Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Εργασία για το μάθημα "Δομές Δεδομένων"

Σύστημα Παρακολούθησης Ενεργών Πτήσεων

Δρόσου Ειρήνη, Π/06047
<eirinidrosou@gmail.com>

Κουζούπης Αντώνης, Π/06073
<kouzoupis.ant@gmail.com>

Πειραιάς, Δευτέρα 11 Ιουλίου 2011

Περιεχόμενα

1 Εισαγωγή

Η εφαρμογή υλοποιεί ένα απλοποιημένο σύστημα προβολής και διαχείρισης των ενεργών πτήσεων. Υλοποιήθηκε στα πλαίσια της δεύτερης εργασίας του μαθήματος *Δομές Δεδομένων*. Υλοποιεί κάποιες λειτουργίες πάνω σε δυαδικό δένδρο αναζήτησης. Η εφαρμογή είναι γραμμένη στη γλώσσα προγραμματισμού Java και οι δομές που χρησιμοποιήθηκαν δεν είναι οι built-in κλάσεις της γλώσσας αλλά υλοποιημένες από εμάς. Δυστυχώς λόγω ακαδημαϊκών υποχρεώσεων δεν προλάβαμε να υλοποιήσουμε το τέταρτο μέρος της εργασίας. Η εφαρμογή δεν χρησιμοποιεί κάποιο είδος persistance καθώς δεν χρειαζόταν από την εργασία.

2 Σχεδίαση

Η εφαρμογή είναι χωρισμένη σε τρία βασικά λειτουργικά κομμάτια. Τα τρία directories–packages της εφαρμογής είναι το business, το entities και το structure. Κάθε ένα package περιέχει κώδικα "ανεξάρτητο" από τα άλλα packages. Παρόλα αυτά χρειάζονται και τα τρία για να λειτουργήσει η εφαρμογή. Το πρώτο πακέτο, το business, περιέχει τον πηγαίο κώδικα για το business logic. Τυπώνει το μενού, συλλέγει τις πληροφορίες από το χρήστης, μετατρέπει τις raw πληροφορίες σε μορφή χρήσιμη και καλεί τις κατάλληλες μεθόδους για λειτουργίες πάνω στο δυαδικό δένδρο αναζήτησης. Στο δεύτερο πακέτο, το entities, υπάρχει ο κώδικας για την υλοποίηση των οντοτήτων του συστήματος. Στη συγκεκριμένη περίπτωση είναι οι πτήσεις. Τέλος στο πακέτο structures υπάρχουν οι κλάσεις των δομών δεδομένων που χρησιμοποιήσαμε οι οποίες προφανώς υλοποιούν τις δομές αλλά και κάποιες λειτουργίες πάνω σε αυτές.

Αναλυτικές πληροφορίες υπάρχουν και στο *JavaDoc* που συνοδεύει την εργασία αλλά και στον πηγαίο κώδικα.

2.1 business package

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις *Business*, *Main* και *Printer*.

Η κλάση *Business* περιέχει τρεις βασικές μεθόδους. Η μία είναι για να φορτώνει κάποιες demo πτήσεις στο σύστημα, η δεύτερη προετοιμάζει τα raw δεδομένα που εισάγει ο χρήστης και καλεί τις κατάλληλες μεθόδους για την προσθήκη μιας νέας πτήσης και τέλος η τρίτη μέθοδος χρησιμοποιείται για την αναζήτηση των ενεργών πτήσεων σε ένα συγκεκριμένο χρονικό διάστημα.

Η κλάση *Main* είναι το entry point της εφαρμογής. Μέσω μεθόδων της κλάσης *Printer* τυπώνει το κεντρικό μενού, τα υπο–μενού και συλλέγει τις πληροφορίες από τον χρήστη.

Τελευταία είναι η κλάση *Printer* που αποτελείται από μεθόδους που προορίζονται αποκλειστικά στην εκτύπωση των διάφορων μενού και "ερωτήσεων" προς το χρήστη.

2.2 entities package

Το πακέτο *entities* περιέχει την κλάση *Flights*.

Η κλάση *Flights* κρατάει όλες τις λεπτομέρειες για μία πτήση. Ο πληροφορίες που κρατάει είναι ο κωδικός πτήσης, η ώρα αναχώρησης και η ώρα άφιξης. Ουσιαστικά ένα instance της κλάσης αυτής αντιπροσωπεύει μία πτήση. Επίσης έχει διάφορες άλλες μεθόδους για να θέτουμε και να διαβάζουμε τις τιμές των παραπάνω μεταβλητών (getters & setters). Τέλος υπάρχει μία μέθοδος για την εκτύπωση των λεπτομερειών μιας πτήσης.

