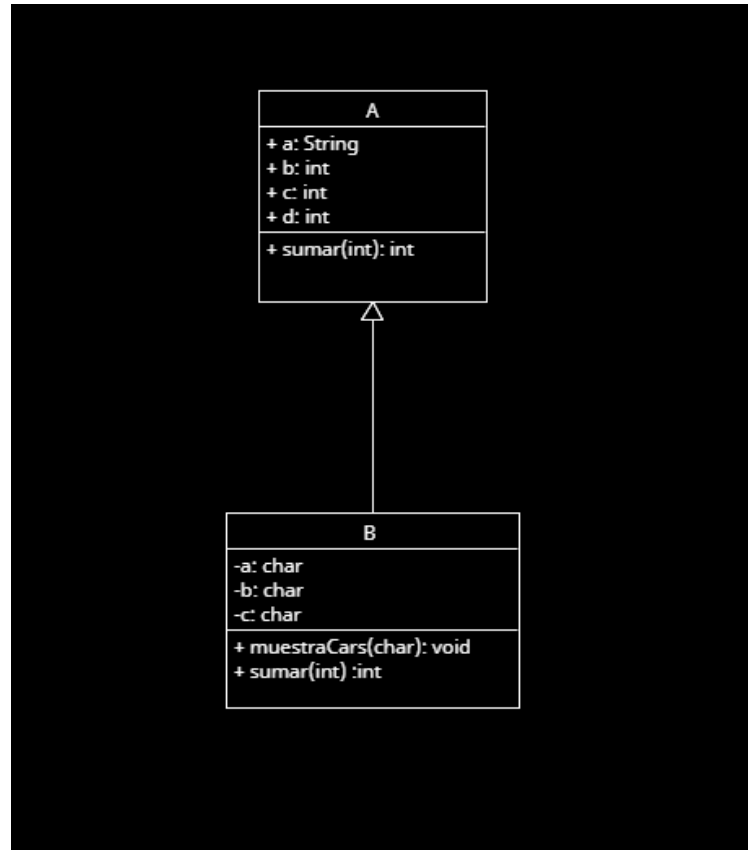


Trabajo Practico Numero 3 Java

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



Preguntas:

1. Se denomina Redefinición de métodos.
2. Se denominaría sobrecarga de métodos
3. **This:** Se refiere a la instancia actual de la clase. Se usa para acceder a atributos y métodos propios.
Super: Se refiere a la superclase. Se usa para llamar a métodos y constructores de la clase padre.
4. Un mejor nombre sería setCarB(char b), ya que establece (set) el valor de b.
5. Si hijo es una instancia de B, hereda todos los atributos y métodos de A y B. Aunque no están creados los constructores correspondientes para el funcionamiento del código

6. El estado de un objeto se define por los valores almacenados en sus atributos en un momento determinado.
7. **Ingeniería directa:** Diseñar un sistema y luego implementarlo.
8. **Ingeniería inversa:** Analizar código existente y extraer su diseño, como el diagrama de clases.
9. **Sobrecarga** es dentro de la misma clase y es cuando un método se reescribe (tienen el mismo nombre) en la misma clase y cambian la cantidad de parámetros.

Redefinición (Sobre-escritura) es entre distintas clases y es cuando un método se invoca en la clase hija o subclase desde la clase super o padre. Su función es la de reutilizar un método ya creado en diferentes clases, cambiando los valores necesarios.

B1) La primer sentencia es válida

B2) Falta el modificador static si se pretende que sea una constante.

B3) Falta el tipo de dato (String, double, int, etc)

Observe responda las siguientes preguntas:

1. El método cocina() está redefinido en las clases Operario, Directivo y Oficial.
2. Están sobrecargados en la clase Empleado el método cocina(int) que es una sobrecarga del método cocina(), y en la clase Operario el método cocinaSuper(int) que es una sobrecarga del método cocinaSuper().
3. cocinaSuper() está redefinido en las clases Operario, Directivo y Oficial.
4. Se usa en los métodos cocinaSuper(), cocinaSuper(int) y cocinaSuperSuper() de las clases Operario, Directivo y Oficial para llamar a la versión del método en la superclase
5. super. Se usa para invocar métodos o acceder a atributos de la superclase desde una subclase
6. Directivo no tiene relación directa con Oficial en la jerarquía de clases. Por lo tanto, no puede acceder a ningún atributo o método de Oficial
7. Sentencias para invocar métodos:
 - a. super.cocina()
 - b. super.cocina(5)
 - c. Desde el método main:
 - i. oficial.setEdad(22)
 - ii. int edad = operario.getEdad()

d.

Clase a la que pertenece	Método	Utilidad que sugiere su cabecera	Métodos que activaría según su utilidad
Empleado	mostrarDatos()	Muestra los datos del empleado (nombre, edad u otra información relevante).	getNombre(), getEdad()
Oficial	cocinaSuper()	Llama al método cocina() de la superclase (Empleado o Operario).	super.cocina() de la superclase.
Directivo	cocinaSuper()	Llama al método cocina() de la superclase (Empleado).	super.cocina() de Empleado.
Oficial	cocinaSuperSuper ()	Llama al método cocinaSuper() de su superclase (Operario), que a su vez puede llamar a cocina() de Empleado.	super.cocinaSuper() de Operario, que podría activar super.cocina() de Empleado.

e.

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