

Input / Output in C++

~ part 1 ~

Input / Output in C++ - Streams

- Η είσοδος και η έξοδος δεδομένων, στη C++ βασίζεται στην έννοια των “ρευμάτων” - *streams*
- Ένα ρεύμα είναι ουσιαστικά μια ακολουθία χαρακτήρων (γενικότερα bytes)
- Τα ρεύματα μπορεί να είναι για είσοδο, για έξοδο ή και για τα δύο

Οθόνη & Πληκτρολόγιο

- Σε αντιστοιχία με τα standard ρεύματα εισόδου / εξόδου στη C (*stdin*, *stdout*, *stderr*)
- Ορισμένα από την αρχή του προγράμματος, στο namespace *std* υπάρχουν τα *cin*, *cout*, *cerr*
- Το *cin* είναι για το ρεύμα που είναι συνδεδεμένο με την standard είσοδο

Οθόνη & Πληκτρολόγιο

- Τα ρεύματα cout και cerr είναι συνδεδεμένα με την standard έξοδο
- Το cout προορίζεται για την “κανονική” εκτύπωση μηνυμάτων του προγράμματος
- Το cerr προορίζεται για την εκτύπωση μηνυμάτων λαθών
- Το κάθε ένα ανεξάρτητα από το άλλο μπορεί να ανακατευθυνθεί σε αρχείο

Παράδειγμα standard IO

```
#include <iostream>                // Απαραίτητο για τα ρεύματα εισόδου/εξόδου
using namespace std;
int main(void)
{
    cout << "Give a positive number and a character: "; // Δεν αλλάζει γραμμή
    int number; char c;
    cin >> number >> c;          // Διάβασμα ενός ακεραίου και ενός χαρακτήρα
    if ( i > 0 )
        cout << number << " - " << c << endl; // endl για αλλαγή γραμμής
    else                                // και άδειασμα του buffer εξόδου
        cerr << "Non-positive" << endl;
}
```

Παράδειγμα standard IO

\$./a.out

Give a positive number and a character: 13 c

13 - c

\$./a.out

Give a positive number and a character: -1b

Non-positive

\$./a.out

Give a positive number and a character: 12b

12 - b

ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Για την χρήση των παραπάνω ρευμάτων, πρέπει να γίνει include το <fstream>

Παράδειγμα ifstream

```
#include <iostream>
#include <fstream>
using namespace std;
int main(void) {
    ifstream in("data.txt");
    while (true) {
        int i; char buf[5];
        in >> i >> buf;
        if ( !in.good() ) break;
        cout << i << " " << buf << endl;
    }
}
```


Παράδειγμα ifstream

```
$ cat data.txt
```

```
12 lala
```

```
20 koko
```

```
40 foo
```

```
$ ./a.out
```

```
12 lala
```

```
20 koko
```

```
40 foo
```

ifstream

- Το `ifstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cin`
- Με την συνάρτηση-μέλος `good`, ελέγχουμε αν η ενέργεια διαβάσματος ήταν επιτυχής
- Σε αντιστοιχία με την C, υπάρχει συνάρτηση-μέλος `eof`, που μας ενημερώνει για το τέλος του αρχείου
- Δεν είναι αναγκαστικό να κάνουμε `close` τα ρεύματα των αρχείων, καθώς κάτι τέτοιο γίνεται κατά την καταστροφή τους

ifstream

- Με την συνάρτηση-μέλος *peek* μπορούμε να δούμε τον επόμενο χαρακτήρα στο ρεύμα, χωρίς να αφαιρεθεί από αυτό
- Με την συνάρτηση-μέλος *ignore* μπορούμε να παραβλέψουμε ένα πλήθος χαρακτήρων από το ρεύμα
- Με την συνάρτηση-μέλος *getline* μπορούμε να διαβάσουμε μια γραμμή από το ρεύμα και να την αποθηκεύσουμε σε κάποιον πίνακα χαρακτήρων

ifstream

- Η συνάρτηση-μέλος `getline` μπορεί να αποθηκεύσει το αποτέλεσμα μόνο σε πίνακα χαρακτήρων
- `char title[256]; in.getline (title,256);`
- Για να μπορούμε να χρησιμοποιήσουμε `strings`, υπάρχει η συνάρτηση `getline` ορισμένη στο `<string>`
- `string title; getline(in, title);`

