

9.- Codificar un programa que lea por teclado una cadena de caracteres y nos muestre por pantalla:

- Su longitud.
- El carácter asociado al índice 7.
- El índice que ocupa el primer carácter que sea una 'x'.
- La String transformada en mayúsculas.
- Por último, comprobar si el primer carácter de la cadena es 'E' y mostrar por consola un mensaje que lo indique.

```
1  import java.util.Scanner;
2
3  public class Ejer9_ut6 {
4
5      public static void main(String[] args) {
6          System.out.print("Introduce cadena: ");
7          Scanner teclado = new Scanner(System.in);
8          String cad1 = teclado.nextLine();
9
10         //su longitud
11         System.out.printf( "\nLongitud de cad1: %d", cad1.length() );
12
13         //carácter asociado al índice 7
14         System.out.printf( "\nCarácter 7º: %s", cad1.charAt(7) );
15
16         //índice que ocupa la primera x
17         System.out.printf( "\nPrimera x en: %d", cad1.indexOf('x'));
18
19         //pasar a mayúsculas
20         System.out.printf("\nEn mayusculas es: %s", cad1.toUpperCase() );
21
22         if (cad1.startsWith("E"))
23             System.out.println("\nEmpieza por E");
24         else
25             System.out.println("\nNo empieza por E");
26     }
27
28 }
```

10.- Codificar un programa en java que lea por teclado una cadena de caracteres y nos indique por pantalla cuántas vocales tiene dicha cadena.

Realizarlo a través de una función llamada "esVocal" a la que se le envíe un carácter y nos devuelva un *boolean* en función de que sea vocal o no.

```
1  import java.util.Scanner;
2
3  public class Ejer10_ut6 {
4
5      static boolean esVocal(char c){
6          //buscaremos sólo vocales sin tilde
7          char[] vocales ={'a','e','i','o','u'};
8          boolean esta = false;
9          //pasamos a minúsculas el caracter
10         c=Character.toLowerCase(c);
11         for(int i=0; i<vocales.length; i++){
12             if(c==vocales[i])
13                 esta=true;
14         }
15         return esta;
16     }
17
18     public static void main(String[] args) {
19         System.out.print("Introduce cadena: ");
20         Scanner teclado = new Scanner(System.in);
21         String cad1 = teclado.nextLine();
22
23         int nVocales =0;
24         for (int i=0; i<cad1.length(); i++){
25             if (esVocal(cad1.charAt(i)))
26                 nVocales++;
27         }
28         System.out.printf("Tiene %d vocales",nVocales);
29     }
30 }
31 }
```

11.- Versión A.- Codificar un programa en lenguaje java que lea por teclado una cadena de caracteres con varias palabras separadas por un espacio en blanco, y nos diga cuantas palabras tiene la cadena. No considerar los espacios del principio y del final.

Versión B.- las palabras pueden estar separadas por varios espacios en blanco.

Versión A.- Un espacio en blanco

```
1  import java.util.Scanner;
2
3  public class Ejer11A_ut6 {
4
5      static int contarPalabras(String cadena) {
6          int nPalabras = 1;
7          for (int i=0; i<cadena.length(); i++)
8              if (cadena.charAt(i)==' ')
9                  nPalabras++;
10         return nPalabras;
11     }
12
13     public static void main(String[] args) {
14         System.out.print("Introduce cadena: ");
15         Scanner teclado = new Scanner(System.in);
16         String cadena = teclado.nextLine();
17         System.out.printf("Tiene %d palabras", contarPalabras(cadena));
18     }
19
20 }
```

Versión B.- Varios espacios en blanco

```
1  import java.util.Scanner;
2
3  public class Ejer11B_ut6 {
4
5      public static void main(String[] args) {
6          System.out.print("Introduce cadena: ");
7          Scanner teclado = new Scanner(System.in);
8          String cad1 = teclado.nextLine();
9          String cad2=cad1.trim();
10         //transformamos el String en un array con una palabra en cada posición
11         String[] cadArray = cad2.split(" +");
12         System.out.printf("Tiene %d palabras",cadArray.length);
13     }
14
15 }
```

trim quita espacios al principio y al final.

split transforma un String en un array de Strings, de acuerdo con un separador, en este caso espacios en blanco.

Construimos una expresión regular " +" que representa uno o más espacios en blanco.

