**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercício 01**

Altere a classe Funcionario do programa abaixo:

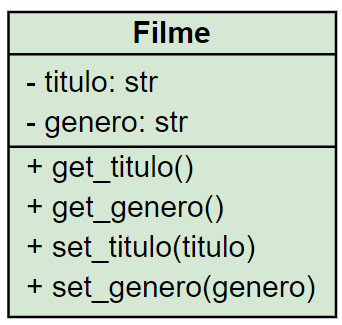
* Transforme os seus atributos em atributos privados.
* Crie os métodos get e set para todos os atributos.
* Faça as alterações necessárias para que o programa principal funcione corretamente, após as alterações feitas na classe.

| class Funcionario:  def \_\_init\_\_(self, nome, cpf, salario):  self.nome = nome  self.cpf = cpf  self.salario = salario  func1 = Funcionario("Pedro", "111222333-22", 1500.0)  func1.salario = 2000.0 # Altera salário  print("Nome:", func1.nome)  print("CPF:", func1.cpf)  print("Salário:", func1.salario) |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercício 02**

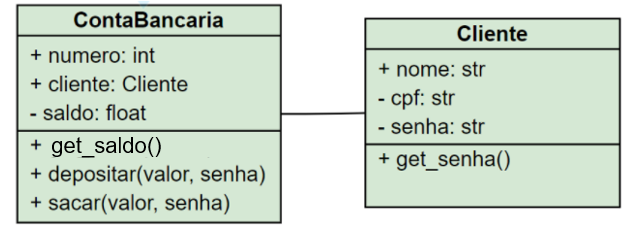
* Implemente a classe Filme, conforme o diagrama de classes abaixo
  + Todos os atributos devem ser *privados*
  + Crie os métodos *get* e *set* para todos os atributos
* No seu programa principal, faça a seguinte implementação:
  + Criar uma lista de filmes vazia
  + Cadastrar 3 filmes (com os dados informados pelo usuário)
  + Armazenar os objetos na lista de filmes
  + Percorrer a lista de filmes e imprimir no terminal os dados de todos os filmes cadastrados



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercício 03**

Implemente as classes ContaBancaria e Cliente, conforme o diagrama de classes abaixo.



**Classe Cliente**

Atributos::

* **nome**: nome do cliente (público)
* **cpf**: cpf do cliente (privado)
* **senha**: senha do cliente (privado)

Métodos:

* **get\_senha**: retorna a senha do cliente

**Classe ContaBancaria**

Atributos:

* **numero**: numero da conta (público)
* **cliente**: objeto Cliente associado à conta (público)
* **saldo**: saldo da conta (privado). Deve ser inicializado com zero.

Métodos:

* **get\_saldo**: retorna o saldo da conta.
* **depositar**: recebe como parâmetros de entrada um valor e uma senha. Acrescenta esse valor ao saldo da conta apenas se a senha for igual à senha do cliente.
* **sacar**: recebe como parâmetros de entrada um valor e uma senha. Subtrai esse valor do saldo da conta, apenas se a senha for igual à senha do cliente.

Utilize o código abaixo para testar as suas classes

| cliente1 = Cliente("João", "111111111", "123")  conta = ContaBancaria(1111, cliente1)  conta.depositar(200, "123")  print(conta.get\_saldo()) # Imprime 200  conta.sacar(50, "123")  print(conta.get\_saldo()) # Imprime 150  conta.depositar(100, "111") # senha inválida  print(conta.get\_saldo()) # Imprime 150  conta.sacar(50, "111") # senha inválida  print(conta.get\_saldo()) # Imprime 150 |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**