

Άσκηση 1

Να γραφεί ένα πρόγραμμα το οποίο θα δημιουργεί μια κλάση με όνομα **biblio**. Η κλάση προορίζεται για τη δημιουργία αντικειμένων – βιβλίων. Για κάθε βιβλίο θα καταχωρίζουμε τον τίτλο, τον αριθμό των σελίδων του και την τιμή του. Το πρόγραμμα θα πρέπει να υλοποιεί τα παρακάτω: 1) η τιμή του βιβλίου θα είναι ιδιωτική, όλα τα υπόλοιπα δημόσια, 2) η δημιουργία ενός αντικειμένου χωρίς όρισμα πρέπει να ορίζει αριθμό σελίδων 0, τιμή 0 και κενό τίτλο, 3) η δημιουργία ενός αντικειμένου με ένα όρισμα πρέπει να ορίζει ως τίτλο του βιβλίου την τιμή του ορίσματος, ως αριθμό σελίδων το 0 και ως τιμή το 0, 4) η δημιουργία ενός αντικειμένου με δυο ορίσματα πρέπει να ορίζει ως τίτλο του βιβλίου την τιμή του πρώτου ορίσματος, ως τιμή του βιβλίου την τιμή του δεύτερου ορίσματος και ως αριθμό σελίδων το 0, 5) πρέπει να υπάρχει μέθοδος που να εμφανίζει όλα τα στοιχεία του βιβλίου, 6) πρέπει να υπάρχει μέθοδος που να ορίζει ως τιμή του βιβλίου τη τιμή της παραμέτρου της, 7) η συνάρτηση main() πρέπει να δημιουργεί 3 αντικείμενα – βιβλία με τέτοιο τρόπο ώστε να χρησιμοποιείται διαφορετική μέθοδος δόμησης για το καθένα. Τα 3 βιβλία πρέπει να έχουν διαφορετικό τίτλο, αριθμό σελίδων 100, 200, 300 και τιμή 20, 30 και 40 αντίστοιχα. Στο τέλος πρέπει να εμφανίζει τα στοιχεία και των τριών βιβλίων.

```
#include <iostream>
#include <string>
using namespace std;
```

```
class biblio
{
    float timi;
public:
    int selides;
    string titlos;
    biblio();
    biblio(string title);
    biblio(string title,int t);
    void info();
    void set_timi(float t);
};

void biblio::info()
{
    cout <<"-----" << endl;
    cout << titlos << endl;
    cout << "Σελίδες : " << selides << endl;
    cout << "Τιμή : " << timi << endl;
}

biblio::biblio()
{
    titlos="";
    timi = 0;
    selides=0;
}

biblio::biblio(string title)
{
    titlos=title;
    timi = 0;
```

Ιδιωτικό μέλος της κλάσης **biblio**.

Δημόσια μέλη της κλάσης **biblio**.

Η κλάση **biblio** διαθέτει τη μέθοδο δόμησης **biblio()** υπερφορτωμένη τρεις φορές με διαφορετικό πλήθος παραμέτρων.

Η μέθοδος **info()** εμφανίζει στην οθόνη τα στοιχεία του αντικειμένου-βιβλίου.

Αυτή η έκδοση της μεθόδου δόμησης καλείται όταν δημιουργείται ένα αντικείμενο χωρίς ορίσματα. Καταχωρίζει αρχικές τιμές στις μεταβλητές-μέλη του αντικειμένου, κενό για τον τίτλο και 0 για τις δύο υπόλοιπες.

Αυτή η έκδοση της μεθόδου δόμησης καλείται όταν δημιουργείται ένα αντικείμενο με ένα όρισμα. Καταχωρίζει την τιμή της παραμέτρου στο μέλος **titlos** του αντικειμένου.

```

    selides=0;
}

biblio::biblio(string title,int t)
{
    titlos=title;
    timi=t;
    selides=0;
}

void biblio::set_timi(float t)
{
    timi=t;
}

int main()
{
    biblio b1,b2("C++ σε βάθος"),b3("Αλγόριθμοι",40);
    b1.titlos="Αντικείμενοστρεφής προγραμματισμός";
    b1.selides=100;
    b2.selides=200;
    b3.selides=300;
    b1.set_timi(20);
    b2.set_timi(30);
    b1.info();
    b2.info();
    b3.info();
    return 0;
}

```

Αυτή η έκδοση της μεθόδου δόμησης καλείται όταν δημιουργείται ένα αντικείμενο με δύο ορίσματα. Καταχωρίζει την πρώτη παράμετρο στο μέλος **titlos** και τη δεύτερη στο μέλος **timi** του αντικειμένου.

