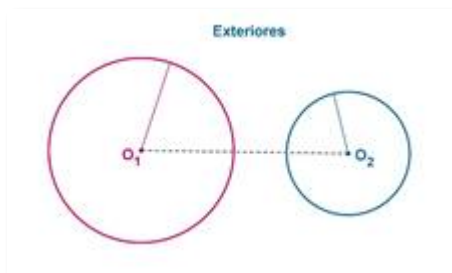


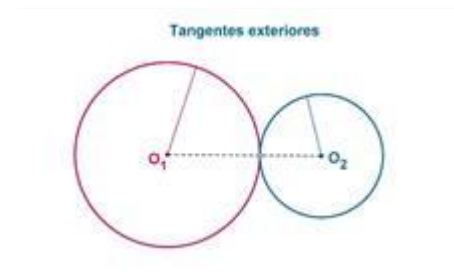
# Ejercicios Java

## Unidad 2: Sentencias de control de Flujo y subprogramas.

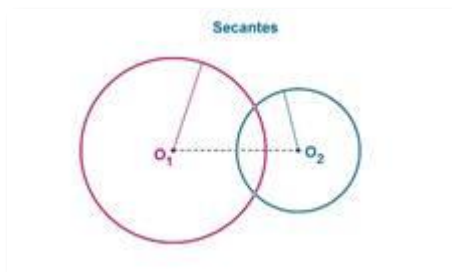
1. Escribe un programa que calcule la división exacta de dos números introducidos por teclado. Si el divisor es cero, se abortará el proceso mostrando un mensaje de error.
2. Escribe un programa que lea por teclado las coordenadas en el plano de los centros de dos circunferencias y la longitud de sus respectivos radios. Finalmente, se mostrará en la consola si ambas son exteriores, tangentes exteriores, secantes, tangentes interiores, interiores o concéntricas. Cada caso se determina de la forma siguiente:



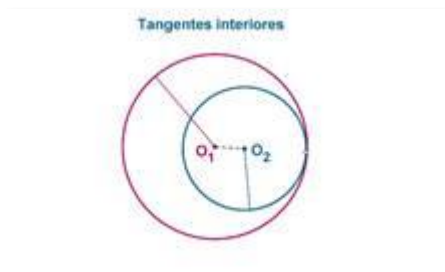
Distancia entre sus centros mayor que la suma de sus radios. Ningún punto en común.



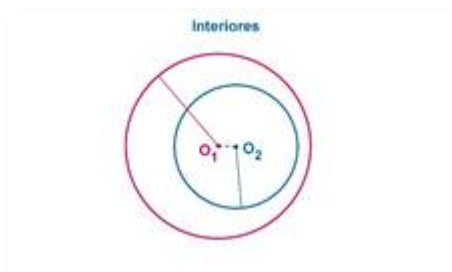
Distancia entre sus centros igual a la suma de sus radios. Un punto en común.



Distancia entre sus centros menor que la suma de sus radios y mayor que su diferencia. Dos puntos en común.



Distancia entre sus centros igual a la diferencia de sus radios. Un punto en común.



Distancia entre sus centros menor que la diferencia de sus radios. Ningún punto en común.



Distancia entre sus centros igual a cero (coinciden). Ningún punto en común.

3. Crea un método que acepte un número entero y retorne true si es par o false si es impar.

Finalmente, escribe un programa que lo ponga a prueba.

4. Escribe un programa que pida por teclado un número de mes y muestre el número de días que tiene. Si el usuario no introduce un valor entre 1 y 12, se mostrará un mensaje de error.

5. Escribe un programa que pida por teclado un nombre de un mes y muestre el número de días que tiene. Si el nombre introducido no es correcto, se mostrará un mensaje de error.

6. Crea un método que acepte un número entero entre 1 y 10 y retorne el resultado de convertirlo a un número romano.

Finalmente, escribe un programa que lo ponga a prueba.

7. Crea un método que acepte tres números enteros que representen la hora, minutos y segundos de un instante cualquiera. A partir de estos datos deberá retornar una cadena que contenga la hora correspondiente al segundo siguiente con el formato "HH:MM:SS".

Finalmente, escribe un programa que lo ponga a prueba, por ejemplo, mostrando un reloj digital que se actualice cada segundo comenzando desde la hora en la que se inicia su ejecución.

8. Crea los métodos siguientes:

- Método que clasifique un triángulo a partir de la longitud de sus lados como equilátero, isósceles, escaleno o imposible (ver teorema de la desigualdad del triángulo).
- Método que clasifique un triángulo a partir del valor de sus ángulos interiores como acutángulo, rectángulo, obtusángulo o imposible (la suma de sus ángulos interiores es distinta de  $180^\circ$ ).

Busca en Internet la información que te permita resolver el problema.

Finalmente, escribe un programa para poner a prueba estos métodos mostrando un menú donde el usuario pueda elegir entre uno de los dos tipos de clasificación de triángulos. Una vez hecha la elección, el programa le pedirá por teclado los datos necesarios y mostrará el resultado. Para que este proceso se pueda repetir indefinidamente, el menú deberá incluir una opción que permita finalizar el programa.

