

# Αντικειμενοστραφής Προγραμματισμός 2017-2018

## Εργασία 4



Σε αυτήν την εργασία θα προσπαθήσουμε να αντιληφθούμε τις πολύ σημαντικές έννοιες της κληρονομικότητας και της υπερφόρτωσης τελεστών. Μια κλάση μπορεί να κληρονομεί από μια άλλη αποκτώντας έτσι πρόσβαση στα public και στα protected μέλη της. Έτσι γλυτώνουμε χρόνο, γιατί δεν χρειάζεται να ξαναγράψουμε ίδιο κώδικα για την καινούρια κλάση που έχουμε φτιάξει. Με την υπερφόρτωση τελεστών όπως είδατε και στο μάθημα, γίνεται εύκολη η ενσωμάτωση νέων τύπων δεδομένων/κλάσεων στο προγραμματιστικό σας περιβάλλον. Επίσης θα διασκεδάσουμε στέλνοντας τους αγαπημένους μας samurai στην σχολή ninja και μετά θα τους δούμε να παλεύουν σε μια όλοι εναντίον όλων μάχη. **Σε αυτήν την άσκηση μην παραδώσετε τη main σας!**

### Ερωτήματα:

1. Χρησιμοποιήστε τον κώδικα της 3ης εργασίας. Αλλάξτε όλες τις μεταβλητές-μέλη των κλάσεων Weapon και Samurai σε protected και τις συναρτήσεις των κλάσεων σε public. Θέλουμε να κληρονομούνται στις κλάσεις-παιδιά. Επίσης ορίστε constructor χωρίς ορίσματα για τις παραπάνω κλάσεις και δηλώστε τους ως explicit. Επίσης, οι constructors και οι destructors των προηγούμενων ασκήσεων να μην εκτυπώνουν πια μηνύματα.
2. Δημιουργήστε μια καινούρια κλάση **ExoticWeapon**. Η κλάση ExoticWeapon πρέπει να είναι απόγονος της Weapon. Η ExoticWeapon να έχει μία μεταβλητή-μέλος τύπου string origin. Να θυμάστε ότι οι κλάσεις που κληρονομούν από άλλες στην ουσία είναι ταυτόχρονα και υλοποιήσεις του παραπάνω τύπου, άρα ο constructor τους μπορεί να αρχικοποιεί και τα πεδία-μεταβλητές του πατέρα αν έχει πρόσβαση. Υλοποιήστε τον constructor ως εξής: **ExoticWeapon(): Weapon()**. Μην ξεχνάτε getter, setter για τη μεταβλητή-μέλος της ExoticWeapon. **Όταν θέλετε να αναθέσετε τιμές στις μεταβλητές μέλη των κλάσεων να χρησιμοποιείτε τους setters και όταν θέλετε να χρησιμοποιήσετε τις μεταβλητές-μέλη να το κάνετε με τους getters.**
3. Δημιουργήστε μια καινούρια κλάση **Ninja**. Η κλάση Ninja να είναι απόγονος της Samurai. Η Ninja να έχει μία μεταβλητή-μέλος τύπου int grade. Όπως πάντα getter, setter.
4. Ορίστε μια καινούρια **void** συνάρτηση στη Ninja την **pickExoticWeapon(ExoticWeapon &a)** που θα θέτει στο samuraiWeapon το όνομα του ExoticWeapon. Όπως καταλαβαίνετε ο