Στη μεταβλητή-μέλος **selides** των τριών αντικειμένων καταχωρίζονται οι τιμές 100, 200 και 300 αντίστοιχα.

Καταχωρίζονται οι τιμές 20 και 30 για τα βιβλία b1 και b2, αντίστοιχα.

Εμφανίζονται τα στοιχεία και των τριών βιβλίων.

- 👉 Στη πρώτη πρόταση της συνάρτησης **main()** δημιουργούνται τρία αντικείμενα της κλάσης **biblio** καλώντας τις αντίστοιχες μεθόδους δόμησης: για καμία, μία και δύο παραμέτρους.
- 👉 ΠΡΟΣΟΧΗ: η μεταβλητή μέλος **timi**, ως ιδιωτική, δεν είναι προσπελάσιμη παρά μόνο μέσω της δημόσιας μεθόδου **set_timi()**.

Άσκηση 2

Θέλουμε να διαχειριζόμαστε κουτιά. Κάθε κουτί προσδιορίζεται από τις διαστάσεις του (ύψος – μήκος – πλάτος), το χρώμα του και το περιεχόμενο του. Να γραφεί μια κλάση με όνομα **box** για τη διαχείριση των αντικειμένων – κουτιών, η οποία θα υλοποιεί τα ακόλουθα: 1) οι διαστάσεις θα πρέπει να είναι ιδιωτικές και όλα τα άλλα μέλη δημόσια, 2) η δημιουργία ενός αντικειμένου χωρίς όρισμα θα πρέπει αυτόματα να ορίζει μηδενικές διαστάσεις, χρώμα άσπρο και μια παύλα για το περιεχόμενο του, 3) η δημιουργία ενός αντικειμένου με ένα όρισμα πρέπει να ορίζει ως τιμή και των τριών διαστάσεων του την τιμή του ορίσματος, 4) η δημιουργία ενός αντικειμένου με τρία ορίσματα πρέπει να ορίζει ως τιμή της κάθε διάστασης του την τιμή του κάθε ορίσματος αντίστοιχα, 5) πρέπει να υπάρχει μέθοδος που να εμφανίζει όλα τα στοιχεία του κουτιού, 6) πρέπει να υπάρχει μέθοδος που να υπολογίζει και να επιστρέφει ως τιμή τον όγκο του κουτιού (γινόμενο 3 διαστάσεων), 7) πρέπει να υπάρχει μέθοδος με τρεις παραμέτρους η οποία να ορίζει ως τιμές των διαστάσεων του αντικειμένου τις τιμές των παραμέτρων της. Η συνάρτηση `main()` πρέπει να δημιουργεί 3 αντικείμενα `k1`, `k2` και `k3` με τέτοιο τρόπο ώστε για το καθένα να χρησιμοποιείται διαφορετική μέθοδος δόμησης: το `k1` κατά την κατασκευή του να αποκτήσει μηδενικές διαστάσεις, το `k2` να αποκτήσει την ίδια τιμή και στις 3 διαστάσεις του `5x5x5` και το `k3` διαστάσεις `10x20x30`. Να μπουν χρώματα και περιεχόμενα της επιλογής σας στα κουτιά `k2` και `k3`. Να εμφανιστεί ο όγκος του κουτιού `k3`, να διπλασιαστούν οι διαστάσεις του κουτιού `k2` και το κουτί `k1` να αποκτήσει διαστάσεις `7x8x9`. Τέλος να εμφανιστούν τα στοιχεία των κουτιών `k1` και `k2`

```

#include <iostream>
#include <string>
using namespace std;
class box
{
    float ypsos;
    float mikos;
    float platos;
public:
    string xroma;
    string perioxomeno;
    box();
    box(float all);
    box(float y, float m, float p);
    void display();
    float ogos();
    void set(float y, float m, float p);
    void set(float kl);
};
box::box()
{
    ypsos=0;
    mikos=0;
    platos=0;
    xroma="ΑΣΗΠΟ";
    perioxomeno="-";
}
box::box(float all)
{
    ypsos=mikos=platos=all;
}
box::box(float y, float m, float p)
{
    ypsos=y;
    mikos=m;
    platos=p;
}

