



Ελληνική Δημοκρατία  
Τεχνολογικό Εκπαιδευτικό  
Ίδρυμα Ηπείρου

# Αντικειμενοστραφής Προγραμματισμός

## Ενότητα 8 : Αρχεία και Streams

Ιωάννης Τσούλος



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Τμήμα Μηχανικών Πληροφορικής Τ.Ε

## Αντικειμενοστραφής Προγραμματισμός

Ενότητα 8 : Αρχεία και Streams

Ιωάννης Τσούλος

Επίκουρος Καθηγητής

Άρτα, 2015





# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





# Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Εισαγωγικά

- **Αρχεία:** Ακολουθίες χαρακτήρων.
- Η C προσφέρει ορισμένες δομές δεδομένων FILE (struct, ο πρόγονος της κλάσης) και ανεξάρτητες συναρτήσεις που έχουν πρόσβαση στα αρχεία.
- Η όλη διαδικασία του προγραμματισμού γίνεται με pointers στις δομές FILE και είναι αρκετά εύκολο να γίνει κάποιο λάθος κατά τη χρήση τους.
- Επιπλέον δε προσφέρουν καμία από τις ευκολίες του αντικειμενοστραφούς προγραμματισμού (κληρονομικότητα, πολυμορφισμό, κλπ).



# Εισαγωγικά

- Για το σκοπό αυτό η C++, υιοθέτησε τη χρήση των **streams**.
- Δεν παρέχεται απευθείας πρόσβαση των δεδομένων των αρχείων, παρά μόνο ως ανταλλαγή bytes από το πρόγραμμα προς το αρχείο και αντίστροφα.
- Η χρήση των streams ως ροές δεδομένων από και προς το αρχείο, έχει το άμεσο όφελος ότι δεν είναι πλέον σημαντικό το μέσον στο οποίο βρίσκεται το αρχείο αλλά μόνο η εναλλαγή της πληροφορίας από και προς το πρόγραμμα.



# Εισαγωγικά

- Τέτοιες κλάσεις είναι η `fstream` και οι πιο συγκεκριμένες `ifstream`, `ofstream`.
  - **`ifstream`**: αναφέρεται στο άνοιγμα αρχείου μόνο για ανάγνωση (input file stream)
  - **`ofstream`**: χρησιμοποιείται μόνο για εγγραφή στο αρχείο (output file stream)
  - **`fstream`**: προσφέρει ταυτόχρονα δυνατότητα για ανάγνωση και εγγραφή



# Η κλάση `fstream`

- Η κλάση `fstream` (δηλώνεται στο αντίστοιχο αρχείο κεφαλίδας), προσφέρει αρκετές μεθόδους για:
  - ανάγνωση, εγγραφή και μετακίνηση στο αρχείο
  - αρκετά `flags` για τροποποίηση των δεδομένων κατά την εγγραφή ή την ανάγνωση.





# Η κλάση `fstream`

| Όνομα  | Λειτουργία   |
|--|--|
| <pre>bool good() bool bad()</pre>                    | <p>Η <code>good()</code> επιστρέφει <code>true</code> αν το αρχείο είναι σε καλή κατάσταση (αν δεν έχει συμβεί κάποιο σφάλμα εισόδου/εξόδου) ή <code>false</code> διαφορετικά. Η <code>bad()</code> έχει την αντίθετη συμπεριφορά.</p> |
| <pre>bool eof()</pre>                                | <p>Επιστρέφει <code>true</code> αν έχουμε φτάσει στο τέλος του αρχείου κατά την ανάγνωση.</p>  |
| <pre>void flush()</pre>                              | <p>Αποθηκεύει όλες τις αλλαγές που έχουν γίνει στο αρχείο και δεν βρίσκονται στους <code>buffers</code> του συστήματος.</p>  |
| <pre>int get()</pre>                                 | <p>Επιστρέφει τον επόμενο χαρακτήρα του αρχείου.</p>   |
| <pre>istream &amp; getline(char *buf, int num)</pre> | <p>Αντιγράφει από το αρχείο στο πίνακα χαρακτήρων <code>buf</code>, το μέγιστο <code>num</code> χαρακτήρες εκτός αν φτάσει στο τέλος της γραμμής ή του αρχείου. Επιστρέφει μια αναφορά στο αρχείο.</p>                                 |



