

Προηγμένα θέματα προγραμματισμού

Προγραμματιστική εργασία Προγράμματα για
επιχείρηση πλυντηρίου αυτοκινήτων

Ονοματεπώνυμο: Γιώργος Καλαντζόπουλος

ΑΜ: 2022201800062

Ονοματεπώνυμο: Σπύρος Αποστολόπουλος

ΑΜ: 2022201800016

Περιεχόμενα

Προηγμένα θέματα προγραμματισμού	1
Περιεχόμενα.....	2
Σχεδίαση των κλάσεων	3
Client	3
Client.java:.....	3
ClientUI.java:	3
clientActions.java:	3
CheckBoxesType.java:	4
Keyboard.java:.....	4
FindAddress.java	4
Service.java:	4
Services.java:.....	4
UserData.java	5
Server	6
Server.java:	6
ServerUI.java:	6
ServerActions.java:.....	6
ServerSide.java:.....	6
Service.java:	7
Services.java:.....	7
UserData.java:	7
Δομές αρχείων	8
Service.java:	8
Services.java:.....	8
UserData.java:	8
Στοιχεία της υλοποίησης που δεν συμφωνούν με την εκφώνηση	8
Στοιχεία	8

Σχεδίαση των κλάσεων

Client

Client.java:

Δημιουργούμε το αρχικό screen με το text-field του license number και τα δύο buttons του Keyboard και του Next το οποίο δημιουργεί ένα new Window. Επίσης το Next για να γίνει enable πρέπει να καλέσουμε από την κλάση clientActions την συνάρτηση nodeActiveFromTextfield .

Επίσης στην Client.java υλοποιούμε και την συνάρτηση στην οποία δημιουργούμε το νέο παράθυρο στην οποία έχουμε ένα Label, ένα ComboBox με το οποίο δίνουμε την ικανότητα να επιλέξει τον τύπο του οχήματος του ,το συνολικό κόστος και την πινακίδα του. Μετά αν πατήσει το κουμπί Submit Selection του εμφανίζει τις υπηρεσίες ανάλογα με το όχημα του. Αν επιλέξει κάποια υπηρεσία το κόστος αυξάνετε ανάλογα την τιμή της. Τέλος αν πατήσει το Done τα στοιχεία της συναλλαγής στέλνονται στον Server.

Για να σταλούν τα δεδομένα στον Server η Client καλεί την FindAddress.java όπου την εξηγούμε αργότερα. Σε περίπτωση που το πρόγραμμα δεν έχει καταφέρει να βρει τον server όταν ο χρήστης πατήσει το κουμπί αποστολής πινακίδας θα εμφανιστεί ένα άσπρο παράθυρο. Το παράθυρο αυτό θα κλείσει από μόνο του και θα εμφανιστούν οι επιλογές μόλις καταφέρει να συνδεθεί με τον σέρβερ.

ClientUI.java:

Στην ClientUI δημιουργήσαμε ένα σύστημα που αυτόματα δημιουργεί ένα node και το τοποθετεί ανάλογα στο Κέντρο Επάνω η Κάτω. Αυτό το σύστημα το χρησιμοποιήσαμε αρκετά για να δημιουργήσαμε τα αντικείμενα στην Client. Συγκεκριμένα η κλάση που χρησιμοποιήσαμε περισσότερο είναι η fnAdd που δημιουργεί αυτόματα ένα HBOX σώζοντας ακόμα περισσότερο χρόνο στην διαδικασία. Στο ίδιο σύστημα δημιουργήσαμε αντίστοιχες κλάσεις για Get όπου γυρνάνε αυτόματα το παιδί που βρίσκετε μέσα στην κλάση στο Node.

clientActions.java:

Σε αυτή την κλάση βρίσκετε η συνάρτηση η οποία ελέγχει αν ο αριθμός πινακίδας είναι έγκυρος (να είναι μεγαλύτερος από 2 γράμματα και να μην είναι κενός) και ανάλογα με αυτό ενεργοποιείτε το κουμπί Next στην σελίδα στην οποία ο client καταχωρεί τον αριθμό πινακίδας του.

