

# ΑΝΤΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

## ΕΡΓΑΣΤΗΡΙΟ 4

Γεώργιος Τσουμάνης, Άγγελος Δήμητσας, Βασίλειος Ναστος

December 5, 2022

### 1 Δυσδιάστατοι πίνακες

Listing 1: "Στατικός δυσδιάστατος πίνακας"

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

int main()
{
    int table[3][3]={ { 1,2,3},{ 2,3,4},{ 3,4,5} };
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            cout<<"table ["<<i<<" ] ["<<j<<" ]:"<<table[i][j]<<endl;
        }
    }
}
```

Listing 2: "Δυναμικός δυσδιάστατος πίνακας"

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

int main()
{
    int **table=new int*[3];
    for(int i=0;i<3;i++)
    {
        table[i]=new int[3];
    }
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            table[i][j]=i+j;
            cout<<"table ["<<i<<" ] ["<<j<<" ]:"<<table[i][j]<<endl;
        }
    }

    for(int i=0;i<3;i++)
    {
        delete[] table[i];
    }
}
```

```

    delete [] table;
    return 0;
}

```

## 2 Κατασκευαστές

- **Κατασκευαστής:** Ένας κενός προκαθορισμένος κατασκευαστής (δηλαδή ένας κατασκευαστής χωρίς παραμέτρους και με κενό σώμα) δημιουργείται αυτόματα αν δεν οριστεί κάποιος άλλος κατασκευαστής (Δημιουργία αντικειμένων).
  - Έχει το ίδιο όνομα με την κλάση.
  - Δεν έχει τύπο επιστροφής (αν και φαίνεται σαν να επιστρέφει ένα αντικείμενο κλάσης όταν καλείται ρητά).
- **Καταστροφέας:** Ένας κενός καταστροφέας δημιουργείται αν δεν οριστεί καταστροφέας (επίσης με κενό σώμα, αποδιάφθρωση αντικειμένων).
- **Κατασκευαστής αντιγραφής:** Ο σκοπός ενός κατασκευαστή αντιγραφής είναι να αρχικοποιηθεί ένα νέο αντικείμενο ως αντίγραφο ενός άλλου αντικειμένου, δέχεται ένα αντικείμενο ως παράμετρο. Η παράμετρος θα πρέπει να περνά με αναφορά. Αν περνούσε με τιμή τότε λόγω αναδρομικής κλήσης της ίδιας συνάρτησης, δηλαδή του copy constructor θα οδηγούμασταν σε stack overflow

```

#include <iostream>
using namespace std;

class A
{
    int x;
public:
    A(int _x):x(_x) {}
    A(const A &a):x(a.x) {}
    ~A() {}
    int getX() {return this->x;}
};

int main()
{
    A a1(15);
    A a2(a1);
    cout<<&a1<<" x:"<<a1.getX()<<endl;
    cout<<&a2<<" x:"<<a2.getX()<<endl;
    return 0;
}

```

## 3 Friend

- Η δεσμευμένη λέξη friend επιτρέπει σε μια κλάση να χορηγήσει πλήρη πρόσβαση σε μια εξωτερική οντότητα.
- Πλήρης πρόσβαση σημαίνει δυνατότητα προσπέλασης και των ιδιωτικών μελών της κλάσης.
- Εξωτερική οντότητα μπορεί να είναι μια συνάρτηση, ή ακόμα και μια άλλη κλάση.
- Για να αποδοθεί η ιδιότητα friend, θα πρέπει να προστεθεί η δήλωση friend μέσα στη δήλωση της κλάσης, και να ακολουθήσει η αντίστοιχη οντότητα.

Listing 3: Παράδειγμα συνάρτησης friend

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

class Complex
{
    double real;
    double imaginary;
public:
    Complex(double r, double im): real(r), imaginary(im) {}
    ~Complex() {}
    friend Complex add(const Complex &c1, const Complex &c2);
    void display()
    {
        cout<<this->real<<("this->imaginary>0?"+" ":"-")<<this->imaginary<<" i"<<endl;
    }
};

Complex add(const Complex &c1, const Complex &c2)
{
    return Complex(c1.real+c2.real, c1.imaginary+c2.imaginary);
}

