

ΕΝΣΩΜΑΤΩΣΗ ή ΠΕΡΙΕΚΤΙΚΟΤΗΤΑ

Πολυμορφισμός

Ο πολυμορφισμός (*polymorphism*), στον αντικειμενοστραφή προγραμματισμό είναι η δυνατότητα που δίνεται:

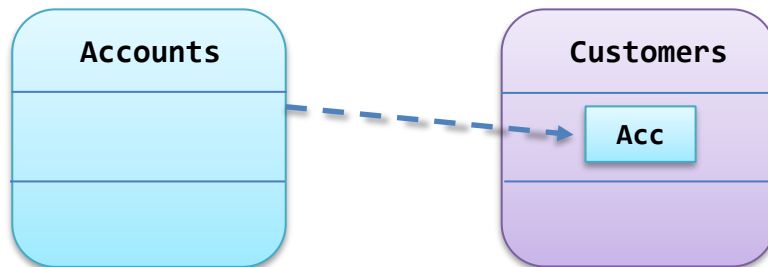
- Σε μία **συνάρτηση** να υπάρχει πολλές φορές μέσα σε μία κλάση με το ίδιο όνομα και τον ίδιο σκοπό, αλλά με διαφορετικά αποτελέσματα. (*Υπερφόρτωση συναρτήσεων*)
- Σε ένα **αντικείμενο** να αναφέρεται σε διαφορετικές κλάσεις (από αυτή από την οποία δημιουργήθηκε) και να επιδρά διαφορετικά σε άλλα αντικείμενα. (*Περιεκτικότητα*)
- Αλλά και σε έναν **τελεστή** να λειτουργεί διαφορετικά μέσα σε ένα πρόγραμμα. (*Υπερφόρτωση τελεστών*)

Περιεκτικότητα

Πολλές φορές χρειάζεται να χρησιμοποιήσουμε την λειτουργικότητα μιας κλάσης από μία άλλη κλάση χωρίς αυτό να συνεπάγεται ότι οι κλάσεις συνδέονται με την ιδιότητα της κληρονομικότητας, χωρίς δηλαδή να επεκτείνουμε την λειτουργικότητα μιας κλάσης σε μία άλλη.

Για παράδειγμα θέλουμε να συσχετίσουμε την κλάση *Accounts* που είδαμε σε προηγούμενες ασκήσεις με την κλάση *Customers* έτσι ώστε να δώσουμε πραγματική υπόσταση στο σενάριο τραπεζικών συναλλαγών που μας δίνει η κλάση *Accounts* για φυσικά πρόσωπα.

Αυτό που κάνουμε είναι να κατασκευάσουμε ένα αντικείμενο *Acc* της κλάσης *Accounts* μέσα στην κλάση *Customers*. Με αυτόν τον τρόπο το αντικείμενο *Acc* μπορεί να εκτελεί μεθόδους της κλάσης *Accounts* μέσα από τη κλάση *Customers*.



Πιο συγκεκριμένα στον κώδικα της κλάσης *Customers* δημιουργούμε το εξωτερικό αντικείμενο *Acc* της κλάσης *Accounts* το οποίο και χειρίζεται το δεδομένο *Balance* με τις μεθόδους του.

Με αυτό τον τρόπο ο οποιοσδήποτε πελάτης πλέον έχει τη δυνατότητα να κάνει τραπεζικές συναλλαγές με την βοήθεια του εξωτερικού αντικειμένου *Acc*.

```
class Customers{
private:
    int ID;
    char firstName[20];
    char lastName[20];
    Accounts Acc; //Δημιουργία εξωτερικού αντικειμένου
public:
    .....
    .....
};
```

Σύνδεση κλάσεων με περιεκτικότητα γίνεται και σε κλάσεις που υπάρχουν σε σχήμα κληρονομικότητας όπου μπορεί να χρησιμοποιηθεί και ταυτόχρονη υπερφόρτωση συναρτήσεων και στην περιεκτικότητα.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΕΡΙΕΚΤΙΚΟΤΗΤΑΣ

Παράδειγμα 1. Περιεκτικότητα σε ανεξάρτητες κλάσεις.

Κλάση Customer.hpp η οποία περιέχει την κλάση Accounts.hpp

```
#include<iostream>
#include<string.h>
#include "Accounts.hpp" //Συμπερίληψη της κλάσης Accounts

class Customers{
private:
    int ID;
    char firstName[20];
    char lastName[40];
    Accounts Acc; //Αντικείμενο της κλάσης Accounts
public:
    Customers(){

    Customers(int _ID,char _firstName[], char _lastName[]){
        ID =_ID;
        strcpy(firstName, _firstName);
        strcpy(lastName, _lastName);
    }

    ~Customers(){

    void takeMoney(){ //Μέθοδος ανάληψης από πελάτη
        float amount;
        cout << "\nPosa xrhmata 8eleis:";
        cin >> amount;

        //Κλήση μεθόδου withdrawal της κλάσης Accounts
        Acc.withdrawal(amount);
    }

    void depositMoney(){ //Μέθοδος κατάθεσης από πελάτη
        float amount;
        cout << "\nPosa xrhmata 8eleis a kata8eseis:";
        cin >> amount;

        //Κλήση μεθόδου deposit της κλάσης Accounts
        Acc.deposit(amount);
    }
    void customerBalance(){ //Μέθοδος ελέγχου υπολοίπου πελάτη

        //Κλήση μεθόδου returnBalance της κλάσης Accounts
        cout << "\nTo trexon ypoloipo einai: " << Acc.returnBalance() << "Euros" << endl;
    }

    void printData(){
        cout << "\nCUSTOMER INFORMATIONS" <<endl;
        cout << "ID: " << ID <<endl;
        cout << "Name: " << firstName << " " << lastName << endl;
        customerBalance(); //Μέθοδος ελέγχου υπολοίπου πελάτη
    }
};
```

