Artigo

Como criar minha primeira classe em Python

Aprenda neste conteúdo como criar sua primeira classe utilizando a linguagem de desenvolvimento Python.

#PraCegoVer - Transcrição dos Slides

Criar uma classe em Python é bastante simples

Para definir o nome da classe usamos a palavra reservada class - class Pessoa: - Após o nome da classe devemos adicionar dois pontos

Para criar um método usamos a palavra reservada def. O construtor é um método reservado chamado _init>. O parâmetro self é obrigatório e os demais são definidor por nós. - def _init_(self, nome, idade):

Aqui está o corpo do método, sempre identado como manda a sintaxe da linguagem - self.nome = nome

self.idade = idade

A partir daí podemos definir quantos métodos precisarmos.

Ir para o código

Primeira classe em Python

No **Python a criação de classes** é bem simples e nos permite definir quais atributos e métodos uma classe irá possuir. Na **Figura 1** podemos visualizar a representação de uma classe com o nome `Pessoa` e quais atributos esta possui:

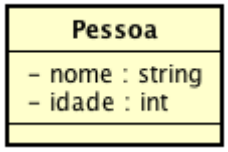

Figura 1. Diagrama de classe Pessoa

```
= idade def setNome(self, nome): self.nome = nome def setIdade(self,

idade): self.idade = idade def getNome(self): return self.nome def

getIdade(self): return self.idade
```

Linha 1: A criação de uma classe começa pelo uso da palavra reservada `class`, seguida do nome da classe e dois pontos;

Linha 2: Aqui temos a definição do construtor da classe, que é um método especial chamado `__init__`. Como todo método em Python, sua declaração começa com `def` e entre parênteses estão os parâmetros, incluindo o parâmetro obrigatório `self`, que está presente em todos os métodos;

Linhas 3 e 4: O corpo do método deve estar identado, como manda a sintaxe da linguagem. Aqui estamos apenas atribuindo os valores recebidos por parâmetro aos atributos da classe;

Linhas 6 a 16: Criamos os métodos get e set de todos os atributos da classe Pessoa que serão responsáveis, respectivamente, por retornar ou modificar os atributos desta classe.

Herança em Python

#PraCegoVer - Transcrição dos Slides

Também podemos implementar herança entre classes no Python: PessoaFisica e PessoaJuridica herdam de Pessoa

Primeiro precisamos importar a classe pai - from pessoa import Pessoa - pessoa é o nome do arquivo em que a classe pai foi criada (sem a extensão .py)

Pessoa é o nome da classe Pai.

Em seguida podemos definir a classe filha herdando da classe pai (entre parênteses) - class PessoaFisica(Pessoa): - onde PessoaFisica é o nome da classe filha e Pessoa é a classe pai

Podemos criar um novo construtor: def _init_(self, CPF, nome, idade):

E invocar o construtor da classe pai, aproveitando seu funcionando padrão - super()._init_(nome, idade)

A classe filha pode ter seus próprios atributos... - self.CPF = cpf

... e também seus próprios métodos:

def setCPF(self, cpf): self.CPF = cpf

def getCPF(self): return self.CPF

Na Programação Orientada a Objetos o conceito de herança é muito utilizado. Basicamente, dizemos que a herança ocorre quando uma classe (filha) herda

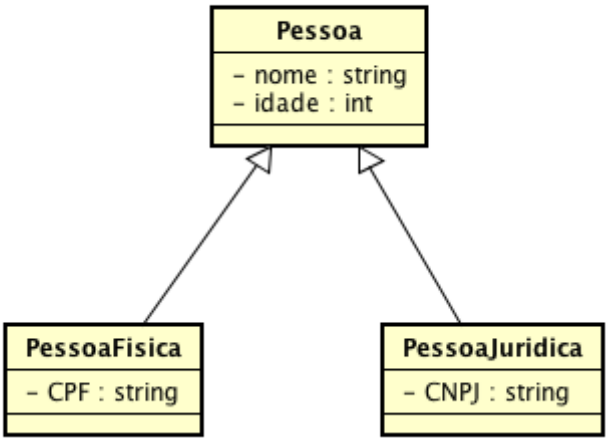

Figura 2. Diagrama de classes com PessoaFisica e PessoaJuridica herdando da classe Pessoa

Abaixo podemos ver como esta relação ocorre em Python:

```
from pessoa import Pessoa class PessoaFisica(Pessoa): def __init__(self, CPF, nome, idade): super().__init__(nome, idade) self.CPF = CPF def getCPF(self): return self.CPF def setCPF(self, CPF): self.CPF = CPF
```

Linha 1: Como vamos herdar da classe `Pessoa` e ela foi definida em outro arquivo (pessoa.py), precisamos importá-la. Para isso usamos a instrução `import` e indicamos o nome do arquivo, sem a extensão .py, seguido do nome da classe que queremos importar;

Linha 3: Para definir que uma classe herdará de uma outra classe, precisamos indicar o nome da classe pai entre parênteses após o nome da classe filha;

isso aproveitamos todo a lógica definida nesse método, que no caso faz a

atribuição dos valores de nome e idade aos atributos da classe. Com isso garantimos que ao ser criada, a classe filha efetuará o mesmo processamento que a classe pai e mais alguns passos adicionais.

Dessa mesma forma podemos criar a classe `PessoaJuridica`:

```
from pessoa import Pessoa class PessoaJuridica(Pessoa): def __init__(self, CNPJ, nome, idade): super().__init__(nome, idade) self.CNPJ = CNPJ def getCNPJ(self): return self.CNPJ def setCNPJ(self, CNPJ): self.CNPJ = CNPJ
```

Observe que o nome das classes começa com letra maiúscula (inclusive quando há mais de uma palavra, como em PessoaFisica). Essa é uma convenção de escrita de código na linguagem, mas não uma obrigatoriedade da sintaxe

Sugestão de conteúdo

- Curso de Python
- Guia Completo de Python

Tecnologias:

POO Python UML

Suporte ao aluno - Tire a sua dúvida.