

CIFP César Manrique.

Programación 1º de Desarrollo de Aplicaciones Web

Profesor: José David Díaz Díaz

Actividades de la Unidad 6: Cadena de caracteres



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional.
Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o
envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Índice

Actividades.....	1
Desarrollo.....	2
Actividades de comprobación.....	2
6.1. En Java, al igual que en otros muchos lenguajes de programación, las secuencias de escape se escriben mediante:.....	2
6.2. La clase Character se encuentra ubicada en el paquete:.....	2
6.3. La aritmética de caracteres permite que exista una fuerte relación entre el tipo char y el tipo int. Sabiendo que el carácter ' a ' tiene una representación numérica de 61, ¿cómo conseguiremos mostrar el carácter ' i ' por consola, a partir de la siguiente variable?.....	2
6.4. Marca la opción que devuelve true:.....	3
6.5. Señala qué opción es cierta :.....	3
6.6. La forma correcta de comparar alfabéticamente dos cadenas es mediante:.....	3
6.7. La forma de extraer el cuarto carácter de la cadena contenida en la variable cad es mediante:.....	3
6.8. La forma de concatenar dos cadenas es mediante:.....	4
6.9. El método que permite eliminar los caracteres blancos del principio y el final de una cadena es:.....	4
6.10. Existe una relación entre las cadenas (clase String) y las tablas de caracteres (char []). ¿Qué métodos permiten convertir un String en un char[] ?.....	4
Actividades de Aplicación.....	5
6.11. Escribe un programa decodificador que muestre un texto codificado con el programa realizado en la Actividad resuelta 6.11.....	5

6.13. El preprocesador del lenguaje C elimina los comentarios (*... *) del código fuente antes de compilar. Diseña un programa que lea por teclado una sentencia en C, y elimine los comentarios.....	7
6.16. Lee una palabra o frase y muestra el proceso en el que cada letra se sustituye por otro símbolo no alfabético. Por ejemplo el carácter ' a ' se podría sustituir por el carácter ' @ ', la ' e ' por ' € ', la ' i ' por ' 1 ', etcétera.....	8
6.17. Construir un programa que convierta una palabra en secuencias de n letras. Por ejemplo, la palabra «destornillador», dividida en secuencias de 4 letras, se mostrará de la siguiente forma :.....	9
6.18. Escribe una aplicación que convierte una frase (que puede estar escrita con cualquier combinación de mayúsculas y minúsculas) en el nombre de una variable que utilice la nomenclatura Camel. Por ejemplo, la frase « Me GUsta merenDAR gaLLEtas », se convertirá en « meGustaMerendarGalletas ».....	11
6.20. Implementa un programa que lea una frase y muestre todas sus palabras ordenadas de forma alfabética. Suponemos que cada palabra de la frase se separa de otra por un único espacio.....	13

Actividades

Actividades de la Unidad 6: Cadena de caracteres.

En este documento se detallan las actividades a realizar. Se entregará al profesor en la plataforma digital dos ficheros. Un primer fichero pdf con todas las actividades a realizar, el nombre del fichero será “unidad2 + nombre del alumno.pdf”. Añadir en el fichero pdf por cada actividad de programación dos capturas de pantalla, una del código y otra de su ejecución. También en el fichero pdf copiar todas las preguntas y las respuestas correctas de las actividades de comprobación. Además, entregar un segundo fichero comprimido con todos los códigos fuentes de cada actividad de programación realizada.

Todas las actividades resueltas se deberán de analizar y no se entregarán.

A continuación, detallamos las actividades a realizar:

- **Actividades propuestas.** Esta unidad no tiene dichas actividades.
- **Actividades de comprobación.** Realizarlas todas. Copiar todas las preguntas y sus respuestas correctas.
- **Actividades de aplicación.** Realizar las siguientes **6.11, 6.13, 6.16, 6.17, 6.18 y 6.20** .
- **Actividades de ampliación.** No realizar ninguna.

Imagen 1:

[Volver al índice](#)

Desarrollo

Actividades de comprobación.

6.1. En Java, al igual que en otros muchos lenguajes de programación, las secuencias de escape se escriben mediante:

- a) Dos puntos (:).
- b) El carácter u mayúscula (U).
- c) El carácter u minúscula (u).
- d) Un barra invertida (\).

[Volver al índice](#)

6.2. La clase Character se encuentra ubicada en el paquete:

- a) java.util.
- b) java.character.
- c) java.lang.
- d) java.unicode.

