

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Χειμερινό εξάμηνο ακ. έτους 2017-2018

Α' ομάδα ασκήσεων

Άσκηση 1: ¹ Η παρούσα άσκηση σκοπό έχει να σας εξασκήσει στις συναρτήσεις κατασκευής-καταστροφής στην περίπτωση που έχουμε σύνθεση κλάσεων και στην προσβασιμότητα και εμβέλεια μεταξύ των μελών των κλάσεων.

Υλοποιήστε σε C++ την προσομοίωση ενός τρένου του μετρό. Το τρένο έχει *totalWaggon*s βαγόνια. Κάθε βαγόνι μπορεί να έχει το πολύ *maxCapacity* επιβάτες. Κάθε επιβάτης έχει ή δεν έχει εισιτήριο, και μπορεί να δικαιούται μειωμένο εισιτήριο ή όχι. Το τρένο λειτουργεί για *N* σταθμούς. Σε κάθε σταθμό αποβιβάζονται ή/και επιβιβάζονται επιβάτες. Ανάμεσα σε κάθε σταθμό, γίνονται έλεγχοι εισιτηρίων σε κάποια από τα βαγόνια, και συλλέγονται τα χρήματα από τους παραβάτες.

Το τρένο αρχικά δεν έχει συλλέξει καθόλου χρήματα, και εκχωρείται σε αυτό το πλήθος των σταθμών, *N*, για τους οποίους θα λειτουργήσει. Όταν δημιουργείται ένα τρένο εκτυπώνεται το μήνυμα "A metro train with <totalWaggon> waggon, was created". Όταν καταστρέφεται το τρένο εκτυπώνεται το μήνυμα "A metro train was destroyed". Το τρένο λειτουργεί (*operate*) για *N* σταθμούς. Σε κάθε σταθμό γίνεται η διαδικασία της επιβίβασης / αποβίβασης για κάθε βαγόνι, και ενδιάμεσα στους σταθμούς γίνεται η διαδικασία του ελέγχου εισιτηρίων για κάποια βαγόνια. Τέλος, στο τρένο εκτυπώνονται στατιστικά (*printStatistics*) σχετικά με τη λειτουργία του καθώς και των βαγονιών του. Στα στατιστικά του τρένου, περιλαμβάνονται τα χρήματα που συλλέχθηκαν κατά τη διάρκεια των ελέγχων εισιτηρίων.

Ένα βαγόνι αρχικά δεν έχει κανένα επιβάτη. Όταν δημιουργείται ένα βαγόνι, εκτυπώνεται το μήνυμα "A waggon with capacity for <maxCapacity> passengers, was created". Όταν καταστρέφεται ένα βαγόνι, εκτυπώνεται το μήνυμα "A waggon was destroyed". Ένα βαγόνι λειτουργεί σε κάθε σταθμό (*inStation*) αποβιβάζοντας και επιβιβάζοντας τυχαίους αριθμούς επιβατών. Κατά την επιβίβαση ενός επιβάτη τον προσθέτει στους επιβάτες του και αυξάνει το αριθμό των επιβατών του κατά ένα. Κατά την αποβίβαση ενός επιβάτη γίνεται η αντίστροφη διαδικασία. Στο στάδιο αυτό επιλέγεται τυχαία και αν το βαγόνι θα υποστεί έλεγχο εισιτηρίων. Ένα βαγόνι λειτουργεί ενδιάμεσα στους σταθμούς (*betweenStations*) και αν έχει επιλεγεί για έλεγχο εισιτηρίων, γίνεται έλεγχος σε κάθε επιβάτη του. Οι παραβάτες αποβιβάζονται από το τρένο. Για κάθε βαγόνι εκτυπώνονται στατιστικά (*printStatistics*) σχετικά με την λειτουργία του. Στα στατιστικά του βαγονιού, περιλαμβάνονται πόσοι επιβάτες βρέθηκαν να παραβατούν, καθώς και πόσοι, ενώ δεν είχαν εισιτήριο, απέφυγαν τυχόν έλεγχο.

Τέλος, σε κάθε επιβάτη επιλέγεται τυχαία κατά την αρχικοποίηση του, αν θα έχει εισιτήριο ή όχι και αν δικαιούται μειωμένο εισιτήριο.

Υλοποιήστε τα παραπάνω μέσω των κατάλληλων κλάσεων, ορίζοντας τα μέλη-δεδομένα που χρειάζονται καθώς και τις συναρτήσεις-μέλη που αναφέρονται μαζί με ό,τι άλλο θεωρήσετε απαραίτητο για να υλοποιηθεί η παραπάνω συμπεριφορά.

Υλοποιήστε επίσης μια συνάρτηση *main*, στην οποία:

1. Θα δημιουργείται ένα τρένο του μετρό που θα λειτουργήσει για *N* σταθμούς.
2. Θα λειτουργεί το τρένο.
3. Θα εκτυπώνονται στατιστικά σχετικά με την λειτουργία του.

