

Συμβολοσειρές – Strings

Είδαμε τον τύπο `char` που απεικονίζει ένα χαρακτήρα, τώρα θα δούμε τις **συμβολοσειρές** (*Strings*) που απεικονίζουν ακολουθίες χαρακτήρων. Οι συμβολοσειρές είναι αντικείμενα της κλάσης *java.lang.String* που χρησιμοποιούνται για την αποθήκευση και απεικόνιση αλφαριθμητικών χαρακτήρων. Μία συμβολοσειρά αποτελεί ένα ξεχωριστό αντικείμενο που αρχικοποιείται με τους παρακάτω τρόπους (διαφορετικοί δομητές):

- 1) **`String MyString = new String();`** Κενή συμβολοσειρά (`MyString`).
- 2) **`String HisName = new String("Nikos Nikas");`** Περιέχει το όνομα `Nikos Nikas`.
- 3) **`String MyString = new String(charArray);`** Περιέχει τους χαρακτήρες του πίνακα `charArray`. Υπάρχει επίσης και ο δομητής για `byteArray`.
- 4) **`String MyString = new String(charArray, index, numb);`** Περιέχει `numb` χαρακτήρες του πίνακα `charArray`, αρχίζοντας από την θέση `index`. Υπάρχει ακριβώς ο ίδιος δομητής και για `byteArray`.
- 5) με **απευθείας ανάθεση**, π.χ. **`String MyName = "Nikos Nikas";`**

Με έμμεσο τρόπο μια συμβολοσειρά μπορεί να δημιουργηθεί:

α) από μια άλλη ήδη δημιουργημένη συμβολοσειρά. Π.χ.

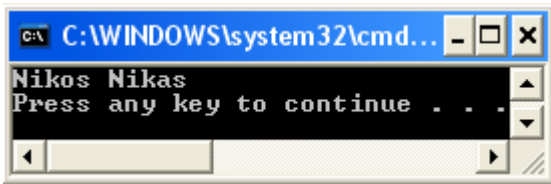
```
String MyToy = new String("Trainaki");  
String HisToy = new String(MyToy);
```

β) με τη συνένωση (*concatenation*) δύο ή περισσότερων ήδη δημιουργημένων συμβολοσειρών. Π.χ.

```
String s1 = "Hello" + " " + "Java";
```

Όπως είδαμε παραπάνω (περίπτωση 3), μια συμβολοσειρά είναι ισοδύναμη με ένα πίνακα χαρακτήρων (πίνακα τύπου `char`). Έτσι το `String MyString` είναι ισοδύναμο με τον πίνακα τύπου `char[]`. Στο παρακάτω παράδειγμα δημιουργείται μία συμβολοσειρά (`NameString`) από ένα πίνακα χαρακτήρων:

```
class TestString {  
public static void main(String[] args) {  
  
char[] NameArray = {'N','i','k','o','s',' ','N','i','k','a','s'};  
String NameString = new String(NameArray);  
  
System.out.println(NameString);  
}  
}
```



Οι συμβολοσειρές είναι αντικείμενα **αμετάβλητα** (*immutable*), δηλαδή δεν μπορούν να αλλάξουν άμεσα, αλλά μόνο έμμεσα με τη δημιουργία ενός νέου αντικειμένου. Επομένως δύο συμβολοσειρές αποτελούν διαφορετικά αμετάβλητα αντικείμενα, ακόμη και αν έχουν το ίδιο περιεχόμενο. Αυτό σημαίνει ότι αν συγκρίνουμε δύο συμβολοσειρές με τον τελεστή ισότητας **==**, ακόμη και αν έχουν το ίδιο περιεχόμενο, θα πάρουμε σαν αποτέλεσμα **false**. Έτσι, όταν θέλουμε να συγκρίνουμε δύο συμβολοσειρές χρησιμοποιούμε την μέθοδο **equals** (δες μεθόδους παρακάτω) και όχι τον τελεστή **==**. Π.χ. `if (password.equals("abcdFGE"))//prosoxi sta kafalaia – mikra grammata`

Παράδειγμα:

