

OOP 2021-2022

Α΄ ομάδα ασκήσεων

ΑΣΚΗΣΗ 1 :

Εντολή μεταγλώττισης: g++ -o assgn1_1 assgn1_1.cpp

Τιμές παραμέτρων για δοκιμές του προγράμματος:

`./assgn1_1 10 50 5 5 ./assgn1_1 3 20 2 3 ./assgn1_1 100 10 100 0`

`./assgn1_1 1 1 20 10 0 ./assgn1_1 30 200 5 20 ./assgn1_1 100 30 30 30`

Πληροφορίες και αιτιολογήσεις σχετικά με τις σχεδιαστικές επιλογές:

Κλάσεις:

Η κλάση **Book** έχει έναν constructor ο οποίος έχει κάποιες default τιμές ορισμάτων (title = "Untitled", writer = "Unnamed", ESNB = 0) έτσι ώστε στην περίπτωση που ο χρήστης πάει να φτιάξει ένα βιβλίο χωρίς να δώσει κάποια ορίσματα, να μην φτιαχτεί κάποιο "dummy" βιβλίο, αλλά να έχει αρχικοποιηθεί. Επίσης έχει κάποιους "getters" οι οποίοι επιστρέφουν τα αντίστοιχα δεδομένα της κλάσης (χρησιμοποιούνται από την κλάση Shelf για να γίνουν print τα στοιχεία του βιβλίου).

Η κλάση **Shelf** έχει σαν δεδομένα έναν πίνακα δεικτών σε Book, έναν μετρητή (shelfbooks) του πλήθους των βιβλίων που περιέχει το συγκεκριμένο ράφι και έναν ακέραιο (maxnum) που είναι η μέγιστη χωρητικότητα του ραφιού (NMax). Ο constructor της δημιουργεί δυναμικά έναν πίνακα από δείκτες σε Book μεγέθους maxnum (το οποίο έχει πάρει σαν όρισμα μέσω initializer list) και αρχικοποιεί τα στοιχεία του πίνακα με NULL (εφόσον κατά την κατασκευή τους τα ράφια δεν περιέχουν κανένα στοιχείο). Η συνάρτηση place_book (επιστρέφει τύπο bool σχετικά με το αν πέτυχε ή όχι) παίρνει ένα βιβλίο (δείκτη σε βιβλίο) και το τοποθετεί, εάν χωράει, στην αμέσως επόμενη κενή θέση του ραφιού. Η συνάρτηση take_book (επιστρέφει τύπο bool σχετικά με το αν πέτυχε ή όχι), εάν το ράφι δεν είναι άδειο, αφαιρεί το βιβλίο που μπήκε τελευταίο στο ράφι. Η συνάρτηση print εκτυπώνει τα στοιχεία όλων των βιβλίων που περιέχει το ράφι.

Η κλάση **Closet** έχει σαν δεδομένα δυο ράφια (τύπου Shelf δηλαδή). Ο constructor της παίρνει σαν όρισμα το NMax και μέσω initializer list το «στέλνει» στους constructor του καθενός από τα δυο ράφια της. Η συνάρτηση place_book ανάλογα με το όρισμα number, που λαμβάνει, αν είναι 1 (πρώτο ράφι του ντουλαπιού) ή 2 (δεύτερο ράφι του ντουλαπιού) καλεί την place_book του αντίστοιχου ραφιού. Ομοίως και η συνάρτηση take_book.

Η κλάση **Bookcase** έχει σαν δεδομένα μια βάση, ένα ντουλάπι και τρία ράφια, καθώς και έναν μετρητή (N) του πλήθους των βιβλίων που περιέχει η βιβλιοθήκη. Ο constructor της παίρνει σαν όρισμα το NMax και μέσω initializer list το «στέλνει» στους constructor του ντουλαπιού και των τριών ραφιών της. Ο constructor επίσης αρχικοποιεί το N σε 0 (εφόσον κατά την κατασκευή της η βιβλιοθήκη είναι άδεια). Η συνάρτηση place_book ανάλογα με το όρισμα number, που λαμβάνει, αν είναι 1 (πρώτο ράφι), 2 (δεύτερο ράφι), 3 (τρίτο ράφι), 4 (πρώτο ράφι ντουλαπιού) ή 5 (δεύτερο ράφι ντουλαπιού) καλεί την place_book του αντίστοιχου ραφιού ή την place_book του ντουλαπιού (δίνοντας σαν όρισμα 1 αν πρόκειται για το 4^ο ράφι δηλαδή το 1^ο του ντουλαπιού ή 2 αν πρόκειται για το 5^ο ράφι δηλαδή το 2^ο του ντουλαπιού), καθώς και αυξάνει σε κάθε περίπτωση τον μετρητή N κατά 1. Ομοίως και η συνάρτηση take_book (μειώνοντας τον μετρητή N κατά 1).