[Volver al índice](#)

6.3. La aritmética de caracteres permite que exista una fuerte relación entre el tipo char y el tipo int. Sabiendo que el carácter ' a ' tiene una representación numérica de 61, ¿cómo conseguiremos mostrar el carácter ' i ' por consola, a partir de la siguiente variable?

```
int codepoint 0x61
```

- a) System.out.println ((char) (codepoint + ' 8 '));
- b) System.out.println ((char) (codepoint - 18 '));
- c) System.out.println ((char) (codepoint - 8));
- d) System.out.println ((char) (codepoint + 8));

[Volver al índice](#)

6.4. Marca la opción que devuelve true:

- a) `Character.isLetter (' 2 ');`
- b) `Character.isUpperCase (' 2 ');`
- c) `Character.isLowerCase (' 2 ');`
- d) `Character.isLetterOrDigit (' 2 ');`

[Volver al índice](#)

6.5. Señala qué opción es cierta :

- a) ' a ' es un carácter.
- b) ' a ' es una cadena de caracteres.
- c) “ a ” es un carácter.
- d) Todas las opciones anteriores son ciertas.

[Volver al índice](#)

6.6. La forma correcta de comparar alfabéticamente dos cadenas es mediante:

- a) El operador `==`.
- b) El método `equal()` de `String`.
- c) El método `equal()` de `Character`.
- d) Todas permiten comparar dos cadenas.

6.7. La forma de extraer el cuarto carácter de la cadena contenida en la variable `cad` es mediante:

- a) `cad.indexOf (3).`
- b) `cad.charAt (3).`
- c) `cad. position (3).`
- d) `cad.extract (3).`

[Volver al índice](#)

6.8. La forma de concatenar dos cadenas es mediante:

- a) El operador +.
- b) El operador + =.
- c) El método concat().
- d) Todas permiten concatenar cadenas.**

[Volver al índice](#)

6.9. El método que permite eliminar los caracteres blancos del principio y el final de una cadena es:

- a) isWhiteSpace().
- b) deleteWhiteSpace().
- c) strip().**
- d) stripLeading().

[Volver al índice](#)

6.10. Existe una relación entre las cadenas (clase String) y las tablas de caracteres (char []). ¿Qué métodos permiten convertir un String en un char[] ?

- a) toCharArray().**
- b) valueof().
- c) convertString().
- d) empty() .

[Volver al índice](#)

Actividades de Aplicación.

6.11. Escribe un programa descodificador que muestre un texto codificado con el programa realizado en la Actividad resuelta 6.11.

```
1  /**
2   * Click https://pbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3   * to change this license
4   * Click https://pbhost/SystemFileSystem/Templates/Classes/Class.java to edit
5   * this template
6   */
7  package Main;
8
9  /**
10   * @author juancfm
11   */
12  import java.util.Scanner;
13
14  public class Main {
15
16      /**
17       * Escribe un programa descodificador que muestre un texto codificado con el
18       * programa realizado en la Actividad resuelta 6.11.
19       *
20       * A efectos prácticos se modificó el programa de la actividad resuelta 6.11
21       * de manera que este programa pudiera codificar y/o decodificar un texto
22       * introducido por el usuario, haciendo uso de una misma función, según la
23       * opción seleccionada por dicho usuario.
24       */
25      /**
26       * public static void main(String[] args) {
27       *     String option = "1";
28       *     Scanner sc = new Scanner(System.in);
29       *     System.out.println("¡Hola!");
30       *
31       *     // Mientras las opciones sean 1 o 2 el programa se seguirá ejecutando,
32       *     // en caso contrario se detiene su ejecución.
33       *     while (option.equals("1") || option.equals("2")) {
34       *         System.out.println("¡Indique la opción deseada: ");
