent. Μετά ακολουθεί η δήλωση των μεταβλητών μου. Αυτές είναι το totalOrderCost (κόστος της συνολικής παραγγελίας), δύο πίνακες foodCost, drinkCost, τύπου double, έξι και τεσσάρων θέσεων αντίστοιχα, που έχουν το κόστος της κάθε μονάδας του κάθε προϊόντος (πχ Πορτοκαλάδα 1 ευρώ, Γύρος 3.20 ευρώ), και δύο πίνακες availableFoodImages, availableDrinksImages τύπου Image, έξι και τεσσάρων θέσεων αντίστοιχα, που αποθηκεύουν τις εικόνες οι οποίες θα εμφανιστούν έπειτα από τις απαραίτητες ενέργειες (όπως στο onclick των στοιχείων ενός στοιχείου του ComboBox). Έπειτα υπάρχει η μεταβλητή df2 τύπου DecimalFormat που βοηθάει στην στρογγυλοποίηση ενός αριθμού σε κάποια δεκαδικά ψηφία και στην συγκεκριμένη περίπτωση σε 2.   
Στην πορεία ακολουθούν οι μεταβλητές των διάφορων τύπων της javafx. Αυτές είναι οι εξής:

|  |  |
| --- | --- |
| **ΤΥΠΟΣ ΜΕΤΑΒΛΗΤΗΣ** | **ΟΝΟΜΑ ΜΕΤΑΒΛΗΤΗΣ** |
| TableView | totalOrder |
| Spinner | posotitaF(posotitaFood) |
| Spinner | posotitaP(posotitaPotwn) |
| ObservableList<String> | dishes |
| ObservableList<String> | drinks |
| TableColumn | Proion |
| TableColumn | Posotita |
| TableColumn | Kostos |
| TextField | synolikoKostos |
| ComboBox | drinksComboBox |
| ListView | foodList |
| ImageView | foodImage |
| ImageView | drinksImage |

Στη μέθοδο initialize, δίνω τις τιμές των διάφορων πιάτων που είναι αποθηκευμένες στην μεταβλητή dishes στο ListView foodList, με την εντολή foodList.getItems().addAll(dishes), ενώ στην συνέχεια αρχικοποιώ το Spinner posotitaF με την τιμή 0, και του δίνω ανώτατη τιμή το 100. Αντίστοιχες ενέργειες γίνονται και για το ComboBox drinksComboBox, που του δίνονται οι τιμές που είναι αποθηκευμένες στη μεταβλητή drinks, και αρχικοποιείται επίσης και το Spinner posotitaP. Ακολουθεί η αρχικοποίηση των δύο πινάκων τύπου Image availableFoodImages και availableDrinksImages, με τις διάφορες εικόνες, και στην συνέχεια οι πίνακες με τις τιμές των διαθέσιμων φαγητών και ποτών. Στο τέλος του initialize γίνεται το setCellValueFactory για τα 3 κελία του πίνακα, που είναι απαραίτητο προκειμένου να μπορούν να προσθεθούν νέες τιμές σε αυτα.

Στη μέθοδο showFood εμφανίζω τις εικόνες στο foodImage έπειτα από το onClickEvent ενός από τα κελία του listView. Ο έλεγχος γίνεται με βάση foodList.getSelectionModel().getSelectedItem(), το οποίο συγκρίνεται με το όνομα του αντίστοιχου φαγητού. Έτσι στο foodList.getSelectionModel().getSelectedItem().equals(‘’Γύρος’’), γίνεται η εμφάνιση της εικόνας που αντιστοιχεί στον γύρο. Παρόμοια διαδικασία γίνεται και με την μέθοδο showDrinks, η οποία καλείται στο onActionEvent του drinksComboBox με τον έλεγχο πχ drinksComboBox.getSelectionModel().getSelectedItem().equals("Κόκα Κόλα").

Επίσης έχουμε τις μεθόδους foodCostPerUnit, και drinkCostPerUnit, τα οποία με τους ελέγχους foodList.getSelectionModel().getSelectedItem().equals(‘’’’), και drinksComboBox.getSelectionModel().getSelectedItem().equals(“”), επιστρέφουν τις τιμές ανά τεμάχιο για το οποιοδήποτε προϊόν.

Με τις μεθόδους foodPriceMul και drinksPriceMul, γίνεται ο πολλαπλασιασμός του κόστους του κάθε προϊόντος, με την αντίστοιχη ποσότητα. Ο έλεγχος για το είδος του προϊόντος γίνεται με τον τρόπο που αναφέρθηκε προηγουμένως, και η πράξη αυτή γίνεται με τα Double.parseDouble(posotitaF.getValue().toString()) \* foodCost[0] Double.parseDouble(posotitaP.getValue().toString()) \* drinkCost[0], μιας και από το Spinner επιστρέφεται η ποσότητα με τύπο String. Έτσι σε περίπτωση όπου έχουμε τον αριθμό 2 στο Spinner των ποτών, και επιλεγμένη την Κόκα Κόλα, θα γίνει η επιστροφή του 2 \* 1.20 που είναι το κόστος ανά μονάδα Κόκα Κόλας.

Τέλος έχουμε την μέθοδο addToTable, η οποία καλείται στο onClick του Button prosthikiProiontwn. Σε αυτή την μέθοδο γίνεται αρχικά ο έλεγχος της περίπτωσης να μην έχει επιλεχθεί κάποιο είδος φαγητού από το foodList με τον έλεγχο foodList.getSelectionModel().getSelectedItem() != null, ενώ στην συνέχεια μέσω της function foodPriceMul(), ελέγχουμε εάν το κόστος των φαγητών που έχουν προσθεθεί είναι μηδενικά (με άλλα λόγια εάν το Spinner έχει Value = 0), και σε περίπτωση που δεν είναι δημιουργούμε την νέα μας παραγγελία δημιουργόντας κάθε φορά ένα νέο αντικείμενο τύπου Paraggelia, και κάνοντας χρήση των function που προαναφέρθηκαν για την δόμηση αυτού του αντικειμένου. Ταυτόχρονα εμφανίζουμε αυτή την νέα παραγγελία στο Table. Όλη η διαδικασία που περιγράφεται γίνεται με την εξής εντολή Μετά προσαυξάνω το totalOrderCost με το foodPriceMul().   
Ακριβώς η ίδια διαδικασία ακολουθεί και για την προσθήκη ποτών στον πίνακα.  
Τέλος μηδενίζω τα Spinner PosotitaF και PosotitaP της ποσότητας φαγητού και ποτών αντίστοιχα, και δίνω την νέα συνολική τιμή στο synolikoKostos, περιορίζοντας το format σε 2 δεκαδικά ψηφία μέσω του DecimalFormat.