int main()
{
    Complex c1(2,5);
    Complex c2(4,7);
    Complex c3=add(c1, c2);
    c1.display();
    c2.display();
    c2.display();
    return 0;
}
```

- Συναρτήσεις Vs φίλες συναρτήσεις
  - Η επιλογή του εάν θα χρησιμοποιηθεί συνάρτηση μέλος ή friend συνάρτηση είναι θέμα προγραμματιστικού στυλ.
  - Διαφορετικοί προγραμματιστές μπορεί να έχουν διαφορετικές απόψεις για το θέμα. Στη συνέχεια παρουσιάζονται για σύγκριση των δύο κλήσεων ο ένας κώδικας ο ένας μετά τον άλλο: [language=c++] f3 = Add(f1,f2); // κλήση στη friend συνάρτηση Add f3 = f1.Add(f2); // κλήση στη συνάρτηση μέλος Add της κλάσης Fraction
  - Ωστόσο, αξίζει να σημειωθεί ότι οι δύο παραπάνω κλήσεις δεν είναι πάντα ισοδύναμες. Στην έκδοση με τις friend συναρτήσεις η Add λαμβάνει αντίγραφα από τα f1 και f2(συνεπώς η συνάρτηση δεν μπορεί να αλλάξει τα αρχικά αντικείμενα). Στην έκδοση με τη συνάρτηση μέλος, η συνάρτηση Add μπορεί να αλλάξει το αντικείμενο f1.

## 4 Κληρονομικότητα

- Δύο βασικές σχέσεις μεταξύ αντικειμένων είναι η σχέση HAS-A και η σχέση IS-A.
- Σχέση IS-A
  - Ένας ComputerScienceStudent είναι ένας (IS-A) UniversityStudent.
  - Ένας UniversityStudent είναι ένας (IS-A) Student.

- Η «κληρονομικότητα» (inheritance) στη C++ επιτρέπει στους προγραμματιστές να ορίσουν IS-A σχέσεις:
  - Μια κλάση μπορεί να οριστεί ως παραγόμενη από μια άλλη κλάση (που ονομάζεται βασική κλάση).
  - Η παραγόμενη κλάση γίνεται ένα είδος/εξειδίκευση της βασικής κλάσης.
  - Εναλλακτικά, μπορεί να θεωρηθεί ότι η παραγόμενη κλάση είναι ότι είναι η βασική κλάση και (πιθανά) περισσότερα.
  - Αυτό επιτρέπει, σε ορισμένες περιπτώσεις και εφόσον απαιτείται, τη χρήση της παραγόμενης κλάσης σαν να ήταν η βασική κλάση.
- Η σχέση IS-A υλοποιείται μέσω της κληρονομικότητας χρησιμοποιώντας τον ακόλουθο τρόπο δήλωσης, που σημαίνει ότι τα αντικείμενα της παραγόμενης κλάσης είναι αντικείμενα της βασικής κλάσης (μπορούν να χρησιμοποιούν όλες τις συναρτήσεις της διεπαφής της βασικής κλάσης)

Listing 4: "Παράδειγμα με χρήση κληρονομικότητας"

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

class Vehicle
{
    string registration_num;
    int fuel_capacity;
public:
    Vehicle(string rg,int fc):registration_num(rg),fuel_capacity(fc) {}
    string get_registration_number()
    {
        return this->registration_num;
    }
    int get_fuel_capacity()
    {
        return this->fuel_capacity;
    }
};

class Car:public Vehicle{
    double cc;
public:
    Car(double cc,string rg,int fc):cc(cc),Vehicle(rg,fc) {}
    ~Car() {}
    void display()
    {
        cout<<this->cc<<" "<<this->get_registration_number()<<" "<<this->get_fuel_capacity()<<endl;
    }
};

int main()
{
    Car c(1500,"AIO9087",75);
    c.display();
    return 0;
}
```

## 5 Επίπεδα προστασίας

- **public:** μέλη που μπορούν να προσπελαστούν με το όνομά τους στο εσωτερικό.

- **private:** Μέλη που μπορούν να προσπελαστούν στο εσωτερικό συναρτήσεων μελών και
- **protected:** Μέλη που μπορούν να προσπελαστούν στο εσωτερικό συναρτήσεων μελών και από συναρτήσεις που έχουν δηλωθεί ως φίλες συναρτήσεις (friend) της κλάσης και επίσης μπορούν να προσπελαστούν στο εσωτερικό των συναρτήσεων μελών της παραγόμενης κλάσης και στις friend συναρτήσεις της παραγόμενης κλάσης. από συναρτήσεις που έχουν δηλωθεί ως φίλες συναρτήσεις (friend) της κλάσης. οποιασδήποτε συνάρτησης.

## 6 Ασκήσεις

1. Γράψτε ένα πρόγραμμα που να ορίζει μια κλάση A με ένα ιδιωτικό μέλος δεδομένων που να είναι δείκτης προς πίνακα ακεραίων. Συμπληρώστε έναν κατασκευαστή αντιγραφής που να πραγματοποιεί deep copy στα δεδομένα του πίνακα. Χρησιμοποιήστε τον κατασκευαστή αντιγραφής στη main.
2. Γράψτε συνάρτηση που να δέχεται ως παραμέτρους έναν διαστάσιμο πίνακα ακεραίων, τον αριθμό γραμμών και τον αριθμό στηλών του και να εμφανίζει τα περιεχόμενα του πίνακα. Καλέστε από το κύριο πρόγραμμα τη συνάρτηση για έναν δυναμικό πίνακα.
3. Γράψτε πρόγραμμα που να ορίζει τη βασική κλάση pet (κατοικίδιο) και τις παραγόμενες κλάσεις cat και dog. Η κλάση pet να είναι αφηρημένη, να ορίζει το προστατευμένο πεδίο name, να ορίζει την ιδεατή συνάρτηση string sound() που θα επιστρέφει την τιμή "hello", να ορίζει την ιδεατή συνάρτηση void make\_sound() που θα εμφανίζει το όνομα του κατοικιδίου και ένα λεκτικό με τον χαρακτηριστικό του ήχο (π.χ. "woof", "miao"). Πειραματιστείτε με ανάθεση αντικειμένων cat και dog σε αναφορές και δείκτες σε pet
4. Να κατασκευάσετε μία κλάση με το όνομα Vehicle και τις υποκλάσεις αυτής Car και Truck. Η κλάση Vehicle θα έχει τα εξής προστατευόμενα μέλη.