Main:

Όσον αφορά το **α' ερώτημα** δημιουργείται μια βιβλιοθήκη (τύπου Bookcase δηλαδή) δίνοντας της σαν όρισμα το NMax (ώστε μέσω του constructor της να «καταλήξει» στο κάθε ράφι).

Όσον αφορά το **β' ερώτημα** δημιουργείται δυναμικά ένας πίνακας μεγέθους L που περιέχει δείκτες σε Book. Επίσης δημιουργούνται δυναμικά L βιβλία (Book), παίρνοντας κάποιες τυχαίες τιμές τίτλος-συγγραφέας-ESBN, και αναθέτονται στον πίνακα. Ο τίτλος δημιουργείται μέσω της συνάρτησης title_gen η οποία: περιέχει ένα πίνακα από 30 τίτλους και επιλεγεί τυχαία έναν από αυτούς επιστρέφοντας τον. Ο συγγραφέας δημιουργείται μέσω της συνάρτησης writer_gen η οποία: περιέχει έναν πίνακα από 20 μικρά ονόματα και έναν πίνακα από 20 επώνυμα, επιλεγεί τυχαία ένα από τον πρώτο πίνακα και ένα από τον δεύτερο πίνακα και τα «συνενώνει», έτσι ώστε να φτιαχτεί ένα ολόκληρο όνομα, επιστρέφοντας το. Το ESBN είναι απλά ένας τυχαίος ακέραιος από το 1 έως το 1000.

Όσον αφορά το **γ' ερώτημα** γίνεται μια επανάληψη η οποία σταματάει μετά από K1 φορές ή εάν δεν υπάρχουν άλλα βιβλία για να γίνουν place. Κάθε φορά καλείται η place_book της βιβλιοθήκης δίνοντας σαν ορίσματα το βιβλίο που θα γίνει place (ξεκινάει από το 1^ο και κάθε φορά παίρνει το επόμενο) καθώς και έναν τυχαίο αριθμό από το 1 έως το 5 (που δείχνει σε ποιο ράφι θα γίνει place).

Όσον αφορά το **δ' ερώτημα** γίνεται μια επανάληψη K2 φορές και κάθε φορά καλείται η take_book της βιβλιοθήκης δίνοντας σαν όρισμα έναν τυχαίο αριθμό από το 1 έως το 5 (που δείχνει από ποιο ράφι θα γίνει take).

Όσον αφορά το **ε' ερώτημα** απλά καλείται η print της βιβλιοθήκης (η οποία με την σειρά της καλεί την print του ντουλαπιού και των ραφιών της) και εκτυπώνονται τα βιβλία του κάθε ραφίου της βιβλιοθήκης.

Υ.Γ.:

Επειδή οι διαθέσιμοι τίτλοι που έχω βάλει (στην title_gen) είναι μόνο 30, κάποιιοι τίτλοι βιβλίων ενδεχομένως να είναι ίδιοι μεταξύ τους (ωστόσο ο συγγραφέας και το ESBN, κατά πασά πιθανότητα, θα διαφέρουν).

Οι συναρτήσεις-μέλη get_title, get_writer, get_ESBN της κλάσης Book, καθώς και οι συναρτήσεις-μέλη print των κλάσεων Shelf, Closet και Bookcase, είναι const, εφόσον δεν τροποποιούν το αντικείμενο για το οποίο καλούνται.

Εφόσον τα L και NMax είναι κάποιες ακέραιες μεταβλητές που έχουν πάρει τιμή κατά την εκτέλεση του προγράμματος, δηλαδή οι τιμές τους δεν είναι γνωστές κατά την μεταγλώττιση, δημιουργούμε τους αντίστοιχους πίνακες (biblio[L] - πίνακα L βιβλίων στη main, και biblia[NMax] - πίνακα βιβλίων που περιέχει το κάθε ράφι στην κλάση Shelf) με δυναμικό τρόπο.