

Ejercicios Básicos de Programación Orientada a Objetos

1. Crea una clase Cuenta con los métodos ingreso, reintegro y transferencia. La clase contendrá un constructor por defecto, un constructor con parámetros, y los métodos para obtener información (los también llamados getters. Suelen tener la forma `get_xxxxx`) o para almacenar información (también llamados setters. Tienen la forma de `set_xxxx`).
2. Crea una clase Contador con los métodos para incrementar y decrementar el contador. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters.
3. Crea una clase Libro con los métodos préstamo, devolución y dame_info. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters que consideres oportunos.
4. Crea una clase Fraccion con métodos para sumar, restar, multiplicar y dividir fracciones.
5. Crea una clase Complejo con métodos para sumar, restar, multiplicar y dividir números complejos.
6. Crea una clase Fecha. La clase contendrá además de constructores, métodos set y get y el método para dar información, un método para comprobar si la fecha es correcta y otro para modificar la fecha actual por la del día siguiente.
7. Crea una clase Empleado. Utiliza tu imaginación para definir un modelo. Define al menos dos atributos y dos métodos.
8. Crea una clase NIF. Los atributos serán el número de DNI y la letra. La clase contendrá un método que calcule la letra del NIF a partir del número de DNI
9. Crea una clase Persona. La clase contendrá un atributo fechaNacimiento que será un objeto de la clase Fecha creada en el ejercicio 6 anterior.

10. Crear una clase Rectángulo, con atributos base y altura. Crear también el constructor de la clase y los métodos necesarios para calcular el área y el perímetro. Crear otra clase PruebaRectángulo que pruebe varios rectángulos y muestre por pantalla sus áreas y perímetros.
11. Se conoce de un alumno: número de matrícula, nombre y tres notas parciales (nota1, nota2, nota3). El programa debe imprimir: la matrícula, nombre, nota final e indique con un mensaje si el alumno aprobó (nota final $\geq 4,8$) o no aprobó (nota final $< 4,8$) la asignatura.
12. Crear una clase Coche, a través de la cual se pueda conocer el color del coche, la marca, el modelo, el número de caballos, el número de puertas y la matrícula. Crear el constructor del coche, así como los métodos que considere necesarios. Crear una clase PruebaCoche que instancie varios coches, cambiándole el color a lo largo de la vida a algunos de ellos y mostrándolo por pantalla.
13. Crear una clase Tiempo, con atributos hora, minuto y segundo, que pueda ser construida indicando los tres atributos, sólo la hora y minuto o sólo la hora. Crear además los métodos necesarios para modificar la hora en cualquier momento de forma manual. Mantenga la integridad de los datos en todo momento. Crear una clase PruebaTiempo que pruebe una hora concreta y la modifique a su gusto mostrándola por pantalla.
14. Construir una clase Cronometro que permita determinar el tiempo transcurrido entre dos eventos y una clase PruebaCronometro que muestre su funcionamiento.
15. Se quiere crear una clase Cuenta la cual se caracteriza por tener asociado un número de cuenta y un saldo disponible. Además, se puede consultar el saldo disponible en cualquier momento, recibir abonos y pagar recibos. Crear además una clase Persona, que se caracteriza por un DNI y un array de cuentas bancarias. La Persona puede tener asociada hasta 3 cuentas bancarias, y debe tener un método que permite añadir cuentas (hasta 3 que es el máximo). También debe contener un método que devuelva si la persona es morosa, i.e., si tienen alguna cuenta con saldo negativo. Crear una clase PruebaCuentas que instancie un objeto Persona con un dni cualquiera, así como dos objetos cuenta, una sin saldo inicial y otra con 700 euros. La persona recibe la nómina mensual, por lo que ingresa 1100 euros en la primera cuenta, pero tiene que pagar el alquiler de 750 euros con la segunda. Imprimir por pantalla el si la persona es morosa. Posteriormente hacer una transferencia de una cuenta a otra y comprobar mostrándolo por pantalla que cambia el estado de la persona.
16. Queremos mantener una colección de los libros que hemos ido leyendo, poniéndoles una calificación según nos haya gustado más o menos al leerlo. Para ello, crear la clase Libro, cuyos atributos son el título, el autor, el número de páginas y la calificación que le damos entre 0 y 10. Crear los métodos típicos para poder modificar y obtener los atributos si tienen sentido. Posteriormente, crear una clase ConjuntoLibros, que almacena un conjunto de libros (con un array de un tamaño fijo). Se pueden añadir libros que no existan (siempre que haya espacio), eliminar libros por título o autor, mostrar por pantalla los libros con la mayor y menor calificación dada y, por último, mostrar un contenido de todo el conjunto. Crear una clase

PruebaLibros, que realice varias pruebas con las clases creadas. En concreto, pruebe a: crear dos libros, añadirlos al conjunto, eliminarlos por los dos criterios hasta que el conjunto esté vacío, volver a añadir un libro y mostrar el contenido final.

17. Desarrolle una clase que realice la abstracción de un sistema planetario, debiendo tener en cuenta para cada cuerpo del sistema aspectos como: masa, densidad, diámetro, distancia al sol (suponga órbitas circulares), un número de identificador único y un nombre textual. Incluya métodos que calculen la atracción gravitatoria entre dos cuerpos cualesquiera del sistema.
18. Se quiere implementar el control de un Ferry. Para ello cree una clase Vehículo. Incluya aspectos comunes a todos los vehículos como número de pasajeros, presencia o ausencia de tripulación, número de ruedas, fecha de matriculación, medio por el que se desplaza, etc. Incluya los métodos que considere oportunos. Realice un programa capaz de leer por teclado los datos de 10 vehículos y los liste a continuación por pantalla.
19. A partir de la clase anterior, y mediante herencia cree nuevas clases que permitan diferenciar si el objeto es un coche, una moto, un camión, una bicicleta, una lancha. Incluya en cada clase algún atributo específico que la diferencie de las demás. Por ejemplo, para el coche un booleano que determine si es de tres o cinco puertas, para el camión la altura (indiferente en los otros vehículos), etc. Realice un programa como el del ejercicio anterior capaz de trabajar con los distintos tipos de vehículos.
20. Mira los ejercicios de la siguiente página e intenta hacerlos: <https://byte-mind.net/curso-python-poo/>
21. Busca información en internet y define con tus propias palabras los siguientes conceptos o características de la Programación Orientada a Objeto: Abstracción, Modularidad, Encapsulación, Herencia, Polimorfismo. También busca información sobre: Referencias, Atributos, Métodos, Constructores, Sobrecarga y Sobreescritura.
22. Construir una clase Factura que descienda de la clase Precio y que incluya dos atributos específicos llamados emisor y cliente y, al menos, un método llamado imprimirFactura.
23. Un ejemplo clásico de herencia es el de las figuras cerradas en el plano, con un método para calcular el área. En este caso, la clase base (padre) no tiene comportamiento definido ni atributos, dado que cada figura tiene atributos muy distintos (radio en el caso del círculo, base y altura en el caso del triángulo, etc.), y en cuanto al cálculo del área, cada figura tiene una fórmula diferente. Define las clases Figura, Cuadrado, Triángulo, etc.. Tienes una posible solución en <https://uniwebsidad.com/libros/algoritmos-python/capitulo-15/herencia>
24. Intenta realizar los ejercicios de la siguiente página web: <https://realpython.com/python3-object-oriented-programming/#review-exercises-1>