OBJECT ORIENTED PROGRAMMING

Lab 10: Strings, Tokenization, Characters

### Exercise 1:

What results does the following program provide?

**public class Chaine**

**{ public static void main (String args[])**

**{ String ch1 = new String();**

**System.out.println ("A - ch1 =:" + ch1 + ":") ;**

**String ch2 = "hello" ;**

**System.out.println ("B - ch2 =:" + ch2 + ":") ;**

**String ch3 = new String ("bonjour") ;**

**System.out.println ("C - ch3 =:" + ch3 + ":") ;**

**String ch4 = new String (ch3) ;**

**System.out.println ("D - ch4 =:" + ch4 + ":") ;**

**ch3 = "bonsoir" ;**

**System.out.println ("E - ch4 =:" + ch4 + ": ch3 =:" + ch3 + ":") ;**

**ch4 = ch3 ;**

**ch3 = "au revoir" ;**

**System.out.println ("F - ch4 =:" + ch4 + ": ch3 =:" + ch3 + ":") ;**

**}**

**}**

Answers

**The program displays the following results:**

**A - ch1 =::**

**B - ch2 =:hello:**

**C - ch3 =:bonjour:**

**D - ch4 =:bonjour:**

**E - ch4 =:bonjour: ch3 =:bonsoir:**

**F - ch4 =:bonsoir: ch3 =:au revoir: {**

### Exercise 2:

Write a program that reads a string on the keyboard and displays it:

* One character of two (the first being displayed),
  + For example :

give a string: java est plus portable que C++

desired result: jv s lspral u +

* the first and last character.

Answers

**We can use the class Clavier**

**You just have to use the length and charAt methods of the String class.**

**public class CarCh**

**{ public static void main (String args[])**

**{ System.out.print ("donnez une chaine : ") ;**

**String ch = Clavier.lireString() ;**

**System.out.print ("un caractere sur deux : ") ;**

**for (int i = 0 ; i<ch.length() ; i+=2)**

**System.out.print (ch.charAt(i)) ;**

**System.out.println () ;**

**System.out.println ("Premier caractere = " + ch.charAt(0)) ;**

**System.out.println ("Dernier caractere = " +**

**ch.charAt(ch.length()-1)) ;**

**}**

**Note that the last character of the string ch has the index ch.length-1. An attempt to access the character of index ch.length would result in a StringIndexOutOfBoundsException exception.}**

### Exercise 3:

Write a program that reads an integer from the keyboard and displays it vertically as in the following example:

**Please insert an integer : 785412**

**7**

**8**

**5**

**4**

**1**

**2**

Answers

**You can convert an integer to a string using the valueOf method of the String class. Access to the characters in the string is done with the charAt method, hence the program:**

**public class Conver**

**{ public static void main (String args[])**

**{ System.out.print ("donnez un nombre entier : ") ;**

**int n = Clavier.lireInt() ;**

**String ch = String.valueOf(n) ;**

**for (int i=0 ; i<ch.length() ; i++)**

**System.out.println (ch.charAt(i)) ; // for (char c : ch)**

**} // System.out.println(c) ;**

**}**

**Here we have used the valueOf method to convert an integer to a string. We could also have exploited the property of the + operator which, when one of its two operands is of type String, converts the other to that same type. This is how we could have written (somewhat artificially) ch = "" + n. Note, however, that the direct assignment of ch = n would not be correct since the type int is not compatible by assignment with the type String.**

### Exercise 4:

Write a program that reads on the keyboard a French verb from the first group (it will make sure it ends with er) and displays its conjugation in the present indicative. We will assume that this is a regular verb. In other words, we will assume that the user does not provide a verb such as ***Manger*** (to eat) (in this case, the program will display “nous mangons”!).

The results will look like this:

**donnez un verbe regulier du premier groupe : chanter**

**je chante**

**tu chantes**

**il/elle chante**

**nous chantons**

**vous chantez**

**ils/elles chantent**

Answers

**We will of course read the verb as a string. Using the substring method, we extract the end of it and compare it with the string "er".**

**The different people of the conjugation are obtained by adding to the verb, deprived of its last two characters, one of the desired endings provided here by an array of endings strings. They are preceded by a subject, also extracted from an array of subject strings.**

**public class Conjug**

**{**

**public static void main (String args[])**

**{**

**final String sujets[] = { "je", "tu", "il/elle", "nous", "vous",**

**"ils/elles"} ;**

**final String terminaisons [] = { "e", "es", "e", "ons", "ez", "ent" } ;**

**String verbe ;**

**int nbLettres ;**

**System.out.print ("donnez un verbe regulier du premier groupe :") ;**

**while (true)**

**{ verbe = Clavier.lireString() ;**

**nbLettres = verbe.length() ;**

**String fin = verbe.substring (nbLettres-2, nbLettres) ;**

**if (fin.equals("er")) break ;**

**System.out.print**

**("\*\*\* il ne se termine pas par er - donnez-en un autre :") ;**

**}**

**String rad = verbe.substring(0, nbLettres-2) ;**

**int n = terminaisons.length ;**

**for (int i=0 ; i<n ; i++)**

**System.out.println (sujets[i] + " " + rad + terminaisons[i]) ;**

**}**

**}**

### Exercise 5:

Write a program that reads a series of words on the keyboard and displays them in alphabetical order. It will be assumed that these words contain only unaccented letters (upper or lower case). The number of words will be given as data and the execution will look like this:

**Combien de mots ? 5**

**donnez vos mots**

**javaScript**

**Pascal**

**BaSiC**

**Java**

**ADA**

**Liste par ordre alphabetique :**

**ADA**

**BaSiC**

**Java**

**javaScript**

**Pascal**

Note that the words are displayed with their original "case" but that this does not affect the sorting which respects the traditional alphabetical order (which does not distinguish upper case from lower case).

Answers

**public class TrisMots**

**{**

**public static void main (String args[])**

**{ // lecture des donnees**

**System.out.print ("Combien de mots ? ") ;**

**int nMots = Clavier.lireInt() ;**

**String [] mots = new String[nMots] ;**

**System.out.println ("donnez vos mots") ;**

**for (int i=0 ; i<nMots ; i++)**

**mots[i] = Clavier.lireString() ;**

**// conversion de chaque mot en minuscules**

**String [] motsMin = new String[nMots] ;**

**for (int i=0 ; i<nMots ; i++)**

**motsMin[i] = mots[i].toLowerCase() ;**

**// tri par reorganisation des references (mots d'origine et en**

**minuscules)**

**// (on compare chaque mot (minuscule) a tous ses suivants)**

**String temp ;**

**for (int i=0 ; i<nMots-1 ; i++)**

**for (int j=i+1 ; j<nMots ; j++)**

**if (motsMin[i].compareTo(motsMin[j]) >= 0)**

**{ temp = motsMin[i] ;**

**motsMin[i] = motsMin[j] ;**

**motsMin[j] = temp ;**

**temp = mots[i] ;**

**mots[i] = mots[j] ;**

**mots[j] = temp ;**

**}**

**// affichage des chaines triees**

**System.out.println ("Liste par ordre alphabetique :") ;**

**for (int i=0 ; i<nMots ; i++) System.out.println (mots[i]) ;**

**} // System.out.println(mot) ;**

**}**