UNIVERSIDAD LA SALLE CHIHUAHUA

**Ecuaciones diferenciales**Image

**Guia de estudio primer parcial**

Ing. Jorge Luis Romo

Nombre del alumno: César Chavira Luna

Matricula: 12366

Fecha de entrega: 2/23/25

1. Una clase en Java es un modelo o plantilla que define atributos y métodos que los objetos pueden tener. Es fundamental en la programación orientada a objetos porque permite la reutilización de código.
2. Una instancia de una clase  
   * UserData objUserData = new UserData();

String fullName = objUserData.fullName("Cesar", "Chavira");

System.out.println(fullName);

1. public static void main(String[] args) es el punto de entrada de un programa Java. public indica que es accesible desde cualquier parte, static permite ejecutarlo sin instanciar la clase, void significa que no devuelve valor y String[] args recibe argumentos de la línea de comandos.
2. Declaración de variables en Java:
   * int numero = 10;
   * double decimal = 5.5;
   * String texto = "Hola";
   * boolean booleano = true; = false;
3. int[] numeros = new int[2];
4. Recorrido de arreglo de enteros:
   * double[] sales = {100.0, 200.0, 150.0};
   * for (int i = 0; i < sales.length; i++) {
   * double sale = sales[i] \* .9;
   * System.out.println(sale);
   * salessum += sale;
   * }
5. Con el operador %
6. int[][] arreglo = new int[2][4];
7. Con Scanner:
   * Scanner objScanner = new Scanner(System.in);
   * System.out.print("Ingresa el radio del circulo: ");
   * double radius = objScanner.nextDouble();
   * objScanner.close();
8. PascalCase
9. camelCase
10. Forma de organizar y almacenar datos para su acceso.
11. Alamcena datos del mismo tipo en posiciones contiguas de memoria.
12. Permiten manejar grandes volúmenes de información de manera eficiente.
13. Editor de código con varias herramientas como depuradores para facilitar el desarrollo de software.
14. Netbeans, VS Code, Eclipse, InteliJ IDEA
15. .length
16. Arreglo dinámica que permite cambiar su tamaño. A diferencia de un arreglo sencillo, puede crecer y reducirse automáticamente.
17. El stack almacena variables locales y llamadas a funciones, mientras que el heap almacena objetos creados dinámicamente.
18. Secuencia contigua de celdas
    * // Representación en memoria:
    * // Dirección 100 -> 10
    * // Dirección 104 -> 20
    * // Dirección 108 -> 30
19. Si, excepto ara estructuras de control como if o for.
20. Función con parámetros de retorno:
    * public class UserData {
    * public String fullName(String firstName, String lastName) {
    * return String.format("%s %s", firstName, lastName);
    * }
    * }
21. Ejemplo de modficadores de arraylist:
    * import java.util.ArrayList;
    * ArrayList<String> lista = new ArrayList<>();
    * lista.add("Elemento 1"); // Agregar
    * lista.remove(0); // Eliminar
    * lista.set(0, "Nuevo valor"); // Modificar
    * int tamano = lista.size(); // Obtener tamaño
22. Ejercicios
    * Ejercicio 1: <https://gist.github.com/CesarCHL/2db917ff6f2365cc57fc51785205d6db>
    * Ejercicio 2: <https://gist.github.com/CesarCHL/2f6f47da64d14d3ebbfd24cc14584a65>
    * Ejercicio 3: https://gist.github.com/CesarCHL/a1325d821e2482839a3eb0dabd374cea
    * Ejercicio 4: https://gist.github.com/CesarCHL/65454b7b83aba8ae1f7b6e29aeb3c30e
    * Ejercicio 5: https://gist.github.com/CesarCHL/db128a956a5b20919f0adebea055b126