ΤΕΙ ΣΕΡΡΩΝ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εικονικές, Φίλες Συναρτήσεις - Αρχεία

Ευάγγελος Γ. Ούτσιος
Θεόδωρος Γ. Λάντζος

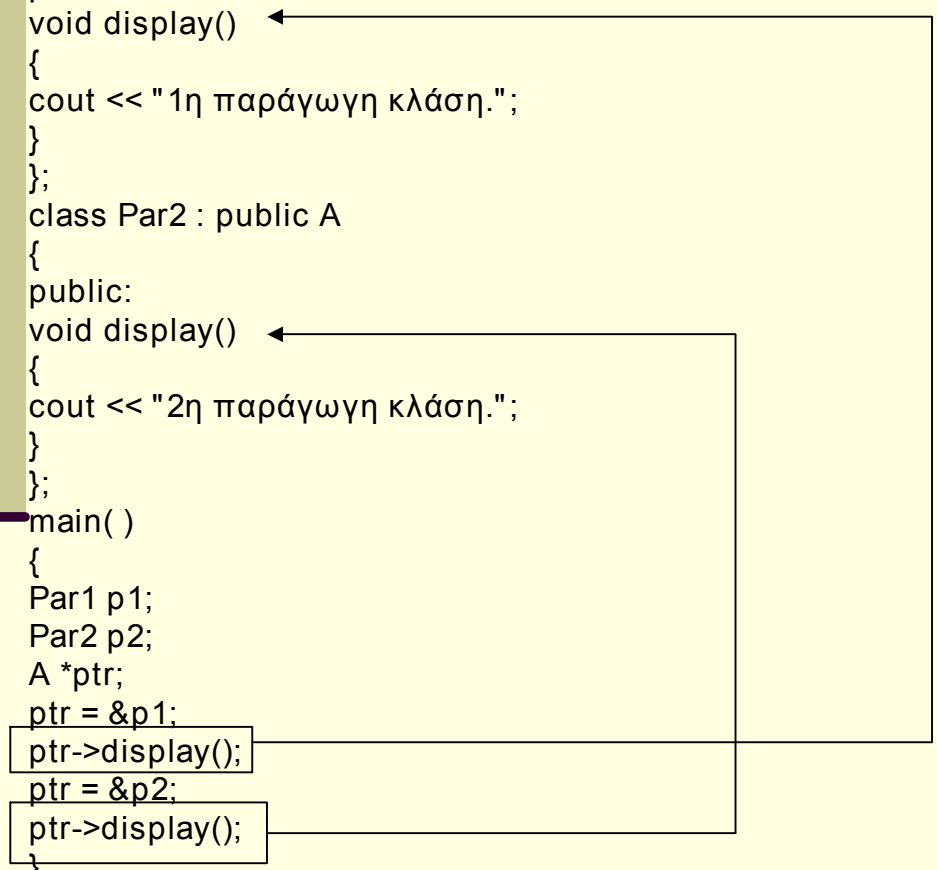
Διάλεξη Νο8

Εικονικές Συναρτήσεις (Virtual Functions)

- Εικονική συνάρτηση είναι μια συνάρτηση που στην πραγματικότητα δεν υπάρχει, αλλά εμφανίζεται σαν πραγματική σε ορισμένα τμήματα ενός προγράμματος
- Δηλώνεται με την δεσμευμένη λέξη virtual μπροστά από το όνομα της συνάρτησης
- Η αρχή των εικονικών συναρτήσεων στηρίζεται στην βάση ότι ο μεταγλωττιστής επιλέγει τη συνάρτηση με βάση το περιεχόμενο του δείκτη και όχι τον τύπο του δείκτη

Παράδειγμα Εικονικής Συναρτήσης

```
#include <iostream.h>
class A
{
public:
virtual void display()
{
cout << "Βασική κλάση.";
}
};
class Par1 : public A
{
public:
void display()
{
cout << "1η παράγωγη κλάση.";
}
};
class Par2 : public A
{
public:
void display()
{
cout << "2η παράγωγη κλάση.";
}
};
main( )
{
Par1 p1;
Par2 p2;
A *ptr;
ptr = &p1;
ptr->display();
ptr = &p2;
ptr->display();
}
```



Έξοδος

1η παράγωγη κλάση

2η παράγωγη κλάση

Φίλες Συναρτήσεις

- Είναι οι συναρτήσεις οι οποίες δεν ανήκουν στις συναρτήσεις μέλη της κλάσης αλλά μπορούν να προσπελάσουν τα ιδιωτικά ή προστατευμένα δεδομένα μιας κλάσης.
- Για να λειτουργήσει μια φιλική συνάρτηση σε μια ή περισσότερες κλάσης πρέπει να δηλώσουμε τον τύπο της συνάρτησης εντός της κλάσης ή των κλάσεων με την δεσμευμένη λέξη friend μπροστά από τον τύπο της συνάρτησης
- Ο κώδικας της συνάρτησης να βρίσκεται εκτός την περιγραφή της κλάσης ή των κλάσεων
- Αποφυγή χρήσης φιλικών συναρτήσεων διότι καταρρέει η αντικειμενοστραφή θεμελίωση του προγράμματος
- Χρήση φιλικών συναρτήσεων μόνο σε εξειδικευμένες περιπτώσεις και κυρίως όταν η σχέση κληρονομικότητας δεν μπορεί να λειτουργήσει ανάμεσα στις αναφερόμενες κλάσεις όπου να οριστεί σχέση κληρονομικότητας από μια κοινή βασική κλάση

