

ΕΡΓΑΣΤΗΡΙΟ - 9

Συμβολοσειρές - Strings

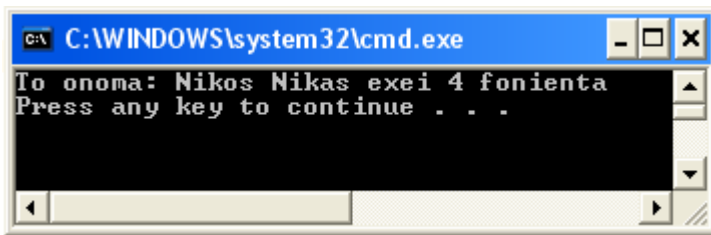
Προσοχή !!! Να εκτελεστούν πρώτα όλες οι ασκήσεις τις Θεωρίας

Άσκηση – 1^η

Να γίνει το πρόγραμμα Java που μετρά τον αριθμό των φωνηέντων ενός ονοματ/μου ή οποιασδήποτε συμβολοσειράς. Στην προτεινομένη λύση χρησιμοποιείται η **charAt()** για να ελεγχθεί ο κάθε χαρακτήρας. Ο αλγόριθμος αυτός, με την κατάλληλη τροποποίηση, μπορεί να χρησιμοποιηθεί για το μέτρημα οποιουδήποτε χαρακτήρα (αριθμού, γράμματος, κλπ.).

```
class MetrisiFonionton {  
    public static void main(String[] args) {  
        char gramma;  
  
        String name = "Nikos Nikas";  
        int arXaraktiron = name.length();  
        int metritis = 0;  
        for (int i = 0; i < arXaraktiron; i++) {  
            gramma = name.charAt(i);  
            if (gramma == 'A' || gramma == 'a' || gramma == 'E' || gramma == 'e' ||  
                gramma == 'I' || gramma == 'i' || gramma == 'O' || gramma == 'o' ||  
                gramma == 'U' || gramma == 'u') {  
                metritis++;  
            }  
        }  
        System.out.println("To onoma: " + name + " exei " + metritis + " fonienta");  
    }  
}
```

Το αποτέλεσμα:



Άσκηση – 2^η

Να γίνει η παραλλαγή του προγράμματος 1, που αντί να μετρά τα φωνήεντα μιας συμβολοσειράς, τα αντικαθιστά με ένα χαρακτήρα που ορίζουμε εμείς. Η προτεινόμενη λύση μοιάζει με αυτή της άσκησης 1, και χρησιμοποιεί επίσης την **charAt()** για τον έλεγχο του χαρακτήρα και την **setCharAt()** για την αντικατάσταση του χαρακτήρα.

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

        StringBuffer sb;
        String s="Nikas Nikos";

        int len;
        char gramma;

        sb = new StringBuffer(s);

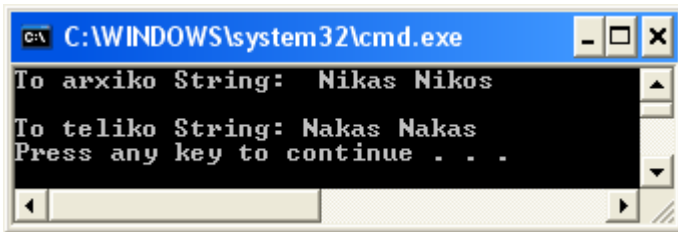
        len = sb.length();

        for (int i = 0; i < len; i++) {
            gramma = sb.charAt(i);

            if (gramma == 'A' || gramma == 'a' || gramma == 'E' || gramma == 'e' ||
                gramma == 'I' || gramma == 'i' || gramma == 'O' || gramma == 'o' ||
                gramma == 'U' || gramma == 'u') {

                sb.setCharAt(i, 'a');
            }
        }
        System.out.println( "To arxiko String: " + s + "\n");
        System.out.println( "To teliko String: " + sb );
    }
}
```

Το αποτέλεσμα:



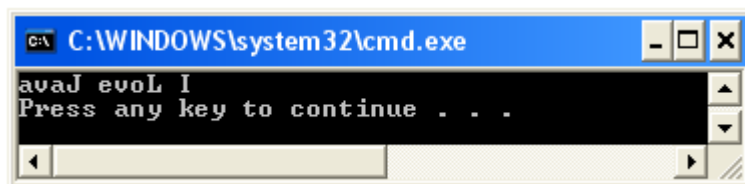
```
C:\WINDOWS\system32\cmd.exe
To arxiko String: Nikas Nikos
To teliko String: Nakas Nakas
Press any key to continue . . .
```

Άσκηση – 3^η

Στην άσκηση αυτή γίνεται η αντιστροφή των στοιχείων μιας συμβολοσειράς με τη χρήση της **StringBuffer**. Προσέξτε την επεξεργασία της συμβολοσειράς με την **StringBuffer** και μετά την μετατροπή των περιεχομένων σε συμβολοσειρά με τη μέθοδο **toString**.

