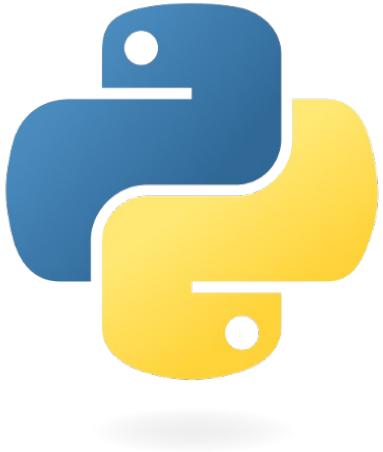
SHIFT

FIAP



python™

Classes + Relaciomentos

O paradigma que revolucionou o mindset dev

AGENDA

1. Relacionamento, o que é isso ?
2. Associação na Classe
3. Agregação vs Composição
4. Herança ... o poder de compartilhar :-)

1

Relacionamento

Como podemos manter um
RELACIONAMENTO DE CLASSE ?

Representa uma ligação entre classes.

Cada tipo de relacionamento descreve um comportamento referente ao conceito.

- Associação é opcional
- Agregação / Composição indica que é parte classe
- Herança representa a especialização da classe “copia” é uma descrição nomeada como superclasse

Class
<relationship>

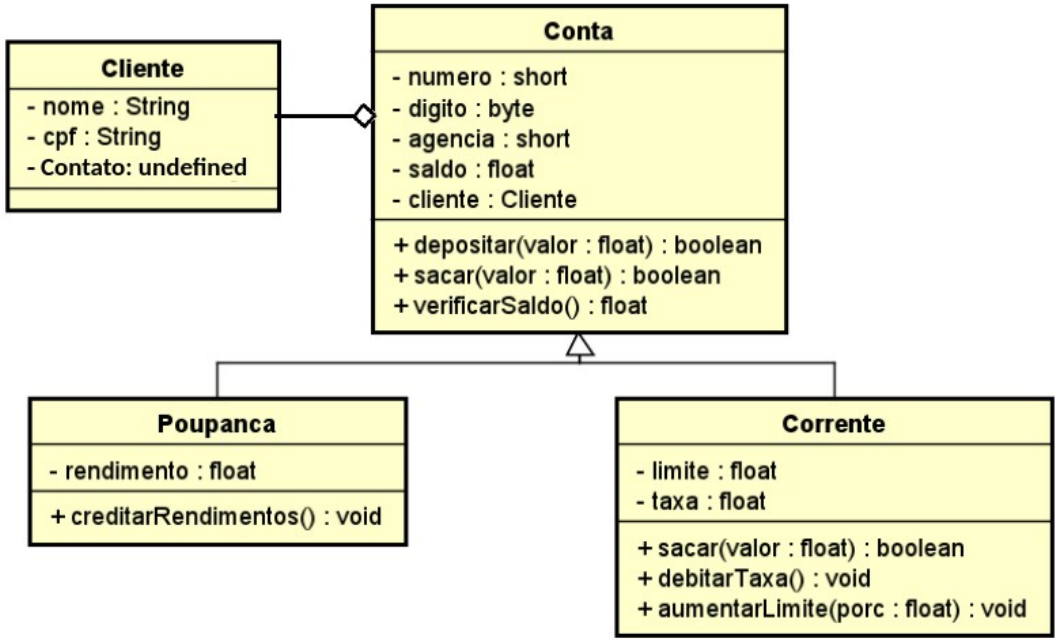
```
"""
Associação – Usa | Agregação – Tem | Composição – É dono | Herança – É
"""
```

A HORA DO DESAFIO

A vida como ela é...

Você recebeu outra “missão”..!!! A rotina de controle financeiro precisa ficar melhor, sendo assim temos as seguintes informações...

Veja ao lado ;-)



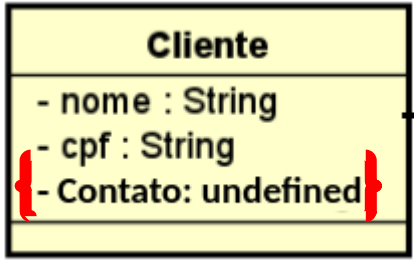
2

Associação

A vida como ela é...

O processo de contato pode ocorrer por e-mail ou telefone. Mas isso é opcional...

