

OOP 2021-2022

Β' ομάδα ασκήσεων

### ΑΣΚΗΣΗ 3 :

Εντολή μεταγλώττισης: g++ -o assgn3 main.cpp creatures.cpp

Τιμές παραμέτρων για δοκιμές του προγράμματος:

`./assgn3 5 10 2 3 3      ./assgn3 2 4 5 2 3`  
`./assgn3 10 5 2 4 5      ./assgn3 4 20 2 3 4`

Πληροφορίες και αιτιολογήσεις σχετικά με τις σχεδιαστικές επιλογές:

Κλάσεις:

Η κλάση **creature** έχει σαν έξτρα μέλη-δεδομένα έναν ακέραιο(index) που είναι η θέση του πλάσματος στην κοινότητα που ανήκει, και έναν δείκτη(pointer) σε `creature_society`, έτσι ώστε να ξέρει το πλάσμα σε ποια κοινότητα ανήκει. Ο destructor της είναι δηλωμένος σαν virtual έτσι ώστε όταν διαγραφτεί ένα καλό/κακό πλάσμα να καλεστεί πρώτα ο αντίστοιχος destructor(και εννοείται μετά καλείται αυτόματα και αυτός της creature). Η συνάρτηση `is_a_good` είναι δηλωμένη σαν pure virtual εφόσον μεταθέτει τον έλεγχο στις παράγωγες κλάσεις και κάνει την creature abstract. Η συνάρτηση `clone` είναι δηλωμένη σαν virtual ώστε όταν γίνει clone κάποιο καλό/κακό πλάσμα να καλεστεί η αντίστοιχη clone του καλού/κακού. Η συγκεκριμένη δεν κάνει τίποτα περισσότερο από το να εκτυπώνει ένα μήνυμα και να επιστρέφει τον δείκτη που έλαβε σαν όρισμα(μόνο και μόνο ώστε να έχει ίδιο ορισμό με την αντίστοιχη clone του good/bad creature). Επίσης η συνάρτηση `bless` είναι δηλωμένη σαν virtual ώστε όταν γίνει bless κάποιο καλό/κακό πλάσμα να καλεστεί η αντίστοιχη bless του καλού/κακού. Επιπλέον έχει κάποιες έξτρα συναρτήσεις-μέλη `get_life`, `get_index`, `get_pointer`, καθώς και μια `print`(που εκτυπώνει το όνομα και την ζωή του πλάσματος).

Η κλάση **good\_creature** έχει σαν μέλος-δεδομένο το `good_thrsh`. Η συνάρτηση `bless` αρχικά καλεί την `bless` της "μαμάς" creature(ώστε να γίνει bless) και κατόπιν εάν η ζωή του πλάσματος είναι μεγαλύτερη του `good_thrsh` τότε καλεί την `clone_next` της society που το περιέχει, με όρισμα την θέση του. Η συνάρτηση `clone` παίρνει σαν όρισμα το πλάσμα που θα αντικατασταθεί από το τρέχον και επιστρέφει το νέο αυτό πλάσμα(δηλαδή τον κλώνο του τρέχοντος). Η συγκεκριμένη αρχικά καλεί την `clone` της "μαμάς" creature(μόνο και μόνο για να εκτυπωθεί κάποιο μήνυμα) και κατόπιν κρατάει την θέση του πλάσματος, που θα διαγραφτεί, σε μια ακέραια μεταβλητή(thesi), διαγράφει αυτό το πλάσμα και δημιουργεί ένα νέο στην θέση του σαν αντίγραφο του τρέχοντος(στέλνοντας στον copy constructor του good\_creature εκτός από το τρέχον πλάσμα(\*this) που θα κλωνοποιηθεί και την θέση(thesi) στην κοινότητα που θα έχει το νέο αυτό πλάσμα που θα δημιουργηθεί(η οποία θα είναι ίδια με εκείνη του διαγραφμένου πλάσματος).

Η κλάση **bad\_creature** έχει αντίστοιχες λειτουργίες με αυτές της good\_creature(με τη διαφορά ότι αντί για good\_thrsh έχουμε bad\_thrsh).

Η κλάση **creature\_society** έχει σαν μέλη-δεδομένα έναν ακέραιο(ni), που είναι το πλήθος των θέσεων που έχει, και έναν πίνακα δεικτών σε πλάσματα. Μέσα στον constructor της δηλώνονται δυο πίνακες ονομάτων(ένανς κάλων πλασμάτων και ένας κακών πλασμάτων), δημιουργείται ο πίνακας δεικτών δυναμικά και αστικοποιείται με καλά ή κακά