35       *         System.out.println("1: Codificar un texto.");
36       *         System.out.println("2: Decodificar un texto.");
37       *         System.out.print("Indique cualquier otra tecla para...");
38       *         String s = sc.nextLine();
39       *         option = s;
40       *         if (option.equals("1") || option.equals("2")) {
41       *             // Si la opción es 1 o 2 se llama a la función codifica
42       *             codifica(sc, option);
43       *         }
44       *     }
45       *
46       *     // Cerramos el Scanner y paramos la ejecución del programa
47       *     sc.close();
48       *     System.exit(status:0);
49       * }
50
51  }
```

```
52  /**
53   * Codifica o Decodifica el texto introducido por el usuario
54   *
55   * @param sc
56   * @param option
57   */
58  static void codifica(Scanner sc, String option) {
59      /**
60       * Definimos dos conjuntos de caracteres que serán los que
61       * intercambiarán letras con el fin de lograr la codificación. Este tipo
62       * de cifrado se conoce como cifrado por sustitución. Más info:
63       * https://es.wikipedia.org/wiki/Cifrado\_por\_sustituci%C3%B3n
64       */
65      final char conjunto1[] = {'e', 'i', 'k', 'm', 'p', 'q', 'r', 's', 't',
66                               'u', 'v'};
67      final char conjunto2[] = {'p', 'v', 'i', 'u', 'm', 't', 'e', 'r', 'k',
68                               'q', 's'};
69
70      String texto;
71      System.out.print("Introduzca un texto a codificar: ");
72      texto = sc.nextLine();
73      texto = texto.toLowerCase();
74
75      char[] arrayTexto = texto.toCharArray();
76      int pos;
77
78      // Guardamos el conjunto correspondiente a la opción del usuario en una
79      // cadena, este conjunto es el que se va a comparar con el texto del
80      // usuario
81      String origen = String.valueOf(
82          option.equals("1")
83              ? conjunto1
84              : conjunto2
85      );
86
87      // Seleccionamos cual va a ser el conjunto de reemplazo, este conjunto
88      // reemplazará a los que se encuentren en el texto del usuario que
89      // coinciden con los que están en la cadena origen
90      char[] destino = (option.equals("1")
91          ? conjunto2
92          : conjunto1);
93
94      // Procedemos a hacer la sustitución
95      for (int i = 0; i < arrayTexto.length; i++) {
96          pos = origen.indexOf(arrayTexto[i]);
97          if (pos != -1) {
98              arrayTexto[i] = destino[pos];
99          }
100      }
101      // Usamos la función presentarValor para imprimir el resultado
102      presentarValor(arrayTexto, sc);
103  }
```

```
104  /**
105   * Imprime el resultado del proceso de codificación/decodificación por
106   * pantalla entre dos líneas hechas con el caracter "-" para que se
107   * visualice mejor, además introduce una pausa antes de continuar con la
108   * ejecución
109   *
110   * @param arrayTexto
111   * @param sc
112   */
113  private static void presentarValor(char[] arrayTexto, Scanner sc) {
114
115      String wait;
116
117      System.out.print("\n");
118      for (int i = 0; i < 40; i++) {
119          System.out.print("-");
120      }
121      System.out.println("\nEl resultado es:\t" + String.valueOf(
122          arrayTexto));
123      for (int i = 0; i < 40; i++) {
124          System.out.print("-");
125      }
126      System.out.print("\n");
127      System.out.println("Presione <<Enter>> para continuar...");
128      wait = sc.nextLine();
129      System.out.println("\n\n");
130  }
```

