



**Πανεπιστήμιο Πελοποννήσου
Πληροφορικής και Τηλεπικοινωνιών
Προηγμένα Θέματα Προγραμματισμού
Τελική Εργασία**

**Μότση Πολυξένη
AM 2022201800125**

**Αντώνιος Λινάρδος
AM 2022201700100**

ΠΕΡΙΕΧΟΜΕΝΑ

1. CLIENT - ΠΡΟΓΡΑΜΜΑ ΥΠΟΔΟΧΗΣ ΟΧΗΜΑΤΩΝ	3
a. Παρουσίαση της σχεδίασης των κλάσεων	3
b. Δομή Αρχείων	10
c. Στοιχεία που πραγματοποιήθηκαν	10
2. SERVER - ΠΡΟΓΡΑΜΜΑ ΤΑΜΕΙΟΥ	
a. Παρουσίαση της σχεδίασης των κλάσεων	11
b. Δομή Αρχείων	23
c. Στοιχεία που πραγματοποιήθηκαν	23
3. ΣΥΝΔΕΣΗ CLIENT - SERVER	24

CLIENT - ΠΡΟΓΡΑΜΜΑ ΥΠΟΔΟΧΗΣ

Παρουσίαση της σχεδίασης των κλάσεων

Στο πρόγραμμα ταμείου έχουμε τις εξής 5 κλάσεις:

1. *Client.class*

Στην κλάση Client γίνεται η δημιουργία του γραφικού περιβάλλοντος του προγράμματος υποδοχής και έχει την λειτουργικότητα του πληκτρολογίου.

a. Δεν υπάρχουν πεδία

b. Μέθοδοι

Μέθοδος	Περιγραφή
start()	Μέσα στην start φτιάχνεται το αρχικό παράθυρο του προγράμματος στο οποίο ο χρήστης εισάγει τον αριθμό κυκλοφορίας του οχήματος του.
createLogo()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>HBox</i> Με αυτήν την μέθοδο δημιουργούμε το logo του προγράμματος το οποίο θα μπει σε ένα HBox διότι θέλουμε να εμφανίσουμε και την εικόνα αλλά και το όνομα "IncomeBook".
secondaryScene()	TYPE → <i>Public</i> PARAM → <i>void</i> Με αυτήν την μέθοδο δημιουργούμε το δεύτερο παράθυρο του προγράμματος στο οποίο θα εμφανίζεται η λίστα των υπηρεσιών και οι τιμές αυτών, που προσφέρει το πλυντήριο αυτοκινήτων.
getProduct()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>Product</i>

	<p>Με αυτήν την μέθοδο φτιάχνουμε μία ObservableList οι οποία περιέχει δεδομένα τύπου Product (Παραπάνω πληροφορίες στην ανάλυση της κλάσης Product παρακάτω) την οποία θα χρησιμοποιήσουμε για να φτιάξουμε την λίστα με τις υπηρεσίες του πλυντηρίου αυτοκινήτων .</p>
productlistWindow()	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>HBox</i></p> <p>Με αυτή την μέθοδο δημιουργούμε μέσα σε ένα HBox την λίστα με όλες τις υπηρεσίες και τις τιμές για κάθε τύπου όχημα καθώς και εξηδικευμένα κουμπιά για τον κάθε τύπο αμαξιού.</p>
buttonPressed()	<p>TYPE → <i>Public</i> PARAM → <i>Button</i> RETURN → <i>void</i></p> <p>Αυτή η μέθοδος καλείται όταν ο χρήστης πατάει ένα από τα κουμπιά που αντιστοιχεί στον τύπο οχήματος του. Στην συνέχεια καλεί την μέθοδο που θα δημιουργήσει το παράθυρο του τύπου αυτού.</p>
carScene()	<p>TYPE → <i>Public</i> PARAM → <i>Button</i> RETURN → <i>void</i></p> <p>Αυτή η μέθοδος δημιουργεί το παράθυρο για τον τύπο οχήματος : Αυτοκίνητο.</p>
jeepScene()	<p>TYPE → <i>Public</i> PARAM → <i>Button</i> RETURN → <i>void</i></p> <p>Αυτή η μέθοδος δημιουργεί το παράθυρο για τον τύπο οχήματος : Τζιπ.</p>
motoScene()	<p>TYPE → <i>Public</i> PARAM → <i>Button</i></p>

	RETURN → <i>void</i> Αυτή η μέθοδος δημιουργεί το παράθυρο για τον τύπο οχήματος : Μηχανή.
radiobuttonbox()	TYPE → <i>Public</i> PARAM → <i>Button</i> RETURN → <i>VBox</i> Αυτή η μέθοδος φτιάχνει τα radiobuttons τα οποία χρησιμοποιούνται και από τα τρία παράθυρα οχημάτων. Επίσης, καλεί και μεθόδους από την κλάση HelperClass με την βοήθεια της οποία γίνεται ο υπολογισμός της συνολικής τιμής των υπηρεσιών

2. HelperClass.class

Στην κλάση HelperClass δημιουργούμε το πληκτρολόγιο και φτιάχνουμε την λειτουργικότητα των radiobutton όλων των παραθύρων.

