



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών

Προγραμματιστικές Τεχνικές

1^η Άσκηση 2015-2016: The Good, the Bad and the Zombie

Σημείωση: Οι ασκήσεις αποτελούν μέρος της εργαστηριακής εξάσκησης στο μάθημα “Προγραμματιστικές Τεχνικές”. Η υποβολή του σχετικού προγράμματος έμμεσα δηλώνει ότι είστε οι μοναδικοί συγγραφείς της λύσης της άσκησης. Εάν το πρόγραμμα ή μέρος του έχει παρθεί από άλλη πηγή θα πρέπει να αναφέρεται η πηγή και ο συγγραφέας του.

Θεωρούμε ότι έχουμε ένα σύνολο από χαρακτήρες (κλάση Character) που ανήκουν σε μια από τις παρακάτω κατηγορίες:

- Οι καλοί χαρακτήρες (κλάση Good).
- Οι κακοί χαρακτήρες (κλάση Bad).
- Τα zombie (κλάση Zombie).

Οι τρεις ομάδες αυτές είναι αντίπαλες μεταξύ τους.

Κάθε χαρακτήρας θα ξεκινάει με 100 hit points, τα οποία θα μειώνονται κάθε φορά που δέχεται χτύπημα από κάποιον αντίπαλο χαρακτήρα. Επίσης, μπορεί να επιτίθεται σε αντίπαλους χαρακτήρες, πράγμα που μειώνει τα hit points του αντιπάλου κατά 10. Κάθε χαρακτήρας μπορεί να επιτεθεί μόνο σε χαρακτήρες διαφορετικής κλάσης.

Οι κακοί χαρακτήρες μπορούν να επιτίθενται μόνο σε καλούς χαρακτήρες και zombie.

Οι καλοί χαρακτήρες μπορούν να επιτίθενται μόνο σε κακούς χαρακτήρες και zombie. Κάθε επίθεση που γίνεται από καλό χαρακτήρα σε zombie, έχει πιθανότητα 20% να γιατρέψει το zombie και να το μετατρέψει είτε σε καλό (με πιθανότητα 40%) είτε σε κακό (με πιθανότητα 60%) χαρακτήρα.

Τα zombie μπορούν να επιτεθούν είτε σε καλούς είτε σε κακούς χαρακτήρες. Οι επιθέσεις των zombie έχουν 10% πιθανότητα να μετατρέψουν τον χαρακτήρα στον οποίον επιτίθενται σε zombie.

Κάθε φορά που κάποιος χαρακτήρας μετατρέπεται σε χαρακτήρα κάποιας άλλης κλάσης, θα διατηρεί τα hit points που είχε πριν από την επίθεση του αντιπάλου.

Για λόγους που σχετίζονται με την ευκολία υλοποίησης, κάθε χαρακτήρας θα πρέπει να δημιουργείται μέσα σε μια κλάση συσκευαστή (wrapper class) με όνομα person, η οποία θα επιτρέπει πρόσβαση στις μεθόδους του χαρακτήρα της. Όποτε γίνεται μετατροπή από ένα είδος χαρακτήρα σε άλλο, απλά θα αλλάζει το αντικείμενο το οποίο περιέχεται στο person, ενώ το αντικείμενο person θα παραμένει σταθερό.

Θεωρούμε κυκλική λίστα από persons που περιέχουν χαρακτήρες, οι οποίοι μπορούν να ανήκουν σε μία από τις παραπάνω κλάσεις. Η λίστα αυτή θα καθορίζει τη θέση των χαρακτήρων στον χώρο.

Θεωρούμε ότι κάθε χαρακτήρας μπορεί να επιτεθεί στον επόμενο του και ο τελευταίος της λίστας στον πρώτο. Για παράδειγμα, αν έχουμε τη λίστα [A, B, C], ο χαρακτήρας A μπορεί να επιτεθεί στον B, ο B μπορεί να επιτεθεί στον C και ο C στον A, εάν φυσικά το επιτρέπουν οι κλάσεις τους.