2.3 structure package

Το πακέτο αυτό περιέχει τις κλάσεις *BinarySearchTree* και *SimplyLinkedList*.

Η κλάση *BinarySearchTree* υλοποιεί αρχικά ένα Δυαδικό Δένδρο Αναζήτησης καθώς και κάποιες επιπλέον λειτουργίες. Κάθε κόμβος του δένδρου αποθηκεύει instances της κλάσης *Flights*, δηλαδή αποθηκεύει πτήσεις. Κάθε κόμβος επίσης έχει ένα πεδίο **MaxArr** το οποίο κρατάει τη μέγιστη ώρα άφιξης του υποδένδρου του. Στην κλάση αυτή υπάρχει και η κλάση *Node* που αντιπροσωπεύει τους κόμβους του δένδρου. Κρατάει πληροφορίες για τον αριστερό κόμβο–παιδί, τον δεξιό κόμβο–παιδί, το πεδίο **MaxArr** και φυσικά τα δεδομένα. Έχει μεθόδους για την εισαγωγή, διαγραφή, αναζήτηση ενός κόμβου και εκτύπωση σε ενδοδιάταξη του δένδρου.

Η κλάση *SimplyLinkedList* υλοποιεί μία μονά συνδεδεμένη λίστα που επίσης χρησιμοποιεί Generics. Αποτελείται και αυτή από δύο κλάσεις, η μία είναι η *SNode* που κρατάει διάφορες πληροφορίες για ένα κόμβο όπως τη τιμή του κόμβου και τον επόμενο κόμβο. Έχει επίσης κάποιες μεθόδους για την προσπέλαση των στοιχείων αυτών. Η κλάση *SimplyLinkedList* υλοποιεί κάποιες λειτουργίες της μονά συνδεδεμένης λίστας όπως την προσθήκη ενός κόμβου είτε στην αρχή, είτε στο τέλος είτε κάπου ενδιάμεσα, τη διαγραφή ενός κόμβου, μέθοδο για την παίρνουμε το μέγεθος της λίστας, μέθοδο για να βλέπουμε αν η λίστα είναι άδεια και μία μέθοδο που τυπώνει τις τιμές όλων των κόμβων της λίστας. Η λίστα χρησιμοποιείται ως στοίβα (stack) για τις λειτουργίες της εισαγωγής και διαγραφής.

2.4 Δομή Εφαρμογής

Όπως αναφέραμε η εφαρμογή κρατάει κάποιες πτήσεις σε μία δομή δυαδικού δένδρου αναζήτησης και εκτελούμε κάποιες λειτουργίες σε αυτές. Μία πτήση χαρακτηρίζεται από τον κωδικό πτήσης, την ημερομηνία αναχώρησης και την ημερομηνία άφιξης. Στο δένδρο η ταξινόμηση γίνεται με βάση την ημερομηνία αναχώρησης. Επίσης κάθε κόμβος στο δένδρο έχει ένα ειδικό field με όνομα **MaxArr**. Αυτό το πεδίο κρατάει την μέγιστη ημερομηνία άφιξης του δένδρου κάτω από αυτό, δηλαδή του υποδένδρου του. Αυτό το πεδίο βοηθάει για διάφορες αναζητήσεις ώστε να μπορούμε να αποφασίζουμε να μην επισκεπτόμαστε κάποια υποδένδρα. Με αυτό τον τρόπο γλιτώνουμε συγκρίσεις. Προφανώς όταν γίνεται κάποια εισαγωγή ή διαγραφή, οι κόμβοι του δένδρου θα πρέπει να επανατοποθετηθούν στη σωστή θέση και να υπολογιστεί ξανά το πεδίο **MaxArr**.

3 Επεξήγηση Λειτουργίας

Η εφαρμογή είναι γραμμένη σε Java οπότε για να τρέξει ή να γίνει compile χρειαζόμαστε το Java Runtime Enviroment. Τα εκτελέσιμα είναι αρχειοθετημένα σε ένα archive τύπου jar για ποιο εύκολη εκτέλεση. Για να ξεκινήσουμε την εφαρμογή αρκεί να πάμε στον κατάλογο που είναι το αρχείο *DataStruct2.jar* και να εκτελέσουμε *java -jar DataStruct2.jar*. Αυτό θα μας εμφανίσει το κεντρικό μενού της εφαρμογής απ' όπου μπορούμε να κάνουμε τις επιλογές μας. Το κεντρικό μενού θα εμφανίζεται μετά από κάθε λειτουργία εκτός αν επιλέγουμε την Έξοδο από την εφαρμογή. Το entry point του συστήματος είναι η κλάση Main που φορτώνει κάποιες demo πτήσεις.

