

Εργαστηριακή άσκηση #6

Θέμα: Εξοικείωση με την χρήση διανυσμάτων και των εντολών επανάληψης. Ανάπτυξη προγραμμάτων ταξινόμησης.

Η εργαστηριακή αυτή άσκηση αποσκοπεί στην περαιτέρω εξοικείωση με τη χρήση διανυσμάτων καθώς και των εντολών επανάληψης. Θα δημιουργήσουμε το έργο **sorter** το οποίο περιέχει την κλάση **Sorter**. Η κλάση **Sorter** παρέχει στατικές μεθόδους ταξινόμησης διανυσμάτων πραγματικών αριθμών. Το έργο **sorter** περιέχει επίσης την κλάση **TestSorter** η οποία χρησιμοποιείται για τον έλεγχο σωστής λειτουργίας της κλάσης **sorter**.

1. Για οικονομία χρόνου, στην εργαστηριακή αυτή άσκηση θα χρησιμοποιήσουμε το έτοιμο έργο **sorter** που είναι διαθέσιμο στη σελίδα του μαθήματος (σύνδεσμος «Κώδικας»). Αντιγράψτε το φάκελο **sorter_labs** στον κοινόχρηστο χώρο του υπολογιστή σας. Κατόπιν ανοίξτε το έργο με το **BlueJ**.
2. Μεταφράστε το έργο. Κατόπιν εκτελέστε τη στατική μέθοδο **main()** της κλάσης **TestSorter**. Παρατηρήστε την έξοδο που δημιουργείται.
3. Μελετήστε και βεβαιωθείτε ότι κατανοείτε τον έτοιμο κώδικα της κλάσης **Sorter**. Ο έτοιμος κώδικας (πιστό αντίγραφο του έτοιμου κώδικα της εργαστηριακής άσκησης #7) περιλαμβάνει τις μεθόδους

▪ void fillRandomData(double[])	Γεμίζει ένα διάνυσμα με τυχαίους αριθμούς
▪ void printH(double[])	Τυπώνει τα στοιχεία ενός διανύσματος, όλα στην ίδια σειρά.
▪ void printV(double[])	Τυπώνει τα στοιχεία ενός διανύσματος, ένα σε κάθε σειρά.
▪ void swap(double[], int, int)	Ανταλλαγή στοιχείων διανύσματος
▪ int maxLocationFrom(double[], int)	Εύρεση θέσης μέγιστου στοιχείου
▪ int minLocationFrom(double[], int)	Εύρεση θέσης ελάχιστου στοιχείου
▪ double[] clone(double[])	Κατασκευή αντίγραφου διανύσματος
▪ double[] merge(double[], double[])	Συγχώνευση ταξινομημένων διανυσμάτων

Παρατηρήστε ότι ο κώδικας περιέχει με μορφή σχολίων (ώστε να μπορεί να μεταφραστεί χωρίς προβλήματα) τις δηλώσεις (signatures) των παρακάτω μεθόδων τις οποίες θα υλοποιήσουμε.

▪ void sortSelection(double[])	Ταξινόμηση διανύσματος με τη μέθοδο “Selection-sort”
▪ void sortBubble(double[])	Ταξινόμηση διανύσματος με τη μέθοδο “Bubble-sort”
▪ void sortInsertion(double[])	Ταξινόμηση διανύσματος με τη μέθοδο “insertion-sort”
▪ void sortMerge(double[])	Ταξινόμηση διανύσματος με τη μέθοδο “merge-sort”

4. Μελετήστε και βεβαιωθείτε ότι κατανοείτε τον έτοιμο κώδικα της κλάσης **TestSorter** (συμπεριλαμβανομένων των σημείων που παρουσιάζονται ως σχόλια). Ο κώδικας απλώς δημιουργεί τέσσερα διανύσματα με δεδομένα, κατόπιν καλεί μεθόδους της κλάσης **Sorter** που τα ταξινομούν και, τέλος, τυπώνει τα αποτελέσματα.
5. Βεβαιωθείτε ότι κατανοείτε τις έτοιμες μεθόδους **minLocationFrom()** και **swap()** της κλάσης **Sorter**.
6. Υλοποιήστε την μέθοδο **sortSelection()** της κλάσης **Sorter**. Η μέθοδος **sortSelection()** δέχεται ως παράμετρο ένα διάνυσμα και το ταξινομεί με τη χρήση της μεθόδου “selection-sort”. Η υλοποίηση να κάνει χρήση των **private** μεθόδων **minLocationFrom()** και **swap()**. Ελέγξτε την ορθότητα της υλοποίησης μέσω της κλάσης

TestSorter (μετατρέποντας τον «σχολιασμένο κώδικα» σε «εκτελέσιμο» διαγράφοντας τους χαρακτήρες σχολίων (« \| » από την αρχή των γραμμών κώδικα).

7. Υλοποιείτε την μέθοδο **sortInsertion()** της κλάσης **Sorter**. Η μέθοδος **sortInsertion()** δέχεται ως παράμετρο ένα διάνυσμα και το ταξινομεί με τη χρήση της μεθόδου “insertion-sort”. Ελέγξτε την ορθότητα της υλοποίησης μέσω της κλάσης **TestSorter** (μετατρέποντας τον «σχολιασμένο κώδικα» σε «εκτελέσιμο» διαγράφοντας τους χαρακτήρες σχολίων (« \| » από την αρχή των γραμμών κώδικα).
8. Υλοποιείτε την μέθοδο **sortBubble()** της κλάσης **Sorter**. Η μέθοδος **sortBubble()** δέχεται ως παράμετρο ένα διάνυσμα και το ταξινομεί με τη χρήση της μεθόδου “bubble-sort”. Ελέγξτε την ορθότητα της υλοποίησης μέσω της κλάσης **TestSorter** (μετατρέποντας τον «σχολιασμένο κώδικα» σε «εκτελέσιμο» διαγράφοντας τους χαρακτήρες σχολίων (« \| » από την αρχή των γραμμών κώδικα).
9. Βεβαιωθείτε ότι κατανοείτε την υπογραφή της μεθόδου **merge()** της κλάσης **Sorter**. Αγνοείτε προς το παρόν την υλοποίηση της.
10. Υλοποιείτε την μέθοδο **sortMerge()** της κλάσης **Sorter**. Η μέθοδος **sortMerge()** δέχεται ως παράμετρο ένα διάνυσμα και το ταξινομεί με τη χρήση της αναδρομικής μεθόδου “merge-sort”. Η υλοποίηση να κάνει χρήση της μεθόδου **merge()** η οποία δίνεται έτοιμη. Ελέγξτε την ορθότητα της υλοποίησης της **sortMerge()** μέσω της κλάσης **TestSorter** (μετατρέποντας τον «σχολιασμένο κώδικα» σε «εκτελέσιμο» διαγράφοντας τους χαρακτήρες σχολίων (« \| » από την αρχή των γραμμών κώδικα).
11. Βεβαιωθείτε ότι κατανοείτε την υλοποίηση της μεθόδου **merge()** της κλάσης **Sorter**.