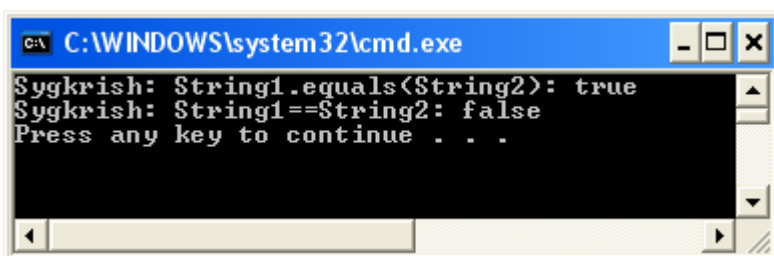
```
class TestString1 {
    public static void main(String[] args) {

        String NameString1 = "Nikos Nikas";
        String NameString2 = new String(NameString1);

        if (NameString1.equals(NameString2))
            System.out.println("Sygkrish: String1.equals(String2) " +
                (NameString1.equals(NameString2)));

        if (NameString1==NameString2)
            System.out.println("Sygkrish: String1==String2 " + (NameString1==NameString2));
        else
            System.out.println("Sygkrish: String1==String2 " + (NameString1==NameString2));
    }
}
```

Το αποτέλεσμα των συγκρίσεων:



Οι κυριότερες μέθοδοι χειρισμού των συμβολοσειρών

Δες αναλυτικά στην τεκμηρίωση της Sun.API

<http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html>

compareTo (<i>String anotherString</i>)	Συγκρίνει 'λεξικογραφικά' δύο συμβολοσειρές, δηλαδή κάθε Unicode χαρακτήρα ξεχωριστά. Επιστρέφει τιμή α) < 0, αν η συμβολοσειρά έχει μικρότερο μήκος από το anotherString, β) =0, αν έχουν το ίδιο μήκος, και γ) >0, αν η συμβολοσειρά έχει μεγαλύτερο μήκος από το anotherString
charAt (<i>int index</i>)	Επιστρέφει τον χαρακτήρα στη συγκεκριμένη θέση (<i>index</i>).
getChars (<i>int srcBegin, int srcEnd, char[] dst, int dstBegin</i>)	Αντιγράφει χαρακτήρες από μια συμβολοσειρά σε ένα πίνακα χαρακτήρων (από -έως).
length ()	Επιστρέφει τον αριθμό των χαρακτήρων της συμβολοσειράς (μετρά και το κενό διάστημα). Π.χ. <code>str1.length()</code> .
equals (<i>Object anObject</i>)	Συγκρίνει τη συμβολοσειρά (ή αντικείμενο) με αυτή που προσδιορίζεται από το <i>anObject</i> . Π.χ. <code>s2.equals(s1)</code> . Επιστρέφει <code>true</code> (ίδιοι χαρακτ., ίδιο μήκος), ή <code>false</code> .
equalsIgnoreCase (<i>String anotherString</i>)	Συγκρίνει τη συμβολοσειρά με μία άλλη χωρίς να λαμβάνεται υπόψη η περίπτωση κεφαλαίων-μικρών.
toUpperCase ()	Μετατρέπει τα γράμματα της συμβολοσειράς σε κεφαλαία, ακολουθώντας τους τοπικούς κανόνες. Π.χ. <code>s2= s1.toUpperCase()</code> .
toLowerCase ()	Μετατρέπει τα γράμματα της συμβολοσειράς σε μικρά, ακολουθώντας τους τοπικούς κανόνες. Π.χ. <code>s2= s1.toLowerCase()</code> .
concat (<i>String str</i>)	Συνενώνει δύο συμβολοσειρές.
indexOf (<i>int ch</i>)	Επιστρέφει την πρώτη θέση που βρίσκεται ο χαρακτήρας (<i>ch</i>) μέσα στη συμβολοσειρά, αλλιώς -1.
indexOf (<i>int ch, int fromIndex</i>)	Όπως και στην <code>indexOf(int ch)</code> , αλλά από τη θέση <i>fromIndex</i> και μετά.
indexOf (<i>String str</i>)	Επιστρέφει τη θέση ενός <code>String</code> μέσα σε ένα άλλο