Actividades de la Unidad 6: Cadena de caracteres

```
run:
    Hola.

    Indique la opción deseada:

    1      Codificar un texto.
    2      Decodificar un texto.

        Indique cualquier otra tecla para salir: 1

    Introduzca un texto a codificar: este es el texto a codificar

    -----
El resultado es:      prkp pr pl kpxko a codvfvcae
    -----
Presione <<Enter>> para continuar...
|
```

```
    Indique la opción deseada:

    1      Codificar un texto.
    2      Decodificar un texto.

        Indique cualquier otra tecla para salir: 2

    Introduzca un texto a codificar: prkp pr pl kpxko a codvfvcae

    -----
El resultado es:      este es el texto a codificar
    -----
Presione <<Enter>> para continuar...
|
```

[Volver al índice](#)

6.13. El preprocesador del lenguaje C elimina los comentarios (/* ... */) del código fuente antes de compilar. Diseña un programa que lea por teclado una sentencia en C, y elimine los comentarios.

Sentencia : if (a == 3) / * igual a tres * / a ++ ; / * incrementamos a */

Salida : if (a == 3) a ++ ;

```
1  /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * to change this license
4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
5  * this template
6  */
7  package Main;
8
9
10 /**
11  * @author juancfr
12  */
13 import java.util.Scanner;
14
15 public class Main {
16
17     /**
18     * 6.13. El preprocesador del lenguaje C elimina los comentarios
19     * /*... */ del código fuente antes de compilar. Diseña un programa que
20     * lea por teclado una sentencia en C, y elimine los comentarios.
21     */
22
23     public static void main(String[] args) {
24         /**
25         * La cadena example contiene el ejemplo del ejercicio, basta copiar el
26         * texto entre comillas dobles y pegarla en donde nos pide introducir
27         * la sentencia
28         */
29         String example = "if(a == 3) /*igual a tres*/ a++;/*incrementamos a */";
30
31         String sentencia = "";
32         Scanner sc = new Scanner( source: System.in );
33
34         sentencia = capturarDatos(sc, sentencia);
35         sentencia = quitarComentarios(sentencia);
36         presentarValor(sentencia);
37
38         // Cerramos el Scanner y paramos la ejecución del programa
39         sc.close();
40         System.exit( status:0 );
41
42     }
43 }
```

```
60
61 /**
62 *
63 * Elimina los comentarios haciendo uso del método replaceAll()
64 * reemplaza el patrón identificado con la expresión regular por una cadena
65 * vacía
66 *
67 * @param sentencia
68 * @return
69 */
70 private static String quitarComentarios(String sentencia) {
71     sentencia = sentencia.replaceAll( regex:"/*(?:.*?)\\*/", replacement:"");
72
73     return sentencia;
74 }
75
76
77 /**
78 * Imprime el resultado del proceso de depuración de comentarios por
79 * pantalla entre dos líneas hechas con el caracter "_" para que se
80 * visualice mejor
81 *
82 * @param sentencia
83 */
84 private static void presentarValor (String sentencia) {
85
86     String wait;
87
88     System.out.print( s: "\n");
89     for (int i = 0; i < 40; i++) {
90         System.out.print( s: "_ ");
91     }
92     System.out.println("\n\nEl resultado es:\t" + sentencia);
93     for (int i = 0; i < 40; i++) {
94         System.out.print( s: "_ ");
95     }
96     System.out.print( s: "\n");
97     System.out.println( x: "\n\n\n");
98
99 }
100
101 }
```

```
44 /**
45 *
46 * Captura la sentencia que será depurada de los comentarios
47 *
48 * @param sc
49 * @param sentencia
50 * @return
51 */
52 private static String capturarDatos(Scanner sc, String sentencia) {