- a. Δεν υπάρχουν πεδία
- b. Μέθοδοι

Μέθοδος	Περιγραφή
keyboardLayout()	TYPE → <i>Public</i> PARAM → <i>GridPane, GridPane, HBox, Map, Button</i> RETURN → <i>void</i> Με την βοήθεια αυτής της μεθόδου δημιουργούμε το πληκτρολόγιο του κυρίου παραθύρου
carRadiobuttons()	TYPE → <i>Public</i> PARAM → <i>ObservableList<RadioButton>, TextField</i> RETURN → <i>void</i> Η μέθοδος αυτή φτιάχνει τους

	περιορισμούς για τα radiobuttons και υπολογίζει την τιμή (in real time) για την κάθε υπηρεσία που επιλέγει ο χρήστης για το Αυτοκίνητο.
jeepRadiobuttons()	TYPE → <i>Public</i> PARAM → <i>ObservableList<RadioButton>, TextField</i> RETURN → <i>void</i> Η μέθοδος αυτή φτιάχνει τους περιορισμούς για τα radiobuttons και υπολογίζει την τιμή (in real time) για την κάθε υπηρεσία που επιλέγει ο χρήστης για το Τζιπ.
motoRadiobuttons()	TYPE → <i>Public</i> PARAM → <i>ObservableList<RadioButton>, TextField</i> RETURN → <i>void</i> Η μέθοδος αυτή φτιάχνει τους περιορισμούς για τα radiobuttons και υπολογίζει την τιμή (in real time) για την κάθε υπηρεσία που επιλέγει ο χρήστης για τη Μηχανή.

3. *Product.class*

Στην κλάση Product φτιάχνουμε τα πεδία τα οποία χρησιμοποιούμε για την δημιουργία την λίστας υπηρεσιών.

a.

Πεδίο	Περιγραφή
Private int product_id ;	Αναγνωριστικό για το TableView

Private String product_name ;	Όνομα υπηρεσίας
Private String car_price ;	Τιμή υπηρεσίας για αυτοκίνητο
Private String jeep_price ;	Τιμή υπηρεσίας για τζιπ
Private String motorbike_price ;	Τιμή υπηρεσίας για μοτοσυκλέτα

b. Μέθοδοι

Μέθοδος	Περιγραφή
getProduct_id()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>Int</i> Παίρνουμε τα id των υπηρεσιών.
setProduct_id (int product_id)	TYPE → <i>Public</i> PARAM → <i>product_id (Int)</i> RETURN → <i>void</i> Αναχωρούμε τα id των υπηρεσιών στο πεδίο services.
getProduct_name()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε το όνομα της υπηρεσίας.
setProduct_name (String product_name)	TYPE → <i>Public</i> PARAM → <i>product_name (String)</i> RETURN → <i>void</i> Αναχωρούμε το όνομα της υπηρεσίας στο πεδίο product_name;
getCar_price()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i>

setCar_price (String car_price)	<p>Παίρνουμε την τιμή της υπηρεσίας για το τύπο οχήματος Αυτοκίνητο.</p> <p>TYPE → <i>Public</i> PARAM → <i>car_price (String)</i> RETURN → <i>void</i></p> <p>Αναχωρούμε την τιμή της υπηρεσίας για το τύπο οχήματος Αυτοκίνητο στο πεδίο car_price;</p>
getJeep_price () setJeep_price (String jeep_price)	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i></p> <p>Παίρνουμε την τιμή της υπηρεσίας για το τύπο οχήματος Τζιπ.</p> <p>TYPE → <i>Public</i> PARAM → <i>jeep_price (String)</i> RETURN → <i>void</i></p> <p>Αναχωρούμε την τιμή της υπηρεσίας για το τύπο οχήματος Τζιπ στο πεδίο jeep_price;</p>
getMotorbike_price () setMotorbike_price (String motorbike_price)	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i></p> <p>Παίρνουμε την τιμή της υπηρεσίας για το τύπο οχήματος Μηχανή.</p> <p>TYPE → <i>Public</i> PARAM → <i>motorbike_price (String)</i> RETURN → <i>void</i></p> <p>Αναχωρούμε την τιμή της υπηρεσίας για το τύπο οχήματος Μηχανή στο πεδίο motorbike_price;</p>

c. Constructor

Δέχεται για παραμέτρους όλες τις πληροφορίες που πρέπει να τοποθετήσει στα πεδία.

