**ΟΝΟΜΑ : ΨΑΡΑ ΣΤΥΛΙΑΝΟΣ**

**Α.Μ: 1151800226**

**Άσκηση 2**

H βάση του προγράμματος μου αποτελείτε από πέντε κλάσις που είναι συνδεδεμένες μεταξύ τους (forum->threads->posts)(forum->tree->list). Σε κάθε κλάση στο private μέρος υπάρχουν οι μεταβλητές και οι συναρτήσεις που είναι υποστηρικτικές και που δεν θέλω να είναι άμεσα προσβάσιμες , από την άλλη στο public κομμάτι βρίσκονται οι constructors, destructors και οι συναρτήσεις που χειρίζομαι μέσο αντικείμενον.

1.

Αρχικά η main δημιουργεί ένα αντικείμενο τύπου forum το οποίο παίρνει ως όρισμα τo string "Αντικειμενοστραϕής Προγραμματισμός" . Σύμφωνα με το composition οι constructors καλούνται από μέσα προς τα έξω οπότε αρχίζει από το post και καταλήγει στο forum.

Ο Constructor του post περιέχει μια static μεταβλητή τύπου int η οποία αυξάνεται για να έχουν τα posts μοναδικό αριθμό και τρεις πίνακές τίτλο, δημιουργό και κείμενο από τους οποίους γίνεται random επιλογή για τα στοιχεία του post και τέλος γίνεται η εκτύπωση του post (η ημερομηνία είναι σταθερή).

Ο constructor του thread περιέχει δύο πίνακες θέμα και δημιουργό από τους οποίους επιλέγονται random τα στοιχεία του thread (η ημερομηνία είναι σταθερή) . Επίσης υπάρχει μια μεταβλητή static τύπου int η οποία είναι 0 μόνο την πρώτη φορά για να δημιουργηθεί σίγουρα το thread με θέμα "Κανόνες και χρήσιμες πληροϕορίες" ακόμη δεσμεύει μνήμη για τον πίνακα αντικείμενων τύπου post ο οποίος είναι δηλωμένος στο private του thread και έχει τυχαίο μέγεθος. Τέλος τυπώνει το μήνυμα τις δημιουργίας του thread.

Ο constructor του forum δεσμεύει μνήμη για τον πίνακα αντικειμένων τύπου threads και για για τον πίνακα τύπου δείκτη tree και τυπώνει το μήνυμα δημιουργίας του forum.

2.

Μέσο της main και της συνάρτηση print\_sorted που βρίσκεται στην κλάση forum μας δίνεται η δυνατότητα να τυπώσουμε όλα τα posts με αύξουσα σειρά αλφαβητικά με βάση τον δημιουργό.

Αρχικά η print\_sorted αποτελείτε από ένα for για το πλήθος των threads το οποίο δεσμεύει μνήμη για την δημιουργία του δέντρου για κάθε thread στέλνοντας όρισμα στoν constructor του tree το αντίστοιχο thread.

Ο constructor του tree είναι υπεύθυνος για την δημιουργία του δέντρου. Αρχικά μέσο τις συνάρτησης get\_ArrayNumber που βρίσκετε στην κλάση threads παίρνουμε τον αριθμό του πλήθους των post για κάθε thread ούτως ώστε να γνωρίζουμε το μέγεθος του δέντρου Στη συνέχεια μέσο της συνάρτησης get\_creator που βρίσκεται στο threads και της getcreator που βρίσκεται στο post παίρνουμε το όνομα για τον δημιουργό του post το οποίο θα χρειαστούμε για να βρούμε την θέση κάθε κόμβου στο δέντρο. Έπειτα δημιουργείτε η ρίζα του δέντρου, αρχικοποιοντας τον δημιουργό και τους δείκτες right και left και μετά δεσμεύει μνήμη για την λίστα με τα post και με την βοήθεια της συνάρτησης get\_post που βρίσκεται στο threads παίρνει σαν όρισμα reference το post του δημιουργού του τρέχον κόμβου (στην get\_creator και την get\_post έχω μια static μεταβλητή η οποία μηδενίζεται κάθε φορά που φτάνει στον τελευταίο κόμβο για εισαγωγή του δέντρου).

Ο constructor του list δημιουργεί τον κόμβο κεφαλή της λίστας που αποτελείτε από ένα δείκτη next και δεσμεύει μνήμη για το αντικείμενο post που αρχικοποίητε μέσο ενός copy constructor παίρνοντας σαν όρισμά ένα reference post.