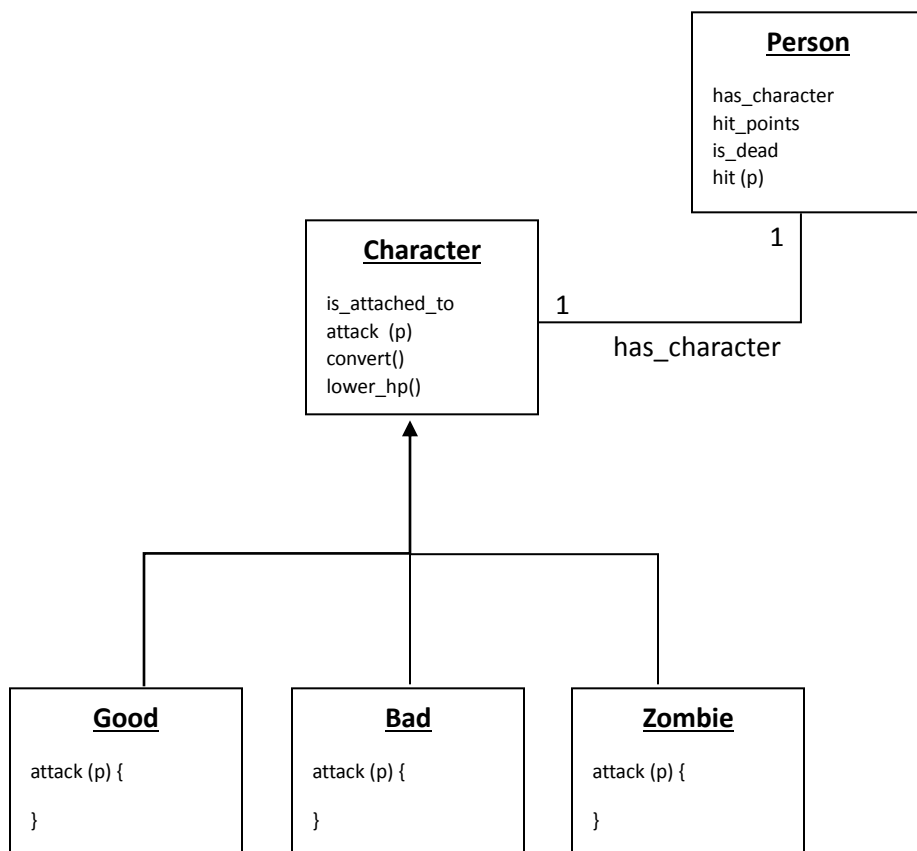
Κάθε χαρακτήρας επιτίθεται με τη σειρά στον επόμενο του. Όταν όλοι έχουν κάνει μια επίθεση θεωρούμε ότι έχει γίνει ένας πλήρης γύρος.

Κάθε φορά που κάποιος χαρακτήρας πεθαίνει (τα hit points του μηδενίζονται), το αντικείμενο person που τον περιέχει πρέπει να διαγράφεται από τη λίστα. Κάθε φορά που ένα αντικείμενο αφαιρείται από τη λίστα θα πρέπει να ελευθερώνεται η μνήμη που έχει δεσμευτεί για αυτό.

Για την υλοποίηση της λίστας μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη list της C++. Εάν δεν χρησιμοποιήσετε την std::list, θα πρέπει να δημιουργήσετε μια καινούρια κλάση – κόμβο για να υλοποιήσετε τη λίστα. Σε κάθε περίπτωση, η υλοποίηση της λίστας πρέπει να είναι ξεχωριστή από την υλοποίηση των κλάσεων των χαρακτήρων (τα αντικείμενα των κλάσεων-χαρακτήρων δεν θα περιέχουν δείκτες ή πληροφορία για τη θέση τους στη λίστα).

Το παιχνίδι τερματίζεται όταν νικήσει μια ομάδα (όταν δηλαδή υπάρχουν χαρακτήρες μόνο αυτής της ομάδας στη λίστα) ή μετά από 100 γύρους. Αν νικήσει μια ομάδα θα πρέπει να εμφανίζεται στον χρήστη ένα ανάλογο μήνυμα.

