



## Προγραμματιστική εργασία

### Προγράμματα για επιχείρηση πλυντηρίου αυτοκινήτων

7 Ιουνίου 2021

#### 1 Περιγραφή

Μία επιχείρηση πλυντηρίου αυτοκινήτων διαθέτει δύο υπολογιστές για τη διαχείριση των οχημάτων που εξυπηρετεί. Ο πρώτος υπολογιστής βρίσκεται στην «υποδοχή» του πλυντηρίου, και καταγράφει τα οχήματα που εισέρχονται καθώς και τις υπηρεσίες που ζητούν οι οδηγοί τους. Ο δεύτερος υπολογιστής βρίσκεται στο ταμείο· λαμβάνει τα δεδομένα των οχημάτων από τον υπολογιστή της υποδοχής, εκδίδει τις αποδείξεις πληρωμής και καταγράφει τα στοιχεία στο ηλεκτρονικό «βιβλίο εσόδων» της επιχείρησης. Οι δύο υπολογιστές επικοινωνούν μέσω του τοπικού δικτύου της επιχείρησης.

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα ζεύγος προγραμμάτων που να υλοποιούν τα παραπάνω, σύμφωνα με τις προδιαγραφές που ακολουθούν.

#### 2 Διεπαφές – Λειτουργικότητα

##### 2.1 Πρόγραμμα υποδοχής οχημάτων

Όπως αναφέρθηκε, το πρόγραμμα υποδοχής των οχημάτων αναλαμβάνει να καταχωρίσει τα στοιχεία κάθε οχήματος που εισέρχεται στο πλυντήριο καθώς και τις ζητούμενες υπηρεσίες, και να τα αποστέλλει στον υπολογιστή του ταμείου.

Η γραφική διεπαφή αυτού του προγράμματος θα πρέπει να υλοποιηθεί έτσι ώστε να μπορούσε να χρησιμοποιηθεί σε οθόνη αφής, χωρίς (φυσικό) πληκτρολόγιο και ποντίκι. Φροντίστε ώστε τα γραφικά στοιχεία να είναι αρκετά μεγάλα ή/και να απέχουν αρκετά μεταξύ τους ώστε να μπορούν να χρησιμοποιηθούν χωρίς τις δυνατότητες λεπτομερούς κατάδειξης του ποντικιού. Υποθέστε ελάχιστο μέγεθος του παραθύρου σας τα  $1024 \times 768$  εικονοστοιχεία και μέγιστο μέγεθος αυτού τα  $1920 \times 1080$  εικονοστοιχεία.

Η βασική οθόνη του προγράμματος θα επιτρέπει την **εισαγωγή του αριθμού κυκλοφορίας** του οχήματος. Θα περιέχει (τουλάχιστον) τα εξής:

- ένα **εικονικό πληκτρολόγιο**, δηλαδή πλήκτρα με τους χαρακτήρες των γραμμάτων και των αριθμών, τα οποία θα μπορεί να πατάει ο χρήστης για να εισαγάγει τον αριθμό κυκλοφορίας, και
- ένα **πεδίο κειμένου** (label, text field ή όποιο άλλο νομίζετε) στο οποίο θα εμφανίζεται ο αριθμός κυκλοφορίας καθώς τον πληκτρολογεί ο χρήστης (με τον τρόπο που εμφανίζονται οι αριθμοί σε ένα κομπιουτεράκι).

Το πληκτρολόγιο αρκεί να περιέχει τους **λατινικούς** (κεφαλαίους) χαρακτήρες (ακόμα και οι αριθμοί κυκλοφορίας στην Ελλάδα χρησιμοποιούν μόνο τους ελληνικούς χαρακτήρες που είναι οπτικά όμοιοι με λατινικούς), τα αριθμητικά **ψηφία**, πλήκτρο **διαστήματος** και πλήκτρο **διαγραφής** (με τη λειτουργία του Backspace). Τα πλήκτρα των χαρακτήρων και του διαστήματος θα είναι διατεταγμένα σε τέσσερις σειρές· για τα πλήκτρα των χαρακτήρων επιλέξτε

είτε σειριακή διάταξη (ABCDE...) είτε τη διάταξη του πληκτρολογίου (QWERTY). Ακόμα και αν χρησιμοποιήσετε διάταξη QWERTY, μπορείτε να διατάξετε τα πλήκτρα σε μορφή πλέγματος (ενώ στο (φυσικό) πληκτρολόγιο δεν είναι στοιχισμένα). Τα πλήκτρα των ψηφίων θα είναι τοποθετημένα δεξιά από τα πλήκτρα των γραμμάτων, σε διάταξη παρόμοια με του αριθμητικού πληκτρολογίου του υπολογιστή. Μπορείτε, τέλος, να προσθέσετε άλλα πλήκτρα που θεωρείτε χρήσιμα.