```
class ReverseAnyString {
    public static void main(String[] args) {
        String myString = "I Love Java";
        StringBuffer sb = new StringBuffer();
        for (int i = myString.length() - 1; i >= 0; i--)
            sb.append(myString.charAt(i));
        String s2=sb.toString();
        System.out.println(s2);
    }
}
```

Το αποτέλεσμα:



```
C:\WINDOWS\system32\cmd.exe
avaJ evoL I
Press any key to continue . . .
```

Άσκηση – 4^η

Να γίνει το πρόγραμμα που αναζητά σε μία συμβολοσειρά την ύπαρξη κάποιας λέξης. Το πρόγραμμα θα μετρά και θα εμφανίζει τον αριθμό επανάληψης της λέξης μέσα στη συμβολοσειρά.

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

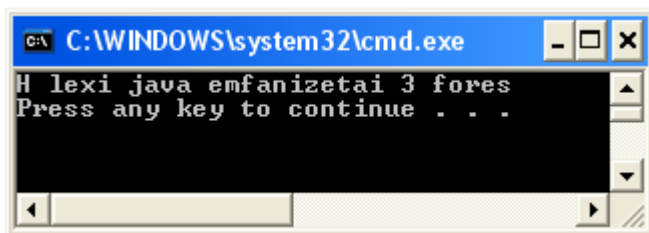
        String s="I love java java java"; //η συμβολοσειρά
        String s1="java"; //η λέξη που ψάχνουμε

        int len = s.length(); //το μήκος της λέξης που ψάχνουμε
        int result = 0;

        if (len > 0) { //ψάχνουμε μόνο αν υπάρχει κείμενο στη συμβολοσειρά
            int start = s.indexOf(s1); //βρίσκουμε την αρχή (θέση) της λέξης μέσα στη συμβολοσειρά

            while (start != -1) { //επανάληψη για εύρεση της λέξης
                result++; //ο μετρητής επανάληψης της λέξης
                start = s.indexOf(s1, start+len); //κάθε φορά από το σημείο που ξεκινά η λέξη + μήκος της
            }
        }
        int i=result; //το αποτέλεσμα
        System.out.println("Η λέξη " + s1 + " εμφανίζεται " + result + " φορές");
    }
}
```

Το αποτέλεσμα:



```
C:\WINDOWS\system32\cmd.exe
H lexi java εμφανίζεται 3 φορές
Press any key to continue . . .
```

Άσκηση – 5^η

Η άσκηση αυτή αποτελεί μία παραλλαγή της προηγούμενης άσκησης (4^{ης}). Δηλαδή, το πρόγραμμα θα ξεχωρίζει τις λέξεις μιας συμβολοσειράς και θα τις εμφανίζει σε διαφορετικές γραμμές. Ο αλγόριθμος της άσκησης είναι η εύρεση της αρχής και του τέλους της κάθε λέξης της πρότασης και η εξαγωγή της με τη χρήση της **substring()**. Κλειδί σε αυτόν τον αλγόριθμο είναι το κενό διάστημα.

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

        char keno = ' ';
        int i, arChar, arxi, telos;
```

```

String lexi;
String protasi = "H protasi ayti exei epta diaforetikes lexeis";

arChar = protasi.length();
i = 0;

//ψάχνουμε να βρούμε την αρχή και το τέλος της κάθε λέξης, ώστε με την
//substring() να τις ξεχωρίσουμε. Κλειδί του αλγόριθμου το κενό διάστημα.

while ( i < arChar ) {

    //βρίσκουμε την αρχή της λέξης, μετρώντας τα κενά διαστήματα
    while (protasi.charAt(i) == keno) {
        i++;
    }

    //κρατάμε την αρχή της λέξης
    arxi = i;

    //βρίσκουμε το τέλος της λέξης
    while (i < arChar && protasi.charAt(i) != keno) { //i < arChar, στα ορια tis protasis
        i++;
    }

    //κρατάμε το τέλος της λέξης
    telos = i;