	(str). Π.χ. <code>int x = s1.indexOf(s2)</code> ; σημαίνει αναζήτηση της συμβολοσειράς <code>s1</code> στην <code>s2</code> , όπου <code>x=η</code> θέση της <code>s1</code> μέσα στην <code>s2</code> .
indexOf (String str, int fromIndex)	Όπως και στην <code>indexOf(String str)</code> , αλλά από τη θέση <code>fromIndex</code> και μετά.
lastIndexOf (int ch)	Επιστρέφει την τελευταία θέση που βρίσκεται ο χαρακτήρας (<code>ch</code>), αλλιώς <code>-1</code> .
lastIndexOf (int ch, int fromIndex)	Όπως και στην <code>lastIndexOf(int ch)</code> , αλλά μέχρι τη θέση <code>fromIndex</code> .
lastIndexOf (String str)	Όπως και στην <code>lastIndexOf(int ch)</code> , αλλά με παράμετρο <code>String</code> .
lastIndexOf (String str, int fromIndex)	Όπως και στην <code>lastIndexOf(int ch, int fromIndex)</code> , αλλά με παράμετρο <code>String</code> .
startsWith (String prefix)	Επιστρέφει <code>true</code> αν η συμβολοσειρά αρχίζει με τους συγκεκριμένους χαρακτήρες, διαφορετικά επιστρέφει <code>false</code> .
endsWith (String prefix)	Επιστρέφει <code>true</code> αν η συμβολοσειρά τελειώνει με τους συγκεκριμένους χαρακτήρες, διαφορετικά επιστρέφει <code>false</code> .
substring (int beginIndex)	Επιστρέφει μια νέα συμβολοσειρά από μία άλλη από την θέση <code>beginIndex</code> και μετά.
substring (int beginIndex, int endIndex)	Επιστρέφει μια νέα συμβολοσειρά από μία άλλη, αρχίζοντας από την θέση <code>beginIndex</code> μέχρι την θέση <code>endIndex</code> .
regionMatches (int offset1, String str2, int offset2, int count)	Επιστρέφει <code>true</code> αν 'μια περιοχή' της τρέχουσας συμβολοσειράς 'ταιριάζει' με 'μια περιοχή' μιας άλλης συμβολοσειράς. Όπου: <code>offset1</code> = η θέση του 1 ^{ου} χαρακ. της τρέχουσας συμβολοσειράς που θα συγκριθεί. <code>str2</code> = η άλλη συμβολοσειρά με την οποία θα συγκριθεί η τρέχουσα. <code>offset2</code> = η θέση του 1 ^{ου} χαρακ. της άλλης συμβολοσειράς που θα συγκριθεί με την τρέχουσα. <code>count</code> = ο αρ. των χαρακ. της 'περιοχής' που θα συγκριθεί.
regionMatches (Boolean ignoreCase, int offset1, String str2, int offset2, int count)	Όπως και η προηγούμενη, αλλά με τη διαφορά ότι δεν λαμβάνεται υπόψη ο τύπος των χαρακτήρων (κεφα-

	λαία – μικρά).
replace (<i>char oldChar, char newChar</i>)	Επιστρέφει μια νέα συμβολοσειρά με αλλαγμένους όλους τους χαρακτήρες (<i>oldChar</i>), στους νέους (<i>newChar</i>).
trim ()	Κόβει τα κενά στην αρχή και στο τέλος της συμβολοσειράς.
toString ()	Επιστρέφει το αντικείμενο (εδώ την συμβολοσειρά). Δες κεφάλαιο 4.

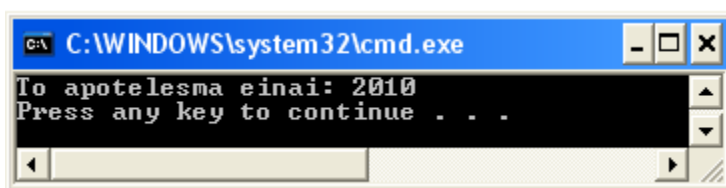
Μη ξεχνάτε τις μετατροπές (*Wrapper Class Conversion Methods*) των αριθμητικών συμβολοσειρών στους αντίστοιχους βασικούς τύπους όπως *int*, *float*, *double*, κλπ. με τις μεθόδους **parseInt**(*String*), **parseFloat**(*String*),..., και **valueOf**() (δες κεφάλαιο 5). Η μέθοδος *valueOf*() μπορεί να χρησιμοποιηθεί και για αντίστροφη μετατροπή, δηλαδή από βασικό αριθμητικό τύπο σε τύπο *String*. Το παρακάτω παράδειγμα δείχνει μια τέτοια μετατροπή:

```
public class MethodValuOf{
    public static void main(String[] args){
        int x = 20;
        int y = 10;

        String S1 = String.valueOf(x);
        String S2 = String.valueOf(y);

        System.out.println("Το αποτελεσμα είναι: "+(S1+S2)); //synenosi tvn String
    }
}
```