Κυρίως πρόγραμμα

```
#include<iostream>
#include"Customers.hpp" //Συμπερίληψη της κλάσης Customers

int main(){

    Customers customer(101,"Marios","Tsoukalas"); //Δημιουργία αντικειμένου πελάτη
    customer.printData(); //Εμφάνιση στοιχείων πελάτη

    customer.depositMoney(1000); //Αίτημα κατάθεσης 1000€ από πελάτη

    customer.customerBalance(); //Αίτημα ερώτησης υπολοίπου από πελάτη
    customer.printData(); //Εμφάνιση στοιχείων πελάτη

return 0;
}
```

Σύντομη εξήγηση του προγράμματος.

Στο πιο πάνω πρόγραμμα έχουμε φτιάξει μία κλάση *Customers* μέσα στην οποία ενσωματώνουμε ένα αντικείμενο *Acc* της κλάσης *Accounts*. Στην κλάση του πελάτη φτιάχνουμε μεθόδους κατάθεσης, ανάληψης και ερώτησης υπολοίπου μέσα στις οποίες εμφανίζεται το εξωτερικό αντικείμενο *Acc* να καλεί τις αντίστοιχες δικές του μεθόδους έτσι ώστε ο πελάτης να κάνει τραπεζικές συναλλαγές.

Μέσα στο κυρίως πρόγραμμα φαίνεται η υλοποίηση του σεναρίου τραπεζικών συναλλαγών όπου το αντικείμενο *customer* χρησιμοποιεί τις δικές του μεθόδους για να κάνει τραπεζικές συναλλαγές

Παράδειγμα 2. Περιεκτικότητα σε ανεξάρτητες κλάσεις με δημιουργία και αποστολή αντικειμένου.

Κλάση *Student2.hpp*

```
#include<iostream>
#include"Lesson2.hpp" //Συμπεριλαμβάνεται η κλάση μαθημάτων
using namespace std;

class Student{
private:
    int ID;
    char name[10];
    char surname[20];
    Lesson newLesson; //Το εξωτερικό αντικείμενο περιέχεται στην κλάση Student
public:
    void setStudentInfos(int anID, char aName[], char aSurname[]){
        ID = anID;
        strcpy(name, aName);
        strcpy(surname, aSurname);
    }

    //Μέθοδος εισαγωγής μαθήματος από αντικείμενα της κλάσης Student
    void setALesson(Lesson aLesson){
        newLesson = aLesson;
    }
    void printStudentInfos(){
        cout << "Student Informations" << endl;
        cout << "ID: " << ID << " - Name: " << name << " - Surname: " << surname << endl;

        //Εμφάνιση στοιχείων μαθήματος από αντικείμενα της κλάσης Student
        newLesson.printLessonInfos();
    }
};
```

Κλάση Lesson2.hpp

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

class Lesson{
private:
    char lessonName[20];
    int semester;

public:
    void setLesson(char aName[], int aSemester){
        strcpy(lessonName, aName);
        semester = aSemester;
    }
    void printLessonInfos(){
        cout << "Lesson Informations" << endl;
        cout << "Name: " << lessonName << " - Semester: " << semester << endl;
    }
};
```

Κυρίως πρόγραμμα

```
#include<iostream>
#include"Student2.hpp" //Συμπεριλαμβάνεται η κλάση φοιτητών
using namespace std;

int main(){
    Lesson newLesson; //Δημιουργία νέου μαθήματος
    newLesson.setLesson("OOP", 2); //Απόδοση τιμών στο νέο μάθημα

    Student newStudent; //Δημιουργία νέου φοιτητή
    newStudent.setStudentInfos(101, "Marios", "Tsoukalas"); //Απόδοση τιμών στον νέο φοιτητή
    newStudent.setALesson(newLesson); //Απόδοση μαθήματος στον φοιτητή

    newStudent.printStudentInfos(); //Εμφάνιση τιμών του φοιτητή

    return 0;
}
```

Σύντομη εξήγηση του προγράμματος.

Στο πιο πάνω πρόγραμμα έχουμε φτιάξει μία κλάση *Student* η οποία συνδέεται με τη ιδιότητα της περιεκτικότητας με την κλάση *Lesson*. Μέσα στην κλάση *Student* έχουμε φτιάξει ένα αντικείμενο της κλάσης *Lesson* το οποίο χειρίζεται τις μεθόδους του για την εισαγωγή νέου μαθήματος στην κλάση φοιτητής.

Στο κυρίως πρόγραμμα φτιάχνουμε ένα νέο μάθημα στο οποίο δίνουμε τιμές μέσω της μεθόδου *setLesson()* της κλάσης *Lesson* και έναν νέο φοιτητή στον οποίο δίνουμε τιμές με την μέθοδο *setStudentInfos()* της κλάσης *Student* αλλά και εγγραφή σε μάθημα με την μέθοδο *setALesson()* της οποίας δίνουμε ως όρισμα το αντικείμενο *newLesson*, δηλαδή το μάθημα που φτιάξαμε. Της στέλνουμε δηλαδή το αντικείμενο *newLesson*.