

Ejercicios POO - Python

Clase 1 de POO

1. Implementar una clase llamada Persona que tendrá como atributo (variable) su nombre y dos métodos (funciones), uno de dichos métodos inicializará el atributo nombre y el siguiente método mostrará en la pantalla el contenido del mismo. Definir dos objetos de la clase Persona e incorporar una variable de clase (piernas).
2. Implementar una clase llamada Alumno que tenga como atributos su nombre y su nota. Definir los métodos para inicializar sus atributos, imprimirlos y mostrar un mensaje que indique: "Promocionó" (nota ≥ 7), "Rinde final" (nota ≥ 4) o "Desaprobó". Definir tres objetos de la clase Alumno, cada uno con una condición de aprobación distinta.
3. Confeccionar una clase que represente un Animal. Definir como atributos su nombre, tipo y edad y desarrollar un método `__init__` para inicializarlos. Crear un método para imprimir los datos del objeto y crear dos objetos. Eliminar uno de ellos a través del método `__del__()`. Sustituir el método para imprimir por el método `__str__()`
4. Plantear una clase Operacion que solicite en el método `__init__` la carga de dos enteros e inmediatamente muestre su suma, resta, multiplicación y división. Hacer cada operación en otro método de la clase Operación y llamarlos desde el mismo método `__init__`.
Opcional: chequear que en la división no se genere un error.

Ejercicios adicionales:

1. Desarrollar una clase que represente un punto en el plano y tenga los siguientes métodos: inicializar los valores de x e y que llegan como parámetros, imprimir en que cuadrante se encuentra dicho punto (concepto matemático, primer cuadrante si x e y son positivas, si $x < 0$ e $y > 0$ segundo cuadrante, etc.)
 2. Plantear una clase que administre dos listas de 5 nombres de alumnos y sus notas. Mostrar un menú de opciones que permita:
 - 1- Cargar alumnos.
 - 2- Listar alumnos.
 - 3- Mostrar alumnos con notas mayores o iguales a 7.
 - 4- Finalizar programa.
-

Ejercicios POO - Python

Clase 2 de POO

5. Un banco tiene 3 clientes que pueden hacer depósitos y extracciones. El banco necesita obtener, al final del día, un reporte de la cantidad de dinero que sus clientes han depositado.
6. Plantear un programa que permita jugar a los dados. Las reglas de juego son: se tiran tres dados y si los tres salen con el mismo valor se debe mostrar un mensaje que diga "ganó", sino "perdió".
7. Definir una clase Cliente que almacene un código de cliente y un nombre. En la clase Cliente definir una variable de clase de tipo lista que almacene todos los clientes que tienen suspendidas sus cuentas corrientes. Imprimir por pantalla todos los datos de clientes y el estado que se encuentra su cuenta corriente.
8. Definir una clase Película que gestione datos de películas. La clase Película tiene los atributos título, duración y lanzamiento, e implementa `__str__` para mostrar sus datos. Definir la clase Catalogo, que administra una lista de películas. Estas películas son objetos de la clase Película. Debe implementar los métodos, mostrar y agregar películas.

Ejercicios adicionales:

3. Declarar una clase llamada Familia. Definir como atributos el nombre del padre, madre y una lista con los nombres de los hijos. Definir el método especial `__str__` que retorne un string con el nombre del padre, la madre y de todos sus hijos. Una salida esperada podría ser la siguiente:

Padres: Juan y Ana
No tienen hijos

Padres: Pablo y Noelia
Hijos: Julieta Paula

Padres: Fernando y Luisa
Hijo: Tobías