- registration\_number:string
- owner\_name:string
- cc:double
- reg\_years:int
- engine\_capacity:int
- price:double

Για την κλάση Car ορίστε μέλος δεδομένο number\_of\_doors ενώ για την κλάση Truck ορίστε μέλος δεδομένο max\_weight.

- Κατασκευάστε constructors και για τις τρεις κλάσεις έτσι ώστε να αρχικοποιούν όλα τα μέλη δεδομένων που διαθέτει η κάθε μια με τιμές παραμέτρων που θα δέχονται.
- Κατασκευάστε getters setters για όλα τα πεδία.
- Ορίστε τη virtual συνάρτηση μέλος traffic\_tax (τέλη κυκλοφορίας) στην κλάση vehicle η οποία θα υπολογίζει τους φόρους με τον εξής τρόπο.
  - Με βάση το engine\_capacity

Engine Capacity(EC)	Tax
$\leq 600$	$200 * 0.782$
$\leq 1000$	$200 + 0.125(EC - 600)0.782$
$\leq 1600$	$250 + 0.375(EC - 1000)0.782$
$\leq 3000$	$475 + 0.125(EC - 600)0.782$
$\leq 6000$	$1525 + 1(EC - 600)0.782$
$\leq 600$	$1785 + 0.4 * reg\_years(EC - 6000)0.782$

- Με βάση τα έτη κυκλοφορίας(reg\_years)

Years	Tax
$\leq 5$	$years * 0.05 * price$
$\leq 10$	$years * 0.05 * price$
$> 10$	$(years - 10 - 1)(0.07 * price) + (years - (years - 10 - 1))(0.09 * price)$

- Ορίστε τη συνάρτηση `traffic_tax` στην κλάση `car` έτσι ώστε να προσθέτει επιπλέον τέλη κυκλοφορίας ως εξής: για αυτοκίνητα μέχρι και 1000 κυβικά εκατοστά(`engine_capacity`) τα τέλη είναι 40€ ενώ για πάνω από 1000 κυβικά εκατοστά για κάθε συμπληρωμένα πλήρως 100 κυβικά εκατοστά άνω των 1000 κυβικών εκατοστών υπάρχει επιπλέον χρέωση 10€ (π.χ. για 1532 κυβ.εκ. η χρέωση είναι  $40€ + 5 * 10€ = 190€$ )
- Ορίστε τη συνάρτηση `traffic_tax` στην κλάση `truck` έτσι έτσι ώστε να προσθέτει επιπλέον τέλη κυκλοφορίας ως εξής: για φορτηγά με μέγιστο βάρος φόρτωσης μέχρι και 3000 κιλά η χρέωση είναι 100€, για φορτηγά με μέγιστο βάρος φόρτωσης μέχρι και 6000 κιλά η χρέωση είναι 150€ και για ακόμα μεγαλύτερα φορτηγά η χρέωση είναι 300€.
- Υλοποιήστε συνάρτηση μέλος `to_string` ώστε να εμφανίζει τα μέλη δεδομένα και τους φόρους ως αλφαριθμητικά και για τις τρεις κλάσεις.
- Γράψτε συνάρτηση `total_tax` που να δέχεται έναν πίνακα 5 θέσεων με δείκτες προς αντικείμενα είτε `car` είτε `truck` και να επιστρέφει το σύνολο τελών κυκλοφορίας που πρέπει να πληρωθούν από όλα τα οχήματα.
- Γράψτε `main` που:
  - Δηλώνει πίνακα 5 θέσεων που θα έχει τη δυνατότητα να δεχθεί δείκτες προς αντικείμενα είτε `car` είτε `truck`.
  - Για κάθε ένα από 5 οχήματα να δέχεται τιμές από τον χρήστη έτσι ώστε ο χρήστης να επιλέγει το εάν το όχημα που εισάγει είναι αυτοκίνητο ή φορτηγό, εν συνεχεία να δέχεται τα στοιχεία του οχήματος και να το τοποθετεί στον πίνακα.
  - Να καλεί τη συνάρτηση `total_tax` και να εμφανίζει το αποτέλεσμα που επιστρέφει.
  - Να εμφανίζει με χρήση κατάλληλης μεταβλητής τα αντικείμενα τύπου `Vehicle` που δημιουργήθηκαν.