| <b>Όνομα</b>   | <b>Λειτουργία</b>   |
|--|---|
| <code>int gcount()</code>  | Επιστρέφει τον αριθμό των χαρακτήρων που αναγνώστηκαν κατά την προηγούμενη εκτέλεση των μεθόδων <code>get()</code> , <code>read()</code> .  |
| <code>istream &amp;read(char *buf, int num)</code>   | Αντιγράφει από το αρχείο στο πίνακα χαρακτήρων <code>buf</code> , το μέγιστο <code>num</code> χαρακτήρες εκτός αν φτάσει στο τέλος του αρχείου. Επιστρέφει μια αναφορά στο αρχείο.  |
| <code>ostream &amp;write(char *buf, int num)</code>  | Αντιγράφει από το πίνακα χαρακτήρων <code>buf</code> στο αρχείο, το μέγιστο <code>num</code> χαρακτήρες. Επιστρέφει μια αναφορά στο αρχείο.   |
| <code>void seekg(int offset, origin)</code><br><code>void seekp(int offset, origin)</code> | Ορίζει τη τρέχουσα θέση του δρομέα για ανάγνωση ( <code>seekg</code> ) ή για εγγραφή ( <code>seekp</code> ). Τα <code>streams</code> στη C++ χρησιμοποιούν δύο διαφορετικούς δείκτες, ένα για ανάγνωση δεδομένων και ένα για εγγραφή. |
| <code>int tellg()</code><br><code>int tellp()</code>                                       | Επιστρέφει την αντίστοιχη θέση του δρομέα για ανάγνωση ( <code>tellg</code> ) ή για εγγραφή ( <code>tellp</code> ).   |



# Η κλάση `fstream`

- Απλό παράδειγμα προγράμματος αντιγραφής δύο αρχείων, σε δύο εκδοχές:
  - A. Παράδειγμα προγράμματος αντιγραφής δύο αρχείων **ανά χαρακτήρα (byte)**
  - B. Παράδειγμα προγράμματος αντιγραφής δύο αρχείων **αντιγράφοντας blocks των 256KB κάθε φορά**



1/2

```
#include <fstream>
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {
    // Έλεγχος για το αν το πρόγραμμα δέχεται 3 παραμέτρους.
    // Σημειώνουμε ότι στη C++ η πρώτη παράμετρος argv[0] είναι
    // πάντα το ίδιο το εκτελέσιμο πρόγραμμα.
    if (argc != 3) {
        cout << "Usage: CopyFile <from> <to>" << endl;
        return 0;
    }
    // Δοκιμάζουμε να ανοίξουμε το αρχείο <from> για ανάγνωση (παράμετρος
    // argv[1]).
    // Η διαδικασία θα αποτύχει αν το αρχείο δεν υπάρχει ή αν δεν έχουμε
    // πρόσβαση σε αυτό.
    ifstream fin(argv[1]);
    if (fin == 0) {
        cout << "Error: Input file cannot be opened for reading!" << endl;
        return 10;
    }
    // Δοκιμάζουμε να ανοίξουμε το αρχείο <to> για εγγραφή (παράμετρος
    // argv[2]).
    // Η διαδικασία θα αποτύχει αν δεν έχουμε πρόσβαση ή αν δεν υπάρχει
    // χώρος στο δίσκο.
    ofstream fout(argv[2]);
    if (fout == 0) {
        cout << "Error: Output file cannot be opened for writing!" << endl;
        return 10;
    }
}
```



```
// Για να μάθουμε το τέλος του αρχείου, χρησιμοποιούμε τις μεθόδους
// seekg() και tellg(). Η seekg() αλλάζει την τρέχουσα θέση του αρχείου,
// ενώ η tellg() επιστρέφει αυτή τη θέση.
// Πρακτικά αυτό που κάνουμε είναι να πάμε το δρομέα (cursor) στη θέση που
// έχει offset θ από το τέλος του αρχείου (ios::end) και να διαβάσουμε τη
// θέση που επιστρέφει η tellg(). Αυτό είναι και το μέγεθος του αρχείου.
fin.seekg(θ, ios::end);
size_t finsize = fin.tellg();
cout << "Input file size: " << finsize << endl;
// Επειδή θέλουμε να αντιγράψουμε το αρχείο, επιστρέφουμε τον δρομέα στην
// αρχή του αρχείου, offset θ bytes από τη θέση ios::beg).
fin.seekg(θ, ios::beg);
// Επειδή θέλουμε να αντιγράψουμε το αρχείο, επιστρέφουμε τον δρομέα στην
// αρχή του αρχείου, offset θ bytes από τη θέση ios::beg).
fin.seekg(θ, ios::beg);
// Θα αντιγράψουμε τώρα τα περιεχόμενα του αρχείου fin στο fout.
// Η αντιγραφή θα γίνει byte προς byte και θα τυπώνουμε την πρόοδο
int c;
int percent = 0;
while (fin.eof() == false) { // έχουμε φτάσει στο τέλος του αρχείου;
c = fin.get(); // διάβασε ένα byte
if (fin.eof() == false) // αν δεν έχουμε φτάσει στο τέλος
fout.put(c); // γράψε το byte
cout << "Copy Completed : " << 100*percent/finsize << "\r";
percent++;
}
fin.close();
fout.close();
return 0;
}
```

