Estrutura de dados
Orientação a Objetos

Métodos Estáticos

Profª. Drª. Alana Morais

Fizeram o exercício de classes abstratas?

Praticar faz a diferença ...

Um desenvolvedor Java acabou de conhecer Python e gostaria de saber como definir métodos privados nessa linguagem.

Assinale a alternativa que transforma o método a seguir em privado:

```
def temSaldoDisponivelPara(self, valor):
        return valor < (self.__saldo + self.__limite)
```

```
def temSaldoDisponivelPara(self, valor):
        return valor < (self.__saldo + self.__limite)
```

a)
```
def private temSaldoDisponivelPara(self, valor):
```

b)
```
@private
  def temSaldoDisponivelPara(self, valor):
```

c)
```
def _temSaldoDisponivelPara(self, valor):
```

d)
```
def __temSaldoDisponivelPara(self, valor):
```

Métodos convencionais

Como se estruturam os métodos normalmente em Python?

```
def nome_metodo(self, valorArgumento):
    return (10 * valorArgumento * self.__valorInterno)
```

Métodos convencionais

Como se estruturam os métodos normalmente em Python?

```
def nome_metodo(self, valorArgumento):
    return (10 * valorArgumento * self.__valorInterno)
```

E se meu método fosse organizado da seguinte maneira:

```
def nome_metodo(self, valorArgumento):
    return (10 * valorArgumento)
```

Métodos convencionais

Como se estruturam os métodos normalmente em Python?

```
def nome_metodo(self, valorArgumento):
    return (10 * valorArgumento * self.__valorInterno)
```

E se meu método fosse organizado da seguinte maneira:

```
def nome_metodo(self, valorArgumento):
    return (10 * valorArgumento)
```

O self seria útil neste método?

Outra possibilidade para essa implementação?

Podemos usar **métodos estáticos:**

- São métodos que podem ser chamados a partir de classes não de objetos.
- Reaproximação do mundo procedural e distanciamento do mundo OO.
- Mesmo sem o objeto, conseguimos executar o método. Como assim?

Exemplo

Dado nosso sistema de Contas Bancárias.

Agora precisaremos incluir uma informação associada à agência da Conta.

Estamos trabalhando na classe que lida com Conta Corrente e a agência sempre será '1212'

Como incluiremos esse ponto na nossa classe Conta?

Exemplo

```
class Conta:
    def __init__(self, numero, titular, saldo, limite):
        print("Construindo objeto ... {}".format(self))
        self.__numero = numero
        self.__titular = titular
        self.__saldo = saldo
        self.__limite = limite
        self.__agencia = '1212'


    @property
    def codigo_agencia(self):
        return self.__agencia
```

Exemplo - Poderia ser assim?

```
class Conta:
    def __init__(self, numero, titular, saldo, limite):
        print("Construindo objeto ... {}".format(self))
        self.__numero = numero
        self.__titular = titular
        self.__saldo = saldo
        self.__limite = limite
        self.__agencia = '1212'


    @property
    def codigo_agencia(self):
        return '1212'
```

Não é a mesma situação do problema inicial? O self é importante aqui?

Exemplo

Imagine que o objeto da classe Conta ainda não foi criado.

E você gostaria de saber qual é o código do banco

- Precisamos criar o objeto primeiro.
- É necessário alguma informação que varia de objeto em objeto?

Métodos Estáticos

- Esse métodos que conseguem ser chamados sem uma referência recebem o nome de estáticos, porque eles fazem parte da classe.
- Todas as linguagens orientadas a objeto trabalham com métodos estáticos, mas para que eles sejam utilizados, é necessário configurar os métodos (sintaxe varia por linguagem).
- Exemplo:
    - O código do banco não depende do objeto.
    - É uma informação associada à classe Conta.

Métodos Estáticos

```
class Conta:
    def __init__(self, numero, titular, saldo, limite):
        print("Construindo objeto ... {}".format(self))
        self.__numero = numero
        self.__titular = titular
        self.__saldo = saldo
        self.__limite = limite
        self.__agencia = '1212'


    @staticmethod
    def codigo_agencia():
        return '1212'
```

Sem self. Por quê?

Métodos Estáticos

```
class Conta:
    def __init__(self, numero, titular, saldo, limite):
        print("Construindo objeto ... {}".format(self))
        self.__numero = numero
        self.__titular = titular
        self.__saldo = saldo
        self.__limite = limite
        self.__agencia = '1212'

    @staticmethod
    def codigo_agencia():
        return '1212'
```

É recomendado remover a informação do construtor

Métodos Estáticos

Agora para testarmos no console, nosso código funcionará corretamente:

- `Conta.codigo_banco()`

Observe que nenhum objeto foi criado, mas conseguimos chamar o método estático.

Nós especificamos o nome da classe, e depois de acessá-la, chamamos o método.

Poderia ser ampliado ...

```
class Conta:
    …
    @staticmethod
        def codigos_agencias():
            return {'BB': '001', 'Caixa': '104',
    'Bradesco': '237', 'Alana': '1212'}


print(Conta.codigos_agencias()['Alana'])
```

Exercício Casa

Escreva a classe Conversao_area em Python com métodos estáticos para conversão das unidades de área segundo a lista abaixo.

- 1 metro quadrado = 10.76 pés quadrados
- 1 pé quadrado = 929 centímetros quadrados
- 1 milha quadrada = 640 acres
- 1 acre = 43.560 pés quadrados

Dúvidas?
alanamm.prof@gmail.com