4. FileHelper.class

Με τη βοήθεια της κλάσης FileHelper δημιουργούμε και γράφουμε τα δεδομένα στο αρχείο που θα στείλουμε στο πρόγραμμα ταμείου.

a. Δεν έχει πεδία

b. Μέθοδοι

Μέθοδος	Περιγραφή
createFile()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>void</i> Δημιουργεί το αρχείο στο οποίο αποθηκεύουμε τα δεδομένα που θα σταλούν στη συνέχεια στο ταμείο
WriteToFile(String data)	TYPE → <i>Public</i> PARAM → <i>data (String)</i> RETURN → <i>void</i> Με την βοήθεια αυτής της μεθόδου γράφουμε τα δεδομένα στο αρχείο μας.

5. ClientFile.class

a. Constructor

Ανοίγει το αρχείο και στέλνει τα δεδομένα στον Server

Δομή Αρχείων

_____ Για το πρόγραμμα υποδοχής υπάρχει 1 αρχείο.

CarWash.txt: Εδώ αποθηκεύονται οι πληροφορίες των οχημάτων καθώς και οι πληροφορίες των υπηρεσιών που επέλεξε ο χρήστης στην μορφή *MJKI 45,Car,1-2-7-8,173*, όπου *MJKI 45* η πινακίδα του οχήματος, *Car* ο τύπος του οχήματος, *1-2-7-8* οι υπηρεσίες που διάλεξε ο χρήστης και τέλος *173* η συνολική τιμή που καλείται να πληρώσει.

Στοιχεία που πραγματοποιήθηκαν

Πραγματοποιήθηκαν όλα τα στοιχεία που ζητήθηκαν για τον Client.

SERVER - ΠΡΟΓΡΑΜΜΑ ΤΑΜΕΙΟΥ

Παρουσίαση της σχεδίασης των κλάσεων

Στο πρόγραμμα ταμείου έχουμε τις εξής 6 κλάσεις:

6. *Server.class*

Ο Server ουσιαστικά ενώνει τα κομμάτια για την εμφάνιση και τα εμφανίζει και επίσης έχει λειτουργικότητα για την μπάρα αναζήτησης, για το κουμπί refresh και για όταν το πρόγραμμα κλείνει να αποθηκευτούν οι νέες πληροφορίες από το πρόγραμμα υποδοχής στο αρχείο SavedCars.txt.

- a. Δεν υπάρχουν πεδία
- b. Δεν υπάρχουν μέθοδοι πέρα από τις βασικές που υπήρχαν ήδη.

7. *mainUI.class*

Εδώ είναι μόνο η εμφάνιση της αρχικής οθόνης.

- a. Δεν υπάρχουν πεδία
- b. Μέθοδοι

Μέθοδος	Περιγραφή
createMainPage()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>VBox</i> Με αυτήν την μέθοδο δημιουργούμε την αρχική εμφάνιση του προγράμματος. Δηλαδή που θα τοποθετηθούν το logo , το search bar και το TableView .
createLogo()	TYPE → <i>Public</i>

	<p>PARAM → <i>void</i> RETURN → <i>HBox</i></p> <p>Με αυτήν την μέθοδο δημιουργούμε το logo του προγράμματος το οποίο θα μπει σε ένα HBox διότι θέλουμε να εμφανίσουμε και την εικόνα αλλά και το όνομα "IncomeBook".</p>
<p>createRefreshButton(Button refreshButton)</p>	<p>TYPE → <i>Public</i> PARAM → <i>refreshButton (Button)</i> RETURN → <i>HBox</i></p> <p>Με αυτήν την μέθοδο φτιάχνουμε το κουμπί όπου θα το χρησιμοποιήσουμε για να πάρουμε τις νέες πληροφορίες από το πρόγραμμα υποδοχής (αν υπάρχουν) και να ανανεώσουμε το TableView. Επιστρέφουμε HBox για να το προσθεσουμε στα δεξιά της οθόνης.</p>
<p>createSearchField()</p>	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>TextField</i></p> <p>Με αυτήν την μέθοδο φτιάχνουμε την μπάρα αναζήτησης. Επιστρέφουμε το TextField που θα χρησιμοποιήσουμε.</p>