2/2



# Η κλάση `fstream`

- Η αντιγραφή ενός αρχείου περίπου 40MB διήρκεσε 20 δευτερόλεπτα. Με μια μικρή αλλαγή στον κώδικα θα δείξουμε πώς η ίδια διαδικασία μπορεί να διαρκέσει μόλις 4 δευτερόλεπτα στον ίδιο υπολογιστή και για το ίδιο αρχείο! Συγκεκριμένα θα αλλάξουμε τον κωδικά του βρόχου `while`:

```
// Θα αντιγράψουμε τώρα τα περιεχόμενα του αρχείου fin στο fout.
// Η αντιγραφή θα γίνει ανά block και θα τυπώνουμε την πρόοδο
size_t bufsize = 262144; // το μέγεθος του block
char buf[bufsize]; // ορίζουμε το buffer από bytes
int count; // ο μετρητής των bytes που γράφονται
int total = 0;
while (fin.eof() == false) { // έχουμε φτάσει στο τέλος του αρχείου;
    fin.read(buf, bufsize); // διάβασε το πολύ bufsize bytes
    count = fin.gcount(); // μέτρα πόσα πραγματικά διαβάστηκαν
    if (count) // αν έχει διαβαστεί έστω και 1 byte
        fout.write(buf, count); // γράψε τα στο αρχείο fout
    cout << "Copy Completed : " << 100*total/fin.size() << "%\r";
    total += count;
}
```



# Οι τελεστές <<, >>

- Η `fstream` προσφέρει έναν πολύ **εύχρηστο τρόπο ανταλλαγής δεδομένων με το αρχείο**, μέσω των τελεστών `<<` και `>>` που δηλώνουν την κατεύθυνση από (`>>`) και προς (`<<`) το αρχείο.
- Τους έχουμε ήδη χρησιμοποιήσει με την πρότυπη έξοδο (κονσόλα) `cout`. Αυτό ισχύει γιατί και η `cout` είναι ένα **προκαθορισμένο αντικείμενο** `ostream` (η γονική κλάση της `ofstream`) που αντιστοιχεί στην πρότυπη έξοδο (`stdout`).
- Αντίστοιχα η πρότυπη είσοδος `cin` είναι ένα `istream` (η γονική κλάση της `ifstream`) που δίνει πρόσβαση στην `stdin`, και η πρότυπη έξοδος λαθών `cerr` που αντιστοιχεί στην `stderr`.

# Οι τελεστές <<, >>

- **Τρόπος χρήσης.** Έστω `fin`, `fout` δύο αντικείμενα `ifstream` και `ofstream` αντίστοιχα:

```
#include <fstream>
#include <iostream>
using namespace std;
int main() {
    ifstream fin("data.txt");
    if (fin == 0) {
        cout << "Error: could not open file data.txt"
        << endl;
        return 10;
    }
    ofstream fout("output.txt");
    if (fout == 0) {
        cout << "Error: could not open file output.txt"
        << endl;
        return 10;
    }
}
```

```
// Δήλωση των μεταβλητών που θα χρησιμοποιηθούν.
string name;
double age;
// όσο δεν έχουμε φτάσει στο τέλος του αρχείου
while (fin.eof() == false) {
    // διάβασε κάθε στήλη στην αντίστοιχη μεταβλητή
    fin >> name >> age;
    if (fin.eof() == false) {
        // τύπωσε τα στην κονσόλα
        cout << "0 " << name << " είναι " << age << "
        ετών." << endl;
        // και γράψε το ίδιο κείμενο στο αρχείο
        output.txt
        fout << "0 " << name << " είναι " << age << "
        ετών." << endl;
    }
}
```





