

ΑΡΧΕΙΑ

ΔΗΜΙΟΥΡΓΙΑ ΑΡΧΕΙΟΥ ΚΑΙ ΕΓΓΡΑΦΗ

```
#include <iostream>
#include <fstream> //Κλάση διαχείρισης ρευμάτων διπλής κατεύθυνσης
using namespace std;
int main()
{
    ofstream fileA; //Συμβολικό όνομα (αντικείμενο της κλάσης ροής εξόδου ofstream)
    fileA.open("f1.txt",ios::out|ios::trunc); //Άνοιγμα stream - Πραγματικό όνομα (f1.txt)
    char string[30]="Το αρχείο είναι το: ";
    char fileName[20]="testFile.txt";
    int intNumber=-210;
    double doubleNumber=100.234;

    //Τα παρακάτω οδηγούνται στο αρχείο
    fileA<<"@"<<endl;
    fileA<<intNumber<<endl;
    fileA<<doubleNumber<<endl;
    fileA<<string<<fileName<<endl;
    fileA.close(); //Κλείσιμο stream

    //Τα παρακάτω τυπώνονται στην έξοδο
    cout<<"ΟΛΑ ΤΑ ΠΑΡΑΠΑΝΩ"<<endl;
    cout<<"ΥΠΑΡΧΟΥΝ ΣΤΟ ΑΡΧΕΙΟ askhsh11.txt"<<endl;
    return 0;
} //Τέλος της main
```

Σύντομη εξήγηση του προγράμματος.

Στο πιο πάνω πρόγραμμα δηλώνουμε ένα συμβολικό όνομα αρχείου **fileA** (αντικείμενο) της κλάσης **fstream**, το οποίο στην συνέχεια ανοίγει (**open**) το ρεύμα (**stream**) εξόδου (**ios::out**) για εγγραφή στο αρχείο (**f1.txt**), το οποίο θα δημιουργηθεί στον φάκελο που αποθηκεύουμε τον κώδικα μας. η σημαία (**ios::trunc**) χρησιμοποιείται για να διαγράψει τις πληροφορίες του αρχείου εάν αυτό υπάρχει ήδη.

Στην συνέχεια στέλνουμε στο αρχείο που ανοίξαμε έναν χαρακτήρα ('z'), τους 2 αριθμούς (x και y) και τις 2 συμβολοσειρές (ch και file_name) με την χρήση του τελεστή εξαγωγής ροής << και κλείνουμε το ρεύμα (**close**). Οι 2 τελευταίες εντολές είναι μηνύματα προς τον χρήστη τα οποία θα εμφανιστούν μόλις τρέξουμε το πρόγραμμα.

Μόλις τρέξουμε το πρόγραμμα λοιπόν θα γίνουν τα εξής:

1. Δημιουργία και άνοιγμα ρεύματος,
2. Δημιουργία αρχείου f1.txt και εγγραφή σε αυτό πληροφοριών,
3. Εκτύπωση μηνυμάτων.

Η έξοδος του προγράμματος είναι:

```
OLA TA PARAPANW
YPARCHOYN STO ARXEIO f1.txt
Press any key toy continue
```

Εάν ανοίξουμε το αρχείο f1.txt (με έναν text editor) θα δούμε τα παρακάτω:

```
A
10
1.234
to arxeio einai to: f1.txt
```

τα οποία είναι οι πληροφορίες που στείλαμε από το πρόγραμμα.

Παρατηρήσεις:

1. Αν αλλάξουμε κάποια από τις μεταβλητές που στέλνουμε στο αρχείο και ξανατρέξουμε το πρόγραμμα τότε το αρχείο θα έχει τις νέες τιμές.

Π.χ η **x** να γίνει **35** και ο χαρακτήρας **A** να γίνει **z** τότε το αρχείο θα πάρει τις νέες τιμές:

```
Z
35
1.234
to arxeio einai to: f1.txt
```

2. Αν θέλουμε να γράψουμε τις παραπάνω αλλαγές μετά το τέλος του αρχείου, μέσα στην δήλωση ανοίγματος του ρεύματος θα χρησιμοποιήσουμε την σημαία (flag) **app** της ios δηλαδή θα γράψουμε:

```
fileA.open("f1.txt",ios::app);
```

Προσέξτε ότι δεν ξαναχρησιμοποιούμε το flag **out** της ios.

