



ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ(Θ)

Ενότητα : ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΔΙΔΑΣΚΩΝ: ΠΑΡΙΣ ΜΑΣΤΟΡΟΚΩΣΤΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



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 παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- 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.



Ενότητα

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΔΙΔΑΣΚΩΝ: ΠΑΡΙΣ ΜΑΣΤΟΡΟΚΩΣΤΑΣ

Περιεχόμενα ενότητας

1. Η ΚΛΑΣΗ string
2. ΜΕΘΟΔΟΙ ΣΤΗ ΚΛΑΣΗ string
3. ΤΕΛΕΣΤΕΣ ΣΤΗ ΚΛΑΣΗ string
4. ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ ΤΕΛΕΣΤΕΣ ΣΤΗ ΚΛΑΣΗ string
5. ΕΙΣΟΔΟΣ / ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ
6. ΚΛΑΣΕΙΣ ΓΙΑ ΕΙΣΟΔΟ / ΕΞΟΔΟ ΑΠΟ ΑΡΧΕΙΑ
7. ΕΛΕΓΧΟΙ ΣΕ ΑΡΧΕΙΑ
8. ΕΞΟΔΟΣ ΣΕ ΑΡΧΕΙΟ
9. ΕΙΣΟΔΟΣ ΑΠΟ ΑΡΧΕΙΟ
10. ΔΕΙΚΤΕΣ ΡΟΗΣ get, put
11. ΜΕΘΟΔΟΙ ΣΕ ΡΟΕΣ
ΕΙΣΟΔΟΥ / ΕΞΟΔΟΥ
12. ΜΕΘΟΔΟΙ ΓΙΑ ΕΙΣΟΔΟ/ΕΞΟΔΟ ΑΠΟ BINARY ΑΡΧΕΙΟ

Σκοποί ενότητας

Η ΚΛΑΣΗ string

```
#include <string>
using namespace std;

void main() {
    string first_name = "Bjarne";
    string last_name;
    last_name = "Stroustrup";
    string names = first_name + " " + last_name;
    cout << names << endl;
    names = last_name + ", " + first_name;
    cout << names << endl;
}
```

ΜΕΘΟΔΟΙ ΣΤΗ ΚΛΑΣΗ string

len, size

```
string str = "Hello";  
string::size_type len;  
len = str.length(); // len == 5  
len = str.size(); // len == 5
```

c_str // Μετατροπή σε συμβολοσειρά της C

```
string filename;  
cout << "Enter file name: ";  
cin >> filename;  
ofstream outfile (filename.c_str());  
outfile << "Data" << endl;
```

insert

```
string str11 = "abcdefghi";  
string str12 = "0123";  
str11.insert (3,str12);  
cout << str11 << endl; // "abc0123defghi"  
str12.insert (1,"XYZ");  
cout << str12 << endl; // "0XYZ123"
```


ΜΕΘΟΔΟΙ ΣΤΗ ΚΛΑΣΗ string

erase

```
string str13 = "abcdefghi";  
str12.erase (5,3);  
cout << str12 << endl; // "abcdei"
```

replace

```
string str14 = "abcdefghi";  
string str15 = "XYZ";  
str14.replace (4,2,str15);  
cout << str14 << endl; // "abcdXYZghi"
```

find, rfind (find βρίσκει την πρώτη θέση, rfind βρίσκει την τελευταία

```
string str16 = "abcdefghi";  
string str17 = "def";  
string::size_type pos = str16.find (str17,0);  
cout << pos << endl; // 3  
pos = str16.find ("AB",0);  
if (pos == string::npos) // string::npos - ειδικός χαρακτήρας σε περίπτωση  
                        // αποτυχίας της μεθόδου  
    cout << "Not found" << endl;
```

substr

```
string str18 = "abcdefghi";  
string str19 = str18.substr (6,2);  
cout << str19 << endl; // "gh"
```