8. Vehicle.class

Εδώ απλά θα μπουν οι πληροφορίες για κάθε όχημα.

a. Πεδία

Πεδίο	Περιγραφή
Private int id ;	Αναγνωριστικό για το TableView
Private String arrival_time ;	Ωρα αναχώρησης του οχήματος
Private String vehicle_number ;	Αριθμός πινακίδας του οχήματος
Private int cost ;	Κόστος των υπηρεσιών που επέλεξε ο πελάτης
Private String date ;	Ημερομηνία αναχώρησης του οχήματος
Private String vehicleType ;	Είδος οχήματος.
Private String services ;	Υπηρεσίες γραμμένες στην μορφής "2-4-6", "3", "2-5" κτλ.

b. Μέθοδοι

Μέθοδος	Περιγραφή
getServices()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε τις υπηρεσίες.

setServices (String services)	TYPE → <i>Public</i> PARAM → <i>services (String)</i> RETURN → <i>void</i> Αναχωρούμε τις υπηρεσίες στο πεδίο services.
getVehicleType () setVehicleType (String vehicleType)	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε το είδος του οχήματος. TYPE → <i>Public</i> PARAM → <i>vehicleType (String)</i> RETURN → <i>void</i> Αναχωρούμε το είδος του οχήματος στο πεδίο vehicleType;
getId () setId (int id)	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>int</i> Παίρνουμε το αναγνωριστικό. TYPE → <i>Public</i> PARAM → <i>id (int)</i> RETURN → <i>void</i> Αναχωρούμε το id στο πεδίο id;
getArrival_time () setArrival_time (String arrival_time)	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε την ώρα αναχώρησης. TYPE → <i>Public</i> PARAM → <i>arrival_time (String)</i>

	<p>RETURN → <i>void</i> Αναχωρούμε την ώρα αναχώρησης στο πεδίο arrival_time.</p>
<p>getVehicle_number()</p> <p>setVehicle_number(String vehicle_number)</p>	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε τον αριθμό πινακίδας του οχήματος.</p> <p>TYPE → <i>Public</i> PARAM → <i>vehicle_number (String)</i> RETURN → <i>void</i> Αναχωρούμε τον αριθμό πινακίδας στο πεδίο vehicle_number.</p>
<p>getCost()</p> <p>setCost(int cost)</p>	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>int</i> Παίρνουμε το κόστος των επιλεγμένων υπηρεσιών.</p> <p>TYPE → <i>Public</i> PARAM → <i>cost (int)</i> RETURN → <i>void</i> Αναχωρούμε το κόστος στο πεδίο cost.</p>
<p>getDate()</p> <p>setDate(String date)</p>	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Παίρνουμε την ημερομηνία αναχώρησης.</p> <p>TYPE → <i>Public</i> PARAM → <i>date (String)</i> RETURN → <i>void</i></p>

	Αναχωρούμε την ημερομηνία στο πεδίο date.
--	---

c. Constructor

Δέχεται για παραμέτρους όλες τις πληροφορίες που πρέπει να τοποθετήσει στα πεδία.

9. **IncomeBook.class**

Εδώ υπάρχει η λειτουργικότητα για ότι έχει να κάνει με το βιβλίο εσόδων όπως για παράδειγμα η λειτουργικότητα για τα κουμπιά πληρωμής και διαγραφής οχήματος που βρίσκονται σε κάθε γραμμή, η ανάγνωση οχημάτων από τα αρχεία CarWash.txt και SavedCars.txt και μετά προσθήκη στο TableView, η διαγραφή οχημάτων από την λίστα και από το αρχείο SavedCars.txt. Επιπρόσθετα, υπάρχει και η λειτουργικότητα για τα κουμπιά πληρωμή και ακύρωση από την δεύτερη οθόνη.

a. Πεδία

Πεδίο	Περιγραφή
Private ObservableList<Vehicle> vehicles ;	Λίστα για την προσθήκη των προσωρινών οχημάτων που θα πρέπει να φαίνονται στο TableView.
Private TableView table ;	Ο πίνακας TableView που θα εμφανιστεί στην αρχική οθόνη.
Private int lastID ;	Μετρητής για το id. Κάθε φορά που φορά προστίθεται ένα όχημα αυτό αυξάνει κατά ένα και το όχημα παίρνει την τιμή του. Όταν σβήνεται ένα