Για τον αριθμό κυκλοφορίας θα γίνονται μόνο απλοί **έλεγχοι εγκυρότητας**: να μην είναι κενός και να έχει τουλάχιστον δύο χαρακτήρες. Υποθέτουμε ότι μπορεί να έρθουν και οχήματα με ξένους αριθμούς κυκλοφορίας, και η δομή αυτών των αριθμών μπορεί να είναι πρακτικά οποιαδήποτε.

Μετά την εισαγωγή του αριθμού, σε νέα οθόνη (ή περισσότερες οθόνες, όπως προτιμάτε) ο χρήστης θα **εισάγει τις υπηρεσίες** που ζητούνται για το όχημα αυτό. Ένας ενδεικτικός κατάλογος των διαθέσιμων υπηρεσιών, με τις αντίστοιχες τιμές, είναι ο ακόλουθος:

	Αυτοκίνητο	Τζιπ	Μοτοσυκλέτα
Πλύσιμο εξωτερικό	7	8	6
Πλύσιμο εσωτερικό	6	7	–
Πλύσιμο εξωτ.+εσωτ.	12	14	–
Πλύσιμο εξωτ. σπέσιαλ	9	10	8
Πλύσιμο εσωτ. σπέσιαλ	8	9	–
Πλύσιμο εξωτ. + εσωτ. σπέσιαλ	15	17	–
Βιολογικός καθαρισμός εσωτ.	80	80	–
Κέρωμα-Γυάλισμα	80	90	40
Καθαρισμός κινητήρα	20	20	10
Πλύσιμο σασί	3	3	–

Παρατηρήστε ότι κάποιες υπηρεσίες μπορούν να συνδυαστούν μεταξύ τους (π.χ. πλύσιμο εξωτερικό και κέρωμα), ενώ άλλες όχι (π.χ. πλύσιμο εσωτερικό-εξωτερικό και πλύσιμο εξωτερικό, ή πλύσιμο αυτοκινήτου και πλύσιμο μοτοσυκλέτας). Θα πρέπει να προγραμματίσετε τη διεπαφή σας ώστε να **επιβάλλει τους αναγκαίους περιορισμούς** (π.χ. χρησιμοποιώντας radio buttons ή απενεργοποιώντας στοιχεία που δεν μπορούν να συνδυαστούν με τα επιλεγμένα).

Κατά την επιλογή των υπηρεσιών, το πρόγραμμα θα εμφανίζει στην οθόνη **το συνολικό κόστος τους**, για ενημέρωση του πελάτη.

Όταν ο χρήστης ολοκληρώσει και την επιλογή των υπηρεσιών, θα **επιβεβαιώνει** τις επιλογές του και το πρόγραμμα θα τις **στέλνει** στο πρόγραμμα του ταμείου. Δεν χρειάζεται να αποθηκεύονται πουθενά τα στοιχεία αυτά στο πρόγραμμα υποδοχής.

Τέλος, σε κάθε σημείο (οθόνη) του προγράμματος, ο χρήστης θα πρέπει να έχει τη δυνατότητα να **ακυρώσει** τις επιλογές του και να επιστρέψει στην αρχική οθόνη εισαγωγής του αριθμού κυκλοφορίας.

## 2.2 Πρόγραμμα ταμείου

Όπως αναφέρθηκε στην περιγραφή της εργασίας, το πρόγραμμα του ταμείου θα λαμβάνει τα στοιχεία των οχημάτων μέσω δικτύου από το πρόγραμμα της υποδοχής, και θα έχει δύο βασικές λειτουργίες:

- Την έκδοση αποδείξεων πληρωμής από τους πελάτες.
- Την καταχώριση των στοιχείων στο βιβλίο εσόδων της επιχείρησης.

