

Ασκήσεις 9 – Κλάσεις-2 (Δόμηση, υπερφόρτωση τελεστών)

9Α.1 - Μέθοδοι Δόμησης

Να ορισθεί κατάλληλη κλάση για αντικείμενα τύπου φοιτητή που να περιλαμβάνει:

Ιδιότητες:

- Όνομα, αριθμός μητρώου (ακέραιος) και μέση βαθμολογία (πραγματικός αριθμός).

Μέθοδοι (συναρτήσεις-μέλη)

- μια μέθοδο για την εκτύπωση-εμφάνιση στην οθόνη των παραπάνω ιδιοτήτων.
- τρεις μεθόδους δόμησης (δομητές-constructors) για την απόδοση –αν θέλουμε- αρχικών τιμών στα αντικείμενα κατά την δήλωσή τους, ούτως ώστε να μπορούμε να έχουμε δηλώσεις των τριών παρακάτω μορφών:

```
foititis x("Nikos", 6023, 8.3), y("Yiannis"), z;
```

- Η πρώτη μέθοδος δόμησης θα αφορά, δηλ. θα εκτελείται, σε δηλώσεις αντικειμένων σαν το x παραπάνω. Θα δέχεται *τρία* ορίσματα –συγκεκριμένα ένα string, έναν ακέραιο και έναν double- και θα τα αποδίδει στις ιδιότητες (μεταβλητές-μέλη) του δηλούμενου-- δημιουργούμενου αντικειμένου της κλάσης.

Οδηγίες:

Ο κατασκευαστής-δομητής είναι μια μέθοδος (συνάρτηση) με το όνομα της κλάσης -χωρίς όμως να έχει επιστρεφόμενη τιμή- η οποία καλείται και εκτελείται κατά την δήλωση-δημιουργία ενός αντικειμένου.

Για παράδειγμα, για την περίπτωση δηλώσεων αντικειμένων σαν το x παραπάνω:

Η δήλωση πρωτοτύπου για τη μέθοδο δόμησης μέσα στην κλάση μπορεί να είναι:

```
foititis (string, int, double );
```

Στη συνέχεια, έξω από την κλάση, θα τοποθετηθεί ο ορισμός της μεθόδου δόμησης, π.χ..

```
foititis::foititis (string name, int mitrwo, double v)
{
    ...κώδικας..., π.χ. onoma = name , κλπ...
}
```

Σημείωση: Θα μπορούσε, εναλλακτικά ο ορισμός της μεθόδου δόμησης να τοποθετηθεί εξ αρχής μέσα στην κλάση, οπότε περιττεύει η δήλωση πρωτοτύπου, π.χ.

```
foititis (string name, int mitrwo, double v)
{
    ...κώδικας..., π.χ. onoma = name , κλπ...
}
```

- Η δεύτερη συνάρτηση δόμησης θα αφορά, δηλ. θα εκτελείται, σε δηλώσεις αντικειμένων σαν το y παραπάνω. Θα έχει ως μοναδικό όρισμα μια συμβολοσειρά και θα την αποδίδει στην ιδιότητα *όνομα* του δηλούμενου-δημιουργούμενου αντικειμένου της κλάσης. Οι άλλες δυο ιδιότητες του αντικειμένου θα έχουν τυχαίες τιμές ή προκαθορισμένες τιμές.

Οδηγίες:

Η δήλωση πρωτοτύπου για τη μέθοδο δόμησης μέσα στην κλάση μπορεί να είναι:

```
foititis (string );
```

Στη συνέχεια, έξω από την κλάση, θα τοποθετηθεί ο ορισμός της μεθόδου δόμησης, π.χ..

```
foititis::foititis (string name)
{
    ...κώδικας..., π.χ. onoma = name; και αν θέλουμε να αποδοθούν
    προκαθορισμένες τιμές στις υπόλοιπες ιδιότητες, π.χ. am=-1111
}
```

- Η τρίτη μέθοδος δόμησης αφορά δηλώσεις σαν το z παραπάνω. Δεν θα έχει κανένα όρισμα. Οι τρεις ιδιότητες του αντικειμένου θα έχουν τυχαίες ή προκαθορισμένες τιμές ανάλογα με το τι θα επιλέξει ο προγραμματιστής. **Οδηγίες:**

Οδηγίες:

Η δήλωση πρωτοτύπου για τη μέθοδο δόμησης μέσα στην κλάση μπορεί να είναι:

```
foititis ( );
```

Στη συνέχεια, έξω από την κλάση, θα τοποθετηθεί ο ορισμός της μεθόδου δόμησης, π.χ..

```
foititis::foititis ()
{
    Μπορεί και να μην υπάρχει καθόλου κώδικας ή αν θέλουμε μπορεί να
    αποδοθούν να αποδοθούν προκαθορισμένες τιμές σε ιδιότητες
}
```

Κυρίως πρόγραμμα:

Στο κυρίως πρόγραμμα να γίνει δήλωση τεσσάρων αντικειμένων (φοιτητών).