Παράδειγμα Φιλικής Συναρτήσης στις κλάσεις A και B

```
class B;
class A
{
  private:
        int x;
  public:
     A()
     {
       x = 5 ;
     }
  friend int ff(A a1, B b1) ;
};
class B
{
  private:
        int y;
  public:
        B()
        {
          y = 4 ;
        }
  friend int ff(A a1, B b1) ;
};
int ff(A a1, B b1)
 {
    return (a1.x+b1.y);
 }
main()
{
  A a;
  B b;
  cout << "Άθροισμα = " << ff(a, b);
 }
```

Δήλωση στην κλάση A

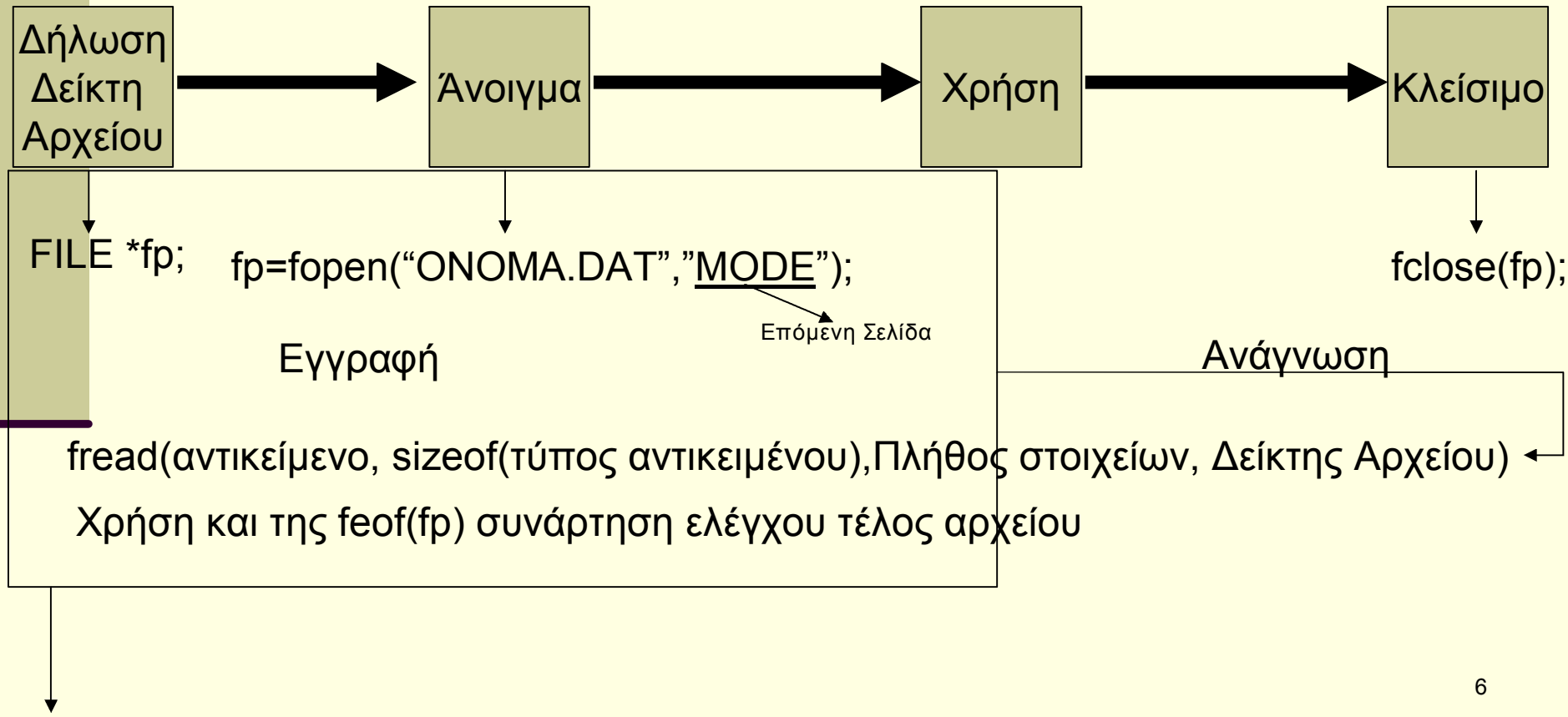
Δήλωση στην κλάση B

Ορισμός

Χρήση

Αρχεία - Δυαδικά

- Η δυνατότητα αποθήκευσης και ανάκτησης αντικειμένων σε μόνιμες μονάδες αποθήκευσης



