

3^η Προγραμματιστική Εργασία

Ημερομηνία παράδοσης: 29.10.2022

Στόχοι: Εξοικείωση με την υλοποίηση κλάσεων, χρήση κλάσης ως τύπου δεδομένων, κλήση μεθόδου αντικειμένου, τοπικές μεταβλητές.

Υλοποιήστε την κλάση `Point3D` για την αναπαράσταση ενός σημείου στον χώρο των τριών διαστάσεων. Για την υλοποίηση της κλάσης `Point3D` χρησιμοποιήστε τα παρακάτω (ιδιωτικά) πεδία:

```
int xCoord;  
int yCoord;  
int zCoord;
```

Αντικείμενα της κλάσης `Point3D` υποστηρίζουν τις παρακάτω (δημόσιες) μεθόδους:

<code>Point3D()</code>	Κατασκευάζει ένα σημείο με συντεταγμένες (0,0,0)
<code>Point3D(int x, int y, int z)</code>	Κατασκευάζει ένα σημείο με συντεταγμένες (x,y,z)
<code>setX(int newX)</code>	Δίνει την τιμή newX στη X-συντεταγμένη του σημείου
<code>setY(int newY)</code>	Δίνει την τιμή newY στη Y-συντεταγμένη του σημείου
<code>setZ(int newZ)</code>	Δίνει την τιμή newZ στη Z-συντεταγμένη του σημείου
<code>getX()</code>	Επιστρέφει την X-συντεταγμένη του σημείου
<code>getY()</code>	Επιστρέφει την Y-συντεταγμένη του σημείου
<code>getZ()</code>	Επιστρέφει την Z-συντεταγμένη του σημείου
<code>getPosition()</code>	Επιστρέφει τις συντεταγμένες του σημείου ως συμβολοσειρά (string), π.χ., "(3,1,2)"

Για την υλοποίηση της τελευταίας μεθόδου συμβουλευθείτε την υλοποίηση της αντίστοιχης μεθόδου `getTime()` της κλάσης `Timer` που σας έχει δοθεί. Επίσης, μην ξεχάσετε να συμπεριλάβετε την βιβλιοθήκη `string` της C++: `#include <string>`.

Στο ίδιο αρχείο, υλοποιήστε την κλάση `Cuboid` για την παράσταση ενός παραλληλεπίπεδου διαστάσεων `length x width x height` του οποίου η μπροστά - κάτω - αριστερή κορυφή του (σημείο αναφοράς) βρίσκεται στο σημείο `corner`. Για την υλοποίηση της κλάσης `Cuboid` χρησιμοποιήστε τα παρακάτω (ιδιωτικά) πεδία:

```
int length;  
int width;  
int height;  
Point3D corner;
```

Η κλάση `Cuboid` υποστηρίζει τους παρακάτω (δημόσιους) κατασκευαστές:

```
Cuboid()  
Κατασκευάζει ένα 1x1x1 παραλληλεπίπεδο με σημείο αναφοράς το (0,0,0)
```

Cuboid(int l, int w, int h, int x, int y, int z)

Κατασκευάζει ένα $l \times w \times h$ παραλληλεπίπεδο με σημείο αναφοράς το (x, y, z)

Ο κώδικας για τον τελευταίο κατασκευαστή είναι:

```
Cuboid(int l, int w, int h, int x, int y, int z) : corner(x, y, z) {  
    length = l;  
    width = w;  
    height = h;  
}
```

Εισάγετε τον παραπάνω κώδικα και βεβαιωθείτε για την ορθότητα του. Ακολούθως, υλοποιήστε τον προκαθορισμένο κατασκευαστή **Cuboid()**.

Η κλάση **Cuboid** υποστηρίζει τις παρακάτω μεθόδους:

setLength(int len)	Θέτει το μήκος στην τιμή της παραμέτρου
setWidth(int wid)	Θέτει το πλάτος στην τιμή της παραμέτρου
setHeight(int hei)	Θέτει το ύψος στην τιμή της παραμέτρου
getLength()	Επιστρέφει το μήκος του παραλληλεπίπεδου
getWidth()	Επιστρέφει το πλάτος του παραλληλεπίπεδου
getHeight()	Επιστρέφει το ύψος του παραλληλεπίπεδου
setCornerX(int newX)	Δίνει την τιμή newX στη X-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
setCornerY(int newY)	Δίνει την τιμή newY στη Y-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
setCornerZ(int newZ)	Δίνει την τιμή newZ στη Z-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
getCornerX()	Επιστρέφει την X-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
getCornerY()	Επιστρέφει την Y-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
getCornerZ()	Επιστρέφει την Z-συντεταγμένη του σημείου αναφοράς του παραλληλεπίπεδου
getSurfaceArea()	Επιστρέφει το εμβαδόν της επιφάνειας του παρ/δου
getVolume()	Επιστρέφει τον όγκο του παραλληλεπίπεδου
printDetails()	Τυπώνει το σημείο αναφοράς και τις διαστάσεις του παρ/δου

Να υλοποιηθούν και να γίνει έλεγχος ορθότητας των παραπάνω μεθόδων. Να εμπλουτίσετε τον κώδικα των κλάσεων σας με σχόλια και να ελέγξετε την ορθότητα του χρησιμοποιώντας την παρακάτω συνάρτηση-οδηγό **main**.

```
int main() {  
    Point3D p(1,1,1);  
    cout << "Point is at position " << p.getPosition() << endl;  
    Cuboid c(1,1,1,1,1,1);  
    c.setHeight(2);  
    c.setCornerX(2);  
    c.printDetails();  
    cout << "Volume: " << c.getVolume() << endl;  
    cout << "Area: " << c.getSurfaceArea() << endl;  
}
```