# Οι τελεστές <<, >>

- Αν το αρχείο *data.txt* έχει π.χ. τα εξής δεδομένα:

```
Κώστας 29.7  
Νίκος 34.4  
Γιαννάκης 6.5  
Μαθουσάλας 803.4
```

- Το αποτέλεσμα του προγράμματος θα αποθηκευτεί στο αρχείο *output.txt* και θα είναι:

```
0 Κώστας είναι 29.7 ετών.  
0 Νίκος είναι 34.4 ετών.  
0 Γιαννάκης είναι 6.5 ετών.  
0 Μαθουσάλας είναι 803.4 ετών.
```



# Streams

- Στη γλώσσα C, η μορφοποίηση ενός `string` μέσω παραμέτρων γίνεται με την εντολή `sprintf`, που λειτουργεί με τον ίδιο τρόπο όπως η `printf` για την πρότυπη έξοδο (`stdout`) ή η `fprintf` για την έξοδο σε αρχείο.
- **Αντίστοιχα**, η λειτουργία των `streams` και η ευκολία των τελεστών `<<` και `>>` παρέχεται στη C++ και για `strings`.
- Για το σκοπό αυτό έχει υλοποιηθεί η κλάση `stringstream` (στο αρχείο κεφαλίδας `sstream`), και χρησιμοποιείται όπως ακριβώς ένα `fstream`.



# Stringstreams

```
#include <string>
#include <sstream>
#include <iostream>
using namespace std;
int main() {
    stringstream formatted;
    // Δήλωση των μεταβλητών που θα χρησιμοποιηθούν.
    int data[] = {10, 5, 4, 3, 8, 11};
    string names[] = {"one", "two", "three", "four",
                     "five", "six"};
    for (int i=0; i < 6; i++) {
        formatted << "Name: " << names[i] << ", value: "
        << data[i] << endl;
    }
    cout << formatted.str();
}
```

→ cout << formatted.str();



```
Name: one, value: 10
Name: two, value: 5
Name: three, value: 4
Name: four, value: 3
Name: five, value: 8
Name: six, value: 11
```

- Η μέθοδος `str()` του `stringstream` “παγώνει” τα περιεχόμενά του και επιστρέφει ένα αντικείμενο `string` το οποίο μπορούμε να τυπώσουμε.
- Δεν είναι αποδεκτό να τυπώσουμε ένα αντικείμενο `stringstream` απευθείας.



# Βιβλιογραφία

1. Εγχειρίδιο της C++, 2η Ελληνική έκδοση, Jesse Liberty, Γκιούρδας.
2. Μάθετε τη C++, 2η Ελληνική έκδοση, Jesse Liberty , Γκιούρδας.
3. Προγραμματισμός με τη γλώσσα C++ Μέρος Α, Αλεβίζος Θ., Έκδοση ΤΕΙ Καβάλας
4. C++ Αντικειμενοστραφής Προγραμματισμός Υπολογιστών Τομαράς Α., , Εκδόσεις Νέων Τεχνολογιών.
5. Ανακαλύψτε τη γλώσσα C, J. Purdum, Εκδόσεις Δίαυλος.
6. Εισαγωγή στο Συστηματικό Προγραμματισμό και στη γλώσσα C++, Σ. Μπαλτζής, εκδόσεις πανεπιστημίου Ιωαννίνων.
7. C++ From the beginning, Jan Skansholm, Addison Wesley.
8. The design and analysis of computer algorithms, A.V. AHO, J.E. HOPCROFT, J.D. ULLMANN, Addison Wesley 1974.
9. Structure and Interpretation of Computer Programs, H. ABELSON, G.J. SUSSMAN, J. SUSSMAN, MIT Press, Mc Graw Hill Book Company, 1985
10. The art of computer programming, D.E. KNUTH, Addison-Wesley.



# Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Ιωάννης Τσούλος.

Αντικειμενοστραφής Προγραμματισμός.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή  
διεύθυνση:

<http://eclass.teiep.gr/courses/COMP113/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



# Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης  
Άρτα, 2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο

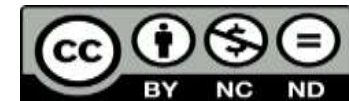


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Τέλος Ενότητας

## Αρχεία και Streams



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

