

Ejercicio 01: Digits

Los objetivos de este ejercicio son:

- Uso de la clase String
- Uso de Java Documentation

El siguiente código usa distintos métodos de la clase String. Ejecútalo e indica cual es la salida por consola. ¿Qué métodos de la clase String se utilizan? ¿Son estáticos o de instancia? ¿Qué función tiene cada uno de estos métodos? ¿Qué devuelven? ¿Qué parámetros de entrada tienen?

```
public class Main {  
    public static void main(String[] args) {  
        double d = 858.48;  
        String s = String.valueOf(d);  
        int dot = s.indexOf('.');  
        System.out.println(dot+" digits before decimal point");  
        System.out.println((s.length()-dot-1)+" digits after decimal point");  
    }  
}
```

Ejercicio 02: Substrings

Los objetivos de este ejercicio son:

- Uso de la clase String
- Uso de Java Documentation

Completa el siguiente código donde se define una variable String que contiene la ruta a un fichero, para que muestre por consola:

- la extensión del fichero
- el nombre del fichero

```
public class Main {  
    public static void main(String[] args) {  
        String path = "home/user/index.html";  
        //insert code here  
    }  
}
```



Hint:

Consulta el método substring de la clase String en la documentación de Java, verás que permite obtener subcadenas de caracteres de una string dada.

Diagram illustrating string extraction from the path `/home/user/index.html`:

- `substring (sep+1, dot)` extracts the filename `index`.
- `substring (dot+1)` extracts the extension `.html`.
- `sep = lastIndexOf ("/")` finds the position of the last slash.
- `dot = lastIndexOf (".")` finds the position of the last dot.

Comprueba que para obtener la extensión del fichero (`.html` en este ejemplo), tenemos que localizar el carácter `'.'` dentro de la cadena de texto que referencia la ruta del fichero y extraer la substring desde esta posición hasta el final de la cadena

Comprueba que para obtener el nombre del fichero (`index` en este ejemplo), tenemos que localizar el carácter `'/'` dentro de la cadena de texto que referencia la ruta del fichero y extraer la substring desde esta posición hasta el carácter `'.'`

Ejercicio 03: Upper/Lower case

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de la clase String
- Uso de Java Documentation

Desarrolla un programa en Java que solicite al usuario un texto y lo muestre en mayúsculas y en minúsculas.



Hint:

Antes de programar una funcionalidad la recomendación es consultar los métodos predefinidos de la clase (en este caso la clase String) para comprobar si esta implementada. Consulta la documentación de Java de la clase String para buscar si dispone de métodos que permitan convertir los caracteres una cadena de texto a mayúsculas / minúsculas.

Ejercicio 04: Contains

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de la clase String
- Uso de sentencias condicionales
- Uso de Java Documentation

Desarrolla un programa en Java que solicite al usuario un texto.

- Si el texto introducido por el usuario está vacío (no contiene caracteres) el programa termina.
- Si el texto introducido por el usuario no está vacío, solicita al usuario una palabra y busca esta palabra en el texto, mostrando por consola si se ha localizado o no.

Ejercicio 05: Search Book

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de sentencias condicionales
- Uso de bucles
- Uso de arrays

Ejercicio 05 Tarea #1

Completa el siguiente programa en Java que dispone de la colección de libros de una biblioteca, para que solicite al usuario el título de un libro, y le devuelva si existe en el catálogo o no.

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        String[] catalogo = {"Los muertos vivientes", "Brevísima historia del tiempo", "La clave secreta del universo", "Mi universo", "Patria", "De mayor quiero ser feliz", "Empieza por los zapatos"};
        //Insert code here
    }
}
```

En la siguiente imagen se muestra un ejemplo de ejecución del programa:

```
Bienvenido a nuestra biblioteca
Escribe el título del libro que quieres buscar:
Los muertos vivientes
Se ha localizado el libro
```



Hint:

Nos gustaría que si el usuario introduce "los muertos vivientes" y en el catálogo el título del libro está registrado como "Los muertos vivientes", el resultado de la búsqueda sea positivo. Es decir, nos interesa que no se tenga en cuenta mayúsculas/minúsculas en la búsqueda.

Para conseguir esto la clase String dispone de, además del método equals, del método equalsIgnoreCase

Ejercicio 05 Tarea #2

Habitualmente las búsquedas no se realizan introduciendo exactamente el título, sino palabras del título. De esta forma se facilita al usuario la búsqueda. En esto va a consistir la tarea, en modificar el código de la tarea anterior para que dada una palabra o palabras del título introducidas por el usuario, el programa busque en el catálogo todos los libros que contengan esta palabra o palabras, devolviendo que títulos coinciden con el criterio de búsqueda.

En la siguiente imagen se muestra un ejemplo de ejecución del programa:

```
Bienvenido a nuestra biblioteca
Escribe palabra o palabras del título del libro que quieres buscar:
universo
La clave secreta del universo
Mi universo
```



Hint:

La clase String dispone de métodos que permiten comparar cadenas de caracteres de forma exacta (`equals`, `equalsIgnoreCase`, `compareTo`), pero también dispone de un método (`contains`) que permite saber si una palabra o palabras están contenidas en una cadena de texto.

Ejercicio 06: Reversed

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de bucles
- Uso de String
- Concatenación de strings
- Uso Java Documentation

Desarrolla un programa en Java que solicite al usuario un texto y lo muestre al revés. En la siguiente imagen se muestra un ejemplo de ejecución del programa:

Introduce un texto y pulsa enter:

Hello world

Texto invertido: dlrow olleH



Hint:

En este ejercicio es necesario acceder a cada carácter del texto original para poder invertir el orden. La clase String dispone del método `charAt(int index)`, que permite obtener un carácter dada una posición.

Ejercicio 07: Question

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de sentencias condicionales
- Uso de String
- Uso Java Documentation

Desarrolla un programa en Java que presente una pregunta al usuario con 4 posibles respuestas (a,b,c,d). Sólo una de las respuestas será correcta. Permite que el usuario seleccione la respuesta (a, b, c, d) que considera correcta e indícale si ha acertado o no. Si no introduce un valor válido (a, b, c, d) el programa termina. En las siguientes imágenes se muestran unos ejemplos de ejecución del programa:

Respuesta correcta	<p>Indica cual de las siguientes opciones no es un tipo primitivo en Java:</p> <p>a. int b. char c. boolean d. String</p> <p><i>d</i></p> <p>Respuesta correcta!</p>
Respuesta incorrecta	<p>Indica cual de las siguientes opciones no es un tipo primitivo en Java:</p> <p>a. int b. char c. boolean d. String</p> <p><i>a</i></p> <p>Respuesta incorrecta, la respuesta correcta es: d</p>
Respuesta no válida	<p>Indica cual de las siguientes opciones no es un tipo primitivo en Java:</p> <p>a. int b. char c. boolean d. String</p> <p><i>f</i></p> <p>Respuesta no válida: a, b, c, d</p>

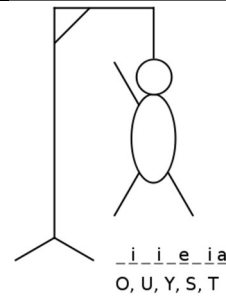
Ejercicio 08: Mini ahorcado

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de sentencias condicionales
- Uso de bucles
- Uso de String
- Uso Java Documentation

El juego del ahorcado es un juego de adivinanzas de uno o más jugadores donde un jugador piensa en una palabra o frase y el resto trata de adivinarla sugiriendo letras. Si la letra que sugiere no está en la palabra o frase, se dibuja un elemento del "ahorcado".

El juego termina bien porque se ha acertado la palabra o frase o bien porque se ha dibujado por completo el "ahorcado"



Ejercicio 08 Tarea #1

Desarrolla un programa en Java que disponga de la palabra o frase a adivinar, y muestre por consola tantos * como letras contenga la palabra o frase. Si la palabra o frase contiene un espacio en blanco no se dibujará * sino que se mantendrá el espacio en blanco. Como pista siempre daremos el primer carácter de la palabra o frase.

En la siguiente imagen se muestra uno ejemplo de ejecución del programa (la película es Forrest Gump):

```
Bienvenido al Mini juego del ahorcado
Intenta adivinar la pelicula
F***** ****
```

Ejercicio 08 Tarea #2

Permite ahora que el usuario sugiera una letra. Si está en la palabra o frase a adivinar muéstrala en la secuencia. En la siguiente imagen se muestra uno ejemplo de ejecución del programa:

```
Bienvenido al Mini juego del ahorcado
Intenta adivinar la pelicula
F***** ****
Introduce una letra:
r
F*r**** ****
```

Ejercicio 08 Tarea #3

Amplia el programa para que el usuario tenga 3 intentos (pueda llegar a sugerir 3 letras).

Ejercicio 09: Counter

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de sentencias condicionales
- Uso de bucles
- Uso de String
- Uso Java Documentation

Desarrolla un programa en Java que permita al usuario introducir un texto y muestre:

- Cuantas palabras tiene. Para simplificar el ejercicio, el texto termina con un punto y la separación entre palabras es un espacio en blanco
- Cuantas vocales tiene



Hint:

Consulta la documentación de Java de la clase String: métodos split, indexOf y lastIndexOf

Ejercicio 10: Classroom

Los objetivos de este ejercicio son:

- Uso de la línea de comandos para interactuar con el usuario
- Uso de bucles
- Uso de sentencias condicionales
- Uso de arrays de strings

Desarrolla un programa en Java que declare e inicialice un array con los nombres de los alumnos de una clase. El tamaño del array y su contenido lo escoges tú.

Ejercicio 10 Tarea #1

Muestra por consola los nombres de la clase (=contenido del array) en una misma línea separados por comas.



Hint:

Observa un detalle. Si en el array tenemos los siguientes nombres: Paula , Santiago , Pascual , Jorge , Saul , Enrique. Lo que queremos que se imprima es esto y no

Paula , Santiago , Pascual , Jorge , Saul , Enrique ,. La diferencia está en no imprimir la última coma. Para evitar esto tendrás que hacer uso de la sentencia IF.

Ejercicio 10 Tarea #2

Haz que el programa solicite al usuario qué posición quiere cambiar.

- Si la posición existe (por ejemplo en un array de 5 elementos son válidas las posiciones 0...4) mostrará al usuario su valor (el nombre) y solicitará al usuario el nuevo nombre. Cambiará el nombre antiguo por el nuevo y mostrará el contenido del array.
- Si la posición no existe (por ejemplo en un array de 5 elementos no es válida la posición 6), indicará al usuario que la posición entrada no existe.