πλάσματα(τυχαία ένα από τα δυο κάθε φορά),(το όνομα του κάθε πλάσματος που δημιουργείται λαμβάνεται τυχαία από τον αντίστοιχο πίνακα ονομάτων, προσαρτώντας του τον αριθμό που αντιστοιχεί στην θέση του στην κοινότητα ώστε να γίνει μοναδικό). Η συνάρτηση beat παίρνει σαν όρισμα την θέση του πλάσματος που θα γίνει beat και κάνει beat στο πλάσμα αυτό. Ομοίως και η συνάρτηση bless. Η συνάρτηση clone\_next παίρνει σαν όρισμα την θέση του πλάσματος που θα κλωνοποιηθεί, ελέγχει αν αυτή η θέση είναι πριν την τελευταία και αν είναι τότε κάνει clone αυτό το πλάσμα στην επόμενη θέση της κοινότητας. Αν η θέση είναι η τελευταία τότε κάνει clone στην πρώτη θέση της κοινότητας. Η συνάρτηση clone\_zobies παίρνει σαν όρισμα την θέση του πλάσματος που θα κλωνοποιηθεί, ελέγχει αν αυτή η θέση είναι πριν την τελευταία και αν είναι τότε κάνει clone αυτό το πλάσμα στις επόμενες συνεχόμενες θέσεις της κοινότητας που περιέχουν ζόμπι. Αν η θέση είναι η τελευταία τότε δεν κάνει clone. Η συνάρτηση no\_of\_good επιστρέφει το πλήθος των κάλων πλασμάτων(που δεν είναι ζόμπι). Επιπλέον έχει μια έξτρα συνάρτηση-μέλος print(που εκτυπώνει την κοινότητα καλώντας την print του κάθε πλάσματος που περιέχει).

#### Main:

Στην main έχω προσθέσει(σε σχόλια-για testing) σε τρία σημεία(μια στην αρχή, μια σε κάθε επανάληψη που γίνεται beat/bless και μια στο τέλος) να καλείται μια print της κοινότητας, η οποία την εκτυπώνει(εκτυπώνει την θέση, το όνομα και την ζωή του κάθε πλάσματος). Όσον αφορά το **α' ερώτημα** δημιουργείται μια κοινότητα με N πλάσματα, περνώντας της τα ορίσματα N, L(που θα «καταλήξει» στην κάθε κλάση creature), good\_thrsh(που θα «καταλήξει» στην κάθε κλάση good\_creature), bad\_thrsh(που θα «καταλήξει» στην κάθε κλάση bad\_creature). Όσον αφορά το **β' ερώτημα** επιλέγεται τυχαία μια θέση της κοινότητας και επιλέγεται τυχαία και μια ενέργεια(beat ή bless) εφαρμόζοντάς την στο πλάσμα που βρίσκεται σε αυτήν την θέση. Αυτό επαναλαμβάνεται M φορές. Όσον αφορά το **γ' ερώτημα** εκτυπώνονται τα αντίστοιχα μηνύματα αναλόγως το αν η κοινότητα αποτελείται μόνο από καλά πλάσματα(που δεν είναι ζόμπι) ή μόνο από ζόμπι.

#### Υ.Γ.:

Σχετικά με το πως συνέδεσα τα στιγμιότυπα της κλάσης creature με την κλάση creature\_society στην οποία ανήκουν, υλοποίησα τον δεύτερο εναλλακτικό τρόπο που μας δίνεται, μετασχηματίζοντας τον λίγο. Δηλαδή, η κοινότητα περιέχει μια ακολουθία δεικτών σε πλάσματα και τα πλάσματα έναν δείκτη στην κοινότητα που ανήκουν. Επιπλέον, τα πλάσματα περιέχουν έναν ακέραιο(index) που είναι η τάξη της θέσεως του πλάσματος στην κοινότητα.

Οι συναρτήσεις-μέλη is\_a\_zombie, get\_life, get\_index, get\_pointer, print της κλάσης creature, is\_a\_good της κλάσης good\_creature, is\_a\_good της κλάσης bad\_creature, no\_of\_good, no\_of\_zombies, print της κλάσης creature society είναι const, εφόσον δεν τροποποιούν το αντικείμενο για το οποίο καλούνται.

Εφόσον το N είναι μια ακέραια μεταβλητή που έχει πάρει τιμή κατά την εκτέλεση του προγράμματος, δηλαδή η τιμή του δεν είναι γνωστή κατά την μεταγλώττιση, δημιουργούμε τον αντίστοιχο πίνακα (pinakas[N] -πίνακα πλασμάτων(δεικτών) που περιέχει η society) με δυναμικό τρόπο.

Η κοινότητα έχει σαν μέλος-δεδομένο έναν πίνακα από δείκτες σε creature, και όχι απλά ένα πίνακα από creatures, έτσι ώστε να μπορέσει να δημιουργηθεί δυναμικά το κάθε καλό/κακό πλάσμα που θα τοποθετηθεί σε αυτόν και να μπορέσει να καλεστεί ο αντίστοιχος constructor του με τα ορίσματα που θέλουμε(και να λειτουργήσει σωστά η κληρονομικότητα).

Θεόδωρος Μωραΐτης, AM: 1115202000150