3.1 Add Flight

Η πρώτη επιλογή είναι για τη προσθήκη μιας πτήσης στην εφαρμογή. Επιλέγοντας λοιπόν την πρώτη επιλογή, μας εμφανίζεται ένα prompt για να εισάγουμε τα χαρακτηριστικά της πτήσης. Αφού τα εισάγουμε, η ημερομηνία αναχώρησης και ημερομηνία άφιξης μετατρέπονται σε χρόνο EPOCH. Έπειτα δημιουργείται ένα νέο instance της κλάσης *Flights* και καλείται η μέθοδος *add* της κλάσης *BinarySearchTree*. Αυτή η μέθοδος αποφασίζει ποιο είναι το σωστό σημείο στο δένδρο για να μπει και ρυθμίζει κατάλληλα τους δείκτες προς το αριστερό και δεξιό παιδί. Σε ένα stack κρατάει τους κόμβους που έκανε traverse μέχρι να βρει τη σωστή θέση ώστε μετά να ελέγξει αν χρειάζεται επανυπολογισμός του πεδίου **MaxArr** σε κάθε ένα κόμβο από το stack.

3.2 List Flights

Δεύτερη επιλογή είναι για την προβολή όλων των πτήσεων. Η επιλογή αυτή καλεί τη μέθοδο *toString* της κλάσης *BinarySearchTree* και αφού μετατρέψει το δένδρο σε μία λίστα με ενδοδιάταξη, τυπώνει όλα τα περιεχόμενα.

3.3 Delete Flight

Με την επιλογή *Delete Flight* μπορεί ο χρήστης να διαγράψει μία πτήση από το σύστημα. Προφανώς αφού διαγραφεί ένας κόμβος, το δυαδικό δένδρο αναζήτησης θα πρέπει να έχει τη σωστή δομή και το πεδίο **MaxArr** να έχει τη σωστή τιμή σε όλους τους κόμβους. Αρχικά ο χρήστης δίνει ένα κωδικό πτήσης, καλείται η μέθοδος *delFlight* της κλάσης *BinarySearchTree* όπου βρίσκει σε ποιο κόμβο αντιστοιχεί ο συγκεκριμένος κωδικός πτήσης. Έπειτα καλείται η μέθοδος *remove* της ίδιας κλάσης όπου διαγράφει τον κόμβο και αποφασίζει ποιος κόμβος θα πάρει τη θέση του στο δένδρο. Επίσης κρατάει σε μία λίστα όλους τους κόμβους που έχουν γίνει traversed μέχρι τον κόμβο προς διαγραφεί έτσι ώστε μετά να ελέγξει αν χρειάζεται αλλαγή το πεδίο **MaxArr**.

3.4 Search within a period

Επόμενη επιλογή είναι η αναζήτηση πτήσεων σε εξέλιξη σε ένα συγκεκριμένο χρονικό διάστημα. Ο χρήστης δίνει την αρχή και το τέλος του χρονικού διαστήματος και επιστρέφονται οι πτήσεις που άρχισαν στο διάστημα αυτό ή ολοκληρώθηκαν στο διάστημα αυτό. Αφού ο χρήστης δώσει την αρχή και το τέλος καλείται η μέθοδος *searchFlight* της κλάσης *Business*, μετατρέπει τις ημερομηνίες σε χρόνους EPOCH και καλεί τη μέθοδο *searchPeriod* της κλάσης *BinarySearchTree*. Η μέθοδος αυτή επιστρέφει μία λίστα με όλες τις πτήσεις που είναι σε εξέλιξη στη συγκεκριμένη χρονική περίοδο και τελικά τις τυπώνει.

3.5 Exit

Τελευταία επιλογή είναι η *Exit* με την οποία μπορούμε να τερματίσουμε και να βγούμε από την εφαρμογή.

4 Περιεχόμενα CD

Στο CD που συνοδεύει το παρόν εγχειρίδιο υπάρχουν τα παρακάτω:

src Ο πηγαίος κώδικας της εφαρμογής. Περιέχει τρεις καταλόγους που αντιστοιχούν στα τρία πακέτα. Το κάθε πακέτο έχει τον κώδικα από τις κλάσεις του.

doc Το Javadoc της εφαρμογής. Περιγραφή της λειτουργίας κάθε μεθόδου της εφαρμογής. Μέσα στον κατάλογο αυτό υπάρχει το αρχείο *index.html* το οποίο το ανοίγετε με ένα browser. (Σε ορισμένους browsers ίσως χρειαστεί να θέσετε με το χέρι την κωδικοποίηση σε UTF-8)

Documentation.pdf Το παρόν εγχειρίδιο.

DataStruct2.jar jar archive το οποίο περιέχει τον εκτελέσιμο κώδικα.

README Αρχείο με πληροφορίες για την εκτέλεση της εφαρμογής.