Το αποτέλεσμα:



Στο παρακάτω παράδειγμα θα κάνουμε χρήση των κυριότερων μεθόδων χειρισμού των συμβολοσειρών.

```
public class MethodsExample{
    public static void main(String[] args){
        String S1 = "I love Java";
```

```

int a = S1.length(); //a=11
System.out.println("O ar. xaraktiron einai: "+ a);

char b = S1.charAt(8);
System.out.println("O xaraktiras einai: "+b); //epistrefei -->a

int c = S1.indexOf('a');
System.out.println("H 1h emfanish toy xaraktira 'a': "+c); //emfanisi tou 'a'-->8

//emfanisi tou 'a' apo tin 9th-->10
int e = S1.indexOf('a', 9);
System.out.println("H emfanisi tou 'a' apo tin 9thesi kai meta: "+e);

int f = S1.indexOf("Java");
System.out.println("H lexi Java apo thn thesi: "+f); //"Java"-->apo thn 7

//"ava" xekinontas apo thn 3 thesi--> 8
int g = S1.indexOf("ava", 3);
System.out.println("H lexi 'ava' apo thn 3h thesi kai meta: "+g);

int h = S1.lastIndexOf('v');
System.out.println("Teleytaios 'v' xaraktiras: "+h); //teleytaio 'v'-->9

//teleytaio mexri ti thesi 6, 'v'-->4
int i = S1.lastIndexOf('v', 6);
System.out.println("Teleytaio 'v' mexri thn thesi 6: "+i);

int k = S1.lastIndexOf("Java");
System.out.println("Teleytaia emfanish lexis 'Java': "+k); //"Java"-->apo thn 7

//"ava" mexri ti thesi 5 --> -1 (den yparxei)
int i = S1.lastIndexOf("ava", 5);
System.out.println("Emfanisi lexis 'ava' mexri thn 5h thesi: "+i);

String s2 = S1.substring(5);
System.out.println("To substring apo tin thesi 5: "+ s2); //-->e Java

String s3 = S1.substring(7, 9);
System.out.println("To substring apo ton 7 eos 9: "+s3); //-->Ja

String s4 = S1.replace('o', 'u');
System.out.println("H antikatastasi tou 'o' me to 'u': "+s4); //-->I luve Java

String s5 = ("    My trim test    ");
String s6 = s5.trim();
System.out.println("To String xoris ta kena: "+s6); //-->My trim test

boolean x = s5.equals(S1);
System.out.println("Einai ta dyo Strings isa? "+x); //h apantisi-->>false

int m = s5.compareTo(S1);
System.out.println("H diafora tvn dyo Strings se sygkrisi: "+m);

```

```

boolean y = S1.startsWith("I love");
System.out.println("Arxizei to String me tis lexeis 'I love' ? "+y);

boolean z = S1.endsWith("ok");
System.out.println("Teleivnei to String me to 'ok' ? "+z);
}
}

```

Τα αποτελέσματα:

```

C:\WINDOWS\system32\cmd.exe
O ar. xaraktiron einai: 11
O xaraktiras einai: a
H lh emfanish toy xaraktira 'a': 8
H emfanisi tou 'a' apo tin 9thesi kai meta: 10
H lexi Java apo thn thesi: 7
H lexi 'ava' apo thn 3h thesi kai meta: 8
Televtaios 'u' xaraktiras: 9
Televtaios 'u' mexri thn thesi 6: 4
Televtaiia emfanish lexis 'Java': 7
Emfanisi lexis 'ava' mexri thn 5h thesi: -1
To substring apo tin thesi 5: e Java
To substring apo ton 7 eos 9: Ja
H antikatastasi tou 'o' me to 'u': I luve Java
To String xoris ta kena: My trim test
Einai ta dyo Strings isa? false
H diafora ton dyo Strings se sygkrisi: -41
Arxizei to String me tis lexeis 'I love' ? true
Teleivnei to String me to 'ok' ? false
Press any key to continue . . . =