ΤΕΛΕΣΤΕΣ ΣΤΗ ΚΛΑΣΗ string

Οι τελεστές που χρησιμοποιούνται σε αντικείμενα της κλάσης string είναι:

=, +, +=, !=, <, >, <=, >=, <<, >>

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ ΤΕΛΕΣΤΕΣ ΣΤΗ ΚΛΑΣΗ string

```
string str1 = "Hello ";  
string str2 = "there";  
string str3 = str1 + str2; // "Hello there"
```

```
string str1 = "Hello ";  
str1 += "there"; // Hello there
```

```
str1 < str2;
```

```
string str1 = "Hello there";  
cout << str1 << endl;
```

```
string str1;  
cin >> str1;
```

ΕΙΣΟΔΟΣ / ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

- Όταν θέλουμε να διαβάσουμε / αποθηκεύσουμε / τυπώσουμε από/προς `stdin` / `stdout` ή σε αρχεία οι βιβλιοθήκες που πρέπει να εισάγουμε είναι οι :
- `#include<iostream>`
 - Βιβλιοθήκη για είσοδο / έξοδο.
- `#include<iomanip>`
 - Βιβλιοθήκη για C++ manipulators. Γενικά χρήσιμη βιβλιοθήκη.
- `#include<fstream>`
 - Βιβλιοθήκη για είσοδο / έξοδο με χρήση αρχείων.

ΚΛΑΣΕΙΣ ΓΙΑ ΕΙΣΟΔΟ / ΕΞΟΔΟ ΑΠΟ ΑΡΧΕΙΑ

- `ofstream`: Κλάση για έξοδο δεδομένων σε αρχείο
- `ifstream`: Κλάση για διάβασμα από αρχείο
- `fstream`: Κλάση για διάβασμα και έξοδο δεδομένων σε αρχείο
- Για να ανοίξουμε ένα αρχείο (για είσοδο ή έξοδο) χρησιμοποιούμε την εντολή

```
void open (const char * filename, openmode1 | openmode2 | .. openmodek);
```

openmode: `ios::in`, `ios::out`, `ios::ate`, `ios::trunc`, `ios::app`, `ios::binary`

Παράδειγμα:

```
ofstream file ("example.bin", ios::out | ios::app | ios::binary);
```

ΕΛΕΓΧΟΙ ΣΕ ΑΡΧΕΙΑ

- Μπορούμε να ελέγξουμε εάν ένα αρχείο έχει ανοίξει σωστά με την μέθοδο `is_open()`
- Κλείνουμε ένα αρχείο με τη μέθοδο `close()`
- Όταν έχουμε αρχεία κειμένου (`text files`) τότε μπορούμε να χρησιμοποιούμε τις ίδιες μεθόδους με τα `cin`, `cout` όπως και όταν έχουμε είσοδο/έξοδο από το πληκτρολόγιο (είσοδο), ή την οθόνη (έξοδο).

ΕΞΟΔΟΣ ΣΕ ΑΡΧΕΙΟ

```
// Έξοδος δεδομένων σε αρχείο
#include <fstream.h>

int main () {
    ofstream examplefile ("example.txt");

    if (examplefile.is_open()) {
        examplefile << "This is a line.\n";
        examplefile << "This is another line.\n";
        examplefile.close();
    }
    return 0;
}
```

Μιας και δεν δίνουμε openmode, το openmode εδώ είναι από default ios::out | ios::trunc

Το αρχείο `example.txt` θα περιέχει:

This is a line.
This is another line.

