String Handling

ΓΛΩΣΣΙΚΉ ΤΕΧΝΟΛΟΓΊΑ

Tokenization

- Tokenization είναι η διαδικασία διαχωρισμού τμημάτων ενός string από χαρακτήρες εισόδου.
- Το tokenization συνήθως εφαρμόζεται για την μορφοποίηση της εισόδου ενός άλλου τμήματος επεξεργασίας.
- Στην επεξεργασία κειμένου χωρίζουμε το κείμενο σε λέξεις για να μορφοποιήσουμε την είσοδο του μορφοσυντακτικού αναλυτή.

Παράδειγμα tokenization

- Είσοδος: “Uranium has atomic number 92.”
  Tokens:
  {“Uranium”, “has”, “atomic”, “number”, “92”, “.”}
- Είσοδος: “In nature, uranium atoms exist as uranium-238.”
  Tokens:
  {“In”, “nature”, “,”, “uranium”, “atoms”, “exist”, “as”, “uranium-238”, “.”}
- “Uranium-238 has a half-life of 4.51*10^9 years.”
  Tokens:
  {“Uranium-238”, “has”, “a”, “half-life”, “of”, “4.51*10^9”, “years”, “.”}

.NET string handling

- Τύπος αλφαριθμητικού: built-in type string
- Κυριολεκτικά με διπλά εισαγωγικά:
  string hi = “Hello World”
- Κάθε string είναι ένα σύνολο χαρακτήρων
- Ο τύπος χαρακτήρα είναι char
- Κυριολεκτικό χαρακτήρα με μονά εισαγωγικά:
  char c = ‘h’;
- Χαρακτήρας διαφυγής: \
  πχ \n, \t, \\ κλπ.

Operators

- Σύγκριση:
  - == και != ορίζονται κανονικά για ισότητα και ανισότητα
- + τελεστής για συγκόλληση strings.
  πχ string hi = “Hello ” + “World”;
- []: επιστρέφει τον χαρακτήρα σε ένα string.
  πχ hi[o]==‘H’

Searching in strings

- String Members:
  - StartsWith(string) – Ελέγχει εάν το string αρχίζει με το string που δίνεται ως παράμετρος.
    ```
    string hi = "Hello World";
    if(hi.StartsWith("Hello"))
                Console.WriteLine("Hello");
    ```
  - EndsWith(string) – Ελέγχει εάν το string τελειώνει με το string που δίνεται ως παράμετρος.
    ```
    string hi = "Hello World";
    if(hi.EndsWith("World"))
                Console.WriteLine("The whole world?");
    ```
  - Contains(string) – Ελέγχει εάν το string περιέχει το string που δίνεται ως παράμετρος.
    ```
    string hi = "Hello Vivi";
    if(hi.EndsWith("Vivi"))
                Console.WriteLine("Are you talking to me?");
    ```

Searching in strings - Indices

- IndexOf(string) – Εντοπίζει στο string το string που δίνεται ως παράμετρος και επιστρέφει το index που βρέθηκε.
- IndexOfAny(char[]) – Εντοπίζει στο string οποιονδήποτε από τους χαρακτήρες που δίνονται ως παράμετροι και επιστρέφει το index του πρώτου που θα βρει.
- LastIndexOf(string) – Εντοπίζει την τελευταία εμφάνιση του string που δίνεται ως παράμετρος και επιστρέφει το index.

Searching in strings - Indices

```
static void SearchExample()
{
    string path = "C:\\test\\temp\\t.txt";
    Console.WriteLine(path.IndexOf('\\'));
    //prints 2
    char[] chars = { 't', 'e' };
    Console.WriteLine(path.IndexOfAny(chars));
    //prints 3
    Console.WriteLine(path.LastIndexOf('\\'));
    //prints 12

}
```

Formatting strings

- Join(string, string[]) (static)– Παρεμβάλλει έναν string seperator μεταξύ των στοιχείων ενός string array, δίνοντας ένα τελικό string
- ToUpper(string) & ToLower(string) – Επιστρέφει το string με κεφαλαίους ή πεζούς χαρακτήρες αντίστοιχα.
- Trim() & TrimEnd() & TrimStart() – Αφαιρούν τα κενά από τις άκρες, το τέλος και την αρχή του string αντίστοιχα.
- Replace(string, string) – Αντικαθιστά στο string τις εμφανίσεις του πρώτου ορίσματος με το δεύτερο.
- Insert(int, string) – Εισάγει στο index που δίνεται ως είσοδος το string που δίνεται ως παράμετρος.