Αυτή η εφαρμογή προβλέπεται να χρησιμοποιείται με πληκτρολόγιο και ποντίκι.

Με το που θα λαμβάνονται τα στοιχεία ενός οχήματος από το πρόγραμμα υποδοχής, θα **εγγράφονται** στο αρχείο του βιβλίου εσόδων μαζί με την (πραγματική) **ημερομηνία και ώρα άφιξης** (λήψης των στοιχείων στο ταμείο).<sup>1</sup>

Επίσης τα στοιχεία (τουλάχιστον αριθμός κυκλοφορίας, κόστος, ώρα άφιξης) θα προστίθενται σε έναν **πίνακα** στο **παράθυρο** της εφαρμογής.<sup>2</sup>

Όταν ο πελάτης έρχεται να πληρώσει, ο ταμίας θα επιλέγει τη γραμμή του πίνακα που αντιστοιχεί στο όχημα του πελάτη και πατώντας ένα κουμπί της οθόνης (ή το **Enter** στο πληκτρολόγιο) θα καταχωρίζει την πληρωμή. Για τους σκοπούς της εφαρμογής, το πρόγραμμα απλώς θα εμφανίζει ένα **παράθυρο μηνύματος** με τον αριθμό κυκλοφορίας, τις υπηρεσίες και το κόστος τους καθώς και το συνολικό κόστος, και θα επιτρέπει **καταχώριση ή όχι** της πληρωμής. Με την καταχώριση της πληρωμής, το πρόγραμμα θα αναζητεί στο αρχείο του βιβλίου εσόδων την εγγραφή του συγκεκριμένου οχήματος και θα **προσθέτει** σε αυτή την ώρα αναχώρησης και το ποσό πληρωμής. Αφού όλα ολοκληρωθούν επιτυχώς, προφανώς η εγγραφή του οχήματος θα **αφαιρείται** από τον πίνακα της οθόνης.

Θα πρέπει ακόμη να υπάρχει η δυνατότητα **ακύρωσης** κάποιου οχήματος: ο ταμίας επιλέγει τη γραμμή του στον πίνακα και πατάει κατάλληλο κουμπί ώστε να ακυρωθεί το συγκεκριμένο όχημα. Τότε αυτό αφαιρείται εντελώς από το αρχείο (με τρόπο που θα επιλέξετε).

Αν ο χρήστης προσπαθήσει να **κλείσει** το παράθυρο του προγράμματος ενώ υπάρχουν ακόμη οχήματα στον πίνακα, αυτό δεν θα επιτρέπεται και θα εμφανίζεται μήνυμα σφάλματος.

Ως επιπλέον στοιχείο μπορεί να υλοποιηθεί το εξής: δεδομένου ότι τα στοιχεία των οχημάτων αποθηκεύονται στο αρχείο με το που φτάνουν στο πρόγραμμα του ταμείου, κάθε φορά που **εκκινείται** το πρόγραμμα του ταμείου θα μπορούσε να **διαβάζει** το αρχείο και να γεμίζει τον πίνακα με τα στοιχεία των οχημάτων που τυχόν δεν έχουν αναχωρήσει. Έτσι θα υπάρχει μία σχετική προστασία από απροειδοποίητες διακοπές του προγράμματος (και θα μπορούσε να επιτρέπεται και το κλείσιμό του ακόμα και αν υπάρχουν ακόμη οχήματα στον πίνακα).

### 3 Υλοποίηση

#### 3.1 Βασικές αρχές της υλοποίησης

Σε προγράμματα όπως αυτά που ζητούνται εδώ, που διαχειρίζονται κάποια δεδομένα και τα παρουσιάζουν στον χρήστη μέσω γραφικής διεπαφής, είναι καλή πρακτική να διαχωρίζεται η υλοποίηση σε τρεις ομάδες (ή «επίπεδα») κλάσεων:

1. στις κλάσεις αναπαράστασης των δεδομένων,
2. στις κλάσεις υλοποίησης της λειτουργικότητας της εφαρμογής,
3. στις κλάσεις που αφορούν τη διεπαφή.