Ξανατρέχοντας το πρόγραμμα το αρχείο θα πάρει την πιο κάτω μορφή:

```
X
10
1.234
to arxeio einai to: f1.txt
z
35
1.234
to arxeio einai to: f1.txt
```

ΑΝΑΓΝΩΣΗ ΠΕΡΙΕΧΟΜΕΝΩΝ ΠΡΟΚΑΘΟΡΙΣΜΕΝΟΥ ΑΡΧΕΙΟΥ

```
#include <iostream.h>
#include <fstream.h>
void main()
{
    ifstream fileB; //αντικείμενο της κλάσης ροής εισόδου ifstream
    char chA;
    fileB.open("f1.txt",ios::in); //άνοιγμα ρεύματος εισόδου

    /* μέθοδος get μέσα σε βρόχο while για την επιστροφή των
    πληροφοριών του αρχείου */
    //λειτουργία βρόχου έως το τέλος του αρχείου
    while(!fileB.eof()){
        fileB.get(chA);
        cout<<chA;
    } //τέλος της while

    fileB.close(); //κλείσιμο ρεύματος
} //τέλος της main
```

Σύντομη εξήγηση του προγράμματος

Στο παραπάνω πρόγραμμα δηλώνω ένα συμβολικό όνομα αρχείου **fileB** (αντικείμενο) της κλάσης **ifstream**. Μετά ανοίγω το ρεύμα εισόδου με flag (**ios::in**) για ανάγνωση του αρχείου f1.txt. Τέλος για να διαβάσω ένα-ένα τα περιεχόμενα (χαρακτήρες) του αρχείου και να τα στέλνω στην οθόνη, χρησιμοποιώ την συνάρτηση **get(ch)** μέσα σε μία δομή επανάληψης while. Για όσο δεν επιστρέφεται **eof** (enf of file), δηλαδή μέχρι να βρεθεί το τέλος του αρχείου, η get() επιστρέφει χαρακτήρες οι οποίοι τυπώνονται στην οθόνη. Μόλις φτάσει το τέλος του αρχείου, σταματά να λειτουργεί η while και το ρεύμα κλείνει με την μέθοδο **close()**.

Σημ. χρησιμοποιούμε τη μεταβλητή χαρακτήρα **chA** για να συλλέξουμε πληροφορίες από το αρχείο διότι ότι είναι γραμμένο σε αυτό αντιμετωπίζεται ως ASCII χαρακτήρας.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ifstream fileB; //αντικείμενο της κλάσης ροής εισόδου ifstream
    char chA;
    char file_name[20];
    cout<<"Dose onoma arxeiou gia anagnosh: ";
    cin>>file_name;

    /* άνοιγμα stream με το όνομα αρχείου
    Που θα δώσει ο χρήστης */
    fileB.open(file_name,ios::in);

    /* συνάρτηση get() μέσα σε βρόχο while
    Για την επιστροφή των πληροφοριών του αρχείου */
    while(!fileB.eof())
    {
        fileB.get(chA);
        cout<<chA;
    } //τέλος του while

    fileB.close(); //κλείσιμο stream
    return 0;
} //Τέλος της main
```

Σύντομη εξήγηση του προγράμματος

Στο παραπάνω πρόγραμμα ο χρήστης δίνει το όνομα του αρχείου του οποίου τα δεδομένα θέλουμε να προσπελάσουμε. Η διαδικασία είναι ίδια με το προηγούμενο πρόγραμμα

ΑΡΧΕΙΑ ΚΑΙ ΚΛΑΣΕΙΣ

ΔΗΜΙΟΥΡΓΙΑ ΑΡΧΕΙΩΝ ΜΕ ΔΕΔΟΜΕΝΑ ΑΠΟ ΑΝΤΙΚΕΙΜΕΝΑ ΚΛΑΣΕΩΝ 1

Το παρακάτω πρόγραμμα δηλώνουμε μία κλάση ημερομηνιών με σκοπό να εισάγουμε ημερομηνίες από αντικείμενα σε αρχείο και να εμφανίσουμε ημερομηνίες από αντικείμενα από αρχείο.

```
#include <iostream.h>
#include <fstream.h>
class date //klash
{
private:
    int givenDay, givenMonth, givenYear; //dedomena
public:
    void setDate() //me8odos gia thn eisagvgh
    {
        cout<<"Give Day: "; cin>>givenDay;
        cout<<"Give Month: "; cin>>givenMonth;
        cout<<"Give Year: "; cin>>givenYear;
    }
    void printDate() //me8odos gia thn ektyposh
    {
        cout<< givenDay <<"/"<< givenMonth <<"/"<< givenYear <<endl;
    }
}; //telos klashs
```