Για τους δυο πρώτους οι τιμές των ιδιοτήτων τους θα ορίζονται κατά την δήλωσή τους.

Για τον τρίτο θα αρχικοποιείται μόνο το όνομα.

Ο τέταρτος θα δηλωθεί χωρίς αρχικοποίηση.

Στη συνέχεια εμφανίστε τα στοιχεία αυτών των φοιτητών καλώντας την μέθοδο της εμφάνισης-εξόδου που δημιουργήσατε παραπάνω.

Η δομή του προγράμματος φαίνεται παρακάτω:

Οδηγίες προς τον προεπεξεργαστή (#include κλπ)

Ορισμός κλάσης φοιτητή

{ Ιδιωτικά μέλη (εξ ορισμού)

Δήλωση ιδιοτήτων

Δημόσια μέλη:

Δήλωση πρωτοτύπου της μεθόδου (συνάρτησης-μέλους) για την εκτύπωση-εμφάνιση των στοιχείων ενός φοιτητή

Δήλωση πρωτοτύπου πρώτης μεθόδου δόμησης

Δήλωση πρωτοτύπου δεύτερης μεθόδου δόμησης

Δήλωση πρωτοτύπου τρίτης μεθόδου δόμησης

}

Ορισμός της μεθόδου για την εκτύπωση των στοιχείων ενός φοιτητή

Ορισμός της πρώτης μεθόδου δόμησης

Ορισμός της δεύτερης μεθόδου δόμησης

Ορισμός της τρίτης μεθόδου δόμησης

Κυρίως πρόγραμμα

{

Δηλώσεις των τεσσάρων αντικειμένων-φοιτητών (οι δυο πρώτοι με αρχικοποίηση όλων των ιδιοτήτων τους, ο τρίτος με αρχικοποίηση μόνο του ονόματος, ο τέταρτος χωρίς καμία αρχικοποίηση)

Εκτύπωση των στοιχείων τους με την μέθοδο (συνάρτηση) εκτύπωσης

}

Μέρος 2 -Υπερφόρτωση Τελεστών

Αφού υλοποιηθεί και ελεγχθεί η σωστή λειτουργία του παραπάνω μέρους της άσκησης, προχωρήστε στη σταδιακή υλοποίηση των παρακάτω:

9B.1 Υπερφόρτωση τελεστή > (περίπτωση Α: φοιτητής σύγκριση με double)

Να γίνει υπερφόρτωση του τελεστή σύγκρισης > ούτως ώστε να επιστρέφει true αν ο βαθμός του αντικειμένου που προκάλεσε την κλήση του (δηλ. που βρίσκεται αριστερά του) είναι μεγαλύτερος από μια πραγματική τιμή που είναι στα δεξιά του, αλλιώς να επιστρέφει false. Παραδείγματα κλήσης:

if (x > 7.0)) ή if (y > v) όπου v μια πραγματική μεταβλητή και x και y αντικείμενα τύπου φοιτητή

Οδηγίες:

Ο κώδικας που πρέπει να γραφεί για τον τελεστή έχει την δομή μιας συνάρτησης-μεθόδου, μόνο στη θέση του ονόματος της μεθόδου βάζουμε τη λέξη-κλειδί operator ακολουθούμενη από το

σύμβολο του τελεστή που μας ενδιαφέρει. Έτσι, όπως και στις μεθόδους-συναρτήσεις, έχουμε τη δήλωση πρωτοτύπου και τον ορισμό του κώδικα.

Η δήλωση πρωτοτύπου για τον κώδικα για τον τελεστή > μέσα στην κλάση θα είναι:

```
bool operator > (double);
```

Ορισμός του κώδικα

Έξω από την κλάση, θα τοποθετηθεί ο ορισμός του κώδικα για τον τελεστή, π.χ.

```
bool foititis::operator > (double v)
{ ...
    Κώδικας, π.χ. if ( b > v) κλπ...
    ...
}
```

Όπου:

bool είναι ο τύπος της επιστρεφόμενης τιμής (αποτελέσματος)

foititis:: προσδιορίζει την κλάση του αντικειμένου που βρίσκεται αριστερά από τον τελεστή και το οποίο προκαλεί την κλήση του.

operator είναι δεσμευμένη λέξη (λέξη-κλειδί).

