# Ejercicios Tema 2. Programación Orientada a Objetos.

## Parte 1

1. Crear una clase que describa un rectángulo que se pueda visualizar en la pantalla del ordenador, cambiar de tamaño.

| Rectángulo |
| --- |
| - alto: float<br>- ancho: double |
| + getAncho(): double<br>+ setAncho(nuevo_ancho: double): void<br>+ getAlto(): float<br>+ setAlto(nuevo_alto: float): void<br>+ cambiarAltoAncho((nuevo_alto: float, nuevo_ancho: double): void |

2. Construir una clase llamada Luz que simule un semáforo. El atributo color de la clase debe cambiar de Verde a Amarillo y a Rojo y de nuevo regresar a Verde mediante la función cambioDeColor(). Cuando un objeto Luz se crea, su color inicial será Rojo.

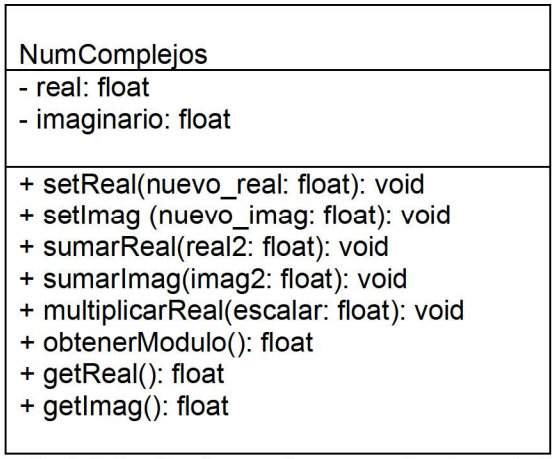
| Luz |
| --- |
| - color: String |
| + iniciarSemaforo(): void<br>+ cambioDeColor (): void<br>+ getColor(): String |

   Note que no es necesario que siempre exista un getter o setter si ya existe una función o método que se encargue de modificar los atributos internos.

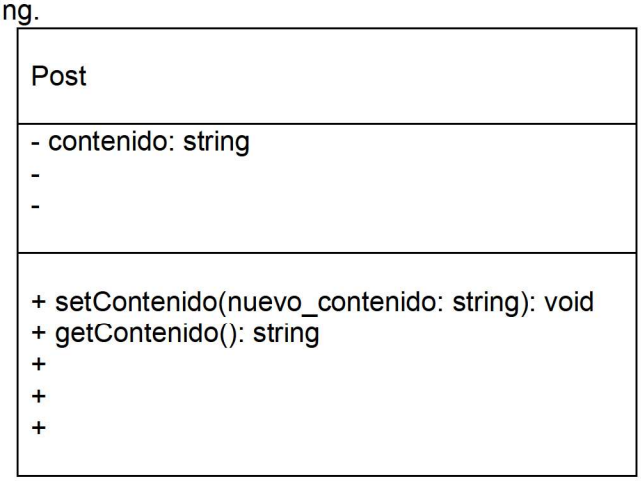
3. Representar una clase ascensor (elevador) que tenga las funciones usuales seleccionarPlanta(), abrirPuerta(), cerrarPuerta().

| Ascensor |
| --- |
| - plantaActual: int<br>- puertaAbierta: bool |
| + seleccionarPlanta(planta: int): void<br>+ abrirPuerta(): bool<br>+ cerrarPuerta(): bool<br>+ getPlantaActual(): int<br>+ isPuertaAbierta(): bool<br>+ isSobrecargado(): bool |

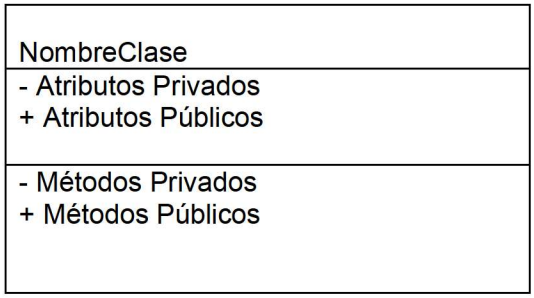
4. Son números complejos aquellos que contienen una parte real y una parte imaginaria. Los elementos esenciales de un número complejo son sus coordenadas y se pueden realizar con ellos numerosas operaciones, tales como sumar con otro entero, multiplicar con un escalar, obtener módulo, etc. Represente la clase NumComplejos.




5. Cree una clase a partir del diseño de un diagrama de clase de los datos de un post de una red social (TikTok/Twitter/Instagram/etc) con todas las características que puede ver en la app, por ejemplo para un post de facebook se tiene: fecha: string, contenido: string, likes: int, comentarios: string[], compartidos: int. El contenido siempre es un string.




6. Desarrolle un programa que modele el comportamiento de una ***Persona***. Esta persona viene definida por un **nombre**, y además, puede **dormir** (diría "zzz zzz zzz"), **hablar** (diría "bla bla bla"), **contarHasta** (diría "1 2 3 4 …"), recibir un nombre, e incluso decir su nombre. Modele e implemente la clase necesaria para resolver este programa. Defina un método principal donde se prueben los métodos creados. Recuerde que el diagrama de clase es de la siguiente forma:




7. Construir una definición de clase que se pueda utilizar para representar un **empleado** de una compañía. Cada empleado se define por un número entero **ID**, un

**salario** y el número máximo de **horas** de trabajo por semana. Los servicios que debe proporcionar la clase al menos deben permitir introducir datos de un nuevo empleado, visualizar los datos existentes de un nuevo empleado.