Error flags in streams

- Τα ρεύματα έχουν 3 bits τα οποία παρέχουν πληροφορίες για την κατάσταση τους
- *eofbit* – δείχνει αν έχει τελειώσει το αρχείο με το οποίο ήταν συνδεδεμένο το ρεύμα (EOF)
- *badbit* – δείχνει αν υπάρχει πρόβλημα με το ίδιο το ρεύμα
- *failbit* – δείχνει αν υπήρξε πρόβλημα με την τελευταία ενέργεια πάνω στο ρεύμα (π.χ. διάβασμα ακεραίου, ενώ δεν υπήρχε ακέραιος για να διαβαστεί)

Error flags in streams

- Για κάθε bit, υπάρχει και αντίστοιχη συνάρτηση-μέλος (*eof*, *fail*, *bad*)
- Η συνάρτηση-μέλος *good* ελέγχει αν δεν είναι set (1) κανένα από τα τρία αυτά bits
- Η συνάρτηση-μέλος *clear* μπορεί να αλλάξει τιμές στην κατάσταση του stream, και αν κληθεί χωρίς ορίσματα, κάνει unset (0) όλα τα error flags

ifstream & μετακίνηση στο αρχείο

- Μαζί με κάθε input stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *get pointer*
- Με την συνάρτηση-μέλος *tellg* μαθαίνουμε για το που βρίσκεται ο get pointer
- Με την συνάρτηση-μέλος *seekg* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

Παράδειγμα ofstream

```
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{
    ofstream out("data2");
    out << "Hello world" << endl;
    out << 42 << " - " << (5/10) << endl;
    out.close();
}
```


Παράδειγμα ofstream

```
$ ./a.out
```

```
$ cat data2.txt
```

```
Hello world
```

```
42 - 0
```

ofstream

- Το `ofstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cout`
- Τα δεδομένα δεν είναι αναγκαστικό να γραφτούν αμέσως στο αρχείο (buffered IO)
- Υπάρχει η συνάρτηση-μέλος *flush()* που γράφει τα περιεχόμενα του `buffer` στο αρχείο, αμέσως μετά την κλήση της
- Κατά το κλείσιμο του αρχείου, γίνεται και γράψιμο του `buffer` στο αρχείο

ofstream & μετακίνηση στο αρχείο

- Μαζί με κάθε output stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *put pointer*
- Με την συνάρτηση-μέλος *tellp* μαθαίνουμε για το που βρίσκεται ο put pointer
- Με την συνάρτηση-μέλος *seekp* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

Παράδειγμα fstream

```
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{
    fstream file("data3.txt", ios::in | ios::out);
    file << 42 << "koko" << endl;
    file.flush();
    int i;
    file.seekg(0, ios::beg);
    file >> i;
    cout << i << endl;
}
```

Παράδειγμα `fstream`

```
$ touch data3.txt
```

```
$ ./a.out
```

```
42
```

Παράδειγμα `fstream`

- Το προηγούμενο παράδειγμα δεν θα δουλέψει αν το αρχείο `data3.txt` δεν υπάρχει ήδη
- Για να γίνει αυτό, πρέπει πρώτα να ανοίξουμε το αρχείο για γράψιμο ώστε να δημιουργηθεί και στη συνέχεια να το ανοίξουμε για διάβασμα και γράψιμο
- Περισσότερα [εδώ](#)

Παράδειγμα fstream

```
#include <iostream>

#include <fstream>

using namespace std;

int main(void) {
    fstream temp("data3.txt", ios::out); temp.close();
    fstream file("data3.txt", ios::in | ios::out);
    file << 42 << "koko" << endl;
    file.flush();
    int i;
    file.seekg(0, ios::beg);
    file >> i;
    cout << i << endl;
}
```

IO & binary files

- Με τα `ifstream` και `ofstream` μπορούμε να χειριστούμε και δυαδικά αρχεία (αρχεία που έχουν bytes και όχι κείμενο)
- Η συνάρτηση-μέλος *read* του `ifstream`, διαβάζει από το αρχείο ένα πλήθος από bytes και το αποθηκεύει σε έναν πίνακα χαρακτήρων
- Η συνάρτηση-μέλος *write* του `ofstream`, γράφει στο αρχείο ένα πλήθος από bytes, που τα διαβάζει από έναν πίνακα χαρακτήρων

Reference – Χρήσιμοι Σύνδεσμοι

- cplusplus.com
- `ifstream`
- `ofstream`
- `fstream`
- `getline` για string