Στη συνέχεια στον constructor του tree για κάθε κόμβο μέσο ενός while γίνετε σύγκριση του ονόματος του τρέχον κόμβου με τους κόμβους που ηδη υπάρχουν στο δέντρο και ανάλογα αν είναι αλφαβητικά μεγαλύτερο η μικρότερο μεταφέρετε ο έλεγχος δεξιά η αριστερά αντίστοιχα και όταν βρεθεί η θέση του δημιουργείτε νέος κόμβος και λίστα με την ίδια διαδικασία που δημιουργήθηκε η ρίζα. Σε περίπτωση που ο δημιουργός είναι ίδιος με κάποιων που υπάρχει ηδη μέσo της συνάρτηση next\_post που βρίσκεται στο list παίρνει σαν όρισμα reference post και το προσθέτει στην ηδη δημιουργημένη λίστα.

Η συνάρτηση next\_post τρέχει στην ηδη δημιουργημένη λίστα μέχρι να φτάσεις στο τέλος της και προσθέτει ένα κόμβο που δημιουργεί με την ίδια διαδικασία που δημιουργήθηκε ο κόμβος κεφαλή.

Μετά την δημιουργία των δέντρων η print\_sorted με την βοήθεια ενός for και της συνάρτησης add\_tree που παίρνει σαν όρισμα το πρώτο δέντρο μας δίνει την δυνατότητα να ενισχύσουμε το το πρώτο δέντρο με τους κόμβους των άλλον.

Η add\_tree καλεί την συνάρτηση get\_node που παίρνοντας σαν όρισμα τη ρίζα του δέντρου και το πρώτο δέντρο μέσο μιας αναδρομικής διαδικασίας POST ORDER απομονώνει τους κόμβους από το τέλος, μετά η enhance παίρνει σαν όρισμα τον κόμβο και ενισχύει το πρώτο δέντρο με την ίδια διαδικασία που προσθέτει κόμβους σε ένα δέντρο. H διαφορά στη δημιουργία του κόμβου είναι ότι για την δημιουργία της λίστας ο constructor παίρνει σαν όρισμα το δείκτη της λίστας του κόμβου που θέλουμε να προσθέσουμε δημιουργεί τον κόμβο κεφαλή και στην συνέχεια μέσο ενός while στέλνει σαν όρισμα στην next\_post το δείκτη του επόμενου post και το προσθέτει με τον ίδιο τρόπο στην λίστα. Επίσης στην συνάρτηση enhance όταν ο δημιουργός είναι ίδιος καλείτε η συνάρτηση add\_list με όρισμα την λίστα που θέλουμε να προσθέσουμε και μέσο ενός while τρέχει την ηδη δημιουργημένη λίστα μέχρι να φτάσει στο τέλος και στη συνέχεια δημιουργεί και προσθέτει κόμβους για όλα τα posts της λίστας .

Η συνάρτηση print\_sorted για τέλος εκτυπώνει το μέσο της της Call\_print το ενισχυμένο δέντρο .

Η Call\_print καλεί την συνάρτηση print δίνοντας της όρισμα τη ρίζα του δέντρου και μέσο μιας αναδρομικής διαδικασίας IN ORDER τυπώνει τον δημιουργό και καλεί την συνάρτηση print\_list που μέσο ενός while καλεί την συνάρτηση print\_list\_post και τυπώνει όλα τα posts της λίστας.

3.

Τέλος σύμφωνα με το composition οι destructors καλούνται από έξω προς τα μέσα .Αρχικά έχουμε τον destructor του forum ο οποίος βγάζει μήνυμα διαγραφής και μεσο ενός while διαγράφει τα δέντρα έτσι καλείτε ο destructor του tree που με τη σειρά του καλεί την συνάρτηση delete\_tree και μέσο μιας αναδρομής POST ORDER καλεί την DeleteList η οποία διαγράφει τα posts και τους κόμβους της λίστας και εκτυπώνεται το μήνυμα του destructor του post και τέλος διαγραφεί τους κόμβους του δέντρου και την λίστα. Στη συνέχεια ο destructor του forum διαγράφει την μνήμη που δέσμευσε για τον πίνακα των tree και thread έτσι καλείτε ο destructor του thread βγάζει μήνυμα διαγραφής και διαγράφει την μνήμη που δέσμευσε για τα posts τέλος καλείτε ο destructor του post που βγάζει μήνυμα διαγραφής για όλα τα posts.

**Για τη μεταγλώττιση του προγράμματος:**

g++ c\_post.cpp c\_threads.cpp c\_forum.cpp c\_BTree.cpp c\_list.cpp main.cpp