9. Crea un método que acepte un número entero que represente un día de la semana y retorne una cadena que represente el nombre de ese día. Si el método recibe un número que no representa un día de la semana, retornará un valor que indique el error.

A continuación, escribe un programa que haga uso de este método para permitir que un usuario obtenga nombres de días de la semana introduciendo por teclado los números correspondientes. Si introduce un número no válido, el programa deberá mostrar el número de datos erróneos introducidos hasta ese momento.

10. Crea un método que acepte dos caracteres cualesquiera y muestre en la consola todos los caracteres comprendidos entre ambos.

A continuación, escribe un programa que lo ponga a prueba.

11. Escribe un programa que genere números aleatorios y los muestre en la consola hasta que encontrar un número par. Éste último no se mostrará en la consola, pero antes de que finalice el programa se mostrarán los mensajes siguientes:
- “se han encontrado  $n$  números impares” si se han encontrado 2 o más números impares, sustituyendo  $n$  por la cantidad que corresponda.
  - “se ha encontrado 1 número impar” si sólo se ha encontrado un número impar.
  - “no se han encontrado números impares” si no se ha encontrado ningún número impar.
12. Escribe un programa que genere números enteros aleatorios y los vaya sumando hasta que dicha suma supere un valor previamente introducido por teclado.
13. Escribe un programa que genere mil millones de números aleatorios de tipo `long` y muestre cuántos pertenecen al intervalo  $[-1.000.000.000.000.000, 1.000.000.000.000.000]$  y del tiempo empleado en obtener dicho resultado.

14. Escribe un programa que lea las edades de varios alumnos y calcule los datos siguientes:

- La suma de todas las edades.
- La edad media.
- El número de alumnos menores de edad.

Los resultados se mostrarán después de introducir el último dato y pulsar la combinación de teclas ctrl+d.

15. Crea un método llamado `esPar` que acepte un número entero y retorne `true` si ese número es par y `false` si es impar. Escribe un programa que utilice este método para calcular el producto de los doscientos mil primeros números impares.

16. Crea un método que acepte un número entero  $n$  y retorne la suma de la serie  $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$ . Crea un programa que ponga a prueba este método con diferentes valores de  $n$  leídos por teclado y finalice cuando el usuario introduzca un número menor o igual que cero.

17. Escribe un programa para jugar al juego de adivinar el número desconocido:

- El programa genera un número aleatorio comprendido entre 1 y  $N$ , siendo  $N$  otro número aleatorio comprendido entre 1.000 y 100.000.
- Después muestra el mensaje “He pensado un número entre 1 y  $N$ , adivina cuál es: “. El usuario tendrá que introducir números hasta que acierte.

- Por cada respuesta no acertada el programa dirá si el valor introducido es mayor o menor que la respuesta correcta y volverá a preguntar.
- Cuando el usuario responda correctamente, el programa dirá cuántos intentos han sido necesarios para adivinar el número le preguntará al usuario si desea volver a jugar.

18. Escribe un programa que ayude al aprendizaje de las tablas de multiplicar de la forma siguiente:

- El programa preguntará qué tabla se desea repasar (número entre 1 y 9).
- A continuación, preguntará el resultado de cada multiplicación de la tabla elegida, contabilizando cada fallo y mostrando la respuesta correcta.
- Al completar la tabla, el programa le dirá al usuario si ha aprobado (menos de 2 fallos) o ha suspendido (2 o más fallos).
- Después dará la opción de repasar otra tabla o salir del programa.

19. Escribe un programa que genere el triángulo de Floyd para un número de filas introducido por teclado y lo muestre por pantalla. Por ejemplo, si el número de filas es 4 el triángulo de Floyd es:

```

1
2 3
4 5 6
7 8 9 10

```

20. Escribe un programa que pida un número  $n$  por teclado y dibuje en la consola triángulos usando asteriscos. Por ejemplo, para  $n=4$  se dibujarán los triángulos siguientes:

```

****      ****      *          *          *          *
***        ***        **         **         ***        *****
**          **         ***        ***        *****        ***
*            *         ****       ****       ****       *

```

21. Crea un método que calcule el  $n$ -ésimo término de la serie de Fibonacci de forma iterativa y también con la fórmula explícita. Busca en Internet la información necesaria.

22. Escribe un programa que simule el lanzamiento de un dado  $N$  veces, siendo  $N$  un número entero que se pedirá por teclado. Después de simular los lanzamientos, el programa mostrará cuantas veces salió cada una de las caras y le dará al usuario la opción de repetir el proceso.

23. Crea un método que acepte las coordenadas de dos puntos en el espacio de dos dimensiones y retorne la distancia entre ambos. Escribe un programa que utilice este método para calcular tantas distancias como desee el usuario, dando la opción a salir del programa después de mostrar cada resultado.