- Samurai δεν μπορεί να χρησιμοποιήσει ExoticWeapon, αλλά ο ninja μπορεί να χρησιμοποιήσει και Weapon και exoticWeapon.
5. Δημιουργήστε μια καινούρια κλάση **NinjaSchool**. Ορίστε μία καινούρια μεταβλητή-μέλος πίνακα 10 θέσεων με όνομα `ninjaClass`, ο οποίος θα πρέπει να γεμίζει με αντικείμενα τύπου `Ninja`. Αρα χρειαζόμαστε το εξής: `Ninja* ninjaClass`. Ορίστε ένα καινούριο πεδίο `int numberOfStudents`. Ο constructor να περιέχει τη δήλωση `ninjaClass = new Ninja[10]`; και να αρχικοποιεί την `numberOfStudents` στο 0.
  6. Για να υπάρχει σωστή δομή στα παραδοτέα, οι κλάσεις να είναι δηλωμένες στο αρχείο που παραδίδετε με την εξής σειρά:
    1. `Weapon`
    2. `ExoticWeapon`
    3. `Samurai`
    4. `Ninja`
    5. `NinjaSchool`
  7. Να υπερφορτώσετε τους μοναδιαίους τελεστές `++` και `--`, και στις δύο μορφές τους (prefix και postfix), ώστε να αυξάνουν ή να μειώνουν τη μεταβλητή-μέλος `numberOfStudents` της `Ninja School`.
  8. Ορίστε μια καινούρια `public` συνάρτηση στη `NinjaSchool` η οποία θα πρέπει να επιστρέφει έναν `Ninja` με όνομα συνάρτησης **`startTrainingNinja(Samurai &a)`**. Η συγκεκριμένη συνάρτηση θα δέχεται ως όρισμα ένα αντικείμενο τύπου `Samurai` και θα φροντίζει ώστε το αντικείμενο `Samurai` να μετασχηματίζεται σε αντικείμενο `Ninja` με σωστή την χρήση των getters/setters. Θέστε την τιμή του `grade` στο 0 όταν δημιουργείτε τον `Ninja`.
  9. Ορίστε μια καινούρια `public` συνάρτηση στη `NinjaSchool` τύπου `void` με όνομα **`addNinjaToNinjaClass(Ninja &a)`**, η οποία θα βάζει έναν καινούριο `Ninja` μέσα στο array της `ninjaClass`. Αν ο `Ninja` που θέλουμε να βάλουμε στην τάξη είναι μικρότερος από 18 χρονών να εμφανίζεται το μήνυμα: **“This class is for grownups!”**.
  10. Ορίστε μια καινούρια συνάρτηση στη `NinjaSchool` τύπου `void` την `brawlWithNinjas()`, η οποία θα χρησιμοποιεί τον πίνακα της `ninjaClass`. Αφού εξετάσει τα στοιχεία του να επιστρέφει το όνομα του `Samurai` με το μεγαλύτερο `grade`.

## Διευκρινίσεις/Παρατηρήσεις

- Όλα τα κενά αποτελούνται από ένα space και στα μηνύματα που χρειάζεται να εκτυπωθούν μην βάζετε `endl` χαρακτήρα.
- Το παραδοτέο να ονομαστεί `XXXX.cpp` και στην πρώτη γραμμή του να περιέχει το `//XXXX` όπου `XXXX` ο Α.Μ. σας.
- Το `XXXX.cpp` να μην περιέχει τη συνάρτηση `main()`.
- Οι τρεις παραπάνω διευκρινίσεις είναι πολύ σημαντικές γιατί οι εργασίες θα βαθμολογηθούν εν μέρει αυτόματα.
- Μη παραδώσετε εργασία που δεν κάνει `compile` σε τουλάχιστον έναν από τους compilers: `Dev C++` (σε Windows συστήματα), `g++` (σε linux/MacOS συστήματα). Αν έχετε την δυνατότητα, προτιμήστε τον `g++`. Εργασίες που δεν μεταγλωττίζονται βαθμολογούνται αυτόματα με μηδέν.
- Η εργασία είναι ατομική. Σε περίπτωση αντιγραφής, όλες οι ίδιες/παρόμοιες εργασίες θα μηδενιστούν και οι παραβάτες θα θεωρηθούν μετεξεταστέοι στο Εργαστήριο, δλδ δεν θα μπορέσουν να πάρουν μέρος στην γραπτή εξέταση για αυτή την ακαδημαϊκή χρονιά.

Καλή επιτυχία