Οι κλάσεις σε κάθε επίπεδο πρέπει να είναι **ανεξάρτητες** από αυτές των *επόμενων* επιπέδων· έτσι, θεωρητικά, θα μπορούσε για παράδειγμα η λειτουργικότητα που έχει υλοποιηθεί στο ενδιάμεσο επίπεδο να προσφερθεί στο χρήστη μέσω μίας εντελώς διαφορετικής διεπαφής.

#### 3.2 Κοινά στοιχεία

Μία κλάση θα αποθηκεύει, σε κάποια δομή δεδομένων που θα αποφασίσετε εσείς, τα στοιχεία των διαθέσιμων υπηρεσιών της επιχείρησης (για κάθε υπηρεσία: κωδικός, ονομασία και κόστος). Η δομή αυτή θα πρέπει να χρησιμοποιείται και από τα δύο προγράμματα που θα αναπτύξετε.

<sup>1</sup> Αναζητήστε πώς μπορείτε να λάβετε την τρέχουσα ημερομηνία και ώρα χρησιμοποιώντας κλάσεις της βιβλιοθήκης της Java.

<sup>2</sup> Βρείτε στοιχεία για την κλάση `TableView` της `JavaFX`.

Η δομή αυτή μπορεί να γεμίζει μέσω κώδικα, με προκαθορισμένα και σταθερά στοιχεία όπως αυτά που δίνονται παραπάνω. Εναλλακτικά, τα στοιχεία θα μπορούσαν να είναι αποθηκευμένα σε κάποιο αρχείο και να διαβάζονται από αυτό.

Ο κωδικός κάθε υπηρεσίας πρέπει να είναι ένα μοναδικό αναγνωριστικό. Μπορεί να είναι είτε ένας απλός ακέραιος αριθμός είτε ένα αλφαριθμητικό ειδικής μορφής που θα σκεφτείτε.

### 3.3 Πρόγραμμα υποδοχής οχημάτων

Το πρόγραμμα υποδοχής οχημάτων θα διαρθρώνεται:

- σε μία κλάση που θα υλοποιεί τη λειτουργία του, και
- σε μία ή περισσότερες κλάσεις που θα υλοποιούν τη διεπαφή του.

Η κλάση υλοποίησης της λειτουργίας του προγράμματος θα έχει σκοπό να παίρνει τα στοιχεία από τη διεπαφή του προγράμματος (άρα θα χρησιμοποιεί τις σχετικές κλάσεις) και να τα στέλνει στο πρόγραμμα του ταμείου (το πρόγραμμα της υποδοχής θα είναι επομένως ο client και το πρόγραμμα του ταμείου θα είναι ο server σε αυτή την επικοινωνία). Θα πρέπει να φροντίσετε ώστε η κλάση αυτή να έχει αρκετές μεθόδους ώστε να μπορούσε να λειτουργήσει και αυτόνομα, ανεξάρτητα από τη διεπαφή που θα αναπτύξετε.

Το πρωτόκολλο της δικτυακής επικοινωνίας μεταξύ των δύο εφαρμογών θα το αποφασίσετε εσείς. Θα πρέπει να αποστέλλονται ο αριθμός κυκλοφορίας του οχήματος και οι υπηρεσίες που έχουν ζητηθεί. Ασφαλώς δεν είναι απαραίτητο να αποστέλλονται πληροφορίες που διαθέτει ο server (υπολογιστής του ταμείου), δηλαδή οι ονομασίες των υπηρεσιών και τα αντίστοιχα κόστη.

### 3.4 Πρόγραμμα ταμείου

Το πρόγραμμα του ταμείου θα πρέπει, παρόμοια με το πρόγραμμα υποδοχής, να διαρθρωθεί σε πολλαπλές κλάσεις. Εδώ μία ενδεικτική διάρθρωση των κλάσεων θα ήταν η εξής:

- Κλάση που διαχειρίζεται το βιβλίο εσόδων. Περιλαμβάνει λειτουργίες που αναφέρονται στις απαιτήσεις που έχουν περιγραφεί παραπάνω, όπως αρχική καταχώριση των στοιχείων όταν ληφθούν από το πρόγραμμα υποδοχής, ενημέρωση των στοιχείων κατά την πληρωμή, διαγραφή στοιχείων σε περίπτωση ακύρωσης.

