



Programación Orientada a Objetos (1ACC0201)

Ciclo 2025-10

Hoja de ejercicios semana 6

Tema: Clases y objetos en Python

1. Crear la clase Loro que tendrá como atributos el nombre y la edad, luego crear dos objetos con los siguientes parámetros: loro1 se llamará "Azul" y tendrá 10 años, el loro2 se llamará "Galán" y tendrá 15 años, finalmente generar un reporte que mostrará lo siguiente:
 - Azul tiene 10 años
 - Galán tiene 15 años
2. Crear la clase círculo, que tenga como atributo el radio y los métodos para calcular el área y la circunferencia. Crear 3 objetos e imprimir sus áreas y circunferencias
3. Escribir una clase llamada CuentaBancaria que tiene lo siguiente:
 - Un campo llamado nombre que almacena el nombre del titular de la cuenta.
 - Un campo llamado saldo que almacena la cantidad de dinero en la cuenta.
 - Un campo llamado tasaInteres que almacena la tasa de interés de la cuenta (como un porcentaje).
 - Un constructor que simplemente establece los valores de los tres campos anteriores.
 - Un método llamado aplicaInteres() que no acepta argumentos y aplica el interés a la cuenta. Solo debe modificar el campo de saldo y no devolver nada. Por ejemplo, si la cuenta tiene 1000 y la tasa de interés es del 3%, entonces la cantidad variable debe cambiarse a 1030 (1000 + 3% de interés).

A continuación, pruebe la clase, mediante la creación de un nuevo objeto llamado CuentaBancaria Para un usuario nombrado Juan De Arona que tiene 1000 al 3% de interés. A continuación, haga lo siguiente:

- Utilice el método aplicaInteres() para aplicar los intereses a la cuenta.
- Imprima cuánto dinero hay ahora en la cuenta después de aplicar los intereses.
- Cambie la tasa de interés de la cuenta a 2%.
- Utilice el método aplicaInteres() para volver a aplicar los intereses a la cuenta.
- Imprima cuánto dinero hay ahora en la cuenta después de aplicar el interés nuevamente.



4. Escriba una clase llamada Ticket que tenga lo siguiente:
- Un campo de costo para el precio del boleto y una hora (como dato de cadena) para la hora de inicio del evento (supongamos que las horas están en formato de 24 horas, como '18:35')
 - Un constructor que establece esas variables
 - Un método **str()** que devuelve una cadena con el formato Ticket(,), donde y se sustituyen por los valores de los campos costo y hora.
 - Un método llamado **esNocheHora()** que devuelve True o False, dependiendo de si la hora se encuentra en el rango de 18:00 a 23:59.
 - Un método llamado **totalDescuento()** que toma un número entero n y devuelve el descuento por comprar n boletos. Debería haber un 10% de descuento por comprar de 5 a 9 boletos, y un 20% de descuento por comprar 10 o más. De lo contrario, no hay descuento. Devuelve estos porcentajes como números enteros
5. Desarrollar un programa en Python para la clase estudiante, que tendrá como atributos, el código del alumno, su nombre, el curso, la nota del examen parcial y la nota del examen final. La clase, además del constructor y el **str()**, tendrá un método para calcular la nota final, según la siguiente fórmula:

$$\text{Nota final} = \text{parcial} * 0.4 + \text{final} * 0.6$$

6. Desarrollar una aplicación que pueda gestionar los clientes (registrar, actualizar, eliminar y reportar información sobre clientes). Cada **cliente** debe tener:
- Un DNI
 - un nombre
 - una dirección
 - un número de teléfono
 - una dirección de correo electrónico.
 - Una indicación de cliente preferente: True o False

El sistema debe permitir generar dos reportes: uno con todos los clientes y otro con clientes filtrados por nombre.

Los clientes están almacenados en una **baseClientes** (la **baseClientes** tiene como atributo una lista de clientes)

Clase cliente (con los atributos descritos y el método **verCliente** [muestra los datos de un cliente])

Clase baseClientes (**listaClientes** (lista que almacena objetos Cliente) y los métodos: **registrarCliente** [agrega un cliente a la **baseClientes**], **actualizarCliente** [actualiza la información de un cliente preexistente], **eliminarCliente** [elimina cliente, se busca por su DNI para eliminar], **visualizarClientes** [ver listado de clientes], **busquedaCliente** [buscar cliente por nombre]).

El programa debe preguntar al usuario por una opción del siguiente menú:



- (1) Añadir cliente
- (2) Buscar cliente
- (3) Actualizar cliente
- (4) Eliminar cliente
- (5) Listar todos los clientes
- (6) Terminar.

En función de la opción elegida el programa tendrá que hacer lo siguiente:

- En la opción 1, preguntar los datos del cliente, según el diseño presentado.
 - Se valida que DNI sea un string de 8 números
 - El nombre no puede tener números
 - El teléfono son 9 dígitos (tip: considere que es un string de 9 dígitos)
 - El correo se forma con la inicial en minúscula del nombre seguido del apellido, luego @ seguido del dominio empsac y terminando con .com
 - Para el preferente se debe ingresar 1 o 0, si es uno se graba True, en caso contrario False
- En la opción 2, preguntar por el DNI del cliente y eliminar al cliente. Debe verificar que el cliente exista
- Para la Opción 5 mostrar un reporte en donde se muestre el DNI y otros datos, similar al siguiente:

DNI	Nombre	Dirección	Teléfono	Email	Preferente
	Robert				
08804832	Fischer	Arges 141	945025205	rFischer@empsac.com	True
	Boris				
12345678	Spassky	Apeliotas 361	998325782	bSpassky@empsac.com	False