12.- Codificar un programa en lenguaje java que lea por teclado una cadena de caracteres y por otro lado, un carácter independiente. El programa reemplazará las ocurrencias del carácter independiente sustituyéndolas por un asterisco, mostrándonos al final la nueva cadena.

```
1  import java.util.Scanner;
2
3  public class Ejer12_ut6 {
4
5      public static void main(String[] args) {
6          System.out.print("Introduce cadena: ");
7          Scanner teclado = new Scanner(System.in);
8          String cadena = teclado.nextLine();
9          System.out.print("\nIntroduce carácter a sustituir: ");
10         char caracter = teclado.nextLine().charAt(0);
11         System.out.println("\nCadena resultante: \n"+ cadena.replace(caracter, '*'));
12     }
13
14 }
```

13.- Codificar un programa en lenguaje java que lea por teclado una cadena de caracteres correspondiente a un NIF, y que verifique si es correcto o no. Si lo es, se mostrará por consola su parte numérica; si no lo es se mostrará el mensaje "NIF no válido". Se tendrá en cuenta lo siguiente: Suponer que los NIFs tienen 8 dígitos y, a continuación, una letra (no importa que sea mayúscula o minúscula).

```
1  import java.util.Scanner;
2
3  public class Ejer13_ut6 {
4
5      static boolean compruebaNif(String nif){
6          boolean correcto=false;
7          if(nif.length()==9){
8              correcto=true;
9              char[] numeros = nif.substring(0,8).toCharArray();
10             int i=0;
11             while(i<numeros.length && correcto==true){
12                 if(!Character.isDigit(numeros[i]))
13                     correcto=false;
14                 i++;
15             }
16             if ( !Character.isLetter(nif.charAt(8)) )
17                 correcto=false;
18         }
19         return correcto;
20     }
21
22     public static void main(String[] args) {
23         System.out.print("Introduce NIF: ");
24         Scanner teclado = new Scanner(System.in);
25         String nif = teclado.nextLine();
26         if (compruebaNif(nif))
27             System.out.println(nif.substring(0,8));
28         else
29             System.out.println("NIF no válido");
30     }
31
32 }
```

toCharArray()
convierte un String en un
array de caracteres

14.- Desarrollar un programa java que pida por teclado, primeramente, el número de palabras a escribir. El programa debe validar que el número de palabras estará entre 2 y 4, ambos inclusive. Todas ellas se almacenarán en un array y se realizarán las siguientes operaciones, mostrando su resultado por consola:

- 1º. Mostrar todas las palabras del array en letras mayúsculas.
- 2º. Mostrar los tres primeros caracteres de cada una de ellas.
- 3º. Mostrar el número de veces que el carácter 'a' aparece en cada palabra.
- 4º. Mostrar la posición en que aparece la última 'a'.
- 5º. Mostrar cada palabra invertida. "Hola" ---> "aloH".

```
1  import java.util.Scanner;
2
3  public class Ejer14_ut6 {
4
5      public static void main(String[] args) {
6          Scanner teclado = new Scanner(System.in);
7          int numPalabras;
8          do{
9              System.out.print("Cuántas palabras? ");
10             numPalabras = teclado.nextInt();
11         }while (numPalabras <2 || numPalabras>4);
12         //limpio el buffer
13         teclado.nextLine();
14         //Leer palabras
15         String[] palabras = new String[numPalabras];
16         for (int i=0; i<numPalabras; i++){
17             System.out.print("\nPalabra: ");
18             palabras[i]=teclado.nextLine();
19         }
20
21         //Mostrar todas las palabras en mayúsculas
22         for (int i=0; i<palabras.length; i++)
23             System.out.println(palabras[i].toUpperCase());
24
25         //Mostrar los tres primeros caracteres
26         for (int i=0; i<palabras.length; i++)
27             System.out.println(palabras[i].substring(0,3));
28
29         //Mostrar cuántas a's en cada una
30         for (int i=0; i<palabras.length; i++){
31             int contador=0;
32             for(int x=0;x<palabras[i].length();x++)
33                 if ( (palabras[i].charAt(x)=='a') )
34                     contador++;
35             System.out.println("Nº de veces que aparece a: "+contador);
36         }
37         //Posición de la última a
38         for (int i=0; i<palabras.length; i++)
39             System.out.println("Posición de la última a: "+palabras[i].lastIndexOf('a'));
40
41         //Cada palabra invertida
42         for (int i=0; i<palabras.length; i++){
43             String cadenaInvertida="";
44             for (int x=palabras[i].length()-1; x>=0; x--)
45                 cadenaInvertida = cadenaInvertida + palabras[i].charAt(x);
46             System.out.println("Palabra invertida: "+cadenaInvertida);
47         }
48     }
49 }
50 }
```

15.- Codificar un programa en java que permita cargar una frase por teclado, luego mostrar cada palabra en una línea distinta.

```
1  import java.util.Scanner;
2
3  public class Ejer15_ut6 {
4
5      public static void main(String[] args) {
6          System.out.print("Introduce frase: ");
7          Scanner teclado = new Scanner(System.in);
8          String cadena = teclado.nextLine();
9
10         String[] palabras = cadena.split(" ");
11         for (int i=0; i<palabras.length; i++)
12             System.out.println("palabra["+i+"]: "+palabras[i]);
13     }
14
15 }
```