Αναφορά για την 1η εργασία στο μάθημα «Προγραμματιστικές τεχνικές» του 2ου Εξαμήνου

Στο πρόγραμμα ορίζονται οι ακόλουθες κλάσεις με τα αντίστοιχα πεδία και μεθόδους:

* Κλάση Person που λειτουργεί ως wrapper class στην κλάση Character. Περιέχει:

Πεδίο int hitpoints: Μεταβλητή η οποία κρατάει τους πόντους ζωής ενός Person, και κατ’ουσίαν ενός χαρακτήρα εφόσον υπάρχει αντιστοιχία ένα προς ένα ανάμεσα σε χαρακτήρες και Person. Από τους πόντους αυτούς αναγνωρίζουμε άμεσα αν ο χαρακτήρας είναι νεκρός η ζωντανός και αν είναι ζωντανός πόσους πόντους ζωής έχει ακόμα.

Πεδίο bool is\_dead: Η τιμή του οποίου δηλώνει το αν είναι ζωντανός ένας Person και κατ’επέκταση ο χαρακτήρας με τον οποίο αυτός είναι συνδεδεμένος. Η τιμή αυτής της μεταβλητής είναι true αν και μόνο αν τα hitpoints είναι ίσο με το μηδέν. Σε κάθε άλλη περίπτωση η μεταβλητή αυτή λαμβάνει την τιμή.

Πεδίο Character \* has\_character: Πολυμορφικός δείκτης στην class Character. Ο δείκτης αυτός χρησιμοποιείται για την εξασφάλιση της ένα προς ένα αναλογίας μεταξύ Person και χαρακτήρων. Μέσω αυτού του δείκτη ένας Person συνδέεται με κάποιον χαρακτήρα και έτσι έχουμε την ζεύξη των δυο από την μεριά του Person.

Constructor: Αρχικοποιεί τα προαναφερθέντα πεδία

Μέθοδο void setHas\_character(Character \*p): Συνδέει το αντικείμενο δίνοντας την τιμή του ρ στο πεδίο has\_character. Δηλαδή πραγματοποιεί επί της ουσίας την σύναψη από την πλευρά του Person.

Getter int getHitpoints(): Επιστρέφει την τιμή του πεδίου hitpoints

Setter void setHipoints(int b): Αρχικοποιεί η αναθέτει μια τιμή στο πεδίο hitpoints, την τιμή της μεταβλητής b.

Bool Is\_dead(): Μέθοδος η οποία επιστρέφει μια τιμή αληθείας αναλόγως της κατάστασης στην οποία έχει περιέλθει ο Person και κατά αντιστοιχία ο χαρακτήρας με τον οποίον αυτός είναι συνδεδεμένος. Αν ο χαρακτήρας είναι νεκρός(μηδέν πόντοι ζωής) τότε επιστρέφει την τιμή true αλλιώς θα επιστρέψει την τιμή false.

Void minus\_hp(Person &p): Μέθοδος η οποία καλεί την μέθοδο lower\_hp() του χαρακτήρα με τον οποίο είναι συνδεδεμένος ο Person p ώστε αυτή να επιτελέσει το έργο της το οποίο θα περιγραφεί στη συνέχεια.

Void hit(Person \*p): Μέθοδος η οποία καλεί την μέθοδο attack του χαρακτήρα με τον οποίο είναι συνδεδεμένος ο Person και της περνά σαν όρισμα τον p, που δέχεται η μέθοδος αυτή στην είσοδο της ως όρισμα.

* Κλάση Character που περιέχει:

Πεδίο Person \* is\_attached\_to: Κρατάει το αντικείμενο στο οποίο είναι συνδεδεμένος ο χαρακτήρας ώστε να υπάρχει αμφίδρομη και ένα προς ένα αντιστοιχία μεταξύ χαρακτήρων και Persons.

virtual string get\_character\_type(Person &p):Μέθοδος η οποία είναι virtual δηλώνεται εντός της κλάσης Character αλλά το κυρίως σώμα της δηλώνεται έξω από την προαναφερθείσα κλάση γιατί μέσα σε αυτή χρησιμοποιούνται πεδία της κλάσης Person η οποία δηλώνεται μετά την κλάση Character.

virtual void attack(Person &p):Μέθοδος η οποία θα οριστεί αναλυτικότερα εντός των κλάσεων Good,Bad,Zombie. Η λειτουργιά αυτής της μεθόδου είναι η πραγματοποίηση επίθεσης από έναν χαρακτήρα σε έναν άλλον.

void lowerhp():Μέθοδος η οποία ορίζεται έξω από την κλάση Character και η χρήση αυτής γίνεται για την μείωση των πόντων ζωής κάποιου χαρακτήρα κατά 10, σύμφωνα με την εκφώνηση.

void convert():Μέθοδος η οποία θα οριστεί αναλυτικότερα εντός των κλάσεων Good,Bad,Zombie. Η λειτουργιά αυτής της μεθόδου είναι η μετατροπή ενός είδους χαρακτήρα σε κάποιο άλλο είδος.

void setIs\_attach\_to(Person \* p) : Δίνει στο πεδίο is\_attached\_to την τιμή p. Ολοκληρώνοντας έτσι την ένα προς ένα σύνδεση τους.

~virtual Character():Ο οποίος είναι ένας virtual Destructor.