CheckBoxesType.java:

Η κλάση αυτή δημιουργεί όλα τα Checkboxes και φροντίζει να μην αφήνουν να επιλεγθούν επιλογές οι οποίες δεν μπορούν να επιλεγθούν μαζί

Keyboard.java:

Η υλοποίηση του keyboard έγινε με την χρήση hbox και vbox .Στην αρχή δημιουργήσαμε ένα Array όπου περάσαμε τα κεφαλαία γράμματα του λατινικού αλφάβητου. Στην συνέχεια υλοποιήσαμε το 0 button ,με τρεις for δημιουργήσαμε τα buttons με τους αριθμούς από το 1 – 9 τους οποίους ομαδοποιήσαμε σε 3 hbox και στο τέλος τα προσθέσαμε σε ένα vbox έτσι ώστε να έχουν την σωστή μορφολογία. Παράλληλα με την δημιουργία των button με τους αριθμούς καλούσαμε την αντίστοιχη συνάρτηση για να την λειτουργία των κουμπιών.

Μετά με 2 for και την χρήση του array με τα γράμματα δημιουργήσαμε τα buttons των γραμμάτων (και καλούμε την συνάρτηση λειτουργία τους) τα οποία σπάσαμε σε δύο hbox . Επίσης υλοποιήσαμε buttons για το backspace και για το space όπου μετά την δημιουργία τους καλούμε κατάλληλη συνάρτηση για την λειτουργία τους. Παρατηρώντας διάφορα UI για keyboard προσθέσαμε το feature να μπορεί να ανοίγει και να κλείνει (με setVisible) το πληκτρολόγιο. Για να πραγματοποιηθεί σωστά έπρεπε να προσθέσουμε άλλα δύο buttons το Close(κλείνει) και το Keyboard(ανοίγει) τα οποία απενεργοποιούνται αντίστοιχα για να αποφύγουμε τυχόν bug. Έτσι το Keyboard button το τοποθετήσαμε δίπλα στο textfield και το Close με το Backspace σε ένα hbox(Μεταξύ του Close και του Backspace προσθέσαμε ένα Spacing) και μετά το hbox αυτο μαζί με τα δύο hbox των γραμμάτων και το Space τα βάζουμε σε ένα vbox. Τέλος σε ένα hbox τοποθετούμε τα δύο vbox και έχουμε το επιθυμητό αποτέλεσμα .

FindAddress.java

Η κλάση αυτή τρέχει σε ένα καινούργιο thread και από την στιγμή που καλείτε από το αρχικό πρόγραμμα προσπαθεί να σκανάρει το τοπικό δίκτυο για να βρει τον server (δηλαδή την μεριά του ταμείου). Για να το πετύχει αυτό ψάχνει ένα ένα τα τοπικά Ip αν ακούνε στην πόρτα 8787 (όπου τοποθετούμε και τον σέρβερ σε αυτήν).

Σε περίπτωση που δεν βρεθεί ο σέρβερ το πρόγραμμα δεν σταματάει και ξαναρχίζει από την αρχή σκανάροντας πάλι όλο το τοπικό δίκτυο και ψάχνοντας τον server

Service.java:

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

Services.java:

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

UserData.java

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

Server

Server.java:

Η κλάση αυτή είναι η αρχική κλάση για τον Σέρβερ μας – ταμείου, σε αυτήν δημιουργούμε όλα μας τα αντικείμενα και τους περνάμε ότι δεδομένα χρειάζονται. Μια από τις υλοποιήσεις που προσπαθήσαμε να πετύχουμε ήταν να δημιουργήσουμε ένα αντικείμενο `ServerActions` που θα το περνάμε και σε όλα τα υπόλοιπα αντικείμενα ώστε να μπορέσουμε εύκολα να κρατάμε κοινές οι αλλαγές που κάνουμε σε κάποια λίστα ή κοινά στοιχεία μεταξύ των αντικειμένων (Αυτό δεν το πετύχαμε σε τέλειο βαθμό αλλά σε έναν αρκετά ικανοποιητικό).