¹Για την ιδέα και τη σύνταξη της άσκησης αυτής, ευχαριστώ θερμά τον Γιώργο Καστρίνη

Το N θα δίνεται από την γραμμή εντολών.

Άσκηση 2:² Να υλοποιήσετε σε C++ μια προσομοίωση αεροδρομίου. Στο αεροδρόμιο εκτελούνται κάποιες πτήσεις (flights) με συγκεκριμένους προορισμούς και συγκεκριμένες ώρες αναχώρησης. Οι άνθρωποι που θέλουν να ταξιδέψουν κάνουν μια αίτηση (application) στην οποία περιγράφουν πού θέλουν να πάνε και τι ώρα θέλουν να έχουν φτάσει. Έπειτα στο αεροδρόμιο αποφασίζεται σε ποια πτήση θα επιβιβαστούν. Καθώς ο χρόνος κυλά, πτήσεις αναχωρούν και αιτήσεις μένουν ανικανοποίητες.

Μια αίτηση περιλαμβάνει το id του επιβάτη, το όνομά του, επώνυμο, προορισμό, χρόνο που θέλει για να φτάσει στο αεροδρόμιο (δεν μπορεί να επιβιβαστεί σε αεροπλάνο με χρόνο αναχώρησης πριν από αυτόν), χρόνο που θέλει να έχει φτάσει στον προορισμό του (δεν μπορεί να επιβιβαστεί σε αεροπλάνο που φτάνει μετά από αυτόν τον χρόνο) και αν θέλει να είναι στην οικονομική θέση του αεροπλάνου ή όχι (θέσεις A και B). Μια αίτηση αρχικοποιείται με όλες τις παραπάνω παραμέτρους. Μπορούμε να ελέγξουμε αν μια πτήση πληρεί τους περιορισμούς μιας αίτησης (matches). Μπορούμε να ελέγξουμε αν μια αίτηση ζητά δέσμευση A θέσης (is_luxury_class). Μπορούμε να ανακτήσουμε την ταυτότητα του επιβάτη που έκανε την αίτηση (get_id). Μπορούμε να ανακτήσουμε το όνομα και το επώνυμο του επιβάτη που έκανε την αίτηση (σαν συμβολοσειρά, χωρισμένα με ένα κενό) (get_name). Μπορούμε να ανακτήσουμε το όριο χρόνου στο οποίο ο επιβάτης επιθυμεί να έχει φτάσει στον προορισμό του (arrived_by).

Μια πτήση αποτελείται από τον προορισμό της, την ώρα αναχώρησης από το αεροδρόμιο, την διάρκεια της πτήσης και το μέγιστο αριθμό επιβατών σε κάθε θέση (θέσεις A και B). Μια πτήση αρχικοποιείται με όλες τις παραπάνω παραμέτρους. Μια πτήση δέχεται μια αίτηση και αν ο αντίστοιχος επιβάτης χωράει στο αεροπλάνο, τον προσθέτει (σαν κλεισμένη θέση) (add_passenger). Δίδοντας ένα id επιβάτη, μπορούμε να ακυρώσουμε όλες τις κλεισμένες θέσεις για αιτήσεις με το συγκεκριμένο id (cancel_reservations). Μπορούμε να ανακτήσουμε την ώρα αναχώρησης της πτήσης (departs_at). Μπορούμε να ανακτήσουμε την ώρα άφιξης της πτήσης στον προορισμό της (arrives_at). Μπορούμε να ανακτήσουμε τον προορισμό της πτήσης (get_destination). Μπορούμε να ανακτήσουμε τον αριθμό των διαθέσιμων θέσεων της πτήσης (ανεξαρτήτως θέσης A ή B) (get_available). Μπορούμε να ανακτήσουμε μια δομή με τις αιτήσεις που έχουν κλειστεί στην συγκεκριμένη πτήση (get_bookings).