Formatting strings

```
static void FormattingExample()
{
    string[] path_parts = { "C:", "test", "temp", "t.txt" };

    string path = string.Join("\\",path_parts);
    Console.WriteLine(path);
    //prints C:\test\temp\t.txt

    Console.WriteLine("Hello".ToUpper());
    //prints HELLO

    Console.WriteLine("Hello".ToLower());
    //prints hello

    Console.WriteLine("S:" + "       Hi       ".Trim() + ":E");
    //prints S:Hi:E

    Console.WriteLine(path.Replace('\\', '/'));
    //prints C:/test/temp/t.txt

    string alt_path = path.Insert(path.IndexOf("\\"), "\\another_step");
    Console.WriteLine(alt_path);
    //prints C:\another_step\test\temp\t.txt

}
```

Splitting strings

- Split(char[]) – Χωρίζει το string με βάση τα delimiters που δέχεται ως είσοδο. Τα delimiters δεν συμπεριλαμβάνονται στα tokens!
- Substring(index) & Substring(index, length)- Επιστρέφει το sustring από το index που προσδιορίζεται μέχρι το τέλος ή στην δεύτερη περίπτωση από το index και για όσους χαρακτήρες προσδιορίζει το length.

Splitting strings

```
static void SplitExample()
{
    char[] delimiters = { ' ', '\t', '\n', '.', ',' };
    string sentence = "Uranium has atomic number 92.";
    string[] tokens = sentence.Split(delimiters);

    foreach (string token in tokens)
    {
        Console.WriteLine(token);
    }
    /*prints
     * Uranium
     * has
     * atomic
     * number
     * 92
    */

    string path = "C:\\test\\temp\\t.txt";
    string filename = path.Substring(path.LastIndexOf('\\') + 1);
    Console.WriteLine(filename);
    //prints t.txt

    string root = path.Substring(0, path.IndexOf('\\'));
    Console.WriteLine(root);
    //prints C:

}
```

Input from strings

- StringReader-Κλάση που υλοποιεί ανάγνωση από string χρησιμοποιώντας streams.

```
static void StreamInputExample()
{
    string input = "Uranium has atomic number 92.\r\nIn nature, uranium atoms" +
                   "exist as uranium-238.\r\nUranium-238 has a half-life of 4.51*10^9 years.\r\n";
    StringReader sr = new StringReader(input);

    int index = 0;
    string line;
    while ((line = sr.ReadLine()) != null)
    {
        Console.WriteLine(index + ":" + line);
        index++;
    }
    /*prints
     0:Uranium has atomic number 92.
     1:In nature, uranium atoms exist as uranium-238.
     2:Uranium-238 has a half-life of 4.51*10^9 years.
    */
    sr.Close();

}
```

Output to strings

- StringBuilder – Κλάση δημιουργίας string. Ο γρηγορότερος τρόπος για συνένωση strings.

```
static void StringOutputExample()
{
    StringBuilder sb = new StringBuilder();

    sb.AppendLine("Uranium has atomic number 92.");
    sb.AppendLine("In nature, uranium atoms exist as uranium-238.");
    sb.AppendLine("Uranium-238 has a half-life of 4.51*10^9 years.");

    string output = sb.ToString();
    Console.Write(output);
    /*
     Uranium has atomic number 92.
     In nature, uranium atoms exist as uranium-238.
     Uranium-238 has a half-life of 4.51*10^9 years.
    */
}
```

PYTHON

Strings - Δήλωση

- Μπορείτε να γράψετε τα δικά σας string περικλείοντας απλά χαρακτήρες και αριθμούς μέσα σε μονά ('...') ή διπλά("...") αυτάκια.

```
>>> 'Hello, World!!'
'Hello, World!!'
>>> "Python is a programming language."
'Python is a programming language.'
>>>
```