> το σύμβολο του συγκεκριμένου τελεστή.

double v μέσα στην παρένθεση, δηλώνεται ως τυπική παράμετρος, ο πραγματικός αριθμός που θα βρίσκεται δεξιά του τελεστή κάθε φορά.

b είναι η ιδιότητα **b** του αντικειμένου που θα βρίσκεται αριστερά του τελεστή σε παραστάσεις που τον περιλαμβάνουν και το οποίο προκαλεί την κλήση του σχετικού κώδικα.

Έτσι π.χ. αν ο κώδικας κληθεί από την παράσταση `if (x>8)` τότε το **b** αναφέρεται στο **b** του **x** (δηλ. **x.b**) και το 8 αποδίδεται στην παράμετρο **v**,

ενώ αν κληθεί από στην παράσταση `if (y>6.5)` τότε το **b** αναφέρεται στο **b** του **y** (δηλ. **y.b**) και το 6.5 αποδίδεται στην παράμετρο **v**

Σημείωση: Θα μπορούσε, εναλλακτικά ο ορισμός της κώδικα για τον τελεστή να τοποθετηθεί μέσα στην κλάση, οπότε σ' αυτήν την περίπτωση, περιττεύει η δήλωση πρωτοτύπου, π.χ. θα μπορούσαμε να είχαμε μέσα στην κλάση:

```
bool operator > (double v)
{ ...
    Κώδικας, π.χ. if ( b > v κλπ...
    ...
}
```

Στο κυρίως πρόγραμμα:

Χρησιμοποιείτε αυτόν τον τελεστή και ελέγξτε αν ο βαθμός του πρώτου φοιτητή που δηλώσατε παραπάνω στο α' μέρος είναι μεγαλύτερος ή όχι πάνω από κάποια τιμή και εμφανίστε κατάλληλα μηνύματα.

9B.2 Υπερφόρτωση τελεστή > (περίπτωση B: φοιτητής σύγκριση με φοιτητή)

Να γίνει υπερφόρτωση του τελεστή σύγκρισης μεγαλύτερο > ούτως ώστε να μπορεί να συγκρίνει δυο αντικείμενα τύπου `foititis` και να επιστρέφει `true` αν ο βαθμός του αριστερού αντικειμένου (που προκάλεσε την κλήση) είναι μεγαλύτερος από τον βαθμό αντικειμένου που θα είναι στα δεξιά του, αλλιώς να επιστρέφει `false`. Παράδειγμα κλήσης:

```
if ( x > f1 ) ... //όπου x, f1 αντικείμενα τύπου φοιτητή.
```

Οδηγίες:

Ο κώδικας για τον τελεστή είναι παρόμοιος με την περίπτωση 9B.1, μόνο που τώρα δεξιά του τελεστή βρίσκεται θα βρίσκεται ένα αντικείμενο `foititis`.

Η δήλωση πρωτοτύπου για τον κώδικα για τον τελεστή > μέσα στην κλάση θα είναι:

```
bool operator > (foititis);
```

Ορισμός του κώδικα

Στη συνέχεια, έξω από την κλάση, θα τοποθετηθεί ο ορισμός του κώδικα για τον τελεστή .

```
bool foititis::operator > (foititis d)
{ ...
  Κώδικας, π.χ. if ( b > d.b) κλπ...
  ...
}
```

Όπου:

`bool` είναι ο τύπος της επιστρεφόμενης τιμής (αποτελέσματος)

`foititis::` προσδιορίζει την κλάση του αντικειμένου που βρίσκεται αριστερά από τον τελεστή και το οποίο προκαλεί την κλήση του.

`operator` είναι δεσμευμένη λέξη (λέξη-κλειδί).

`>` το σύμβολο του συγκεκριμένου τελεστή.

`foititis d` μέσα στην παρένθεση, δηλώνεται ως τυπική παράμετρος, το αντικείμενο που θα βρίσκεται δεξιά του τελεστή κάθε φορά.

`b` η ιδιότητα `b` του αντικειμένου που θα βρίσκεται αριστερά του τελεστή σε παραστάσεις που τον περιλαμβάνουν και το οποίο προκαλεί την κλήση του σχετικού κώδικα.

d.b η ιδιότητα b του όποιου αντικειμένου που θα βρίσκεται δεξιά του τελεστή στην παράσταση που τον περιλαμβάνει.

