OOP 2021-2022

Α΄ ομάδα ασκήσεων

ΑΣΚΗΣΗ 2 :

Εντολή μεταγλώττισης: **g++ -o assgn1_2 assgn1_2.cpp**

Τιμές παραμέτρων για δοκιμές του προγράμματος:

./assgn1_2 1 2 3 2 ./assgn1_2 2 1 15 2 ./assgn1_2 3 2 8 3
./assgn1_2 6 2 7 4 ./assgn1_2 5 3 10 4 ./assgn1_2 2 4 13 6

Πληροφορίες και αιτιολογήσεις σχετικά με τις σχεδιαστικές επιλογές:

Κλάσεις:

Η κλάση **Cashier** έχει σαν "extra private data" έναν ακέραιο limit ο οποίος παίρνει την τιμή του L μέσω του constructor της. Η συνάρτηση free είναι bool ώστε αν καταλήξει στο "The cashier overworked" να επιστρέψει false (και, ως αποτέλεσμα, αυτός που την κάλεσε να ξέρει ότι πρέπει να κλείσει το συγκεκριμένο ταμείο) αλλιώς επιστρέφει true. Η συνάρτηση is_open επιστρέφει bool σχετικά με το αν είναι ανοιχτό(true) ή όχι(false) το συγκεκριμένο ταμείο. Η συνάρτηση is_free επιστρέφει bool σχετικά με το αν είναι εξυπηρετεί(false-αφού δεν είναι ελεύθερο) ή όχι(true-αφού είναι ελεύθερο) το συγκεκριμένο ταμείο.

Η κλάση **Bank** έχει σαν "extra private data" έναν ακέραιο ith που δείχνει το επόμενο ταμείο που θα κάνει serve(δηλαδή είναι το επόμενο ανοιχτό και ελεύθερο ταμείο από αυτό που εξυπηρέτησε την τελευταία φορά-κυκλικά), έναν ακέραιο next που δείχνει το επόμενο ταμείο που θα γίνει open(δηλαδή είναι το επόμενο κλειστό ταμείο) και έναν ακέραιο kappa ο οποίος παίρνει την τιμή του K μέσω του constructor της. Ο constructor της, επιπλέον, δημιουργεί δυναμικά 5 ταμεία και «στέλνει» στο καθένα από αυτά(μέσω constructor) το L. Επίσης, εφόσον ανοίγει το 1ο ταμείο, αρχικοποιεί το ith σε 0(δηλαδή το next cashier to serve γίνεται το 1ο) και το next σε 1(δηλαδή το next cashier to be opened γίνεται το 2ο). Η συνάρτηση check_to_open επιστρέφει bool σχετικά με το αν χρειάζεται να ανοίξει(true) ή όχι(false) ταμείο. Η συνάρτηση open παίρνει σαν όρισμα ένα index που δείχνει ποιο ταμείο να ανοίξει και ανοίγει αυτό το ταμείο(καλώντας την open του ταμείου). Η συνάρτηση check_to_close επιστρέφει bool σχετικά με το αν χρειάζεται να κλείσει(true) ή όχι(false) ταμείο. Η συνάρτηση close παίρνει σαν όρισμα ένα index που δείχνει ποιο ταμείο να κλείσει και κλείνει αυτό το ταμείο(καλώντας την close του ταμείου). Έπειτα ελέγχει μήπως όλα τα ταμεία είναι κλειστά οπότε κάνει το next 0(δηλαδή το next cashier to be opened γίνεται το 1ο). Η συνάρτηση enter επιστρέφει bool σχετικά με το αν κατάφερε να κάνει enter(true) ή όχι(false). Αν χρειαστεί να ανοίξει νέο ταμείο, και δεν είναι όλα ανοιχτά, τότε ανοίγει το ταμείο next και αλλάζει το next στο επόμενο ταμείο(ή στο 1ο αν μόλις άνοιξε το 5ο). Η συνάρτηση serve, αν όλα τα ταμεία είναι κλειστά, ανοίγει το 1ο ταμείο, κάνει το next 1(δηλαδή το next cashier to be opened γίνεται το 2ο) εξυπηρετεί τον πελάτη, ελευθερώνεται(κλείνοντας το ταμείο αν επιστραφεί false-δηλαδή αν το ταμείο "overworked") και κάνει και το ith 1(δηλαδή το next cashier to serve γίνεται το 2ο). Αν δεν είναι όλα τα ταμεία κλειστά τότε ψάχνει να βρει ποιο ταμείο είναι το ith, δηλαδή ποιο θα κάνει serve, και εφαρμόζει την ίδια διαδικασία για αυτό, αλλάζοντας κατόπιν το ith ανάλογα. Έπειτα κάνει exit του πελάτη.

Main:

Όσον αφορά το **α' ερώτημα** δημιουργείται μια τράπεζα, περνώντας της ορίσματα τα L και K.

Όσον αφορά το **β' ερώτημα**, αρχικά αρχικοποιείται ένας συνολικός μετρητής waiting_to_enter σε 0 (αυτός θα μετρήσει πόσοι συνολικά πελάτες, μετά από τις M επαναλήψεις, δεν κατάφεραν να γίνουν enter). Μέσα στη μεγάλη for του M ένας τοπικός μετρητής(w1), της τρέχουσας αυτής μεγάλης for, γίνεται 0 (αυτός θα μετρήσει πόσοι πελάτες(από τους συγκεκριμένους N) δεν κατάφεραν να γίνουν enter). Έπειτα γίνονται N enter. Κάθε φορά που η enter επιστρέφει false(δηλαδή δεν κατάφερε να κάνει enter τον πελάτη) αυξάνονται και οι δυο μετρητές κατά ένα. Κατόπιν γίνονται N-w1 serve(δηλαδή όσοι κατάφεραν επιτυχώς να μπουν, εξυπηρετούνται).

Όσον αφορά το **γ' ερώτημα** επαναλαμβάνεται η ίδια διαδικασία, αλλά για τους waiting_to_enter, μέσα σε μια do while που τρέχει μέχρις ότου να εξυπηρετηθούν όλοι οι πελάτες.

Υ.Γ.:

Ο πίνακας των πέντε ταμείων της τράπεζας είναι τύπου Cashier* και όχι απλά Cashier έτσι ώστε να δημιουργηθεί δυναμικά και να μπορέσει να περαστεί το L σαν όρισμα στον constructor του κάθε ταμείου.

Οι συναρτήσεις-μέλη is_open και is_free της κλάσης Cashier καθώς και οι συναρτήσεις-μέλη check_to_open, check_to_close, waiting_customers και open_cashiers της κλάσης Bank, είναι const, εφόσον δεν τροποποιούν το αντικείμενο για το οποίο καλούνται.

Οι συναρτήσεις-μέλη check_to_open, open, check_to_close, close, exit, waiting_customers και open_cashiers της κλάσης Bank είναι δηλωμένες στο private κομμάτι της κλάσης εφόσον είναι βοηθητικές – δεν καλούνται άμεσα από την main(ο χρήστης καλεί μόνο τις συναρτήσεις enter και serve).

Θεόδωρος Μωραΐτης, ΑΜ: 1115202000150