Python – Orientação a Objetos

* O paradigma de orientação a objetos nasceu para fazer frente ao mundo procedural
* Procedural são programas de procedimento. Funções são chamadas, executam suas funções e terminam.
* Em OO, as classes são como receitas de bolo, contendo ingredientes e modo de preparo.
  + Nem todo objeto precisa ser criado com todos os ingredientes (variáveis) de uma classe
  + Para iniciar uma classe, chamamos a função construtora *def \_\_init\_\_(self):*
  + *self* é a referência que guarda onde, em memória, o objeto foi construído.
  + *Atributos* são as características de uma classe
* Esse é um diagrama de classes, feito com UML.
* conta.py pode ser chamado de ‘módulo’ ou ‘namespace’
* A ‘Conta’ é a classe,
* E dentro dela estão os ‘atributos’
* Dentro da classe também há os métodos: ‘sacar’, ‘depositar’ e ‘extrato’
* É possível encapsular atributos para que ngm acesse diretamente sem os métodos.
  + - * Para encapsular, coloca-se 2 undescore na frente do atributo: *self.\_\_saldo*
* Métodos estáticos são definidos com @staticmethod antes do método
  + Servem para chamadas de método sem precisar da construção de um objeto.