*''' POO em Pytthon  
class NomeClasse:  
 def \_\_init\_\_ (self): # Método construtor  
 self.nome\_atributo  
 def outro\_metodo (self): # Outros métodos da classe  
 return  
if \_\_name\_\_ == '\_\_main\_\_': # main <tab>  
 nome\_objeto1 = NomeClasse() # Cria (instancia) um objeto da classe  
 nome\_objeto2 = NomeClasse()  
 print ("Valor do atributo x: "p.x) # nome\_objeto . nome\_atributo  
- Com base nos modelos acima, implemente estes itens:  
Crie um novo projeto  
Crie a classe Point.  
Crie o método construtor.  
Dentro do construtor, crie os atributos x e y.  
O método construtor recebe o self  
Dentro do contrutor, inicialize os atributos com zero.  
Dentro da função main crie o objeto p (o ponto\_p)  
Mostre o objeto criado, o ponto p.  
Rode a classe  
Mostre o valor do atributo x  
Mostre o valor do atributo y  
Dentro da função main crie o segundo objeto da classe Ponto, objeto m (o ponto\_m)  
Altere o valor do atributo x para 9 do primeiro objeto (objeto p)  
Teste o item anterior  
Rode a classe  
Dentro da função main crie o objeto q passando o valor do atributo x  
Mostre os atributos do ponto q  
Altere o construtor para ele receber o valor do atributo x e do atributo y  
Atualize as atribuições dentro do método construtor  
Dentro da classe, crie os métodos get (consultar) e set (alterar) para os atributos x e y.  
def getNomeAtributo (self): # Modelo do método get  
 return self.nome\_atributo  
def setNomeAtiibuto (self, valor): # Modelo do método set  
 self.nome\_atributo = valor  
No main, teste os métodos gets e sets dos dois atributos da classe Point  
Na classe, crie o método distância do ponto para origem (0, 0) do plano cartesiano.*