53
54     System.out.print( s: "Introduzca la sentencia a depurar: ");
55     sentencia = sc.nextLine();
56
57     return sentencia;
58 }
59
60 }
```

```
run:
Introduzca la sentencia a depurar: if(a == 3) /*igual a tres*/ a++;/*incrementamos a */

-----

El resultado es:      if(a == 3) a ++;

-----
```

[Volver al índice](#)

6.16. Lee una palabra o frase y muestra el proceso en el que cada letra se sustituye por otro símbolo no alfabético. Por ejemplo el carácter 'a' se podría sustituir por el carácter '@', la 'e' por '€', la 'i' por '1', etcétera.

```

1  /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * to change this license
4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
5  * this template
6  */
7
8  package Main;
9
10 /**
11  * @author juancfm
12  */
13 import java.util.Scanner;
14
15 public class Main {
16
17     /**
18     * 6.16. Lee una palabra o frase y muestra el proceso en el que cada letra
19     * se sustituye por otro símbolo no alfabético. Por ejemplo el carácter 'a'
20     * se podría sustituir por el carácter '@', la 'e' por '€', la 'i' por '1',
21     * etcétera.
22     */
23
24     public static void main(String[] args) {
25         String sentencia = "";
26         Scanner sc = new Scanner(System.in);
27
28         // Se llaman a los distintos métodos usados
29
30         presentarValor(
31             sentencia: cambiarLetras(
32                 sentencia: capturarDatos(sc, sentencia)
33             )
34         );
35
36         /**
37         * Cierra el Scanner y sale del programa con status 0 de ejecución
38         * correcta
39         */
40         sc.close();
41         System.exit(status:0);
42     }
43 }

```

```

44 /**
45 * Captura la sentencia que a la que se le cambiarán los caracteres
46 */
47
48 @param sc Scanner usado para la captura de datos
49 @param sentencia Sentencia, frase o palabra a ser procesada
50 @return Retorna un String con la sentencia a procesar
51 */
52 private static String capturarDatos(Scanner sc, String sentencia) {
53
54     System.out.print("Introduzca la sentencia a cambiar: ");
55     sentencia = sc.nextLine();
56
57     return sentencia;
58 }
59
60 }

```

```

61 /**
62 * Cambia las letras por los símbolos.
63 */
64
65 * Para hacer esto de manera diferente a la actividad 6.11 se optó por
66 * cambiar cada carácter por el carácter correspondiente a los
67 * códigos decimales 913 y siguientes (alfabeto griego)
68 */
69
70 @param sentencia Sentencia a la que se le van a cambiar las caracteres
71 @return Retorna un String con las letras de la sentencia cambiadas
72 */
73 private static String cambiarLetras(String sentencia) {
74
75     String letrasCambiadas = "";
76     char c;
77     for (int i = 0; i < sentencia.length(); i++) {
78         c = sentencia.charAt(i);
79         if (c != ' ') c += 913 - 65;
80
81         letrasCambiadas += c;
82     }
83
84     return letrasCambiadas;
85 }
86
87 }
88
89





```

```

89 /**
90 * Imprime el resultado del proceso de cambio de letras por pantalla entre
91 * dos líneas hechas con el carácter "-" para que se pueda visualizar mejor
92 */
93
94 @param sentencia Frase que va a ser presentada por pantalla
95 */
96 private static void presentarValor(String sentencia) {
97
98     System.out.print("\n");
99     for (int i = 0; i < 40; i++) {
100         System.out.print(" ");
101     }
102     System.out.println("\n\nEl resultado es:\t" + sentencia);
103     for (int i = 0; i < 40; i++) {
104         System.out.print(" ");
105     }
106     System.out.print("\n");
107     System.out.println("\n\n");
108 }
109
110 }

