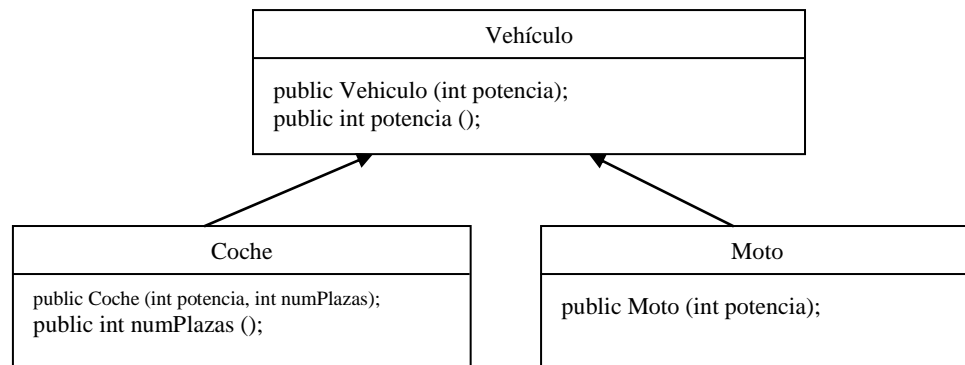


Ejercicios Clases

1. Definir una clase **PERSONA** que contenga información de propósito general común a todas las personas (nombre, dirección, fecha de nacimiento, sexo etc.). Diseñar una jerarquía de clases que contemple las siguientes clases: estudiante (nº expediente, nota, etc.) y empleado (NSS, sueldo etc.).
Escribir un programa que lea datos y los visualice, utilizando varias formas de cargar datos en la clase:
 - a. estudiantes
 - b. empleados
2. Implementar una jerarquía **EMPLEADO** de cualquier tipo de empresa que te sea familiar. La jerarquía debe tener al menos cuatro niveles, con herencia de miembros datos y métodos.
3. Construir una clase base genérica llamada edificio que almacene el número de plantas que tiene el edificio, el número de habitaciones y su superficie total. Crear una clase derivada llamada casa que herede edificio y que almacene también lo siguiente: el número de dormitorios y de baños. Crear también otra clase derivada llamada oficina que herede edificio y que almacene además el número de extintores y de teléfonos. Definir los atributos como privados, y los métodos como públicos.
4. Construir una jerarquía de clases que almacene información sobre aeronaves. Comenzar con una clase base general, llamada aeroplanos, que almacene el número de pasajeros que pueden ser transportados y la cantidad de carga que puede llevar. A continuación crear dos clases derivadas, avión y globos, a partir de aeronaves. Avión debe almacenar el tipo del motor empleado (propulsión o jet) y su rango, en millas. Globo debe almacenar información sobre el tipo de combustible utilizado para la ascensión del globo (hidrógeno o helio) y su altitud máxima (en pies). Construir un pequeño programa que muestre esta jerarquía de clases.
5. Implementar una clase **FECHA** con datos miembros para el día, mes y año. Cada objeto de esta clase representa una fecha, que almacena el día, mes y año como enteros. Se debe incluir un constructor por defecto, constructor parametrizado, funciones de acceso, un método *reiniciar* () que inicie los datos a una fecha concreta, por ejemplo 1-1-1900, un método *adelantar* (int días) que nos de la fecha dentro de una serie de días y un método *visualizar* (). Utilizar un método *validar* () que asegure que los datos miembros están en el rango correcto.
NOTA: Un año es bisiesto si es divisible por 400, o si es divisible por 4 pero no por 100.
6. Realizar una lista enlazada de números enteros.
7. Crear una clase **EMPLEADO** que modele la información que una empresa mantiene sobre cada empleado: número de DNI (entero largo), sueldo base (real), pago de horas extras (real), horas extras realizadas en el mes (real), tipo (entero) de IRPF, casado o no (verdadero o falso) y número de hijos (entero). Al crear cada objeto se podrá proporcionar el DNI correspondiente. Los demás servicios que deberá proporcionar los objetos de la clase serán los siguientes:
 - a. Cálculo y devolución del complemento correspondiente a las horas extras realizadas.
 - b. Cálculo y devolución del sueldo bruto.
 - c. Cálculo y devolución de las retenciones (IRPF) a partir del tipo, teniendo en cuenta que el porcentaje que hay que aplicar es el tipo menos 2 puntos si el empleado está casado y menos 1 punto por cada hijo que tenga; el porcentaje se aplica sobre el sueldo bruto.
 - d. Visualización de la información básica del empleado.
 - e. Visualización de toda la información del empleado, la básica más el sueldo base, el complemento por horas extras, el sueldo bruto, las retenciones de IRPF y el sueldo neto.

8. Crear una clase llamada hora que tenga miembros datos horas, minutos y segundos de tipo int. Un constructor inicializará este dato a 0, y otro lo inicializará a valores fijos. Un método miembro visualizará la hora en formato 11:59:59, Otro método miembro sumará dos objetos de tipo hora pasados como argumentos.
Una clase con el método principal main() crea dos objetos inicializados y uno que no está inicializado. Sumar los dos valores inicializados y dejar el resultado en el objeto no inicializado. Por último, visualizar el valor resultante.
9. Crear una clase llamada empleado que contenga como miembro dato el nombre y el número de empleado, y como métodos miembros *Leerdatos()* y *Verdatos()* que lean los datos del teclado y los visualice en pantalla, respectivamente.
Escribir un programa que utilice la clase, creando un array de tipo empleado y luego llenándolo con datos correspondientes a 5 empleados. Una vez rellenado el array, visualizar los datos de todos los empleados.
10. Dada la siguiente jerarquía de clase:



Diseñar una clase Garaje que:

- En el constructor se indica el número total de plazas de garaje.
- En cada plaza se pueden guardar tanto un coche como una moto.
- Tenga una función que devuelva la cuota mensual de una plaza:
 - Si en dicha plaza hay un coche, la cuota se calcula como la potencia multiplicada por el número de plazas.
 - Si en dicha plaza hay una moto, la cuota se calcula como la potencia multiplicada por 2.
 - Si no hay ningún vehículo en la plaza, la cuota es 0.