	<p>όχημα από τον πίνακα αυτό μηδενίζεται και ξανα αρχίζει πάλι όπως είπα παραπάνω για να είναι το id της μορφής 0,1,2,3 και όχι 0,3,6,7 κτλ</p>
--	---

b. Μέθοδοι

Μέθοδος	Περιγραφή
createTable()	<p>TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>TableView</i></p> <p>Δημιουργούμε τον πίνακα προσθέτοντας του στήλες. Επιπρόσθετα δημιουργεί λειτουργικότητα για το κουμπί πληρωμής και για το κουμπί διαγραφής που θα έχει κάθε γραμμή ξεχωριστά.</p>
getVehicleFromFile(String filename, boolean delete)	<p>TYPE → <i>Public</i> PARAM → <i>filename (String)</i> PARAM → <i>delete (boolean)</i> RETURN → <i>void</i></p> <p>Εδώ πηγαίνουμε στο αρχείο που έχουμε με όνομα το περιεχόμενο της μεταβλητής filename και διαβάζουμε τα οχήματα από εκεί προσθέτοντας τα στην συνέχεια στην λίστα για να εμφανιστούν στο TableView. Άμα η boolean τιμή της μεταβλητής delete είναι true τότε θα καθαρίσει το</p>

	περιεχόμενο του αρχείου αυτού.
getVehicles()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>ObservableList<Vehicle></i> Επιστρέφει ολόκληρη την λίστα.
addVehicleToList(int id)	TYPE → <i>Public</i> PARAM → <i>id (int)</i> RETURN → <i>void</i> Προσθέτουμε το όχημα στην λίστα μας.
deleteVehicle()	TYPE → <i>Public</i> PARAM → <i>id (int)</i> RETURN → <i>void</i> Σβήνουμε το όχημα από το αρχείο Saved.txt και στο τέλος και από την λίστα.
deleteVehicleFromFile (File inputFile,File tempFile,int id, boolean delete)	TYPE → <i>Private</i> PARAM → <i>inputFile (File)</i> PARAM → <i>tempFile (File)</i> PARAM → <i>id (int)</i> PARAM → <i>delete (boolean)</i> RETURN → <i>void</i> Εδώ είναι που σβήνουμε όντως το όχημα από την λίστα. Καλείται από την μέθοδο deleteVehicle().
getVehicleNumber (int id)	TYPE → <i>Public</i> PARAM → <i>id (int)</i> RETURN → <i>String</i> Επιστρέφει τον αριθμό πινακίδα ενός συγκεκριμένου οχήματος στην λίστα μέσω

	του id.
--	---------

c. Constructor

Απλά αρχικοποιεί το *lastID* με τιμή μηδέν και στην συνέχεια καλεί την μέθοδο *getVehiclesFromFile()* για να μπουν να οχήματα στην λίστα.

10. ReceiptUI.class

Εδώ είναι η εμφάνιση της δεύτερης οθόνης όπου θα εμφανίζονται οι πληροφορίες του οχήματος, της απόδειξης, τις υπηρεσίες με τα κόστοι και τέλος το συνολικό ποσό.

a. Πεδία

Πεδίο	Περιγραφή
Private String s_Labels [];	Πίνακας με τις πληροφορίες των υπηρεσιών.
Private s_Car_Labels [];	Πίνακας με τις τιμές του αυτοκινήτου για τις υπηρεσίες.
Private s_Jeep_Labels [];	Πίνακας με τις τιμές του Jeep για τις υπηρεσίες.
Private s_Motor_Labels [];	Πίνακας με τις τιμές της μηχανής για τις υπηρεσίες.
Private Vehicle vehicle ;	Πεδίο για το αντικείμενο της κλάσης Vehicle.
Private String depTime ;	Πεδίο που αποθηκεύεται η ώρα αποχώρησης.
Private String depDate ;	Πεδίο που αποθηκεύεται η ημερομηνία αποχώρησης.
Private Button	Πεδίο που αποθηκεύεται το

b_payButton;	κουμπί για την πληρωμή.
Private b_cancelButton;	Πεδίο που αποθηκεύεται το κουμπί για την ακύρωση.