```

Output - Main (run) x

```

run:
Introduzca la sentencia a cambiar: Esta es la sentencia que vamos a cambiar

-----
El resultado es:      Esta es μα σεξετεγια ρυε φανος α γανβιας
-----

BUILD SUCCESSFUL (total time: 21 seconds)

```

[Volver al índice](#)

6.17. Construir un programa que convierta una palabra en secuencias de n letras. Por ejemplo, la palabra «destornillador», dividida en secuencias de 4 letras, se mostrará de la siguiente forma :

dest

orni

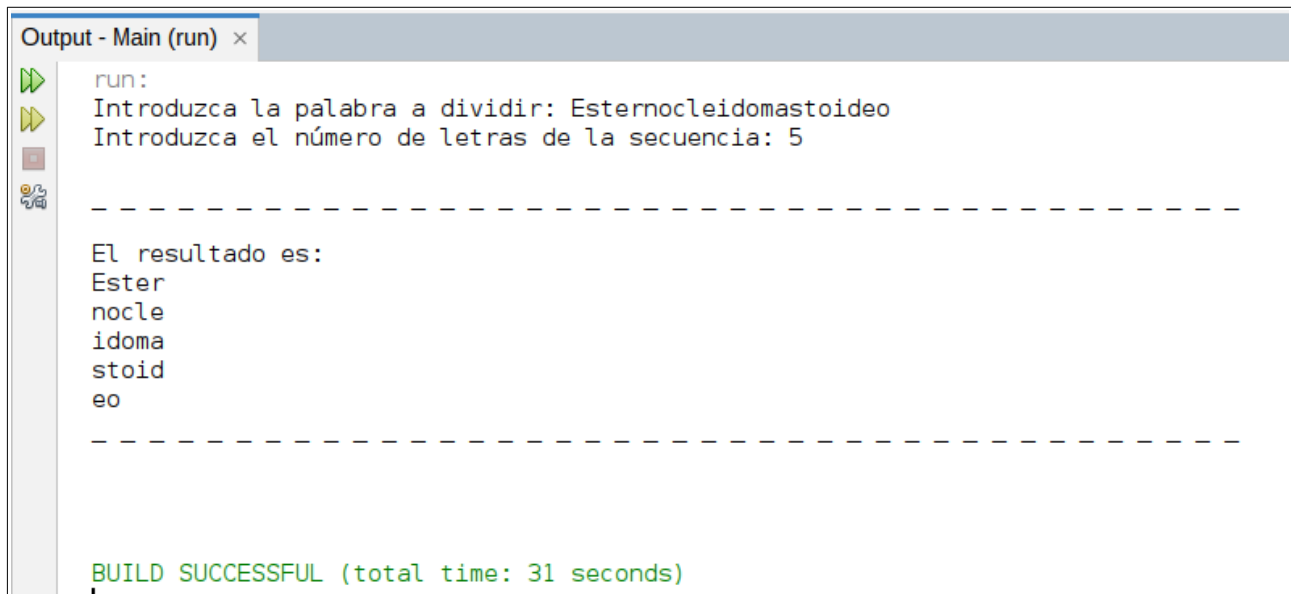
llad

or

```
1  /*
2  * Click https://localhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * to change this license
4  * Click https://localhost/SystemFileSystem/Templates/Classes/Class.java to edit
5  * this template
6  */
7  package Main;
8
9  /**
10   * @author juancfm
11   */
12  import java.util.Scanner;
13
14  public class Main {
15
16      /**
17       *
18       * 6.17. Construir un programa que convierta una palabra en secuencias de
19       * n letras. Por ejemplo, la palabra «destornillador», dividida en
20       * secuencias de 4 letras, se mostrará de la siguiente forma :
21       *
22       * dest
23       * orni
24       * llad
25       * or
26       */
27
28      public static void main(String[] args) {
29          Scanner sc = new Scanner(System.in);
30
31          // Se llaman a los distintos métodos usados
32
33          presentarValor(
34              sentencia: separarPalabra(
35                  datos: capturarDatos(sc)
36              )
37          );
38
39          /**
40           * Cierra el Scanner y sale del programa con status 0 de ejecución
41           * correcta
42           */
43          sc.close();
44          System.exit(status:0);
45
46      }
```

```
47  /**
48   *
49   * Captura la palabra que será separada en secuencias
50   *
51   * @param sc Scanner usado para la captura de datos
52   * @return Retorna un array de String con la palabra a procesar y el número
53   * de letras de las secuencias en las cuales será dividida la palabra
54   */
55  private static String[] capturarDatos(Scanner sc) {
56
57      String[] datos = new String[2];
58
59      System.out.print(s: "Introduzca la palabra a dividir: ");
60      datos[0] = sc.nextLine();
61
62      System.out.print(s: "Introduzca el número de letras de la secuencia: ");
63      datos[1] = sc.nextLine();
64
65      return datos;
66
67  }
```

```
68  /**
69   *
70   * Separa la palabra ingresada por el usuario en secuencias de la longitud
71   * especificada por él
72   *
73   * @param datos Array que contiene la palabra a separar y extensión de la
74   * secuencia
75   * @return Retorna un String con la palabra separada en secuencias de n
76   * letras
77   */
78
79  private static String[] separarPalabra(String[] datos) {
80
81      String myRegex = "(?<=[\\G.{ " + datos[1] + "})";
82
83      return datos[0].split(regex: myRegex);
84
85  }
86
87  /**
88   * Imprime el resultado del proceso de cambio de letras por pantalla entre
89   * dos líneas hechas con el caracter "_" para que se pueda visualizar mejor
90   *
91   * @param sentencia Frase que va a ser presentada por pantalla
92   */
93  private static void presentarValor(String[] sentencia) {
94
95      System.out.print(s: "\n");
96      for (int i = 0; i < 40; i++) {
97          System.out.print(s: "_ ");
98      }
99      System.out.println(x: "\n\nEl resultado es: ");
100
101      for (String word: sentencia) {
102          System.out.println(x: word);
103      }
104
105      for (int i = 0; i < 40; i++) {
106          System.out.print(s: "_ ");
107      }
108      System.out.print(s: "\n");
109      System.out.println(x: "\n\n\n");
110
111  }
```



```
Output - Main (run) x
run:
Introduzca la palabra a dividir: Esternocleidomastoideo
Introduzca el número de letras de la secuencia: 5

-----

El resultado es:
Ester
nocle
idoma
stoid
eo

-----

BUILD SUCCESSFUL (total time: 31 seconds)
```

[Volver al índice](#)

6.18. Escribe una aplicación que convierte una frase (que puede estar escrita con cualquier combinación de mayúsculas y minúsculas) en el nombre de una variable que utilice la nomenclatura Camel. Por ejemplo, la frase « Me GUsta merenDAR gaLLEtas », se convertirá en « meGustaMerendarGalletas ».

Supondremos que cada palabra que compone la frase está separada por un único espacio en blanco.