Έτσι π.χ. αν ο κώδικας κληθεί από την παράσταση `if (x>f1)` τότε το b αναφέρεται στο b του x (δηλ. x.b) και το f1 αποδίδεται στην παράμετρο d,

ενώ αν κληθεί από στην παράσταση `if (f1>y)` τότε το b αναφέρεται στο b του f1 (δηλ. f1.b) και το y αποδίδεται στην παράμετρο d

Σημείωση: Θα μπορούσε, εναλλακτικά ο ορισμός της κώδικα για τον τελεστή να τοποθετηθεί μέσα στην κλάση, οπότε σ' αυτήν την περίπτωση, περιττεύει η δήλωση πρωτοτύπου, π.χ. θα μπορούσαμε να είχαμε μέσα στην κλάση:

```
bool operator > (foititis d)
{
    ...
    Κώδικας, π.χ. if ( b > d.b) κλπ...
    ...
}
```

Στο κυρίως πρόγραμμα:

Χρησιμοποιείτε αυτόν τον τελεστή και ελέγξτε αν ο βαθμός του πρώτου φοιτητή που δηλώσατε παραπάνω στο α' μέρος είναι μεγαλύτερος ή όχι από τον βαθμό του δεύτερου φοιτητή που δηλώσατε. Εμφανίστε κατάλληλα μηνύματα.

9B.3 Υπερφόρτωση τελεστή + (περίπτωση A: αντικείμενο φοιτητής πρόσθεση με double)

Θέλουμε να έχουμε τη δυνατότητα για παραστάσεις σαν τις παρακάτω (όπου x, y αντικείμενα τύπου foititis)

`x = x + 2.0 ;`

`y = x + v; //όπου v μια πραγματική μεταβλητή`

Συνεπώς, πρέπει να γραφεί κώδικας που θα κάνει υπερφόρτωση του τελεστή πρόσθεσης + για την περίπτωση που αριστερά του είναι αντικείμενο της κλάσης foititis και δεξιά του ένας double αριθμός. Το αποτέλεσμα του τελεστή (δηλ. η τιμή της παράστασης) θα είναι τύπου foititis. Συγκεκριμένα θα είναι ένας φοιτητής ίδιος με αυτόν που προκάλεσε την κλήση του κώδικα του τελεστή + (δηλ. που βρίσκεται στα αριστερά του +) μόνο που μετά την πράξη θα έχει βαθμό αυξημένο κατά την πραγματική ποσότητα που βρίσκεται στα δεξιά του τελεστή.

Οδηγίες:

Ο κώδικας για τον τελεστή είναι παρόμοιος με τις παραπάνω περιπτώσεις 9B.1 και 9B.2 και προχωράμε κατά αντίστοιχο τρόπο.

Η δήλωση πρωτοτύπου για τον κώδικα για τον τελεστή + μέσα στην κλάση θα είναι:

```
foititis operator > (double);
```

Στη συνέχεια, έξω από την κλάση, θα τοποθετηθεί ο ορισμός του κώδικα για τον τελεστή .

```
foititis foititis::operator + (double v)
{
    Κώδικας, π.χ.
    foititis f;
    ...
    t.b = b + v;
    κλπ... t.onoma = ...κλπ...
    return f;
}
```

Όπου:

foititis είναι ο τύπος της επιστρεφόμενης τιμής (αποτελέσματος)

foititis:: προσδιορίζει την κλάση του αντικειμένου που βρίσκεται αριστερά από τον τελεστή και το οποίο προκαλεί την κλήση του.

operator είναι δεσμευμένη λέξη (λέξη-κλειδί).

+ το σύμβολο του συγκεκριμένου τελεστή.

double v μέσα στην παρένθεση, δηλώνεται ως τυπική παράμετρος, ο πραγματικός αριθμός που θα βρίσκεται δεξιά του τελεστή κάθε φορά.

b η ιδιότητα b του αντικειμένου που θα βρίσκεται αριστερά του τελεστή του τελεστή σε παραστάσεις που τον περιλαμβάνουν και το οποίο προκαλεί την κλήση του σχετικού κώδικα.

f ένα τοπικό αντικείμενο του οποίου οι τιμές-ιδιότητες θα επιστραφούν ως αποτέλεσμα του τελεστή.

Στο κυρίως πρόγραμμα:

Χρησιμοποιείστε αυτόν τον τελεστή για να αυξήσετε τον βαθμό του πρώτου φοιτητή που δηλώσατε κατά μια ποσότητα που θα διαβάσετε από το πληκτρολόγιο και εκχωρείστε το αποτέλεσμα στον τρίτο φοιτητή. Εμφανίστε το αποτέλεσμα χρησιμοποιώντας την κατάλληλη μέθοδο.

Υλικό:

Διάλεξη/Παράδοση 2015/05/14, 2014/05/15 και 2014/05/22

Συνοπτικό φύλο με παραδείγματα σχετικά με την Β' εξέταση (βρίσκεται στο eclass του εργαστηρίου Προγ. II, στην κατηγορία Έγγραφα > Εξετάσεις)

Βιβλίο: Η γλώσσα C++ σε βάθος, του Ν.Μ. Χατζηγιαννάκη, σελ. 364-371, 394-396, 417-420

Βιβλίο: Αντικειμενοστρεφής προγραμματισμός με τη C++, του R. Lafore, Κλάσεις και αντικείμενα