b. Μέθοδοι

Μέθοδος	Περιγραφή
getDepTime()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Επιστρέφει την ώρα αποχώρησης.
getDepDate()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>String</i> Επιστρέφει την ημερομηνία αποχώρησης.
getPayButton()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>Button</i> Επιστρέφει το κουμπί για την πληρωμή.
getCancelButton()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>Button</i> Επιστρέφει το κουμπί για την ακύρωση.
makeLogo()	TYPE → <i>Private</i> PARAM → <i>void</i> RETURN → <i>Pane</i> Δημιουργεί το logo για την δεύτερη οθόνη.

makeVehicleAndRecei ptInfo()	TYPE → <i>Private</i> PARAM → <i>void</i> RETURN → <i>BorderPane</i> Δημιουργεί την περιοχή κάτω από το Logo όπου θα εμφανίζονται οι πληροφορίες του αμαξιού και της απόδειξης. Τα βάζουμε σε <i>BorderPane</i> έτσι ώστε να πάνε εύκολα το ένα τέρμα δεξιά και το άλλο τέρμα αριστερά.
makeServicesInfo()	TYPE → <i>Private</i> PARAM → <i>void</i> RETURN → <i>BorderPane</i> Δημιουργούμε τις πληροφορίες των υπηρεσιών μαζί με τα κόστος τους και από κάτω το συνολικό ποσό.
makeReceiptUI()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>BorderPane</i> Δημιουργούμε την συνολική εμφάνιση ενώνοντας όλα τα υπόλοιπα από τις παραπάνω κλάσεις και την επιστρέφουμε.

c. Constructor

Αρχικοποιεί το αντικείμενο της κλάσης Vehicle και στην συνέχεια αρχικοποιεί την ώρα και ημερομηνία αποχώρησης,

11. Connention.class

Εδώ είναι η κλάση που θα πραγματοποιηθεί η σύνδεση. Τόσο του ανοίγματος του *ServerSocket* (της σύνδεσης) τόσο και της αποδοχής των αιτήσεων από το πρόγραμμα υποδοχής.

a. Πεδία

Πεδίο	Περιγραφή
ServerSocket serverSocket;	Πεδίο για τον Socket.

b. Μέθοδοι

Μέθοδος	Περιγραφή
closeConnection()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>void</i> Διακόπτει την σύνδεση.
makeConnection()	TYPE → <i>Public</i> PARAM → <i>void</i> RETURN → <i>void</i> Ξεκινάει την σύνδεση παίρνοντας επίσης (όχι από το πρόγραμμα υποδοχής) κάθε φορά που συνδέεται και την ώρα/ημερομηνία αναχώρησης.

c. Constructor

Δημιουργεί τον ServerSocket.

Δομή Αρχείων

_____ Για το πρόγραμμα ταμείου υπάρχουν 3 αρχεία.

CarWash.txt: Εδώ αποθηκεύονται τα οχήματα με το που τα παίρνουμε από το πρόγραμμα υποδοχής στην μορφή `MJKI45,Car,3,6,18 July 2021,18:07`. Στο τέλος, αφού διαβαστούν από το πρόγραμμα και προστεθούν στο TableView διαγράφονται για να μπουν επόμενα οχήματα.

`MJKI45,Car,3,6,18 July 2021,18:07`

Αρ. Πινακίδας, όχημα, υπηρεσίες, κόστος, ημερομηνία αν, ώρα ανχ

SavedCar.txt: Εδώ αποθηκεύονται όλα τα οχήματα που βρίσκονται στο TableView και δεν έχουν πληρωθεί ή ακυρωθεί ακόμα. Ο λόγος που τα έχουμε εκεί είναι για να μπορέσουν να ξαναμπούν στο TableView όταν το πρόγραμμα κλείσει και ξανα ανοίξει. Έχει την ίδια μορφή με το πάνω.

IncomeBook.txt: Εδώ αποθηκεύονται όλα τα οχήματα που έχουν πληρωθεί, προσθέτοντας στο τέλος και την ημερομηνία και την ώρα αποχώρησης.

Στοιχεία που πραγματοποιήθηκαν

Πραγματοποιήθηκαν όλα τα στοιχεία που ζητήθηκαν για τον Server με την μόνη διαφορά ότι το πρόγραμμα υποδοχής μου στέλνει το συνολικό κόστος και το τοποθετώ κατευθείαν στη κλάση Vehicle. Επιπρόσθετα, έχω προσθέσει και ένα search bar για την αναζήτηση οχημάτων στο TableView.

ΣΥΝΔΕΣΙΜΟΤΗΤΑ CLIENT - SERVER

Σε σχόλια στον κώδικα. Τα αρχεία είναι:

CLIENT → ClientFile.java

SERVER → Connection.java