Vamos usar Associação para resolver esse item..



class Fone

class Email

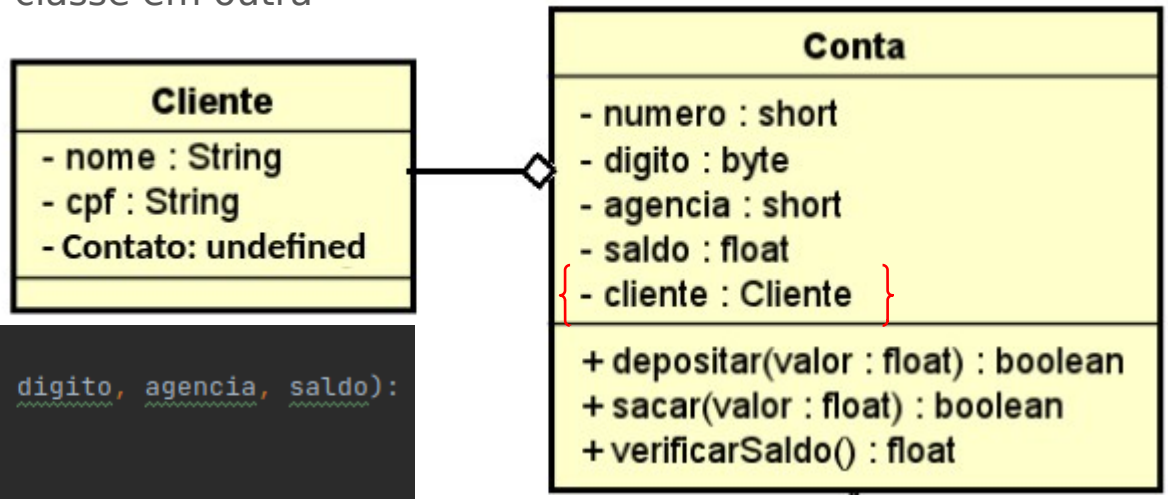
;-)

3

Agregação vs Composição

Afinal a Conta precisa de um CLIENTE ou NÃO ?

Que que tal AJUSTARMOS nossa classe CONTA para obter uma informação de “COMPLEMENTO” indicando o cliente... (isso é Agregar uma classe em outra através de um atributo

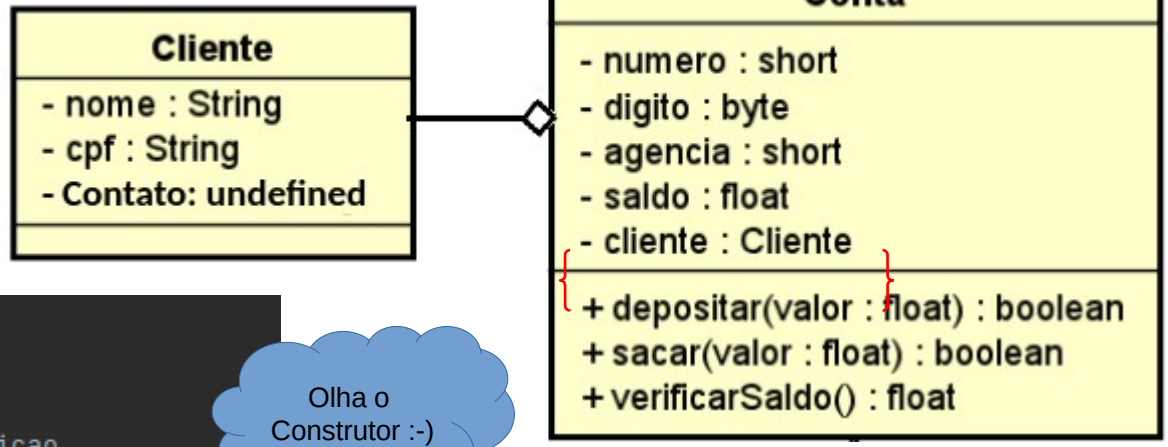



```
class Conta:
    def __init__(self, numero, digito, agencia, saldo):
        self.numero = numero
        self.digito = digito
        self.agencia = agencia
        self.saldo = saldo
        self.cliente = None
```

E para usar COMPOSIÇÃO na Conta?

Isso indica que a Classe Conta será responsavel por “CRIAR o OBJETO CLIENTE” através de algum método da própria classe Conta.

O objeto CLIENTE não existe fora da CONTA




```
#alterando para Composição
@property
def clienteComposicao(self):
    return self.__clienteComposicao

def adicionar_clientecomposicao(self,nome,cpf):
    self.__clienteComposicao = Cliente(nome,cpf)
```

Olha o Construtor :-)