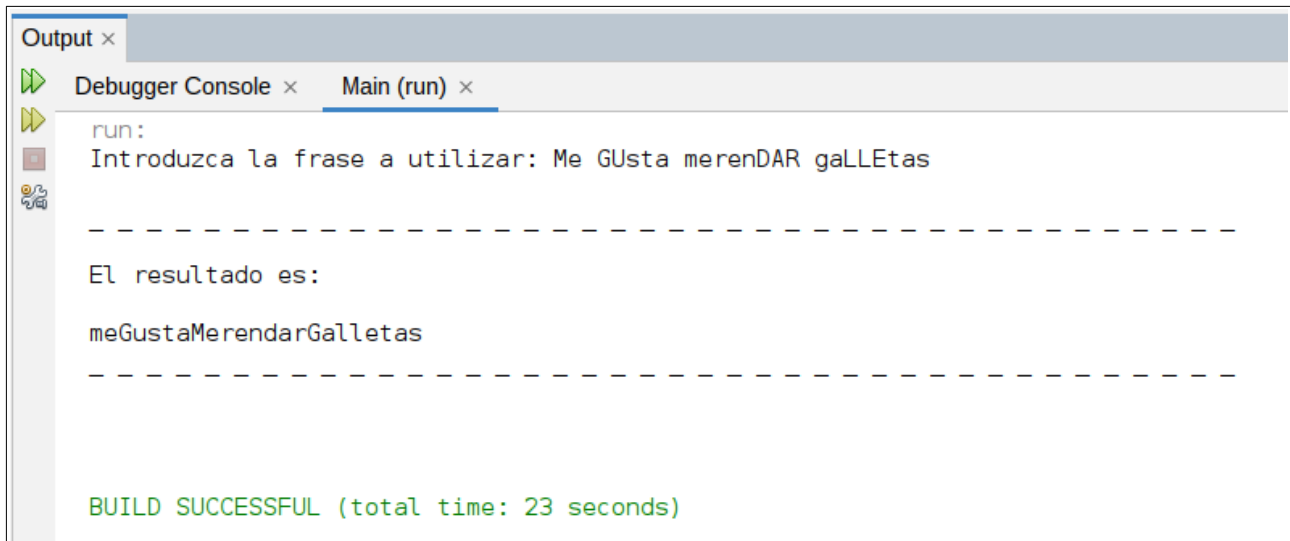
```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * to change this license
4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
5  * this template
6  */
7  package Main;
8
9  /**
10   * @author juancf
11   */
12  import java.util.Scanner;
13
14  public class Main {
15
16      /**
17       *
18       * 6.18. Escribe una aplicación que convierte una frase ( que puede estar
19       * escrita con cualquier combinación de mayúsculas y minúsculas ) en el
20       * nombre de una variable que utilice la nomenclatura Camel. Por ejemplo, la
21       * frase « Me GUsta merenDAR gaLLEtas », se convertirá en
22       * «meGustaMerendarGalletas». Supondremos que cada palabra que compone la
23       * frase está separada por un único espacio en blanco.
24       */
25
26      public static void main(String[] args) {
27          Scanner sc = new Scanner( source: System.in );
28
29          // Se llaman a los distintos métodos usados
30          prepararValor(
31              varName: prepararVarName(
32                  frase: capturarDatos(sc)
33              )
34          );
35
36          /**
37           * Cierra el Scanner y sale del programa con status 0 de ejecución
38           * correcta
39           */
40          sc.close();
41          System.exit( status: 0 );
42
43      }
```

```
44  /**
45   *
46   * Captura la frase que será convertida en el nombre de la variable
47   *
48   * @param sc Scanner usado para la captura de datos
49   * @return frase Retorna un String con la frase
50   */
51  private static String capturarDatos(Scanner sc) {
52
53      String frase;
54
55      System.out.print( s: "Introduzca la frase a utilizar: ");
56      frase = sc.nextLine();
57
58      return frase;
59
60  }
```

```
61  /**
62   *
63   * Primera convertimos todas las letras de la frase en minúsculas, usando el
64   * método split creamos un array con todas las palabras de la frase, y luego
65   * listamos esas palabras una a una convirtiendo su primera letra en
66   * mayúscula, a excepción de la primera palabra que permanecerá en minúscula
67   * todo eso lo concatenamos y lo retornamos
68   *
69   * @param frase String con la frase a tratar
70   * @return varName Nombre de la variable creada a partir de la frase dada
71   */
72
73  private static String prepararVarName(String frase) {
74
75      frase = frase.toLowerCase();
76      String[] words = frase.split( regex: "\\s+" );
77      String varName = "";
78      int cont = 0;
79
80      for (String word : words) {
81
82          cont++;
83
84          if (cont == 1) {
85              varName += word;
86          } else {
87
88              varName
89                  += word.substring( beginIndex: 0, endIndex: 1 ).toUpperCase()
90                  + word.substring( beginIndex: 1 );
91
92          }
93
94      }
95
96      return varName;
97
98  }
```

```
101  /**
102   *
103   * Imprime el resultado de la creación del nombre de la variable
104   *
105   * @param varName Nombre de la variable que va a ser presentada por pantalla
106   */
107  private static void presentarValor(String varName) {
108
109      System.out.print( s: "\n" );
110      for (int i = 0; i < 40; i++) {
111          System.out.print( s: " " );
112      }
113
114      System.out.println( "\n\nEl resultado es: \n\n" + varName );
115
116      for (int i = 0; i < 40; i++) {
117          System.out.print( s: " " );
118      }
119      System.out.print( s: "\n" );
120      System.out.println( x: "\n\n\n" );
121
122  }
```

Actividades de la Unidad 6: Cadena de caracteres



```
run:
Introduzca la frase a utilizar: Me GUSTA merendar gaLLETas

-----

El resultado es:

meGustaMerendarGalletas

-----

BUILD SUCCESSFUL (total time: 23 seconds)
```

[Volver al índice](#)