* Κλάσεις Good, Bad, Zombie οι οποίες κληρονομούν από την κλάση Character και:

Υλοποιούν την μέθοδο string get\_character\_type(Person &p) η οποία επιστρέφει τον τύπο της του χαρακτήρα με τον οποίο είναι συνδεδεμένος ο Person p,βεβαία κάτι τέτοιο θα μπορούσε να γίνεται και χωρίς να δέχεται όρισμα η μέθοδος αυτή εφόσον στην προκειμένη άσκηση η χρήση της περιορίζεται στον εντοπισμό του είδους του χαρακτήρα που την καλεί.

Υλοποιούν την μέθοδο void attack(Person &p): Εκτελεί την επίθεση σύμφωνα με τους κανόνες του παιχνιδιού σε έναν χαρακτήρα με τον οποίον είναι συνδεδεμένος ο Person p ο οποίος εισέρχεται ως όρισμα.

Υλοποιούν την μέθοδο void convert(int k,Person \*p): Μετατρέπει τον χαρακτήρα στον οποίο δείχνει ο p σε άλλον τύπο αναλόγως της τιμής της μεταβλητής κ (Good, Bad, Zombie), διαγράφοντας παράλληλα το προηγούμενο είδος χαρακτήρα δημιουργεί ένα νέο είδος και το συνδέει στο Person ρ και αντίστοιχα συνδέει στον νέο χαρακτήρα το αντικείμενο Person p.

Υλοποιούν την μέθοδο void lower\_hp(): Μειώνει τους πόντους ζωής του χαρακτήρα κατά 10

* Κλάση World η οποία περιέχει:

Πεδίο int number:Που κρατάει τον αριθμό των χαρακτήρων που δίνεται από τον χρηστή ως μέγεθος του κόσμου.

Πεδίο list<Person\*> first: Φυλάσσονται δείκτες τύπου Person ώστε να γίνεται εύκολη η δυναμική παραχώρηση μνήμης σε κλάση τύπου list ώστε να δημιουργεί ο κόσμος χωρίς χρήση περισσότερης μνήμης από την απαιτουμένη.

Μέθοδο void create(): Γεμίζει τη λίστα first με number δείκτες σε κλάσεις Person και στο καθένα από αυτά συνδέει έναν χαρακτήρα Good, Bad ή Zombie με ίση πιθανότητα. Στην συνέχεια εμφανίζει μια φορά την αρχική κατάσταση του κόσμου και ξεκινά να κάνει τις επιθέσεις. Θεωρείται μια μεταβλητή b η οποία μετρά τους γύρους που γίνονται μέσα στο παιχνίδι και μια άλλη μεταβλητή telos η οποία σταματάει την κυκλική επανάληψη όταν εντοπίσει ότι μια φυλή/κλάση/είδος χαρακτήρων έχει επικρατήσει στον κόσμο. Σε κάθε γύρω με την χρήση δεικτών και την διενέργεια των κατάλληλων ελέγχων, ώστε να μην εκτελούν επιθέσεις οι νεκροί και να μην γίνονται επιθέσεις σε αντικείμενα ίδιας κλάσης, γίνονται επιθέσεις με κυκλικό τρόπο. Μετά το τέλος των επιθέσεων αφαιρούνται από την λίστα οι νεκροί χαρακτήρες αποδεσμεύοντας χώρο από την μνήμη και μειώνοντας το μέγεθος της λίστας. Κατόπιν γίνεται κλήση της μεθόδου print η οποία τυπώνει την κατάσταση στην οποία έχει περιέλθει ο κόσμος. Και στη συνέχεια καλείται η μέθοδος print1 η οποία ελέγχει αν έχει επικρατήσει ένα είδος χαρακτήρων και αν αυτό έχει γίνει αλλάζει την τιμή της μεταβλητής telos ώστε να σταματήσουν οι επαναλήψεις. Τέλος αναλόγως της επιστρεφόμενης από την print1 τιμής εμφανίζεται το αποτέλεσμα του κόσμου, δηλαδή ποιο είδος έχει επικρατήσει.

Μέθοδο void print(list<Person\*> &first): Τυπώνει το περιεχόμενο της λίστας first με τον τρόπο που ζητείται από την εκφώνηση.

Μέθοδο int print1(list<Person\*> &first): Ελέγχει αν μια κλάση έχει επικρατήσει στον κόσμο και επιστρέφει μια τιμή αναλόγως(1 για επικράτηση καλων,2 για επικράτηση κακων,3 για επικράτηση ζόμπι και 4 για ισοπαλία).

Μέθοδο void setNumber(int a): Η οποία δέχεται σαν όρισμα το μέγεθος του κόσμου, το εκχωρεί στην μεταβλητή number και στη συνέχεια καλεί την μέθοδο create ώστε να ξεκινήσει η όλη διαδικασία του παιχνιδιού.

Το σχεδιάγραμμα των κλάσεων που περιγράφηκαν ώστε να δηλώνεται η σχέση τους είναι το εξής:

Person

Character

Zombie

Bad

Good

H int main(void) του προγράμματος αρχικοποιεί την τυχαία ακολουθία που θα χρειαστεί για την κατανομή των χαρακτήρων στον κόσμο καθώς και στις επιθέσεις μεταξύ τους (srand(time(NULL)). Ύστερα παίρνει έναν ακέραιο αριθμό από τον χρήστη και φτιάχνει ένα αντικείμενο της κλάσης World και καλεί την μέθοδο setNumber του αντικειμένου αυτό και της περνά ως όρισμα αυτόν τον αριθμό ώστε να ξεκινήσει η όλη διαδικασία του παιχνιδιού. Τέλος διαγράφεται το αντικείμενο World.