Η υλοποίηση των κλάσεων πρέπει να πραγματοποιηθεί με βάση το παρακάτω διάγραμμα:



- Κλάση Person: Η κλάση wrapper των χαρακτήρων, που επιτρέπει την πρόσβαση στις μεθόδους του αντικειμένου από μία άλλη κλάση. Περιέχει:
 - has_character: έναν δείκτη σε αντικείμενο τύπου Character.
 - hit_points: έναν ακέραιο αριθμό που θα ορίζει τα hit points του χαρακτήρα.
 - is_dead: μία λογική μεταβλητή που θα δηλώνει αν ο χαρακτήρας έχει πεθάνει ή όχι.
 - hit (p): μία μέθοδο, η οποία θα δέχεται ως όρισμα έναν δείκτη σε κάποιον χαρακτήρα και θα καλεί τη μέθοδο attack του χαρακτήρα με τον οποίο είναι συνδεδεμένο το αντικείμενο person (has_character) και θα της περνάει ως όρισμα τον p.
 - Όποιες άλλες μεθόδους θεωρείτε απαραίτητες για την ανάκτηση πληροφοριών από ένα αντικείμενο person (π.χ. get_hp()).
- Κλάση Character: Η βασική κλάση του παιχνιδιού, από την οποία κληρονομούν οι υπόλοιπες. Περιέχει:
 - is_attached_to: έναν δείκτη στο αντικείμενο τύπου person, με το οποίο είναι συνδεδεμένος ο χαρακτήρας.
 - get_character_type (p): μία μέθοδο που θα επιστρέφει τον τύπο του χαρακτήρα με τον οποίο είναι συνδεδεμένος ο person p. Για την ανάκτηση του τύπου του χαρακτήρα θα χρησιμοποιηθεί η συνάρτηση typeid() που θα περιγράψουμε παρακάτω.
 - attack (p): μία μέθοδο που πραγματοποιεί την επίθεση στον p. Αυτή η μέθοδος θα καλεί την get_character_type (p), θα πραγματοποιεί τους κατάλληλους ελέγχους και θα καλεί, ανάλογα με την περίπτωση, είτε την lower_hp() είτε την convert() του χαρακτήρα που δείχνει ο person p.
 - lower_hp(): η οποία μειώνει τα hit_points του συνδεδεμένου person (is_attached_to) κατά 10.
 - convert(): η οποία θα πραγματοποιεί τις αλλαγές από έναν τύπο χαρακτήρα σε κάποιον άλλο.
- Κλάση Good: Η κλάση Good κληρονομεί την κλάση Character.
- Κλάση Bad: Η κλάση Bad κληρονομεί την κλάση Character.
- Κλάση Zombie: Η κλάση Zombie κληρονομεί την κλάση Character.

Θα πρέπει να ελέγξετε τί επιστρέφει η μέθοδος get_character_type(), καθώς εξαρτάται από τον τρόπο υλοποίησης των υπόλοιπων κλάσεων. Ο έλεγχος θα γίνει μέσω της συνάρτησης typeid() της C++. Προκειμένου να χρησιμοποιήσετε τη συνάρτηση αυτή, θα πρέπει να κάνετε πρώτα δυο πράγματα:

- Να συμπεριλάβετε την βιβλιοθήκη που περιέχει τη συνάρτηση typeid: #include <typeinfo>.
- Να «δείξετε» στον compiler ότι θα χρησιμοποιηθεί η καινούρια έκδοση της C++: η έκδοση C++11. Για να το κάνετε αυτό πρέπει να **προσθέσετε** την εντολή “ -std=c++11” (μην σβήσετε τίποτα, απλά προσθέστε την στο τέλος και προσοχή να υπάρχει κενό πριν το -std) στο eclipse, στα παρακάτω δυο σημεία:

