

Guia de Objetos III

modelado e implementación

Sobre esta guía

Lea atentamente cada uno de los enunciados y antes de llevar a código cada ejercicio realice en papel o con la ayuda de algún software, el correspondiente diagrama de UML, detallando los atributos, métodos y constructores que considere necesarios. Una vez terminado escriba el código y ejecute las pruebas que se detallan.

Enunciados

1. Para comenzar esta guía, vamos a empezar con un ejemplo sencillo, que nos permita entender el concepto de herencia, superclase y subclase. Para esto vamos a definir el tipo **Círculo**, que posee un radio cuyo valor por defecto al ser inicializado sin valores es 1.0. Además el tipo **Círculo** posee un atributo **color**, por defecto rojo, un método para calcular el área y otro para imprimir sus características. Puede reutilizar la clase **Círculo** de las guías anteriores.

Por otro lado tenemos un tipo **Cilindro**, que extiende a la clase **Círculo**, por ende se convierte en subclase de **Círculo**. El **Cilindro** a diferencia del **Círculo** posee un atributo **altura**, que también se inicializa en 1.0 cuando no le pasamos un valor a su constructor. Y un método para calcular el volumen. Codifique ambas clases y realice las siguientes pruebas.

- Inicializar un **Cilindro** y debuguear analizando los constructores a los que va llamando para inicializar la cadena de herencia. Imprimir por pantalla, el radio del cilindro, la altura, el área de la base y el volumen.
- Inicializar un segundo **Cilindro** esta vez especificando la altura y radio. Imprimir por pantalla el radio, la altura, el área de la base y el volumen.
- Sobrecribir el método **calcular area** declarado en **Círculo**, para que nos permita calcular el área del cilindro.

$(2\pi \times \text{radius} \times \text{height} + 2 \times \text{areaBase})$

Imprimir por pantalla el área y el volumen del cilindro. Vamos a notar que el cálculo del volumen difiere, esto es porque nosotros sobreescribimos el método **calcular area** y ahora nos calcula el área de un cilindro. Modificar el método **calcular volumen** en la clase **cilindro** para que llame al método de la SUPERclase que calcula el área.

- Modificar el método **toString()** de la clase **cilindro** para que imprima por pantalla lo siguiente

Cilindro: subclase de + {toString() de Círculo} + altura= {alturaCilindro};

2. Vamos a diseñar un programa que nos permita gestionar el personal que concurre a un colegio. Para no hacerlo muy extenso vamos a limitarnos a estudiantes y miembros de staff que a diferencia de los estudiantes, estos perciben una remuneración. Ambos tipos comparten la característica de Persona que posee los atributos de dni, nombre, apellido, email y direccion.

Por otro lado tenemos al Estudiante que posee las características de Persona y las de un estudiante, que en este caso son, año de ingreso, cuota mensual y carrera.

Y finalmente tenemos al miembro de Staff que también es una Persona pero con características propias de alguien que trabaja para una institución, por ejemplo salario y turno, este puede ser mañana o noche.

Diagramar el UML identificando superclase y subclasses, crear constructores necesarios, getters y setters, métodos de ayuda como salario anual y toString para facilitar la impresión.

- Inicializar 4 estudiantes diferentes.
- Inicializar 4 miembros de staff con características diferentes.
- Crear un array que permita almacenar juntos los tipos anteriores y almacenar las 8 instancias creadas anteriormente.
- Investigar el uso de la palabra reservada instanceof.
- Recorrer el array y mostrar por pantalla la cantidad de estudiantes y la cantidad de miembros de staff.
- Recorrer el array y sumar el total de ingresos que percibe la institución por parte de la cuota de estudiantes.

3. Necesitamos crear un programa que nos permita crear diferentes tipos de figuras que poseen un color, estas figuras pueden o no estar coloreadas. Además necesitamos que las figuras tengan la posibilidad de calcular area y perimetro. Por el momento nos piden crear 2 Figuras, Círculo y Rectángulo. ambos deben contar con sus respectivas características, ya sea radio, alto, largo, etc. Y deben contar con los métodos de cálculo de area y perimetro implementados.

- Considerar si existe la posibilidad de crear un tipo abstracto que defina las características en común y que permita establecer qué métodos son mandatorios para las subclasses.
- Los tipos deben contener diferentes constructores, por defecto y que inicialice diferentes atributos.
- Crear diferentes instancias de cada tipo e imprimir sus características.
- Nos piden agregar un nuevo tipo de forma, esta vez es un Cuadrado. Analice dónde es conveniente agregarlo en el árbol de herencia. Teniendo en cuenta evitar la duplicación de código innecesario.