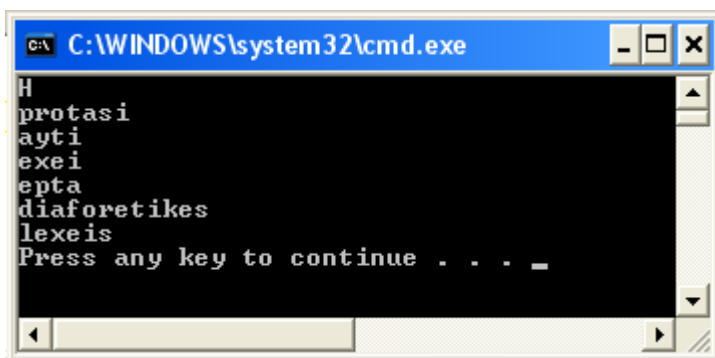
    if (arxi != telos) {

        //βρήκαμε τη λέξη και την εξάγουμε από την πρόταση με την substring()
        lexi = protasi.substring(arxi, telos);

        System.out.println(lexi);
    }
}
}
}
}

```

Τα αποτελέσματα:



```

C:\WINDOWS\system32\cmd.exe
H
protasi
ayti
exei
epta
diaforetikes
lexeis
Press any key to continue . . . -

```

Άσκηση – 6^η

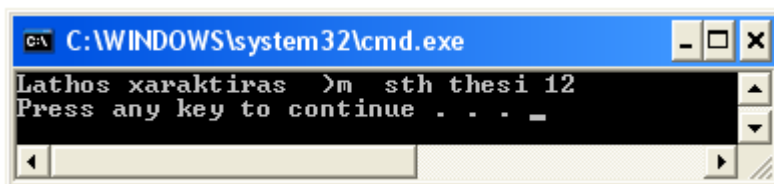
Να γίνει το πρόγραμμα που ελέγχει αν ένα String (π.χ. password) είναι σωστό, χρησιμοποιώντας την **indexOf()** για τον έλεγχο των χαρακτήρων. Αν υπάρχει λάθος, τότε το πρόγραμμα εμφανίζει τον λάθος χαρακτήρα και τη θέση του μέσα στο password.

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

        String Password="123456789kljmno"; //το σωστό password
        String validPassw = "123456789kljMno"; //αυτό που δίνει ο χρήστης

        for (int i=0;i<Password.length();i++) {
            char c = Password.charAt(i); //ο χαρακτήρας της κάθε θέσης
            if (validPassw.indexOf(c)== -1) //ο έλεγχος του χαρακτ. κάθε θέσης
                System.out.println("Lathos xaraktiras []"+c+"] sth thesi "+i);
        }
    }
}
```

Το αποτέλεσμα:



```
C:\WINDOWS\system32\cmd.exe
Lathos xaraktiras >m sth thesi 12
Press any key to continue . . . -
```

Άσκηση – 7^η

Η άσκηση αυτή έχει σκοπό να δείξει τρεις διαφορετικούς τρόπους δημιουργίας συμβολοσειράς. Διαλέξτε τον τρόπο που σας βολεύει.

```
class StringBufferDemo1 {
    public static void main(String[] args) {

        //synenvsi string
        String myString1 = "I" + " " + "love "+"Java";
        System.out.println(myString1);

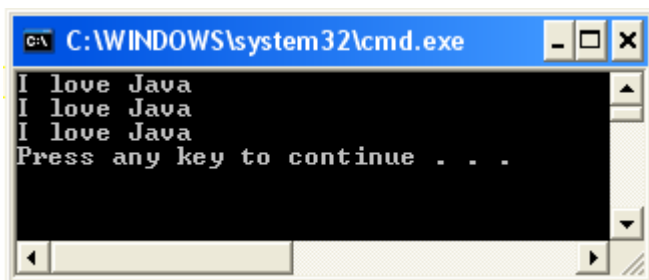
        //me ton StringBuffer, kai prothesi stoixeivn
        StringBuffer myString2 = new StringBuffer();
        myString2.append("I");
        myString2.append(" ");
    }
}
```

```
myString2.append("love ");
myString2.append("Java");

//Metatropi tou periexomenou tou StringBuffer se String
String s = myString2.toString();
System.out.println(s);

// H idia xrisi alla me pio sympiknomeno tropo
StringBuffer myString3 = new StringBuffer().append("I").
    append(" ").append("love ").append("Java");
System.out.println(myString3.toString());
}
```

Τα αποτελέσματα:



```
C:\WINDOWS\system32\cmd.exe
I love Java
I love Java
I love Java
Press any key to continue . . .
```