Το αεροδρόμιο δουλεύει πάνω σε συγκεκριμένο χρονοδιάγραμμα και έχει σαν μια ενεργή παράμετρο την τωρινή ώρα. Τόσο οι πτήσεις όσο και οι αιτήσεις δεν υφίστανται ανεξάρτητα από το αεροδρόμιο. Ο χρόνος 'κυλά' και στο πέρασμά του συμβαίνουν γεγονότα που περιγράφουμε στη συνέχεια. Όταν γίνεται μια αίτηση στο αεροδρόμιο, τότε προσπαθούμε να την καλύψουμε με διαθέσιμη πτήση, αλλιώς μπαίνει σε λίστα αναμονής (μέχρι να οριστεί πτήση που να ταιριάζει με την αίτηση). Όμως, όταν ο τωρινός χρόνος περάσει τον χρόνο άφιξης που ορίστηκε στην αίτηση, η αίτηση ακυρώνεται, διαγράφοντάς την από τη λίστα αναμονής. Το αεροδρόμιο έχει συγκεκριμένο αριθμό από σταθμούς (terminals) και κάθε πτήση καταλαμβάνει απευθείας έναν σταθμό. Αν δεν υπάρχει διαθέσιμος, δεν μπορεί να προστεθεί η πτήση. Επίσης όταν ακυρώνεται μια πτήση, οι αιτήσεις μπαίνουν στην λίστα αναμονής (στην αρχή της). Οι αιτήσεις μένουν στην λίστα αναμονής μέχρι να βρεθεί πτήση που να ταιριάζει ή μέχρι να ακυρωθούν. Ένα αεροδρόμιο αρχικοποιείται με παραμέτρους τον τωρινό χρόνο και τον αριθμό των σταθμών του. Μπορούμε να ανακτήσουμε τον τωρινό χρόνο (get_time). Μπορούμε να δώσουμε όλες τις παραμέτρους μιας αίτησης και να προσθέσουμε την αίτηση στη λίστα αναμονής των αιτήσεων (add_application). Μπορούμε να δώσουμε ένα id και να διαγράψουμε όλες τις αιτήσεις με το συγκεκριμένο id (cancel_applications). Αυτές μπορεί να βρίσκονται στη λίστα αναμονής ή να έχουν προσθέσει επιβάτες σε πτήσεις. Στη δεύτερη περίπτωση, θα ελευθερώσουμε τις θέσεις

²Για την ιδέα και τη σύνταξη της άσκησης αυτής, ευχαριστώ θερμά τον Ορέστη Πολυχρονίου

στις πτήσεις. Μπορούμε να δώσουμε τις παραμέτρους μιας πτήσης και να ορίσουμε μια νέα πτήση και να ανακτήσουμε τον αριθμό του σταθμού στον οποίο βρίσκεται (`add_flight`). Υπάρχει μια διαδικασία ακύρωσης πτήσης (`cancel_flight`). Αυτή παίρνει σαν όρισμα τον αριθμό του σταθμού στον οποίο βρίσκεται η πτήση και τη διαγράφει. Οι κρατήσεις της πτήσης που ακυρώθηκαν εισάγονται στην αρχή της λίστας αναμονής των αιτήσεων. Μπορούμε να ανακτήσουμε τον αριθμό των ακυρωθέντων αιτήσεων ως τώρα συνολικά (`failed_applications`). Μπορούμε να εκτυπώσουμε τις διαθέσιμες πτήσεις καθώς και τις πληροφορίες της κάθε πτήσης, με σειρά ώρας αναχώρησης (`show_timetable`). Μπορούμε να εκτυπώσουμε τα ονοματεπώνυμα αυτών που οι αιτήσεις τους βρίσκονται στην λίστα αναμονής (`show_people_waiting`). Η διαδικασία `flow_time`, με όρισμα το χρονικό διάστημα που θέλουμε να περάσει, 'κινεί' το χρόνο. Κατά την διαδικασία αυτή, κάθε πτήση που είναι να αναχωρήσει εντός αυτού του διαστήματος ελευθερώνει τον σταθμό που καταλαμβάνει και τα ονοματεπώνυμα των επιβατών που επιβαίνουν καθώς και ο προορισμός της εκτυπώνονται. Επίσης, αιτήσεις που αναφέρονται στο διάστημα αυτό και δεν ικανοποιήθηκαν διαγράφονται από τη λίστα αναμονής (ακυρώνονται).

Υλοποιήστε τα παραπάνω, μέσω των κατάλληλων κλάσεων, ορίζοντας τα μέλη-δεδομένα που χρειάζονται καθώς και τις συναρτήσεις-μέλη που αναφέρονται μαζί με ό,τι άλλο θεωρήσετε απαραίτητο για να υλοποιηθεί η παραπάνω συμπεριφορά. Για την λίστα αναμονής μπορείτε είτε να την ορίσετε εσείς είτε να χρησιμοποιήσετε τύπο βιβλιοθήκης.

Υλοποιήστε επίσης μια συνάρτηση `main`, στην οποία:

1. Θα δημιουργείται ένα αεροδρόμιο
2. Θα δίδονται στο αεροδρόμιο στοιχεία πτήσεων.
3. Θα δίδονται στο αεροδρόμιο στοιχεία αιτήσεων.
4. Θα γίνονται σχετικές εκτυπώσεις ώστε να παρουσιάζεται η κατάσταση του αεροδρομίου.
5. Θα κυλά το χρόνο κάποιο χρονικό διάστημα
6. Θα γίνονται σχετικές εκτυπώσεις ώστε να παρουσιάζεται η κατάσταση του αεροδρομίου.

Το χρονικό διάστημα να δίδεται από τη γραμμή εντολής. Τα στοιχεία των πτήσεων και των αιτήσεων να διαβάζονται από αρχεία.

Σημείωση: Για την υλοποίηση του χρόνου να χρησιμοποιήσετε τον τύπο `time_t` οι δηλώσεις του οποίου βρίσκονται στο αρχείο-επικεφαλίδα `ctime`.