

TP: POO & PYTHON

Problemas:

P1.

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.

P2.

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).

P3.

Crear una clase llamada Persona. Sus atributos son: nombre, edad y DNI. Construya los siguientes métodos para la clase:

- * Un constructor, donde los datos pueden estar vacíos
- * Los **setters y getters** para cada uno de los atributos. Se deben validar las entradas de datos
- * Un método ***mostrar()*** que muestre los datos de la persona
- * Un método ***es_mayor_de_edad()*** que devuelva un valor lógico indicando si es mayor de edad.

P4.

Se necesita una porción de código para un inventario de stock, cree una clase inventario con atributos: id, nombre, stock, precio.

*Se deberá contar con metodos agregar, modificar, informacion.

*Utilice diccionarios para almacenar los items, donde el ID será la clave y los datos serán los valores (que a su vez será otro diccionario)

tip: Puede utilizar el operador `in` para verificar que el ID del diccionario se encuentra en el inventario para mostrar la información, caso que no exista deberá mostrar un mensaje que no existe ese item.

Realice la prueba de código y verifique su funcionamiento.

P5.

Crear una clase llamada Cuenta que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Crear los siguientes métodos para la clase:

- * Un constructor, donde los datos pueden estar vacíos
- * Los setters y getters para cada uno de los atributos. El atributo cantidad no se puede modificar directamente, solo ingresando o retirando dinero.
- * mostrar(): Muestra los datos de la cuenta
- * ingresar(cantidad): se ingresa una cantidad a la cuenta, si la misma es negativa no se hará nada.
- * retirar(cantidad): se retira una cantidad de la cuenta. La misma puede tener saldo negativo.

P6.

Escribir un código en python, donde defina la clase Empleado, y dos subclases, Administrador y desarrollador. El empleado tendrá Nombre, antigüedad.

El administrador tendrá variables como numero_equipo y nro_cartera_clientes, el desarrollador tendrá numero_de_proyectos y lenguaje_programacion

- * Crear un método para mostrar la información completa del empleado.
- * Realizar una carga de datos manual y luego llamar la función para mostrar en ambos casos la funcionalidad del código.

tip: utilice la función super() para referirse a la clase padre a la hora de mostrar los datos para las clases hijas heredadas.