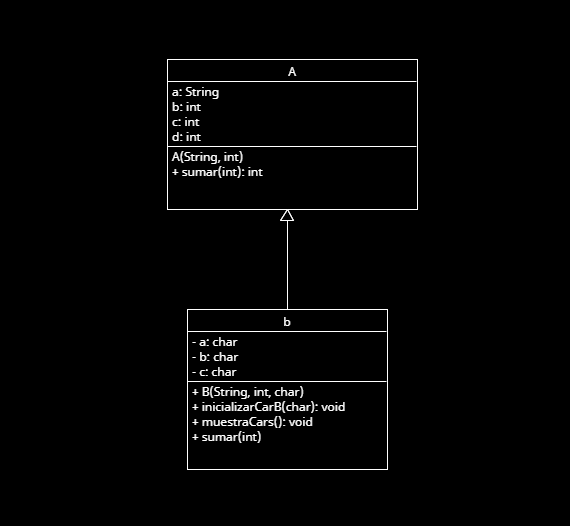
|  |  |
| --- | --- |
| Alumno: Cristian Krahulik, Tomas Mastropietro, Gianfranco Guzman, Alex Filchel, Lautaro Castillo | Fecha de entrega: 2/4/2024 |

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.

El mecanismo es redefinición o sobreescritura

1. ¿Cómo se denominaría el mecanismo, si se implementa el método sumar en la clase B, con otros tipos o cantidad de parámetros?

Ese otro mecanismo se llama sobrecarga

1. ¿Qué significado tienen las palabras reservadas this y super (sin uso de paréntesis?

this hace referencia al objeto que lo esta utilizando. Si se instancia un objeto, cuando se llame al constructor la palabra this hará referencia a ese objeto.

La palabra super hace referencia a la super clase de cual hereda la clase desde la que llamamos al super. Si la clase B hereda de la clase A, Cuando usemos la palabra super en la clase B, hará referencia a los métodos y atributos de la clase padre A.

1. Según el funcionamiento del método inicializaCarB¿Cuál sería el nombre adecuado para el mismo?

El nombre adecuado seria setB()

1. Falta definir algunos métodos, ¿Cuáles son?

Los getters y setters de a,b,c en la clase B

1. ¿Cuál es el estado de un objeto?

El estado de un objeto son los atributos que tiene

1. ¿Qué son la ingeniería directa y la inversa?

La ingeniería directa significa a partir de un esquema o diseño como el UML, pasarlo a código o implementar el diseño de alguna forma.

La inversa seria a partir del código analizar y realizar su esquema

1. ¿Cuál es la diferencia entre sobrecarga y redefinición?

La sobrecarga es escribir varios métodos con el método con la diferencia que uno recibe parámetros distintos, en distinto orden, o mas que el otro.

La redefinición proviene del polimorfismo. Una clase B hereda un método de la clase padre A, pero se puede redefinir para tener otra implementación. Cuando se llame a ese método desde un objeto creado con la clase B, se utilizara el método redefinido en la clase B y no el que hereda en A.

1. Observe las siguientes sentencias y explique por qué no son posibles:
2. \* private static final int VALOR = 9;

No veo error aca.

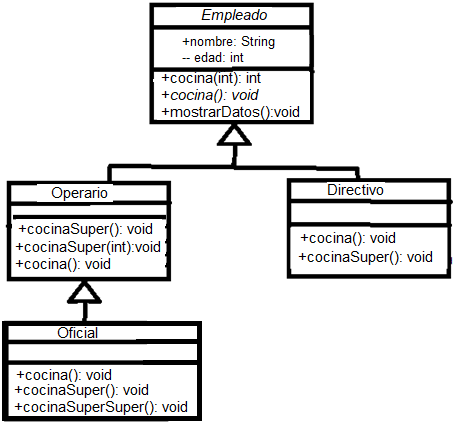
1. \* public final double PI = 6;

El único error es que PI vale 6 y no puede cambiar. Lo cual no tiene sentido

1. \* public static moneda;

Falta el tipo de dato

1. Teniendo el modelo:



Observe responda las siguientes preguntas:

1. ¿En qué clases está redefinido el método cocina()?

El método cocina() esta redefinido en las clases Operario, Directivo y Oficial.

1. ¿En qué clases está sobrecargado?

El método cocina esta sobrecargado en la clase Empleado.

1. ¿Qué otro método está redefinido y en qué clases?

El método cocinaSuper() esta redefinido en la clase Oficial.

1. ¿En qué métodos se utiliza “super()”?

En el constructor de Oficial, Operario y Directivo

1. ¿Para qué se usa “ super**.**”?

Para acceder a la clase padre. A sus atributos y métodos desde la subclase

1. Desde la clase Directivo, ¿a qué atributos de la clase Oficial no se puede acceder directamente invocándolo con su nombre?

No se puede acceder a el atributo edad porque es privado. Se necesita un getter.

1. Indique la sentencia a utilizar para cada uno de los siguientes casos
   1. Desde un objeto de clase Directivo, activar el cocina() de Empleado.

super.cocina();

* 1. Desde un objeto de clase Oficial, activar el cocina(int) de Empleado.

super.cocina(100);

* 1. Desde el método main, creado en otra clase:
     1. Cambiar el valor del atributo edad, de un objeto de Oficial.

Oficial of1 = new Oficial();

of1.setEdad(17);

* + 1. Obtener el valor del atributo edad, de un objeto de la clase Operario.

Operario op1 = new Operario();

op1.getEdad();

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() | Muestra los datos de empleado | System.out.println() |
| Operario | cocinaSuper() | Un método de cocina superior. Heredada de Empleado | Super.cocina() |
| Directivo | cocinaSuper() | Un método de cocina superior. Heredada de Empleado | La implementación que tenga |
| Oficial | cocinaSuperSuper () | Un método de cocina mas avanzado. | This.cocinaSuper() |

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. | V |
| 1. Se podrá acceder desde objetos de cualquier clase, de esta jerarquía de clases. | V |
| 1. Aunque no se pueden crear objetos de la clase Empleado, se la puede nombrar para acceder a la variable. | V |
| 1. Un método definido en la clase Operario, podrá usar su valor sin nombrar a clases ni a objetos. | F |
| 1. Todas las clases de esta jerarquía utilizan como propia a la variable. | V |
| 1. Es correcto utilizar en el main: int a = Oficial.recetas \* 2; | V. Es mala practica pero compila |