

ΗΛΙΑΣ ΣΕΤΤΑΣ 3150156

ΑΛΕΞΑΝΔΡΟΣ ΤΖΟΒΑΡΑΣ 3150171

ΜΕΡΟΣ Α

EmptyListException.java: Περιέχει πιθανά σφάλματα για την λίστα.

ListNode.java: Κατασκευαστής της λίστας.

List.java: Αρχείο για την κατασκευή λίστας και μεθόδων εισαγωγής και εξαγωγής στοιχείων σε αυτήν.

Τα παραπάνω αρχείων προέρχονται από το εργαστήριο 2 με μικρές αλλαγές ώστε να ανταποκρίνονται σε αντικείμενα τύπου PrintJob.

PrintJob.java: Κατασκευαστής αντικειμένων τύπου PrintJob. Περιέχει τους απαραίτητους getters και setters για τις μεταβλητές του, καθώς και την μέθοδο compareTo για σύγκριση αντικειμένων με βάση την προτεραιότητά τους.

MaxPQ.java: Κατασκευαστής ουράς προτεραιότητας η οποία δέχεται αντικείμενα τύπου PrintJob και τα κατατάσει με βάση την προτεραιότητά που θα τους δοθεί κατά την εισαγωγή τους σε αυτήν.

ΜΕΡΟΣ Β

inputB.txt: Εισαγωγή των στοιχείων. Μπορεί να μην χρησιμοποιηθεί.

outputB.txt: Εξαγωγή των στοιχείων. Χρησιμοποιείται από τον αλγόριθμο.

AlgorithmB.java: Αρχικά ο αλγόριθμος αποτελείται από 3 μεθόδους. Δύο μέθοδοι χρησιμοποιούνται για την ανάγνωση των στοιχείων από κείμενο και την εγγραφή τους σε κείμενο. Και οι δύο καλούνται από τη μέθοδο runB. Αρχικά ζητείται από το χρήστη να εισάγει το όνομα του αρχείου για ανάγνωση. Ο αλγόριθμος διαβάσει από το αρχείο και δημιουργεί τα απαραίτητα αντικείμενα PrintJob τα οποία στη συνέχεια αποθηκεύονται σε μια λίστα. Η λίστα χρησιμοποιείται μέχρι το τέλος του αλγορίθμου για την εγγραφή του αποτελέσματος στο αρχείο εξόδου. Η εκφώνηση δεν ζητούσε την εισαγωγή του αρχείου εξόδου από το χρήστη και έτσι φτιάξαμε το outputB.txt. Τα αντικείμενα μεταφέρονται όταν έρθει η χρονική στιγμή από τη λίστα στην ουρά και στην συνέχεια όταν εκτελεστούν επιστρέφονται στη λίστα έχοντας ανανεωμένο χρόνο αναμονής. Η προτεραιότητά τους υπολογίζεται ανάλογα με το μέγεθος τους.

Για παραπάνω πληροφορίες ελέγξτε τον κώδικα και δείτε τα σχόλια στην αρχή του.

ΜΕΡΟΣ Γ

inputC.txt: Εισαγωγή των στοιχείων. Μπορεί να μην χρησιμοποιηθεί.

outputC.txt: Εξαγωγή των στοιχείων. Χρησιμοποιείται από τον αλγόριθμο.

AlgorithmC.java: Η δομή του είναι στο μεγαλύτερο μέρος του παρόμοια με αυτή του B. Όμως κατά την εκτέλεση του runB κάθε 15 δευτερόλεπτα ανανεώνεται η προτεραιότητα των αντικειμένων λαμβάνοντας υπόψη το χρόνο αναμονής τους. Στην πραγματικότητα οι αλλαγές δεν γίνονται κάθε 15 δευτερόλεπτα αλλά κάθε φορά που ολοκληρώνεται ένα αντικείμενο. Με τη χρήση if και loop βρίσκεται η νέα προτεραιότητα κάθε αντικειμένου. Τα αντικείμενα της ουράς μεταφέρονται όλα στην λίστα. Κατά τη μεταφορά γίνονται οι αλλαγές και στη συνέχεια επιστρέφονται στην ουρά. Γίνεται η ανακατάταξή τους και διαλέγεται το 1^ο ως επόμενο αντικείμενο για εκτύπωση. Διαλέξαμε να μην σταματάμε κάθε 15 δευτερόλεπτα μιας και δεν έχει νόημα να αλλάζουμε τις προτεραιότητες όταν η αλλαγή τους δεν θα επιφέρει κάποια αλλαγή στη σειρά εκτύπωσης.

ΜΕΡΟΣ Δ

N1 = [100, 10.000]

N2 = [10.000, 20.000]

N3 = [20.000, 30.000]

N4 = [30.000, 40.000]

N5 = [40.000, 50.000]

Παρατηρούμε ότι ο αλγόριθμος B έχει μικρότερο χρόνο αναμονής και μέγιστο χρόνο αναμονής κατά μέσο όρο των 50 αρχείων από τον Γ. Για περισσότερες πληροφορίες ελέγξτε τους φακέλους random και results.