Η μορφή του αρχείου στο οποίο θα φυλάσσονται τα στοιχεία μπορεί να είναι οποιαδήποτε· μπορείτε ακόμη και να χρησιμοποιήσετε περισσότερα από ένα αρχεία – το ζητούμενο είναι όλες οι πληροφορίες κάθε οχήματος (αριθμός, υπηρεσίες, ώρα εισόδου, ώρα εξόδου, ποσό πληρωμής) να αποθηκευτούν στο δίσκο και (θεωρητικά) να μπορούν να ανακτηθούν συγκροτημένα. Δεδομένων των απαιτήσεων για ενημέρωση των στοιχείων, μία εύχρηστη επιλογή είναι αυτή του **αρχείου τυχαίας προσπέλασης με εγγραφές σταθερού μήκους**.

- Κλάση που υλοποιεί την υπόλοιπη λειτουργικότητα: λήψη των στοιχείων από το πρόγραμμα υποδοχής, αποθήκευσή τους σε μια (προσωρινή) τοπική δομή δεδομένων (μέχρι να γίνει η πληρωμή και να διαγραφούν), κ.λπ. Αυτή η κλάση θα χρησιμοποιεί την προηγούμενη για τον χειρισμό του βιβλίου εσόδων και τις επόμενες για την εμφάνιση της διεπαφής.
- Κλάση ή κλάσεις για την υλοποίηση της διεπαφής.

## 4 Αξιολόγηση

Η εργασία θα αξιολογηθεί σε τρεις άξονες:

#### 4.1 Αξιολόγηση διεπαφής

- Πληρότητα (αν έχουν υλοποιηθεί όλα τα ζητούμενα σε καθένα από τα προγράμματα).
- Ευχρηστία (ως προς τις προδιαγραφές που έχουν τεθεί για καθένα από τα προγράμματα αλλά και γενικότερα ως προς βασικούς κανόνες ευχρηστίας).
- Χρήση `rapes` και γενικότερα συμπεριφορά των παραθύρων, π.χ. όταν αλλάζει το μέγεθός τους.
- Ποιότητα υλοποίησης (π.χ. τρόποι υλοποίησης event handlers).

#### 4.2 Αξιολόγηση υλοποίησης λοιπών κλάσεων

- Διαχωρισμός και ανεξαρτησία κλάσεων.
- Χρήση κατάλληλων δομών δεδομένων από την βιβλιοθήκη της Java για αποθήκευση των στοιχείων (η χρήση απλών πινάκων παντού δεν θα αξιολογηθεί θετικά).
- Αποτελεσματική οργάνωση των αρχείων.

#### 4.3 Γραπτή αναφορά

Η υλοποίηση της άσκησης θα πρέπει να συνοδεύεται από γραπτή αναφορά, η οποία θα περιλαμβάνει:

- Τα στοιχεία των φοιτητών που υλοποίησαν την άσκηση.
- Παρουσίαση της σχεδίασης των κλάσεων: διάρθρωση κλάσεων κάθε προγράμματος, πεδία και μέθοδοι κάθε κλάσης. Η παρουσίαση των πεδίων και μεθόδων των κλάσεων σε μορφή κειμένου μπορεί να αντικατασταθεί από τεκμηρίωσή τους σε μορφή Javadoc (η οποία δεν χρειάζεται να αντιγραφεί στην αναφορά).
- Περιγραφή των δομών αρχείων που χρησιμοποιήθηκαν.
- Περιγραφή όποιων στοιχείων της υλοποίησης δεν συμφωνούν με την εκφώνηση (είτε λείπουν από τα ζητούμενα είτε είναι επιπλέον αυτών).

#### Διαδικαστικά

- Η εργασία πρέπει να εκπονηθεί από **ομάδες δύο ατόμων**. Εργασίες από ένα άτομο θα γίνουν δεκτές, χωρίς όμως ευνοϊκότερη βαθμολόγηση.

**Σε καμία περίπτωση δεν θα γίνουν δεκτές ομάδες με περισσότερα από δύο μέλη.**

- Έχει δημιουργηθεί μία «εργασία» στο GitHub, στο URL:

<https://classroom.github.com/g/FQsgbeaB>

Ακολουθείτε την ίδια διαδικασία με τις προηγούμενες ασκήσεις για την υποβολή της εργασίας.

Δείτε παρακάτω κάποιες ειδικές οδηγίες για την συγκεκριμένη εργασία.

