



Ciclo 1 Fundamentos de programación

Reto 2 – Creación de un programa para guardar clientes y otorgar créditos

Descripción del problema: El sector bancario siempre ha jugado un papel fundamental en la economía, desarrollo de los países y de las personas. Como intermediarios financieros canalizan los recursos excedentes de las personas, que le son consignados en sus diferentes productos de ahorro y cuentas corrientes, en préstamos para atender las diferentes necesidades de los clientes, de consumo, compra de vivienda, carro, viajes y estudios. Dacasa es un banco junior que está empezando a expandir sus servicios, pero para esto Dacasa decide contratar a un ingeniero de Sistemas para realizar un algoritmo en Python que permita guardar sus clientes y otorgar créditos a los clientes. Cada crédito siempre tiene solo un titular. Es decir, no se permiten crear créditos sin un deudor. Los clientes se caracterizan por tener un número de identificación, un nombre completo, un tipo: Preferencial o Regular y una deuda. A los clientes preferenciales se les hace un descuento del 30% en el valor de su deuda si es mayor a 50000000, si esta entre 50000000 y 20000000, se les hace un descuento del 20% y si la deuda es menor a 20000000 se le hace un descuento del 10%. A los clientes Regulares se les hace un descuento del 7% en el valor de su deuda si es mayor a 70000000, si esta entre 70000000 y 50000000, se les hace un descuento del 3% y si la deuda es menor a 50000000 se le hace un descuento del 1%.

Para realizar el ejercicio se realizarán dos funciones, las cuales son:

Primera Función: Esta función recibe los siguientes parámetros: Identificación del cliente, nombre del cliente, tipo del cliente y deuda. Esta función debe retornar un diccionario con la información previamente mencionada.

Segunda Función: Esta función recibe como parámetro único un diccionario, este puede ser el creado por la Primera función, debe retornar una cadena {"ID: {Diccionarios['ID']}, Nombre: {Diccionarios['Nombre']}, Tipo cliente: {Diccionarios['Cliente tipo']}, Deuda: {Diccionarios['Deuda']}" ya con la deuda actualizada.



Ejemplos:

Primera Función:

ID	Nombre	Tipo	deuda	Return
2324234	Carlos	Preferencial	60000000	{'ID': 2324234, 'Nombre': 'Carlos', 'Cliente tipo': 'Preferencial', 'Deuda': 60000000}
3452333	Carolina	Regular	80000000	{'ID': 3452333, 'Nombre': 'Carolina', 'Cliente tipo': 'Regular', 'Deuda': 80000000}
2324234	Juan	Regular	60000000	{'ID': 2324234, 'Nombre': 'Juan', 'Cliente tipo': 'Regular', 'Deuda': 60000000}

Entrada:

Nombre	Tipo	Descripción
ID	int	El identificador del Cliente
Nombre	Str	Nombre del Cliente
Tipo	Str	Tipo de clientes. Preferencial o Regular
deuda	Float	El precio de la deuda, que este se le resta el descuento que se aplica dependiendo del Tipo de cliente

Salida:

Tipo de retorno	Descripción
dic	{'ID': ID, 'Nombre': Nombre, 'Cliente tipo': Tipo, 'Deuda': deuda}

Esqueleto:

```
def CrearDiccionario(ID: int, Nombre: str, Tipo : str, deuda : float) -> dict:
```



Segunda Función:

Diccionarios	Return
{'ID': 2324234, 'Nombre': 'Carlos', 'Cliente tipo': 'Preferencial', 'Deuda': 60000000}	ID: 2324234, Nombre: Carlos, Tipo cliente: Preferencial, Deuda: 42000000.0
{'ID': 3452333, 'Nombre': 'Carolina', 'Cliente tipo': 'Regular', 'Deuda': 80000000}	ID: 3452333, Nombre: Carolina, Tipo cliente: Regular, Deuda: 74400000.0
{'ID': 2324234, 'Nombre': 'Juan', 'Cliente tipo': 'Regular', 'Deuda': 60000000}	ID: 2324234, Nombre: Juan, Tipo cliente: Regular, Deuda: 58200000.0

Entrada:

Nombre	Tipo	Descripción
Diccionario	dic	Los diccionarios previamente retornados por la Primera Función

Salida:

Tipo de retorno	Descripción
dict	{"ID": {Diccionarios['ID']}, Nombre: {Diccionarios['Nombre']}, Tipo cliente: {Diccionarios['Cliente tipo']}, Deuda: {Diccionarios['Deuda']}

Esqueleto:

```
def AplicarDescuentoPorTipo(Diccionarios: dict)->str:
```