```

Άλλα παραδείγματα χρήσης των μεθόδων των συμβολοσειρών

1) Παράδειγμα που δείχνει τη χρήση της μεθόδου substring():

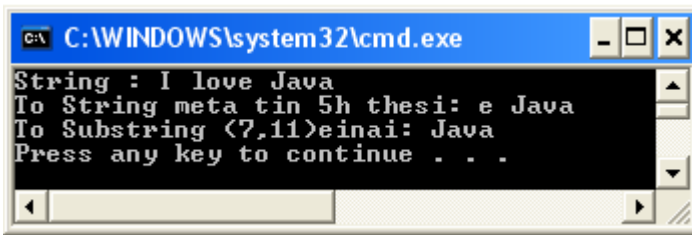
```

public class SubstringExample{
    public static void main(String[] args){
        String myString = "I love Java";
        System.out.println("String : " + myString);
        String mySubString = myString.substring(5);
        System.out.println("To String meta tin 5h thesi: " + mySubString);

        mySubString = myString.substring(7, 11);
        System.out.println("To Substring (7,11)einai: " + mySubString);
    }
}

```

Τα αποτελέσματα:



```
C:\WINDOWS\system32\cmd.exe
String : I love Java
To String meta tin 5h thesi: e Java
To Substring <7,11>einai: Java
Press any key to continue . . .
```

2) Παράδειγμα που δείχνει τη χρήση της μεθόδου concat():

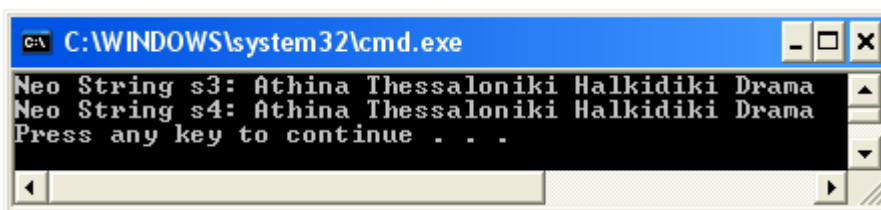
```
class TestConcat {
    public static void main(String[] args) {
        String s1 = "Athina Thessaloniki Halkidiki";
        String s2 = " Drama";

        String s3 = s1.concat(s2);

        // to idio apotelesma me synenosh me to (+)
        String s4 = s1 + s2;

        System.out.println("Neo String s3: " + s3);
        System.out.println("Neo String s4: " + s4);
    }
}
```

Τα αποτελέσματα:



```
C:\WINDOWS\system32\cmd.exe
Neo String s3: Athina Thessaloniki Halkidiki Drama
Neo String s4: Athina Thessaloniki Halkidiki Drama
Press any key to continue . . .
```

3) Παράδειγμα που δείχνει τη χρήση της μεθόδου regionMatch():

```
class TestRegionMatch {
    public static void main(String[] args) {
        String s1 = "Athina Thessaloniki Halkidiki";
        String s2 = "thessaloniki halkidiki athina";

        // Elehei an oi xar. 0-11 (thessaloniki) sto String s1
        // tairiazoun me tous xar. 7 eos 18 sto String s2, agnoontas ta
```



```
// kefalaia kai mikra grammata
```

```
boolean trueFalse = s2.regionMatches(true, 0, s1, 7, 18);  
System.out.println("string2[0 - 11] == string1[7 - 18]? " + trueFalse);  
}  
}
```

Το αποτέλεσμα:



```
C:\WINDOWS\system32\cmd.exe  
string2[0 - 11] == string1[7 - 18]? true  
Press any key to continue . . . -
```

Η κλάση StringBuffer

Η κλάση **StringBuffer** χρησιμοποιείται για την τροποποίηση των συμβολοσειρών. Τα αντικείμενα τύπου **StringBuffer** είναι δυναμικοί χώροι μνήμης (*διάνυσματα*), δηλαδή χώροι μνήμης μεταβλητού μεγέθους (μεταβάλλονται δυναμικά), αποτελούμενοι από χαρακτήρες. Τα περιεχόμενα ενός **StringBuffer** επιστρέφονται με τη μορφή συμβολοσειράς από τη μέθοδο `toString()`. Π.χ.

```
StringBuffer buf = new StringBuffer("Java");  
String s1 = buf.toString();  
System.out.println(s1); //tha emfanisei to String Java
```

Οι δομητές της StringBuffer:

- 1) **StringBuffer()**, δημιουργεί ένα κενό **StringBuffer** με χωρητικότητα 16 χαρακτ. (default).
- 2) **StringBuffer(int i)**, δημιουργεί ένα **StringBuffer** με χωρητικότητα *i* χαρακτ.
- 3) **StringBuffer(String str)**, δημιουργεί ένα **StringBuffer** με αρχική τιμή το **String str**.