Καταστάσεις Ανοίγματος Δυαδικών Αρχείων

“rb”	Άνοιγμα δυαδικού αρχείου για διάβασμα
	(το αρχείο πρέπει να υπάρχει)
“wb”	Άνοιγμα δυαδικού αρχείου για γράψιμο
	(αν το αρχείο υπάρχει, σβήνει το περιεχόμενο του)
“ab”	Προσάρτηση σε δυαδικό αρχείο
	(αν το αρχείο δεν υπάρχει, το δημιουργεί)
“rb+”	Άνοιγμα για διάβασμα και γράψιμο
	(το αρχείο πρέπει να υπάρχει)
“wb+”	Άνοιγμα για διάβασμα και γράψιμο
	(αν το αρχείο δεν υπάρχει, το δημιουργεί, αν υπάρχει, σβήνει το περιεχόμενο του)
“ab+”	Άνοιγμα αρχείου για διάβασμα και προσάρτηση
	(αν το αρχείο δεν υπάρχει, το δημιουργεί)

Εγγραφή Αντικειμένου σε Αρχείο - Παράδειγμα

```
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <conio.h>
class Person
{
protected:
int am;
char name[20];
public:
void setData(int am1, char name1[])
{
am = am1;
strcpy(name, name1);
}
};
void main()
{
Person p;
FILE *outfile;
int armit;
char onoma[20];
outfile = fopen("person.dat", “wb”);
cout << "Give am, 0 to stop:";
cin >> armit;
while (armit != 0)
{
cout << "Give name:";
cin >> onoma;
p.setData(armit, onoma);
fwrite(&p, sizeof(p), 1, outfile);
cout << "Give am, 0 to stop:";
cin >> armit;
}
fclose(outfile);
}
```

Δήλωση Κλάσης

Δήλωση Δείκτη Αρχείου

Άνοιγμα Αρχείου για εγγραφή

Εγγραφή Αντικειμένου

Κλείσιμο Αρχείου

Ανάγνωση Αντικειμένου από Αρχείο - Παράδειγμα

```
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <conio.h>

class Person
{
  private:
      int am;
      char name[20];
  public:
      void printData()
      {
         cout << "A.M.: " << am << endl;
         cout << "Name: " << name << endl;
      }
};
void main()
{
   Person p;
   FILE *infile;

   infile = fopen("person.dat", "rb");
   cout << "The contents of the file are:" << endl;
   fread(&p, sizeof(p), 1, infile);
   while (!feof(infile))
   {
      p.printData();
      cout << endl;
      fread (&p, sizeof(p), 1, outfile)
   }
   getch();
   fclose(infile);
}
```

Δήλωση Κλάσης

Δήλωση Δείκτη Αρχείου

Άνοιγμα Αρχείου για εγγραφή

Χρήση συνάρτηση ελέγχου τέλος αρχείου

Ανάγνωση Αντικειμένου

Κλείσιμο Αρχείου

```
//---------------------------------------------------------------------
-----

#pragma hdrstop

#include "Student.h"
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <string.h>

//---------------------------------------------------------------------
-----
#pragma package(smart_init)

void Student::setData(int am1, char name1[], float
grade1)
{
   am = am1;
   strcpy(name,name1);
   grade = grade1;
}
void Student::printData()
{
  cout << "AM     :" << am << endl;
  cout << "Name   :" << name << endl;
  cout << "Grade  :" << grade << endl;
}
```

```
void Student::writeStudents(FILE *sf)
{
   cout << "Give Student AM (0 to finish):";
   cin >> am;
   while(am != 0)
   {
      cout << "Give Name:";
      cin >> name;
      cout << "Give Grade:";
      cin >> grade;
      fwrite(this, sizeof(*this), 1, sf);
      cout << endl;
      cout << "Give Student AM (0 to finish):";
      cin >> am;
   }
}
void Student::readStudents(FILE *sf)
{
   cout << "The contents of the file are: " << endl;
   fread(this, sizeof(*this), 1, sf);
   while(!feof(sf))
   {
     this->printData();
     cout << endl;
     fread(this, sizeof(*this), 1, sf);
   }
   getch();
}
```

Student.cpp

Student.h

```
//---------------------------------------------------------------------------

#ifndef StudentH
#define StudentH
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
//---------------------------------------------------------------------------
#endif

class Student
{
    private:
            int am;
            char name[20];
            float grade;
    public:
            void setData(int am1, char name1[], float grade1);
            void printData();
            void writeStudents(FILE *sf);
            void readStudents(FILE *sf);
};
```

main

```
#pragma hdrstop
#include <stdio.h>
#include "Student.h"

#pragma argsused
void test()
{
   Student s;
   FILE *fp;

   fp = fopen("student.dat","wb");
   s.writeStudents(fp);
   fclose(fp);
   fp = fopen("student.dat","rb");
   s.readStudents(fp);
   fclose(fp);

}

int main(int argc, char* argv[])
{
    test();
    return 0;
}
```