- Project -> Properties -> C/C++ Build -> Settings -> G++ Compiler -> Miscellaneous -> Other Flags:
έτσι ώστε να γράφει, π.χ. “-c -fmessage-length=0 -std=c++11”.
- Project -> Properties -> C/C++ General -> Preprocessor Include Paths, Macros etc. -> Providers -> CDT GCC Built-in Compiler Settings -> Command to get compiler specs:
Πιθανότατα να είναι επιλεγμένη η επιλογή “Use global provider shared between projects” η οποία πρέπει να βγει. Τελικά θα γράφει κάτι τέτοιο: `{COMMAND} {FLAGS} -E -P -v -Dd “{INPUTS}” -std=c++11`.
- Κλάση World: Είναι η κλάση που υλοποιεί τη στρατηγική του παιχνιδιού. Η κλάση αυτή θα πρέπει να δημιουργεί και να αρχικοποιεί (όπως θα δούμε παρακάτω) τη λίστα με τους χαρακτήρες, να εκτελεί έναν γύρο παιχνιδιού, να ελέγχει για πεθαμένους χαρακτήρες και να τους διαγράφει απ' τη λίστα, να ελέγχει αν έχει τερματιστεί το παιχνίδι και να μπορεί να τυπώνει την παρούσα κατάσταση του κόσμου.
 - Έξω από την κλάση αυτή δεν πρέπει να έχει κανένας πρόσβαση στα στοιχεία της λίστας.
 - Η εκτύπωση της κατάστασης του κόσμου πρέπει να δείχνει στον χρήστη τους χαρακτήρες του κόσμου, με τη θέση και τα hit points τους.
 - Η αρχικοποίηση της λίστας θα γίνει ως εξής: Θα δίνεται στην κλάση World το μέγεθος του κόσμου (ένας integer), και η κλάση θα πρέπει να δημιουργεί μια λίστα με το μέγεθος αυτό και να την γεμίσει με αντικείμενα τύπου Good, Bad ή Zombie, τυχαία με ίση πιθανότητα το καθένα.

Για την απόκτηση τυχαίων αριθμών θα σας δοθεί μία έτοιμη μέθοδος που θα δέχεται ένα seed τύπου int και θα επιστρέφει έναν int στο διάστημα [0,99], τον οποίο θα μπορείτε να χρησιμοποιήσετε για να δημιουργείτε τις απαραίτητες τυχαίες τιμές σύμφωνα με τις πιθανότητες που δόθηκαν.

Είσοδος: Η είσοδος του προγράμματος θα είναι ένας ακέραιος αριθμός ο οποίος θα ορίζει το μέγεθος του κόσμου (αριθμός χαρακτήρων).

Έξοδος: Το πρόγραμμα, όταν τερματιστεί, θα πρέπει να τυπώνει ένα μήνυμα που να ορίζει ποια ομάδα νίκησε. Επίσης θα πρέπει σε κάθε γύρο να τυπώνει την κατάσταση του κόσμου που θα έχει την ακόλουθη μορφή:

[Bad-60, Good-40, Zombie-40, Zombie-50]

δηλαδή, θα εμφανίζει τον τύπο που περιέχεται στο person, ακολουθούμενο από ένα “-” και τον αριθμό hitPoints που απομένουν.

ΣΥΝΘΗΚΕΣ ΤΕΡΜΑΤΙΣΜΟΥ: Το πρόγραμμα τερματίζει είτε μετά από 100 γύρους, είτε όταν απομένουν χαρακτήρες μόνο ενός είδους.

ΟΔΗΓΙΕΣ

- Η εργασία είναι αυστηρώς ατομική. Σε περίπτωση αντιγραφής η άσκηση θα μηδενίζεται.
- Το πρόγραμμα θα πρέπει να υλοποιηθεί σε C++.
- Το πρόγραμμά θα πρέπει να υλοποιεί όλες τις κλάσεις και τις διεπαφές όπως περιγράφηκαν παραπάνω.
- Το πρόγραμμα θα πρέπει να χρησιμοποιεί τα πλεονεκτήματα του αντικειμενοστραφούς προγραμματισμού (ιεραρχία κλάσεων, κληρονομικότητα, πολυμορφισμό κ.λπ.).

Παράδοση:

- Η παράδοση της εργασίας γίνεται μέσω της ιστοσελίδας του μαθήματος στο **mycourses.ntua.gr**, στην ενότητα “Εργασίες”.
- **Παραδοτέα:** Η άσκηση θα παραδίδεται υπό την μορφή ενός ZIP αρχείου, το οποίο θα περιέχει τα ακόλουθα:
 - Τον πηγαίο κώδικα των κλάσεων που υλοποιούν την άσκηση, επομένως μόνο τα αρχεία με κατάληξη .cpp (ή .hpp), **όχι τα εκτελέσιμα αρχεία!!**
 - Μια σύντομη επεξήγηση του προγράμματος σας σε μορφή αρχείου κειμένου (.doc, .pdf, κ.λ.π.), **αυστηρά έως τέσσερις σελίδες**, στο οποίο θα περιέχεται ένα διάγραμμα που δείχνει τη δομή και τις σχέσεις μεταξύ των κλάσεων του σχεδιασμού σας.