ΕΙΣΟΔΟΣ ΑΠΟ ΑΡΧΕΙΟ

// Είσοδος από αρχείο κειμένου

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>

int main () {
    char buffer[256];
    ifstream examplefile ("example.txt");

    if (! examplefile.is_open()) {
        cout << "Error opening file";
        exit (1);
    }

    while (! examplefile.eof() ) {
        examplefile.getline (buffer,100);
        cout << buffer << endl;
    }
    return 0;
}
```

Στην οθόνη τυπώνεται:

**This is a line.
This is another line.**

ΔΕΙΚΤΕΣ ΡΟΗΣ `get`, `put`

- Ο δείκτης ροής είναι το αποτέλεσμα της κλήσης των constructors `istream`, `ifstream`, `ostream`, `ofstream`.
- Όταν μιλάμε για είσοδο τότε ο δείκτης ονομάζεται `get input stream`, ενώ όταν έχουμε να κάνουμε με έξοδο ονομάζεται `put input stream`.
- Υπάρχουν μερικές συναρτήσεις βιβλιοθήκης που εφαρμόζονται συχνά στις δύο αυτές ροές (εισόδου, εξόδου)

ΜΕΘΟΔΟΙ ΣΕ ΡΟΕΣ ΕΙΣΟΔΟΥ / ΕΞΟΔΟΥ

- `tellg()` (για `get` δείκτη) και `tellp()` (για `put` δείκτη)
Επιστρέφουν την θέση που βρισκόμαστε στο αρχείο

`seekg()` and `seekp()` (Κατάλληλες για χρήση με αρχεία κειμένου)
Επιτρέπουν την αλλαγή της θέσης που βρισκόμαστε στο αρχείο

- `seekg (pos_type_position);`
`seekp (pos_type_position);`
Αλλάζουν τη θέση που βρισκόμαστε στο αρχείο κατά σταθερό βήμα ίσο με την τιμή `pos_type_position` και σε σχέση με την αρχή του αρχείου
- `seekg (off_type_offset, seekdir_direction);`
`seekp (off_type_offset, seekdir_direction);`
Αλλάζουν τη θέση που βρισκόμαστε στο αρχείο ίσο με την τιμή `offset` και σε σχέση με το σημείο που περιγράφει η παράμετρος `seekdir_direction`
Οι τιμές της παραμέτρου `seekdir_direction` μπορεί να είναι:
`ios::beg` (αρχή του αρχείου)
`ios::cur` (τρέχων σημείο)
`ios::end` (τέλος του αρχείου)

ΠΑΡΑΔΕΙΓΜΑ

// Υπολογισμός μεγέθους binary αρχείου

```
#include <iostream.h>
#include <fstream.h>
const char * filename = "example.txt";

int main () {
    long l, m;
    ifstream file (filename, ios::in | ios::binary);

    l = file.tellg();
    file.seekg (0, ios::end);
    m = file.tellg();
    file.close();
    cout << "size of " << filename; cout << " is " << (m-l) << " bytes.\n";
    return 0;
}
```

Στην οθόνη τυπώνεται:

size of example.txt is 40 bytes.

ΜΕΘΟΔΟΙ ΓΙΑ ΕΙΣΟΔΟ/ΕΞΟΔΟ ΑΠΟ BINARY ΑΡΧΕΙΟ

- `write (char * buffer, streamsize size);`
Γράφει από τη συμβολοσειρά `buffer` τόσα bytes όσα ορίζονται από τη παράμετρο `size`
- `read (char * buffer, streamsize size);`
Διαβάζει από το αρχείο τόσα bytes όσα ορίζονται από τη παράμετρο `size` και αποθέτει στη συμβολοσειρά `buffer`

ΠΑΡΑΔΕΙΓΜΑ

```
// Είσοδος από binary αρχείο
#include <iostream.h>
#include <fstream.h>

const char * filename = "example.txt";

int main () {
    char * buffer;
    long size;
    ifstream file (filename, ios::in | ios::binary | ios::ate);

    size = file.tellg();
    file.seekg (0, ios::beg);
    buffer = new char [size];
    file.read (buffer, size);
    file.close();
    cout << "the complete file is in a buffer";
    delete [] buffer;
    return 0;
}
```

Στην οθόνη τυπώνεται:

the complete file is in a buffer

Τέλος Ενότητας

