

UT.2.3. EJERCICIOS MÉTODOS

Ejercicio 02 – 41

Crea un método en Java que muestre por consola tu nombre y apellidos. El método no tiene que recibir parámetros ni devolver ningún valor. Tu nombre y apellidos tienen que estar en una constante de la clase (del programa).

Escribe el programa principal (método main) para que llame al método que has creado para mostrar tu nombre y apellidos.

Ejercicio 02 – 42

Crea un método en Java que reciba dos números (con decimales) y muestre en la consola la suma de los dos números. El método tiene que llamarse “suma”, recibirá dos parámetros y no devuelve ningún valor.

Prueba el método invocándolo desde el método main.

Ejercicio 02 – 43

Crea un método “suma” en Java que reciba dos números enteros y devuelva la suma de los dos números. Este método, a diferencia del método del problema anterior, sí que devuelve el valor de la suma, y no muestra nada por pantalla.

Prueba el método invocándolo desde el método main.

Ejercicio 02 – 44

Crea un método “mostrarNVeces” que reciba un mensaje, que será una cadena de texto (String) y un número entero.

Si el número es menor que cero, mostrará en pantalla un mensaje indicando que es un error. Si el número es mayor que cero mostrará el mensaje tantas veces como indique el parámetro entero, una vez en cada línea.

Ejercicio 02 – 45

Crea un método “factorial” que reciba un número entero (n) y devuelva su factorial (n!). El método recibirá un int pero devolverá un long, porque los factoriales pueden ser números muy grandes.

El factorial sólo se puede calcular para números negativos, y se define como la multiplicación de un número por todos los que son inferiores a él, excepto el cero.

$3! = 3 \times 2 \times 1 = 6$ $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ $10! = 10 \times 9 \times \dots \times 2 \times 1 = 3628800$

El factorial de cero es 1.

Ejercicio 02 – 46

Crea un método que devuelva el número de cifras que tiene un número entero positivo, teniendo en cuenta los siguientes requisitos:

- El método recibe un int y devuelve también un int.
- El método tiene que controlar si el número es positivo. Si no lo es devolverá cero.
- Para calcular el número de cifras tiene que, obligatoriamente, convertir el número recibido a un objeto de tipo String y usar métodos del objeto String para calcular el número de dígitos.

En el método main, haz un programa que pida un número al usuario y, usando este método muestre el número de dígitos, o un mensaje de error si el número es negativo. Ojo, no se puede hacer comparación del número pedido con cero en main, tenemos que evaluar lo que devuelve el método creado para controlar el posible error.

Ejercicio 02 – 47

Repite el ejercicio anterior, pero esta vez no uses la clase String en el método. Usa divisiones enteras para saber cuántas veces puedes dividir un número por 10.

Ejercicio 02 – 48

Crea un método que reciba un número entero mayor o igual a cero y devuelva una cadena (String) que represente el número en binario.

Para convertir el número a binario tienes que hacer una serie de divisiones sucesivas por 2 (la base del sistema binario). Ejemplo: transformar el número decimal 131 en binario:

- 131 dividido entre 2 da 65 con resto igual a 1
- 65 dividido entre 2 da 32 con resto igual a 1
- 32 dividido entre 2 da 16 con resto igual a 0
- 16 dividido entre 2 da 8 con resto igual a 0
- 8 dividido entre 2 da 4 con resto igual a 0
- 4 dividido entre 2 da 2 con resto igual a 0
- 2 dividido entre 2 da 1 con resto igual a 0
- el último cociente es 1
- Se ordenan los restos, del último al primero: 10000011. En sistema binario, 131 se escribe 10000011.

Ejercicio 02 – 49

Ya hay métodos en la API de Java para hacer lo que hace el método del ejercicio 48. Y mejor, porque funciona con números negativos. Busca dos formas de hacer lo

mismo, usando métodos de la API de Java, y comprueba si el resultado es el mismo.

Ejercicio 02 – 50

Crea una aplicación que nos calcule el área de un círculo, cuadrado o un triángulo. Esta aplicación:

- Mostrará un menú con cuatro opciones:
 - 1 – Círculo
 - 2 – Cuadrado
 - 3 – Triángulo
 - 0 – Salir
- Si se elige 0, el programa terminará.
- Si se elige 1, el programa pedirá al usuario el radio del círculo, calculará su área y mostrará el resultado.
- Si se elige 2, el programa pedirá al usuario el lado del cuadrado, calculará su área y mostrará el resultado.
- Si se elige 3, el programa pedirá al usuario la base y la altura del triángulo, calculará su área y mostrará el resultado.
- Seguirá preguntando hasta que el usuario introduzca un cero.

Hay que tener en cuenta que:

- Deben utilizarse métodos. Como mínimo (pueden crearse más):
 - Método para mostrar el menú
 - Método para calcular el área de un círculo: $A = (R^2) \cdot \pi$
 - Método para calcular el área de un cuadrado: $A = L^2$;
 - Método para calcular el área de un triángulo: $A = (B \cdot H) / 2$

Ejercicio 02 – 51

Crea un método “pedirNumero” que:

- Recibe cuatro parámetros:
 - Un objeto Scanner, para poder interactuar con el usuario.
 - Un mensaje, que será la pregunta que se hará al usuario cada vez que tenga que introducir el número.
 - Un entero, límite superior inclusivo.
 - Un entero, límite superior exclusivo.
- Devuelve un número entero, que siempre estará entre el límite inferior (incluido) y el límite superior (excluido).

Este método realizará el siguiente proceso:

- Si los valores de los límites no son correctos (está mal si el inferior es mayor o igual al superior), muestra un mensaje de error y devuelve cero.
- Pedirá al usuario, usando el mensaje que ha recibido como parámetro, que introduzca un número.

- Si el número no está entre el límite inferior (incluido) y el exterior (excluido), mostrará un mensaje de error del tipo “El número introducido no está entre xxx (incluido) y zzz (excluido), y volverá a preguntar, hasta que el usuario introduzca el valor correcto.
- Devolverá el valor correcto introducido por el usuario.

Haz un pequeño programa principal con pruebas para el caso en que los límites sean incorrectos y para el caso en que sean correctos.

Ejercicio 02 – 52

Crea un método “pedirConfirmacion” que:

- Recibe dos parámetros:
 - Un objeto Scanner, para poder interactuar con el usuario.
 - Un mensaje, que será la pregunta a la que el usuario tiene que responder “S” o “N”.
- Devuelve un boolean, true si ha respondido “S” y false si ha respondido “N”

Este método realizará el siguiente proceso:

- Pedirá al usuario, usando el mensaje que ha recibido como parámetro, que responda a la pregunta.
- Si el usuario responde con cualquier cosa distinta a “S”, “s”, “N” o “n”, mostrará un mensaje de error del tipo “El valor introducido no es válido. Sólo puede responder S/N”.
- Devolverá el valor adecuado en función de lo que haya introducido el usuario.

Haz un pequeño programa principal para poder probar el método.

Ejercicio 02 – 53

Crea un método “mostrarTablaMultiplicar” que:

- Recibe un parámetro “número”.
- No devuelve ningún valor.

Mostrará en la consola la tabla de multiplicar del número (multiplicado del 1 al 10)

Utilizando este método y el método del ejercicio 51, crea un programa principal que:

- Pida al usuario que introduzca valores entre 0 y 10, ambos incluidos, usando el método del ejercicio 51.
- Si el valor es mayor que cero, muestra la tabla de multiplicar (usando el método previamente descrito) y vuelve a preguntar.
- Cuando el usuario introduzca un cero, el programa termina