ServerUI.java:

Εδώ υλοποιούμε το UI του ταμείου με ένα `Label` και ένα `TableView`. Δημιουργούμε τα `columns` του `table` και περνάμε από το `Observable list` τα αντικείμενα μιας κλάσης η οποία ονομάζεται `UserData`. Μετά με το κουμπί `CancelUser` μπορούμε να ακυρώσουμε πελάτες με την κλήση της συνάρτησης `DeleteItemFromTable` στην οποία βρίσκουμε ποιο `row` είναι επιλεγμένο και μετά με ένα `setOnAction` για το συγκεκριμένο κουμπί διαγράφουμε το `row` και από το `TableView` και από το `ObservableList`. Επιπλέον με το κουμπί `EntryUser` καταχωρούμε έναν επιλεγμένο `row` του `TableView` στο `csv` το οποίο είναι το βιβλίο εισόδων. Αυτό μπορεί να υλοποιηθεί και με το `virtual` κουμπί `EntryUser` αλλά και με το κουμπί `Enter` του πληκτρολογίου με ένα `setOnKeyPressed`.

ServerActions.java:

Όπως περιγράψαμε και στην `Server` η κλάση αυτή ευθύνεται για να κρατάει κοινά στοιχεία μεταξύ των αντικειμένων και να κάνει πράγματα με αυτά. Ποιο συγκεκριμένα η κλάση αυτή χειρίζεται δύο λίστες. Μία που δέχεται από την αρχική κλάση και μία που δημιουργεί η ίδια. Η πρώτη λίστα βασίζεται πλήρως στο αρχείο που δημιουργούμε και έχει όλα τα δεδομένα, καθώς η δεύτερη παίρνει μόνο τα δεδομένα από το αρχείο που δεν έχουν κάποια τιμή στο πεδίο της ώρας που αναχώρησε από το ταμείο (Αυτή χρησιμοποιείται και για το `tableview`).

ServerSide.java:

Η κλάση αυτή φροντίζει να κάνει όλη την σύνδεση με τον `client` και να δεχτεί τα δεδομένα από αυτόν. Η κλάση αυτή αντίστοιχα με την κλάση `FindAddress.java` τρέχει σε διαφορετικό `thread` και κάθε φορά που δέχεται μία σύνδεση διαβάζει το αντικείμενο (`UserData`) και το προσθέτει στην λίστα.

Service.java:

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

Services.java:

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

UserData.java:

Η κλάση αυτή έχει σχέση με την δομή και εξηγείτε αργότερα

Δομές αρχείων

Service.java:

Η αρχική δομή όπου περιέχει το όνομα της υπηρεσίας και την τιμή του για κάθε όχημα

Services.java:

Δεύτερο επίπεδο της δομής όπου περιέχει ένα map από service και ένα String ως το κλειδί με το string να είναι ο κωδικός της υπηρεσίας. Η κλάση αυτή περιέχει αρκετές χρήσιμες μεθόδους όπως την λειτουργία να διαβάζει της υπηρεσίες από ένα αρχείο κλπ.

UserData.java:

Περιέχει τα δεδομένα που στέλνονται και λαμβάνονται μεταξύ των δύο Προγραμμάτων (Πινακίδα , διαλεγμένες υπηρεσίες , Συνολική Τιμή , Ώρα προσέλευσης , Ώρα φυγής)

Στοιχεία της υλοποίησης που δεν συμφωνούν με την εκφώνηση

Στοιχεία

Δεν πιστεύουμε πως υπάρχουν στοιχεία που δεν συμφωνούν με την εκφώνηση παρόλα αυτά λόγω του ότι δεν είχαμε χρόνο να αρχίσουμε την εργασία αρκετά νωρίς είμαστε σίγουροι ότι θα μας έχει ξεφύγει κάτι καθώς και ότι μπορούσαμε να βελτιώσουμε αρκετά πράγματα άλλα δεν είχαμε τον χρόνο να το κάνουμε.