- Χρησιμοποιούνται αυτόματα οι ομάδες των προηγούμενων εργασιών. Αν υπάρχει σοβαρός λόγος αλλαγής ομάδας (π.χ. αποχώρηση ατόμου) παρακαλώ επικοινωνήστε μαζί μου.
- Για την παράδοση της εργασίας σας, απλώς θα κάνετε push στο GitHub το project σας.
- Μέσα στο project σας περιλάβετε και την **αναφορά** σας. Μπορεί να είναι ένα αρχείο Microsoft Word ή LibreOffice Writer ή PDF ή και Markdown.

- **Προθεσμία παράδοσης:** μετά το τέλος της εξεταστικής του Ιουνίου. Η ακριβής ημερομηνία θα ανακοινωθεί αργότερα.
- Θα πραγματοποιηθεί **προφορική εξέταση** της εργασίας, στην οποία θα πρέπει να παρευρίσκονται όλα τα μέλη της κάθε ομάδας. Αν κάποιος δεν εξεταστεί προφορικά, θα λάβει μηδενικό βαθμό για την εργασία. Ο βαθμός μπορεί να διαφέρει μεταξύ των μελών μίας ομάδας, ανάλογα με τις διαπιστώσεις της προφορικής εξέτασης.
- Οι εργασίες θα **ελεγχθούν** για αντιγραφές με αυτοματοποιημένο τρόπο. Εργασίες που θα θεωρηθούν αντιγραμμένες θα μηδενιστούν.
- Η εργασία θα μετρήσει με ποσοστό περίπου 60–70% στον τελικό βαθμό σας, και θα πρέπει να έχετε τουλάχιστον 4/10 σε αυτή για να περάσετε το μάθημα.
- Για οποιαδήποτε απορία γράψτε τις απορίες σας στην σχετική «Περιοχή συζητήσεων» στο e-class ώστε να επωφεληθούν όλοι οι συμφοιτητές σας. **Δεν** θα απαντηθούν απορίες μέσω email.

### Ειδικές οδηγίες για το project αυτής της εργασίας

Όπως και στις προηγούμενες ασκήσεις, παρέχεται με την εργασία αυτή ένας αρχικός κώδικας στο project που δημιουργείται για την ομάδα σας στο GitHub.

Ο κώδικας χωρίζεται σε δύο φακέλους, έναν για καθένα από τα προγράμματα που καλείστε να αναπτύξετε, και οι οποίοι φέρουν τα γενικά ονόματα `Server` και `Client`. Ο κάθε φάκελος δεν περιέχει τίποτε άλλο παρά ένα Maven project για εφαρμογή JavaFX, όμοιο με αυτό που είχατε χρησιμοποιήσει σε προηγούμενες ασκήσεις, και επομένως μπορεί να μεταγλωττιστεί και να τρέξει χωριστά.

Όμως, λόγω περιορισμών του Visual Studio Code, φαίνεται ότι μόνο **ένα** Maven project μπορεί να εκτελείται κάθε φορά μέσα από το περιβάλλον του. Επομένως, για να εργαστείτε αποδοτικά με τον κώδικα, προτείνεται να ανοίξετε καθέναν από τους δύο φακέλους **χωριστά**, τρέχοντας δύο φορές το Visual Studio Code.

Σε σχέση με το git, όλος ο κώδικας (και οι δύο φάκελοι) βρίσκεται σε ένα (κοινό) αποθετήριο. Εφόσον εργαστείτε σε δύο «αντίγραφα» του Visual Studio Code θα μπορείτε σε καθένα από αυτά να δείτε **όλες** τις αλλαγές (όχι μόνο αυτές που αφορούν τον φάκελο στον οποίο εργάζεστε), να κάνετε commit ό,τι θέλετε, κ.λπ. Αυτό συμβαίνει καθώς, γενικότερα, το git αν δεν βρει αποθετήριο (φάκελο `.git`) ακριβώς μέσα στον φάκελο στον οποίο εργαζόμαστε, ψάχνει ένα επίπεδο παραπάνω κ.ο.κ. μέχρι να βρει κάπου.

Αν τυχόν ανακαλύψετε καλύτερη λύση, μπορείτε να την μοιραστείτε μέσω της Συζήτησης της εργασίας.