

Οδηγίες Υλοποίησης :

Point: Η υλοποίηση της **distanceTo()** γίνεται με βάση την τον τύπο της Ευκλείδειας απόστασης μεταξύ δύο σημείων. Η **squareDistanceTo()** επιστρέφει το τετράγωνο της μεθόδου **distanceTo()**.

Rectangle: Η **distanceTo ()** επιστρέφει 0 όταν το σημείο βρίσκεται μέσα στο παραλληλόγραμμο(χρήση **contains()**). Σε διαφορετική περίπτωση βρίσκει ποια είναι η κοντινότερη πλευρά του παραλληλογράμμου και επιστρέφει την απόσταση του σημείου από αυτήν με τη χρήση της βοηθητικής private μεθόδου **distance()** (η οποία υπολογίζει την απόσταση ενός σημείου από μία ευθεία).

LinkedList: Η υλοποίηση αυτής της κλάσης είναι ίδια με την υλοποίηση της **LinkedList** της πρώτης εργασίας.

TwoDTree: Οι μέθοδοι **nearestNeighbor()** και **rangeSearch()** έχουν από μια βοηθητική private μέθοδο οι οποίες χρησιμεύουν για την αναδρομική υλοποίησή τους. Είναι private προκειμένου ο πελάτης να μην γνωρίζει την ύπαρξη του.

Συγκεκριμένα η **nearestNeighbor()** η αναδρομή ξεκινά από τη ρίζα με Rectangle το $[0,100] \times [0,100]$ και minimum το Point που περιέχει η ρίζα. Σε κάθε αναδρομική κλήση ελέγχουμε αν ο κόμβος στον οποίο βρισκόμαστε έχει μικρότερη απόσταση από τον minimum, και ανανεώνουμε το minimum ανάλογα. Στη συνέχεια, υπολογίζουμε το Rectangle που περιέχει το αριστερό παιδί (σύμφωνα με τις x συντεταγμένες) και επαναλαμβάνουμε την ίδια διαδικασία με πάνω. Έπειτα, κάνουμε 2 αναδρομικές κλήσεις στο αριστερό και στο δεξί παιδί του αριστερού παιδιού υπολογίζοντας κάθε φορά τα ανάλογα Rectangle που περιέχουν τα αντίστοιχα παιδιά (σύμφωνα με τις y συντεταγμένες). Η παραπάνω διαδικασία επαναλαμβάνεται και για το δεξί παιδί του κόμβου που βρισκόμαστε. Η αναδρομή τελειώνει είτε όταν ο κόμβος είναι null είτε όταν η απόσταση του query point από το Rectangle που βρισκόμαστε εκείνη τη χρονική στιγμή γίνει αυστηρά μεγαλύτερη από την απόσταση του query point ως τότε minimum σημείου.

Στη **rangeSearch()** η αναδρομή ξεκινά από τη ρίζα με Rectangle το $[0,100] \times [0,100]$ και με μία κενή λίστα. Σε κάθε αναδρομική κλήση εξετάζουμε αν ο κόμβος στον οποίο βρισκόμαστε περιέχεται στο query rectangle, και αν ναι, το προσθέτουμε στη λίστα. Αμέσως μετά ελέγχουμε το αριστερό παιδί με την ίδια λογική αφού πρώτα υπολογίσουμε το Rectangle (σύμφωνα με τις x συντεταγμένες) στο οποίο περιέχεται και συνεχίζουμε αναδρομικά πηγαίνοντας στο αριστερό και στο δεξί παιδί αφού υπολογίσουμε τα αντίστοιχα Rectangles (σύμφωνα με τις y συντεταγμένες). Η παραπάνω διαδικασία επαναλαμβάνεται και για το δεξί παιδί του κόμβου που βρισκόμαστε. Η αναδρομή τελειώνει είτε όταν ο κόμβος είναι null είτε όταν το query rectangle και το Rectangle στο οποίο βρισκόμαστε εκείνη τη χρονική στιγμή δεν έχουν κανένα κοινό σημείο.