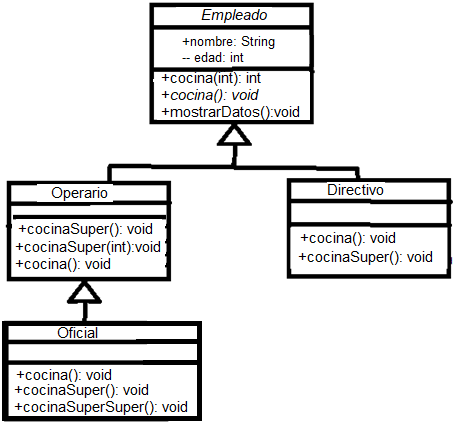
|  |  |
| --- | --- |
| Alumno: | Fecha de entrega: 7/4/2025 |

1. Realice un diagrama de clases que refleje el siguiente código JAVA (ingeniería inversa).

|  |  |
| --- | --- |
| public class A {  String a;  int b,c,d;    public A(String s, int t)  { a = s; b = t; c,d =0;}  public int sumar(int k) {  int auxi = b+c+d+k;  return auxi;  }  } | public class B extends A {  private char a, b, c;    public B(String h, int j, char m )  {  super(h,j);  a = m;  }  public void inicializaCarB(char b)  {  this.b = b;  }  public void muestraCars() {  System.out.println( a );  System.out.println( b );  System.out.println( c );  }  public int sumar(int k) {  int auxi = ((int)b+(int)c+ super.d+k);  return auxi;  }  } |

1. Respecto al diagrama anterior: ¿cómo se denomina el mecanismo utilizado con el método sumar? Compruebe si funcionan ambos correctamente.
2. ¿Cómo se denominaría el mecanismo, si se implementa el método sumar en la clase B, con otros tipos o cantidad de parámetros?
3. ¿Qué significado tienen las palabras reservadas this y super (sin uso de paréntesis?
4. Según el funcionamiento del método inicializaCarB¿Cuál sería el nombre adecuado para el mismo?
5. Falta definir algunos métodos, ¿Cuáles son?
6. ¿Cuál es el estado de un objeto?
7. ¿Qué son la ingeniería directa y la inversa?
8. ¿Cuál es la diferencia entre sobrecarga y redefinición?
9. Observe las siguientes sentencias y explique por qué no son posibles:
10. \* private static final int VALOR = 9;
11. \* public final double PI = 6;
12. \* public static moneda;

1. Teniendo el modelo:



Observe responda las siguientes preguntas:

1. ¿En qué clases está redefinido el método cocina()?
2. ¿En qué clases está sobrecargado?
3. ¿Qué otro método está redefinido y en qué clases?
4. ¿En qué métodos se utiliza “super()”?
5. ¿Para qué se usa “ super**.**”?
6. Desde la clase Directivo, ¿a qué atributos de la clase Oficial no se puede acceder directamente invocándolo con su nombre?
7. Indique la sentencia a utilizar para cada uno de los siguientes casos
   1. Desde un objeto de clase Directivo, activar el cocina() de Empleado.
   2. Desde un objeto de clase Oficial, activar el cocina(int) de Empleado.
   3. Desde el método main, creado en otra clase:
      1. Cambiar el valor del atributo edad, de un objeto de Oficial.
      2. Obtener el valor del atributo edad, de un objeto de la clase Operario.
   4. Observando las cabeceras de los métodos, indique:

|  |  |  |  |
| --- | --- | --- | --- |
| Clase a la que pertenece | Método | Utilidad que sugiere su cabecera | Métodos que activaría según su utilidad |
| Empleado | mostrarDatos() |  |  |
| Oficial | cocinaSuper() |  |  |
| Directivo | cocinaSuper() |  |  |
| Oficial | cocinaSuperSuper () |  |  |

1. Si se agregara en la clase Empleado, **la variable public static int recetas = 0**, responda V o F según corresponda. Para cada respuesta compruebe el funcionamiento, creando el código en el entorno:

|  |  |
| --- | --- |
| 1. Se accedería a ella desde objetos de otras clases (de la misma jerarquía o de otras) nombrando a la clase Empleado. |  |
| 1. Se podrá acceder desde objetos de cualquier clase, de esta jerarquía de clases. |  |
| 1. Aunque no se pueden crear objetos de la clase Empleado, se la puede nombrar para acceder a la variable. |  |
| 1. Un método definido en la clase Operario, podrá usar su valor sin nombrar a clases ni a objetos. |  |
| 1. Todas las clases de esta jerarquía utilizan como propia a la variable. |  |
| 1. Es correcto utilizar en el main: int a = Oficial.recetas \* 2; |  |