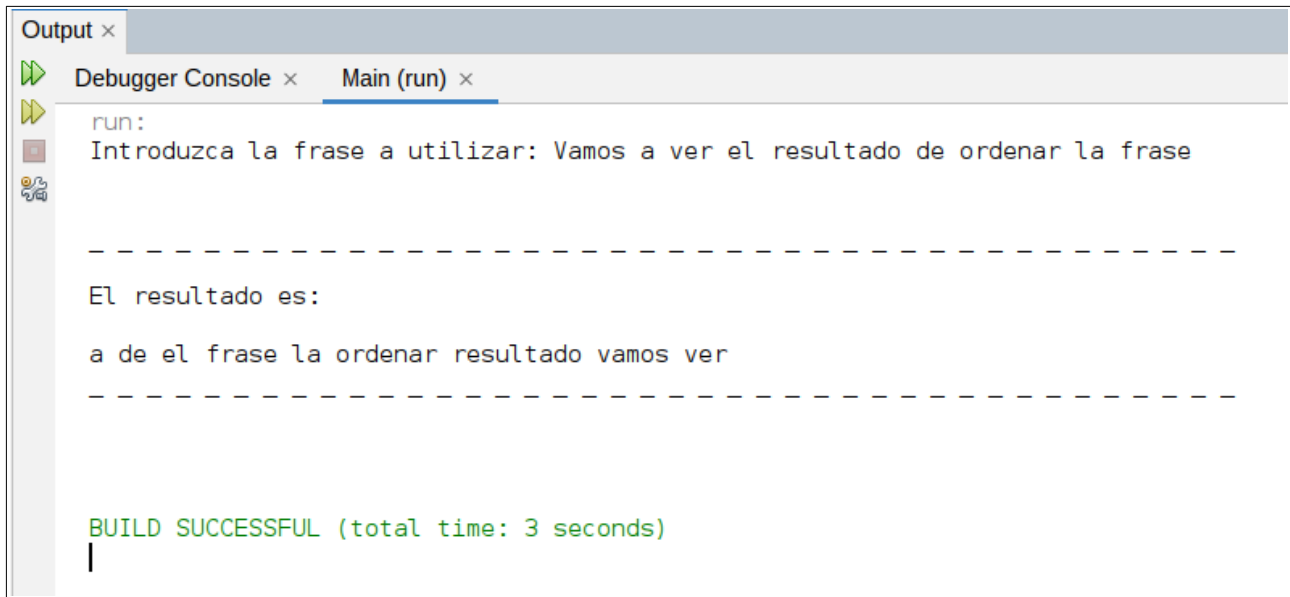
6.20. Implementa un programa que lea una frase y muestre todas sus palabras ordenadas de forma alfabética. Suponemos que cada palabra de la frase se separa de otra por un único espacio.

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
3  * to change this license
4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
5  * this template
6  */
7  package Main;
8
9  /**
10   * @author juancfm
11   */
12  import java.util.Scanner;
13  import java.util.Arrays;
14
15  public class Main {
16
17      /**
18       *
19       * 6.20. Implementa un programa que lea una frase y muestre todas sus
20       * palabras ordenadas de forma alfabética. Suponemos que cada palabra de la
21       * frase se separa de otra por un único espacio.
22       */
23
24      public static void main(String[] args) {
25          Scanner sc = new Scanner(System.in);
26
27          // Se llaman a los distintos métodos usados
28          presentarValor(
29              frase:ordenarDatos(
30                  frase:capturarDatos(sc)
31              )
32          );
33      }
34      /**
35       * Cierra el Scanner y sale del programa con status 0 de ejecución
36       * correcta
37       */
38      sc.close();
39      System.exit(status:0);
40  }
41
```

```
41
42
43  /**
44   *
45   * Captura la frase que se usará para ordenar alfabéticamente sus palabras
46   *
47   * @param sc Scanner usado para la captura de datos
48   * @return frase Retorna un String con la frase
49   */
50  private static String capturarDatos(Scanner sc) {
51
52      String frase;
53
54      System.out.print(s:"Introduzca la frase a utilizar: ");
55      frase = sc.nextLine();
56
57      return frase;
58  }
59
```

```
60
61
62  /**
63   *
64   * Primero
65   *
66   * @param frase String con la frase a tratar
67   * @return frase.strip() frase con palabras ordenadas alfabéticamente
68   * eliminando cualquier carácter blanco al inicio o al final de la frase
69   */
70  private static String ordenarDatos(String frase) {
71
72      frase = frase.toLowerCase();
73      String[] words = frase.split(regex:"\\s+");
74      Arrays.sort(a:words);
75      frase = "";
76
77      for (String word : words) {
78          frase += word + " ";
79      }
80
81      return frase.strip();
82  }
83
84
85  /**
86   *
87   * Imprime el resultado de la ordenación de las palabras de la frase
88   *
89   * @param frase frase ordenada
90   */
91  private static void presentarValor(String frase) {
92
93      System.out.print(s:"\n");
94      for (int i = 0; i < 40; i++) {
95          System.out.print(s:"_ ");
96      }
97
98      System.out.println("\n\nEl resultado es: \n\n" + frase);
99
100      for (int i = 0; i < 40; i++) {
101          System.out.print(s:"_ ");
102      }
103      System.out.print(s:"\n");
104      System.out.println(x:"\n\n\n");
105
106  }
107
```

Actividades de la Unidad 6: Cadena de caracteres



The screenshot shows a debugger console window with two tabs: "Output x" and "Main (run) x". The "Main (run) x" tab is active. The console output is as follows:

```
run:
Introduzca la frase a utilizar: Vamos a ver el resultado de ordenar la frase

-----

El resultado es:

a de el frase la ordenar resultado vamos ver

-----

BUILD SUCCESSFUL (total time: 3 seconds)
```

[Volver al índice](#)