```

Ιδιωτικά μέλη της κλάσης

Δημόσια μέλη της κλάσης

Μέθοδοι δόμησης της κλάσης

Υπόλοιπες μέθοδοι της κλάσης

Μέθοδος δόμησης η οποία εφαρμόζεται όταν δημιουργούνται αντικείμενα χωρίς ορίσματα

Μέθοδος δόμησης η οποία εφαρμόζεται όταν δημιουργούνται αντικείμενα με ένα όρισμα

Μέθοδος δόμησης η οποία εφαρμόζεται όταν δημιουργούνται αντικείμενα με τρία ορίσματα

```
void box::display()
```

```
{  
    cout <<"Κουτί "<<xroma<<" Διαστάσεων: "<<yposos<<"x"<<mikos<<"x"<<platos  
        <<" με "<<periexomeno<<endl;  
}
```

Μέθοδος η οποία εμφανίζει τα στοιχεία των αντικειμένων

```
float box::ogos()
```

```
{  
    return yposos*mikos*platos;  
}
```

Μέθοδος η οποία επιστρέφει ως τιμή τον όγκο των αντικειμένων-κουτιών.

```
void box::set(float y, float m, float p)
```

```
{  
    yposos=y;  
    mikos=m;  
    platos=p;  
}
```

Μέθοδος η οποία θέτει ως διαστάσεις σε ένα αντικείμενο τις τιμές των παραμέτρων της.

```
void box::set(float k1)
```

```
{  
    yposos=yposos*k1;  
    mikos=mikos*k1;  
    platos=platos*k1;  
}
```

Υπερφορτωμένη έκδοση της προηγούμενης μεθόδου η οποία θέτει και στις τρεις διαστάσεις ενός αντικείμενου την τιμή της παραμέτρου της.

```
int main()
```

```
{  
    box k1, k2(5), k3(10, 20, 30);  
    k2.xroma="Κόκκινο";  
    k3.xroma="Πράσινο";  
    k2.periexomeno="Παπούτσια";  
    k3.periexomeno="Αυγά";  
    cout << k3.ogos() <<endl;  
    k2.set(2);  
    k1.set(7, 8, 9);  
    k1.display();  
    k2.display();  
    return 0;  
}
```

Το αντικείμενο **k1** δημιουργείται με την πρώτη έκδοση της μεθόδου δόμησης, το **k2** με τη δεύτερη και το **k3** με την τρίτη.

Καταχωρίζεται χρώμα και περιεχόμενο για τα αντικείμενα **k2** και **k3**.

Εμφανίζεται ο όγκος του αντικείμενου **k3**.

Διπλασιάζονται οι διαστάσεις του **k2** με εφαρμογή της δεύτερης έκδοσης της **set()**.

Το **k1** λαμβάνει συγκεκριμένες διαστάσεις

Εμφανίζονται τα στοιχεία των **k1** και **k2**

Άσκηση 3

- Να γράψετε πρόγραμμα το οποίο θα δημιουργεί την κλάση **circle** η οποία χρησιμοποιείται για τη διαχείριση σχημάτων – κύκλων. Ως private θα δηλώνεται η μεταβλητή της ακτίνας και τα υπόλοιπα ως public. Θα δηλώνονται 2 μέθοδοι δόμησης, μια με παράμετρο την ακτίνα και μια χωρίς παραμέτρους καθώς και 3 μέθοδοι: υπολογισμού εμβαδού, περιμέτρου και ανάθεσης τιμής στην ακτίνα. Το κυρίως πρόγραμμα θα δημιουργεί έναν πίνακα 10 αντικειμένων τύπου **circle** με διαφορετική ακτίνα το καθένα και στο τέλος θα εμφανίζει το εμβαδό και την περίμετρο του κάθε αντικειμένου.

```

#include <iostream>
#define pi 3.141593
using namespace std;
class circle
{
    float aktina;
public:
    circle(float r);
    circle();
    float emvado();
    float perimetros();
    void set_r(float r);
};
float circle::emvado()
{
    return pi*aktina*aktina;
}
float circle::perimetros()
{
    return 2*pi*aktina;
}
void circle::set_r(float r)
{
    aktina=r;
}
circle::circle(float r)
{
    aktina = r;
    cout << "Δημιουργήθηκε κύκλος ακτίνας " << aktina << endl;
}
circle::circle()
{
    aktina = 0;
    cout << "Δημιουργήθηκε κύκλος ακτίνας " << aktina << endl;
}