Οι κυριότερες Μέθοδοι της StringBuffer (Δες αναλυτικά στην τεκμηρίωση της *Sun.API* - <http://java.sun.com/javase/6/docs/api/java/lang/StringBuffer.html>)

append(String str1) append(char ch) append(int i) append(boolean b)	Προσθήκη της συμβολοσειράς <i>str1</i> (ή <i>char</i> , ή <i>int</i> , κλπ.), στο τέλος της τρέχουσας συμβολοσειράς.
--	--

<i>... (10 υπερφορτωμένες διαφορετικές append)</i>	
insert (int offset, String str1)	Προσθήκη της συμβολοσειράς str1 στη θέση offset.
delete (int start, int end)	Διαγραφή χαρακτήρων από τη θέση start μέχρι την end - 1.
replace (int start, int end, String str1)	Αντικατάσταση χαρακτήρων από τη θέση start μέχρι την end - 1 με τη συμβολοσειρά str1.
ensureCapacity (int min)	Επιβεβαιώνει ότι η χωρητικότητα του StringBuffer είναι τουλάχιστον min.
capacity ()	Επιστρέφει τη χωρητικότητα του StringBuffer.
length ()	Επιστρέφει το μήκος του StringBuffer.
setLength (int newlength)	Αλλάζει το μήκος του StringBuffer σε newlength.
charAt (int index)	Επιστρέφει τον χαρακτήρα στη θέση index.
setCharAt (int index, char ch)	Αποθηκεύει τον χαρακτήρα ch στη θέση index.
getChars (start, end, charArray, index)	Αντιγράφει χαρακτ. από τη θέση start μέχρι τη θέση end-1 που τοποθετεί στον πίνακα χαρακτ. charArray, αρχίζοντας από τη θέση index.
reverse ()	Αντιστρέφει την αλυσίδα των χαρακτ. του StringBuffer.

Παράδειγμα:

```

class TestStringBuffer {
    public static void main(String[] args) {

        //epidixi capacity()
        StringBuffer buf = new StringBuffer(10);
        int c = buf.capacity();
        System.out.println("Χαrakt. του buffer: " + c);

        //epidixi append()
        StringBuffer buf1 = new StringBuffer("Onomateponymo: ");
        buf1.append("Nikas Nikos");
        System.out.println("O StringBuffer me append --> " + buf1);

        //epideixi insert()
        buf1.insert(17, "ni");
        System.out.println("O StringBuffer me insert --> " + buf1);

        //epideixi delete()
        buf1.delete(17, 19);
    }
}

```

```

System.out.println("O StringBuffer me delete --> " + buf1);

//epideixi replace()
buf1.replace(16, 17, "a");
System.out.println("O StringBuffer me replace --> " + buf1);

//epideixi toString()
String s1=buf1.toString();
System.out.println("To String apo to StringBuffer --> " + s1);

//epideixi reverse()
buf1.reverse();
System.out.println("O StringBuffer me reverse --> " + buf1);
}
}

```

Τα αποτελέσματα:

```

C:\WINDOWS\system32\cmd.exe
Χαρακτ. του buffer: 10
O StringBuffer me append --> Ονοματεπωνυμο: Nikas Nikos
O StringBuffer me insert --> Ονοματεπωνυμο: Ninikas Nikos
O StringBuffer me delete --> Ονοματεπωνυμο: Nikas Nikos
O StringBuffer me replace --> Ονοματεπωνυμο: Nakas Nikos
To String apo to StringBuffer --> Ονοματεπωνυμο: Nakas Nikos
O StringBuffer me reverse --> sokiN sakaN :omynopetamonO
Press any key to continue . . .

```