8. Construir una definición de clase que se pueda utilizar para representar un coche de una compañía. La clase coche contiene los atributos **color**, **motor** y **velocidadMáxima**, **encedido**. La clase puede agrupar las operaciones **arrancar**, **acelerar** y **frenar**. El coche no puede acelerar si no está **encendido**.
9. Implementar una clase *Hora* con 3 miembros enteros para la hora, el minuto y el segundo. Se debe incluir solamente funciones de acceso (getters), una función para imprimir la hora en pantalla mostrarHora(): void, una función reiniciar*(h: int, m: int, s: int)* para reiniciar la hora de un objeto existente. Además agregue 3 funciones tal que se puedan sumar segundos, minutos u horas a la hora (al objeto) actual:

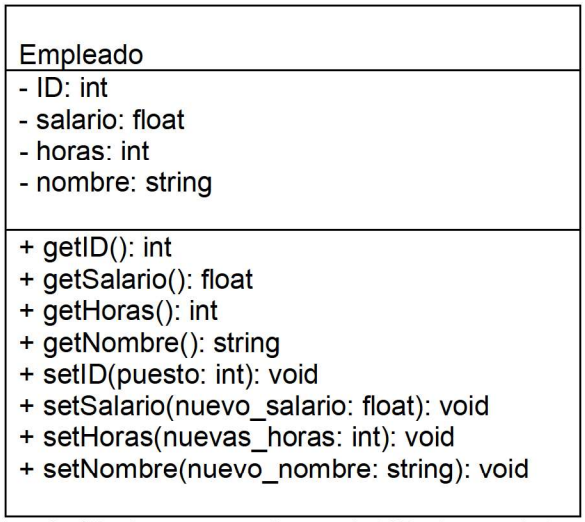
    + sumarHora(h: int)
    + sumarMinutos(m: int)
    + sumarSegundos(s: int)

Ejercicios de cara al examen:

1. Definir los siguientes términos: a) Clase, b) Objeto, c) Sección de declaración, d) Sección de implementación, e) Variable de instancia, f) Función miembro, g) Atributo o miembro dato, h) Niveles de Acceso.
2. Determinar los atributos y operaciones que pueden ser de interés para los siguientes objetos, partiendo de la base que van a ser elementos de un almacén de regalos: un libro, un disco, una grabadora de vídeo, una cinta de vídeo, un televisor, una radio, una tostadora de pan, una cadena de música, una calculadora y un teléfono celular (móvil). Seleccionar dos de ellos y crear sus diagramas de Clase.

## Parte 2

10. A partir de la clase rectángulo, en un programa principal instancie 2 objetos diferentes rectangulo1 y rectangulo2 y utilice cada una de las funciones diseñadas. Imprima en pantalla el área de los rectángulos. A continuación, añada una función para obtener el área dentro de la clase Rectángulo.
11. Escriba un programa principal que utilice la clase Luz para simular dos semáforos en un cruce. Muestre en pantalla cada vez que llame a la función cambioDeColor().
12. Utilice la clase Ascensor dentro de un programa que simule la utilización de 1 ascensor. Despliegue un menú con las opciones de acuerdo con el diagrama de la clase Ascensor y seleccione las distintas funciones miembros con el uso de un switch-case.
13. Mediante el uso de la clase NumComplejos, cree 3 números complejos: 2 + 3i, 1 + 4i y 3 + 5i. Cree un cuarto número complejo que es el doble de la suma de los tres números complejos. Nota: no se deben utilizar el operador + ni el operador * en el programa principal.
14. En una empresa, existen 5 encargados de despacho, 3 jefes de proyecto y 42 programadores. El número de ID de los encargados siempre debe empezar por 11 seguido de un conjunto aleatorio de 5 números (Ej.: 1112345), el de los jefes empieza por 555 seguido de 4 números aleatorios y el de los programadores por 1 seguido de 6 números aleatorios. Modifique la clase Empleado de tal forma que el diagrama sea el siguiente:




Note que el setter de ID ahora no recibe un int ID sino un int puesto. Puesto es un entero igual a 0 para Encargados, 1 para jefes y 2 para Programadores. Modifique la función setID(puesto: int) de tal forma a que se cumpla con el enunciado expuesto. Puede hacer uso de la función rand() de la librería cstdlib, la función entrega un número int que puede utilizarse en conjunto con el operador de resto.

Cree un programa principal para la empresa que tenga un total de 50 empleados (Empleado empleados[50]). A continuación, cree IDs para cada uno de los 5 primeros, que serán los encargados de despacho, cree otros 3 para los jefes y otros 42 para los programadores. Por último, imprima en pantalla todos los IDs que existen dentro de cada uno de los 40 empleados.

Ejercicios de cara al examen:

15. Cree un programa que permita saber cuánto tiempo ha estado en marcha un coche. Para ello debe utilizar dos clases creadas con anterioridad: Coche y Hora. El programa principal debe crear un objeto de la clase Coche y un objeto de la clase Hora. A continuación, debe realizar el siguiente proceso:
    - Encender el Coche y reiniciar la hora a la hora (12, 59, 55).
    - Acelerar el coche durante 5 segundos. Asumir que la aceleración del coche es 4 m/s^2.
    - Frenar el coche durante 10 segundos. Asumir que el freno produce una aceleración del coche de -2 m/s^2.
    En cada paso debe imprimir todos los atributos de la clase coche y la hora en que ha empezado dicho paso.