Mini Ejercicios Java

A continuación, se presenta una colección de pequeños ejercicios de java, para practicar distintos conceptos que hemos ido viendo en clase.

1. Hola Mundo:

- Escribe un programa que imprima "Hola, mundo" en la consola.
- Modifica el programa para que solicite al usuario su nombre y salude personalmente (ej. "Hola, Ana").
- Expande el programa para que después de saludar, pregunte al usuario cómo está y responda a su respuesta.
- Agrega una función que repita el saludo tres veces utilizando un bucle `while`.

2. Variables y Tipos de Datos:

- Declara e inicializa variables de diferentes tipos (int, double, String).
- Imprime estas variables.
- Realiza y muestra conversiones entre tipos de datos (casting).
- Escribe un programa que calcule y muestre el área de un rectángulo usando variables para los lados.
- Crea un programa que convierta grados Fahrenheit a Celsius y viceversa.

3. Operaciones Básicas:

- Realiza y muestra el resultado de operaciones básicas (suma, resta, multiplicación, división).
- Añade operaciones de módulo y elevación a potencia.
- Crea un calculador simple que acepte entradas del usuario y realice estas operaciones.
- Implementa un sistema simple para calcular el promedio de cinco números ingresados por el usuario.

4. Estructuras de Control:

- Escribe un programa que use `if` para comparar dos números e imprimir cuál es mayor o si son iguales.
- Implementa un bucle `for` que imprima números del 1 al 10.
- Usa un bucle `while` para realizar una cuenta regresiva desde 10 hasta 1.
- Desarrolla un programa que pida al usuario un número y utilice un bucle `do-while` para sumar todos los números hasta ese número.
- Agrega un ejemplo que use `switch` para realizar diferentes operaciones matemáticas basadas en la entrada del usuario.

5. Array y Bucles:

- Crea un array de enteros y usa un bucle para sumar todos los valores del array.
- Modifica el array anterior para que imprima todos los valores al revés.
- Utiliza un bucle `for-each` para buscar el mayor número en el array.
- Desarrolla un programa que compare dos arrays de enteros y determine si son idénticos.
- Implementa un programa que encuentre el número más pequeño y el más grande en un array.

6. Funciones y Métodos:

- Escribe un método que tome dos parámetros, los sume y devuelva el resultado.
- Crea un método que reciba un array y devuelva el elemento más grande.
- Desarrolla un método que verifique si un número es par o impar.
- Implementa un método que invierta una cadena de texto y la retorne.
- Añade un método que reciba un string y devuelva el mismo string sin vocales.
- Escribe un método `static` que calcule el factorial de un número.

7. Clases y Objetos:

- Crea una clase `Persona` con propiedades como nombre y edad, e incluye métodos para cambiar y obtener estos valores, usando modificadores `private` para las variables y `public` para los métodos.
- Implementa un método que imprima todos los datos de un objeto `Persona`.
- Añade un constructor que inicialice todos los atributos de `Persona`.
- Escribe un método dentro de la clase que describa si la persona es mayor de edad.
- Diseña un método en `Persona` que compare la edad de dos personas y determine cuál es mayor.

8. Herencia y Clases Abstractas:

- Extiende la clase `Persona` para crear una nueva clase `Estudiante` que tenga grado y especialización como atributos adicionales.
- Sobrescribe un método de la clase padre en la clase hijo y demuestra su uso.
- Añade un método específico para `Estudiante` que muestre su información académica.
- Crea una clase abstracta `Animal` con un método abstracto `hacerSonido`.
- Implementa clases `Perro` y `Gato` que hereden de `Animal` y definan el método `hacerSonido`.

9. Interfaces y Polimorfismo:

- Define una interface `Vehículo` con métodos como `acelerar` y `frenar`.
- Crea clases `Coche` y `Bicicleta` que implementen `Vehículo`.
- Demuestra el uso de polimorfismo con una función que acepte `Vehículo` y ejecute los métodos `acelerar` y `frenar`.
- Utiliza interfaces para implementar múltiples herencias en Java.

10. Excepciones y Control de Errores:

- Escribe un programa que maneje posibles errores de entrada/salida utilizando bloques `try-catch`.
- Crea tu propia excepción y úsala en una situación controlada.
- Implementa manejo de excepciones en métodos que puedan lanzar un error de división por cero.
- Desarrolla un método que lance una excepción personalizada si se intenta añadir un objeto `null` a una colección personalizada.
- Escribe un programa que capture múltiples tipos de excepciones y las maneje de manera específica.

11. Uso de `final`, `static` y `abstract`:

a) Uso de `final`: Crea una clase `ConstantesMatematicas` con un campo `final` que almacene el valor de PI. Intenta modificar este valor en el código y observa qué sucede.

b) Uso de `static`: Crea una clase `Contador` que tenga un campo `static` que cuente cuántas instancias de la clase se han creado. Implementa un método `static` que devuelva el valor del contador.

c) Uso de clases `abstract`: Crea una clase abstracta `Figura` con un método abstracto `calcularArea`. Luego, crea las clases `Circulo` y `Rectangulo` que extiendan de `Figura` e implementen el método para calcular el área.

12. Árbol Binario de Búsqueda:

Implementa una clase `ArbolBinarioBusqueda` que permita insertar valores y realizar un recorrido en orden del árbol binario de búsqueda. Añade métodos para insertar un valor y recorrer el árbol en orden.

13. Algoritmo de Grafos (Búsqueda en Profundidad - DFS):

Crea una clase `Grafo` que represente un grafo mediante listas de adyacencia. Implementa el algoritmo de búsqueda en profundidad (DFS) para recorrer el grafo desde un nodo inicial dado.