4

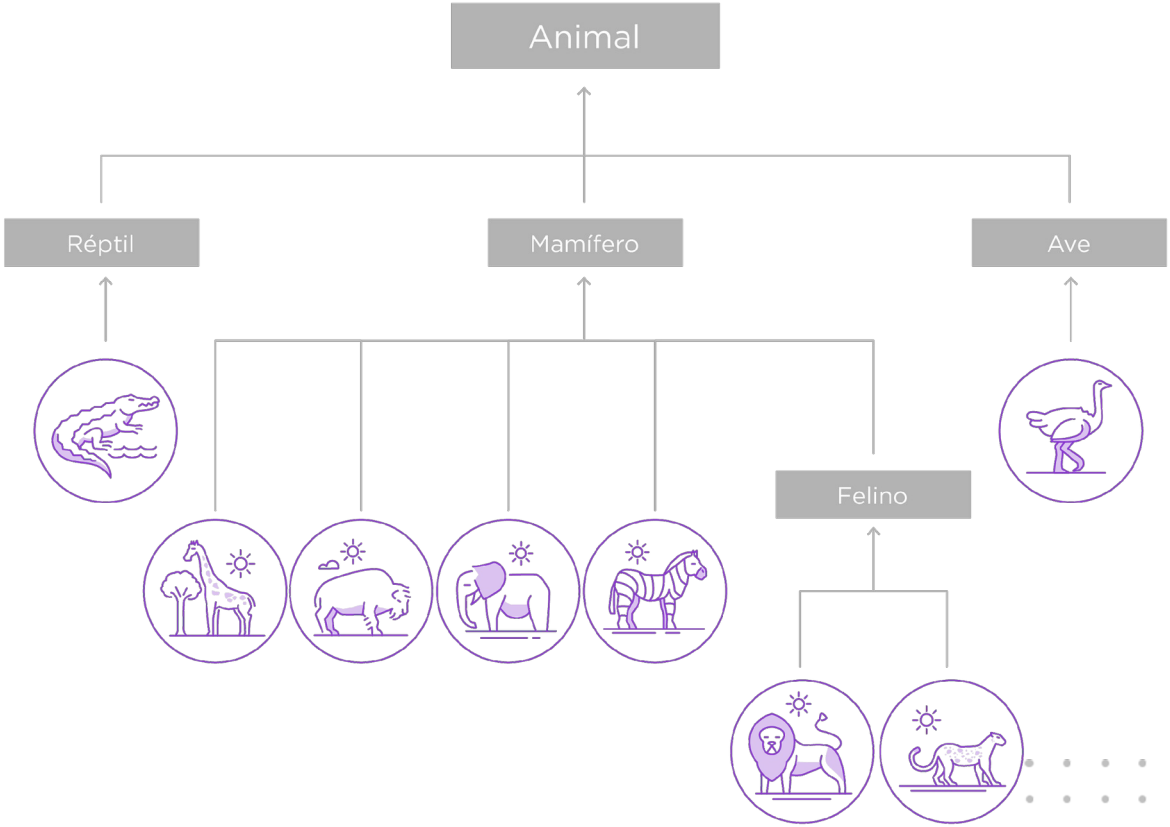
Herança

Compartilhar para conquistar

Aplicar herança sempre envolve basicamente dois elementos: uma superclasse (classe pai) e uma subclasse (classe filha).

Superclasse: Apresenta características genéricas

Subclasse: é a especificação das características

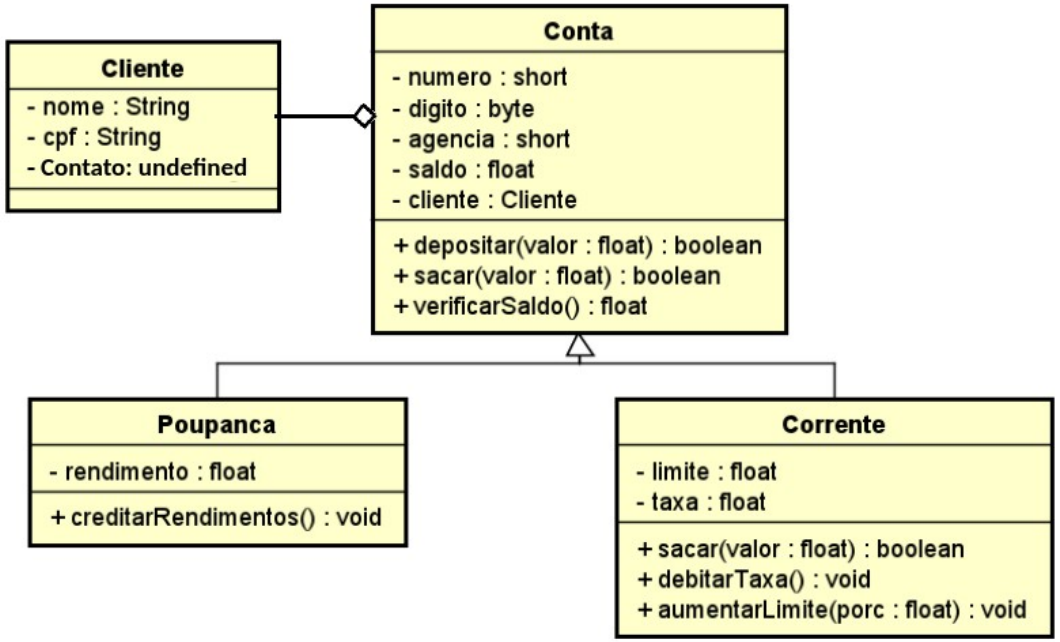


Nossa Missão...

Poupança e Corrente, são informações especializadas do negócio, vamos aplicar HERANÇA para resolver esse problema.

class <nome>(superclasse):

Veja ao lado ;-)



OBRIGADO

FIAP