Strings - Τελεστές

- Μπορείτε να κάνετε χρήση των τελεστών “+” και “*” πάνω σε strings.
  - “+” (concatenation)
  - “*” (repeatition)

```
>>> a = "Hello, World!!"
>>> b = "Python is a programming language."
>>> a + b
'Hello, World!!Python is a programming language.'
>>> a*3
'Hello, World!!Hello, World!!Hello, World!!'
>>> b + a*3 + b
'Python is a programming language.Hello, World!!Hello, World!!Hello, World!!Pyth
on is a programming language.'
>>>
```

Strings – Δεικτες I

- Μπορείτε να φανταστείτε τα string σαν πίνακες χαρακτήρων της C. Ο πρώτος χαρακτήρας έχει δείκτη ο.

Strings – Δεικτες II

- Μπορείτε ακόμη να χρησιμοποιήσετε και αρνητικούς δείκτες.

Strings – Δεικτες III

- Γενικά η δεικτοδότηση στα strings έχει ως εξής:

```
>>> a = "hello"
>>> for i in range(len(a)):
...     print i, a[i], "\t", -i, a[-i]
...
0 h     0 h
1 e     -1 o
2 l     -2 l
3 l     -3 l
4 o     -4 e
>>>
```

Συναρτήσεις Αναζήτησης I

- str.count(sub[,start[,end]])
  - Επιστρέφει τις μη επικαλυπτόμενες εμφανίσεις του substring sub στο διάστημα (προαιρετικά) [start, end].
- str.find(sub[,start[,end]])
  - Επιστρέφει τον μικρότερο δείκτη του string όπου το substring sub εμφανίζεται (προαιρετικά) στο διάστημα [start, end], διαφορετικά -1.
- str.index(sub[,start[,end]])
  - Όμοια με την find αλλά "πετάει" ValueError όταν δεν βρεθεί το substring sub.

Συναρτήσεις Αναζήτησης II

```
>>> a = "abababababa"
>>> a.count("bab")
2
>>> a.find("bab")
1
>>> a.find("bab",4)
5
>>> a.find("c")
-1
>>> a.index("bab",4)
5
>>> a.index("c")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: substring not found
>>>
```

Συναρτήσεις Ταυτοποίησης I

- str.startswith(prefix[,start[,end]])
  - Επιστρέφει boolean τιμή, ανάλογα με το αν το string ξεκινάει με το πρόθεμα prefix (προαιρετικά) στο διάστημα [start, end].
- str.endswith(suffix[,start[,end]])
  - Επιστρέφει boolean τιμή, ανάλογα με το αν το string τελειώνει με το επίθεμα suffix (προαιρετικά) στο διάστημα [start, end].
- Και στις δύο περιπτώσεις, τα prefix/suffix μπορεί να είναι tuple από προθέματα/επιθέματα, αντίστοιχα.

Συναρτήσεις Ταυτοποίησης II

```
>>> a = "abababababa"
>>> a.startswith("c")
False
>>> a.startswith("ab",4,6)
True
>>> a.endswith(("ab","c"))
False
>>> a.endswith(("ab","ba"),0,len(a))
True
>>>
```

Συναρτήσεις Ταυτοποίησης III

- str.isalnum()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο αλφαριθμητικούς χαρακτήρες και δεν είναι το κενό string.
- str.isalpha()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο αλφαβητικούς χαρακτήρες και δεν είναι το κενό string.
- str.isdigit()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο αριθμητικούς χαρακτήρες και δεν είναι το κενό string.

Συναρτήσεις Ταυτοποίησης IV

- str.isspace()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο κενούς χαρακτήρες και δεν είναι το κενό string.
- str.islower()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο lowercase χαρακτήρες και δεν είναι το κενό string.
- str.isupper()
  - Επιστρέφει boolean ανάλογα με το αν το string περιέχει μόνο uppercase χαρακτήρες και δεν είναι το κενό string.