```

void main()
{
    ofstream outDate; //stream exodou
    ifstream inDate; //stream eisodou
    char choice;
    date idate,odate; //antikeimena eisagoghs-exagoghs
    // anoigoume to arxeio gia eggraph dyadikwn dedomenwn
    outDate.open("date.txt",ios::out|ios::binary);
    while(1) //1>0 dhladh panta
    {
        cout<<"EISAGOGH HMEROMHNIAS (n=Nai/o=Oxi): ";
        cin>>choice;
        if(choice=='n') //ean epilexoume to n
        {
            /*to ant hm1 diabazei hmeromhnia
            kai thn apodidei olh mazi me thn synarthsh write
            san omada dyadikwn dedomenwn */
            idate.setDate();
            outDate.write((char *) &idate, sizeof(idate));
        }
        else break; //aliws vgainoume
    }
    outDate.close(); //kleisimo stream eggraphs
    inDate.open("date.txt", ios::in|ios::binary);
    cout<<"Oi hmeromhnies poy edoses einai....."<<endl;
    while(!inDate.eof())
    {
        /*sto ant hm2 erxontai ta dedomena toy arxeioy
        ola mazi me thn synarthsh read
        san omada dyadikwn dedomenwn */
        inDate.read((char *) &odate, sizeof(odate));
        if(!inDate.eof()) //diabasma ews to telos toy arxeioy
        {
            odate.printDate(); //ektyposh dedomenwn arxeioy
        }
    }
    inDate.close();//kleisimo stream anagnoshs
} //telos ths main

```

Σύντομη εξήγηση του προγράμματος

Στο πιο πάνω πρόγραμμα φτιάχνουμε μία κλάση ημερομηνίας με 3 δεδομένα μέρα, μήνα, έτος και 2 μεθόδους για την εισαγωγή και τη εμφάνιση της ημερομηνίας.

Στην main() φτιάχνουμε 2 streams ένα για την εγγραφή (*outDate*) και ένα για την ανάγνωση (*inDate*) του αρχείου που θα καταχωρηθούν οι ημερομηνίες (*date.txt*). Δημιουργούμε 2 αντικείμενα που στην ουσία το πρώτο θα οδηγήσει τα δεδομένα από το πληκτρολόγιο στο αρχείο και το δεύτερο θα οδηγήσει τα δεδομένα από το αρχείο στην οθόνη. Τα δεδομένα θα οδηγηθούν όλα μαζί από το πρόγραμμα στο αρχείο σαν μπλοκ δεδομένων με τη συνάρτηση write και το αντικείμενο idate και από το αρχείο προς το πρόγραμμα με την συνάρτηση read και το αντικείμενο odate.

Στην εισαγωγή των ημερομηνιών έχουμε το δικαίωμα επιλογής εισαγωγής με το βρόχο while και το prompt (n=Nai/o=Oxi) και το αντικείμενο idate να καλεί την μέθοδο εισαγωγής τιμών setDate() της κλάσης ή μη εισαγωγής με την break να μας βγάζει από το βρόγχο.

Κατά την ανάγνωση των δεδομένων του αρχείου χρησιμοποιούμε ένα βρόχο while με συνθήκη επιστροφής eof και κλήση από το αντικείμενο odate της μεθόδου εκτύπωσης των δεδομένων στην οθόνη printDate() .

ΔΗΜΙΟΥΡΓΙΑ ΑΡΧΕΙΩΝ ΜΕ ΔΕΔΟΜΕΝΑ ΑΠΟ ΑΝΤΙΚΕΙΜΕΝΑ ΚΛΑΣΕΩΝ 2

Το παρακάτω πρόγραμμα δηλώνουμε μία κλάση φοιτητής εγγράφοντας και διαβάζοντας δεδομένα από αρχείο με την βοήθεια ενός stream.

```
#include <iostream.h>
#include <fstream.h>
class foithths //dhlosh klashs
{
private:
    char name[20], surname[20];
    int mhtrwo;
public:
    void pareStoixeia() //me8odos eisagoghs stoixeiw
    {
        cout<<"Dose Onoma: "; cin>>name;
        cout<<"Dose Epwnymo: "; cin>>surname;
        cout<<"Dose A.M.: "; cin>>mhtrwo;
    }
    void emfaniseStoixeia() //me8odos emfanishs stoixeiw
    {
        cout<<"Onomatepwnymo"<<endl;
        cout<<surname<<' '<<name<<endl;
        cout<<"A.M.: "<<mhtrwo<<endl;
    }
}; //telos klashs
void main()
{
    fstream File; //dhlosh toy stream
    char ch;
    foithths f1; //antikeimeno ths klashs
    File.open("Foithths.txt",ios::out|ios::binary); //anoigma stream
    cout<<"Eisagogh stoixeiw....."<<endl;
    do
    {
        f1.pareStoixeia(); //eisagogh stoixeiw
        File.write((char *) &f1, sizeof(f1)); //eggrafh sto arxeio
        cout<<"Yparxoy n alloi foithtes? (n=Nai/o=Oxi)";
        cin>>ch;
    }while(ch!='n');
    File.close(); //kleisimo stream eggrafhs
    File.open("Foithths.txt",ios::in|ios::binary); //anoigma stream
    cout<<"======"<<endl;
    cout<<"To arxeio Foithths.txt perixe ta parakatw..."<<endl;
    cout<<"======"<<endl;
    File.read((char *) &f1, sizeof(f1));
    do
    {
        f1.emfaniseStoixeia(); //ektyposh stoixeiw
        File.read((char *) &f1, sizeof(f1)); //anagnosh apo arxeio
    }while(!File.eof());
    File.close(); //kleisimo stream anagnmoshs
} //telos ths main
```

Σύντομη εξήγηση του προγράμματος

Το παραπάνω πρόγραμμα ακολουθεί την ίδια λογική του προηγούμενου με την μόνη διαφορά ότι χρησιμοποιούμε ένα μόνο stream για την εγγραφή και την ανάγνωση των δεδομένων στο αρχείο αλλά και ότι η εισαγωγή και η εμφάνιση των στοιχείων γίνεται με δομή επανάληψης do/while με τα ίδια αποτελέσματα.