```

```
int main()
{
    circle cir[10];
    int i;
    for (i=0;i<10;i++) cir[i].set_r(i);
    for (i=0;i<10;i++)
        cout << "Εμβαδό "<<i<<" = "
        <<cir[i].envado()<<endl;
    for (i=0;i<10;i++)
        cout << "Περίμετρος "<<i<<" = "
        <<cir[i].perimetros()<<endl;
    return 0;
}
```

Άσκηση 4

- Να γραφεί πρόγραμμα το οποίο θα δημιουργεί την κλάση **trionymo**. Κάθε δευτεροβάθμια εξίσωση μοντελοποιείται ως αντικείμενο της κλάσης αυτής. Οι συντελεστές της είναι μεταβλητές – μέλη της κλάσης (Private a,b,c,d) και ο υπολογισμός των ριζών γίνεται από μεθόδους της κλάσης (δόμησης, καταχώρισης τιμών στις μεταβλητές μέλη, ελέγχου ύπαρξης πραγματικών ριζών, επιστροφής πραγματικών ριζών). Το κυρίως πρόγραμμα να δημιουργεί ένα αντικείμενο της κλάσης με συντελεστές 1,2 και -3. Στη συνέχεια μέσω επαναληπτικής διαδικασίας ο χρήστης θα αλλάζει τους παραπάνω συντελεστές από το πληκτρολόγιο και με την εφαρμογή των κατάλληλων μεθόδων θα ελέγχεται η ύπαρξη πραγματικών ριζών οι οποίες θα εμφανίζονται στην οθόνη.

```
#include <iostream>
#include <cstdlib>
#include <math.h>
using namespace std;
class trionymo
{
    double a;
    double b;
    double c;
    double d;
public:
    trionymo(double s1,double s2,double s3);
    void syntelestes(double s1,double s2,double s3);
    bool exei_pragmatikes_rizes();
    double r1();
    double r2();
};

trionymo::trionymo(double s1,double s2,double s3)
{
    a=s1;
    b=s2;
    c=s3;
    d=pow(b,2)-4*a*c;
}
```

```

void trionymo::syntelestes(double s1,double s2,double s3)
{
    a=s1;
    b=s2;
    c=s3;
    d=pow(b,2)-4*a*c;
}
bool trionymo::exei_pragmatikes_rizes()
{
    if (d>0)
        return true;
    else
        return false;
}
double trionymo::r1()
{
    double riz1;
    if (exei_pragmatikes_rizes())
    {
        riz1= (-b+sqrt(d))/(2*a);
        return (riz1);
    }
    else
    {
        cout << "Δεν υπάρχουν πραγματικές ρίζες" << endl;
        exit(1);
    }
}

```

```

double trionymo::r2()
{
    double riz2;
    if (exei_pragmatikes_rizes())
    {
        riz2= (-b-sqrt(d))/(2*a);
        return (riz2);
    }
    else
    {
        cout << "Δεν υπάρχουν πραγματικές ρίζες" << endl;
        exit(1);
    }
}

int main()
{
    trionymo ex1(1,2,-3);
    double sn1,sn2,sn3;
    if (ex1.exei_pragmatikes_rizes())
        cout << "r1="<<ex1.r1()<<" r2="<<ex1.r2()<<endl;
    do
    {
        cout << "Δώσε με τη σειρά τους συντελεστές του τριονύμου:";
        cin>>sn1>>sn2>>sn3;
        if (sn1==0 || sn2==0 || sn3==0) break;
        ex1.syntelestes(sn1,sn2,sn3);
        if (ex1.exei_pragmatikes_rizes())
            cout << "r1="<<ex1.r1()<<" r2="<<ex1.r2()<<endl;
        else
            cout << "Δεν έχει πραγματικές ρίζες"<<endl;
    } while(1);
    return 0;
}

```