Συναρτήσεις Ταυτοποίησης V

```
>>> a = "abababababa"
>>> a.isalnum()
True
>>> a.isalpha()
True
>>> a.isdigit()
False
>>> a.isspace()
False
>>> a.islower()
True
>>> a.isupper()
False
>>> a.istitle()
False
>>>
```

Συναρτήσεις Μορφοποίησης I

- str.lower()
  - Επιστρέφει ένα αντίγραφο του string με όλα του τα στοιχεία μικρά.
- str.upper()
  - Επιστρέφει ένα αντίγραφο του string με όλα του τα στοιχεία κεφαλαία.
- str.capitalize()
  - Επιστρέφει ένα αντίγραφο του string με το πρώτο του στοιχείο κεφαλαίο.
- str.title()
  - Επιστρέφει ένα αντίγραφο του string με όλα τα πρώτα γράμματα κάθε λέξης κεφαλαία.

Συναρτήσεις Μορφοποίησης II

```
>>> a = "abababababa"
>>> a.upper()
'ABABABABABA'
>>> a.title()
'Abababababa'
>>> a.lower()
'abababababa'
>>> a
'abababababa'
>>> b = a.capitalize()
>>> a
'abababababa'
>>> b
'Abababababa'
>>>
```

Συναρτήσεις Αντικατάστασης

- str.replace(old,new[,count])
  - Επιστρέφει ένα αντίγραφο του string όπου το substring old έχει αντικατασταθεί από το substring new. Αν δοθεί το προαιρετικό όρισμα count, πραγματοποιούνται μόνο οι πρώτες count αλλαγές.

```
>>> a = "Hello, World!!"
>>> b = a.replace("World","girl")
>>> c = b.replace("Hello,","OK bye..")
>>> a
'Hello, World!!'
>>> b
'Hello, girl!!'
>>> c
'OK bye.. girl!!'
>>>
```

Συναρτήσεις Διαχωρισμού I

- str.split([sep[,maxsplit]])
  - Επιστρέφει μία λίστα από λέξεις του string χρησιμοποιώντας το sep ως delimeter. Αν δοθεί το προαιρετικό όρισμα maxsplit, τότε μόνο maxsplit "κοψίματα" πραγματοποιούνται.
  - Έχει σημασία αν δοθεί ή όχι το όρισμα sep καθώς σε κάθε περίπτωση εκτελείται διαφορετικός αλγόριθμος.
  - Αν δοθεί (μπορεί να είναι και παραπάνω από ένας χαρακτήρας), χρησιμοποιείται ως έχει, με την ιδιότητα ότι οι delimeters δεν ομαδοποιούνται.
  - Αν όχι, χρησιμοποιείται ως separator το κενό, και με την ιδιότητα συνεχόμενα κενά μέσα στο string να λαμβάνονται ως ένα.

Συναρτήσεις Διαχωρισμού II

```
>>> a = '  1$$2$3 '
>>> a
'  1$$2$3 '
>>> a.split("$")
['  1', '', '2', '3 ']
>>> a.split(" ")
['', '', '1$$2$3', '']
>>> a.split()
['1$$2$3']
>>> b = ""
>>> b
''
>>> b.split(" ")
['']
>>> b.split()
[]
>>>
```

Συναρτήσεις Συνένωσης

- str.join(seq)
  - Επιστρέφει ένα string το οποίο είναι η συνένωση των strings της ακολουθίας seq. Όπου seq, μπορεί να είναι string, tuple, list, κτλ

```
>>> ls = ["Two", "beer", "or", "not", "two", "beer", "?"]
>>> " ".join(ls)
'Two beer or not two beer ?'
>>>
```

Module StringIO

- import StringIO
  - Με το παραπάνω module μπορεί κανείς να μεταχειριστεί τα strings σαν αρχεία, καθώς δύνεται δυνατότητα για ανάγνωση/εγγραφή ενός string buffer.

```
>>> output = StringIO.StringIO()
>>> output.write("First line.\n")
>>> print >>output,"Second line."
>>> contents = output.getvalue()
>>> output.close()
>>> contents
'First line.\nSecond line.\n'
>>>
```