

Polimorfismo

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Polimorfismo é o princípio a partir do qual as classes derivadas de uma única classe base (relação de herança) são capazes de invocar os métodos que, embora apresentem a mesma assinatura, comportam-se de maneira diferente para cada uma das classes derivadas. Portanto, um método polimórfico tem a possibilidade de assumir lógicas diferentes.

Polimorfismo

```
1  class Animal:
2      def comer(self):
3          print('O animal anda')
4
5  class Peixe(Animal):
6      def comer(self):
7          print('O peixe nada')
8
9  peixe = Peixe()
10 peixe.comer()
```

Qual será o resultado?

Polimorfismo

```
1  class Animal:
2      def comer(self):
3          print('O animal anda')
4
5  class Peixe(Animal):
6      def comer(self):
7          print('O peixe nada')
8
9  peixe = Peixe()
10 peixe.comer()
11 super(Peixe, peixe).comer()
```

Qual será o resultado?

Polimorfismo

```
1  class Animal:
2      def comer(self):
3          print('O animal anda')
4
5  class Peixe(Animal):
6      def comer(self):
7          print('O peixe nada')
8
9  peixe = Peixe()
10 peixe.comer()
11 super(Peixe, peixe).comer()
```

O método `super()` é utilizado entre heranças de classes, ele proporciona acionamento dos métodos de uma classe pai para a classe filha. Uso: **`super(classFilha, objetoInstaciado)`**

Exercício 1 (20min)

Construir duas classes filhas, nomeadas como vendedorPJ e uma vendedorCLT, ambas tendo como base a classe Pessoa, com um método de calcularComissaoVendas. Definir o método também nas classes filhas mas com as lógicas:

- Para a classe Pessoa, retornar a comissão equivalente a 10% das vendas
- VendedorCLT, retornar a comissão a 3% das vendas mais um valor fixo de 1100.
- VendedorPJ, retornar uma comissão de 18% das vendas.

Encapsulamento

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Definição

Encapsulamento é um dos pilares da programação orientada a objetos, por meio dele conseguimos proteger informações sigilosas ou sensíveis. Encapsular os dados de uma aplicação significa evitar que estes sofram acesso indevido, para isso usamos estruturas de dados que podem ser utilizadas por outras classes sem causar inconsistência.

Exemplo Prático 2

```
1  class Conta:
2      def __init__(self, nome, agencia, conta):
3          self.agencia = agencia
4          self.conta = conta
5          self.nome = nome
6          self.saldo = 0
7
8      def depositar(self, valor):
9          self.saldo += valor
10
11
12  conta = Conta('Yan', '0001', '1234')
13  conta.saldo = 1000
14  # O saldo da conta é: R$ 1000
15  print(f"O saldo da conta é: R$ {conta.saldo}")
```

As linguagens de programação possuem mecanismos diferentes para implementar o encapsulamento, como exemplo em C# que utiliza a palavra-chave `private` para determinar se o recurso é visível para fora da classe. No Python existe uma convenção de declaração para definir quais recursos o “cliente” pode acessar, essa convenção dita que dados ou métodos que iniciem com dois `_` (underscore) não deveriam ser acessados fora da classe.

Exemplo Prático 2

```
1  class Conta:
2      def __init__(self, nome, agencia, conta):
3          self.agencia = agencia
4          self.conta = conta
5          self.nome = nome
6          self.__saldo = 0
7
8      def depositar(self, valor):
9          self.saldo += valor
10
11
12  conta = Conta('Yan', '0001', '1234')
13  conta.__saldo = 1000
14  print(f"O saldo da conta é: R$ {conta.__saldo}")
```

Que resultado teremos na atribuição do novo saldo?

Mas o uso do underscore apenas sinaliza que o recurso não deveria ser acessado ou modificado diretamente fora da classe, mas não bloqueia de ser acessado/modificado. Para isso, o Python possui o decorator **@property** que nos permite restringir acesso a variáveis de uma classe.

@property é usado para dar funcionalidade "especial" a certos métodos para fazê-los agir como getters, setters ou deleters quando definimos propriedades em uma classe. Sendo um condicional para nossa estabilidade de informações.

Exemplo Prático 2

```
1 class Conta:
2 > def __init__(self, nome, agencia, conta):...
7
8 @property
9 def saldo(self):
10 |     return self.__saldo
11
12 @saldo.setter
13 def saldo(self, novo_saldo):
14 |     raise ValueError("Impossível alterar saldo diretamente. Use a função depositar() ou sacar().")
15
16 def depositar(self, valor):
17 |     self.__saldo += valor
18
19 def sacar(self, valor):
20 |     self.__saldo -= valor
21
22
23 conta = Conta('Yan', '0001', '1234')
24 conta.depositar(2000)
25 print(f"O saldo da conta é: R$ {conta.saldo}")
```

Retornará erro caso tentar definir o saldo diretamente.

Exercício 1 (30min)

Construir aplicação de venda de passagens de um ônibus com 46 lugares, o menu da aplicação deve conter os seguintes itens:

- Comprar Passagem (Mostrar apenas caso ainda não tenha comprado)
- Cancelar Passagem (Mostrar quando tiver poltrona selecionada)
- Alterar Poltrona (Caso já tenha poltrona escolhida)
- Sair

As propriedades de valor da compra e poltrona devem ser construídas com encapsulamento, não permitindo que o cliente altere diretamente o valor da variável. Ao final da aplicação, informar qual poltrona o usuário selecionou e o valor total da compra ou mostrar que a compra foi cancelada sem custo.

- Encapsulamento em Python - <https://acervolima.com/encapsulamento-em-python/>
- Herança e Polimorfismo - https://tmfilho.github.io/pyestbook/objects/03_inhe.html
- Lista de exercícios de POO em Python com resolução - <https://docplayer.com.br/196038860-Lista-de-exercicios-de-poo-em-python-e-java-com-solucao.html>



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>

AGENDA

- Informações do template
- Exemplo de texto (Light mode)
- Exemplo de texto (Dark mode)
- Exemplo de tópicos (Light mode)
- Exemplo de tópicos (Dark mode)
- Exemplo com imagens (Light mode)
- Exemplo com imagens (Dark mode)
- Exemplo com tabelas (Light mode)
- Exemplo com tabelas (Dark mode)

INFORMAÇÕES DO TEMPLATE

- Título da Apresentação:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 34
 - Cor: Branco
- Título do Slide:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 22
 - Cor: Branco
- Parágrafos:
 - Fonte: Open Sans Normal
 - Tamanho: 14 a 18
 - Cores: Branco (Dark Mode) ou Preto (Light Mode)
- Marcadores de tópicos:
 - Formatos: Símbolos ou Alfanuméricos
 - Cor: Laranja
- Padrão de Cores:
 - Cinza - #868584
 - Preto - #1C1C19
 - Branco - #FAFAFA
 - Laranja - #F08305
 - Rosa - #c71d81
 - Azul - #0e1d8e

EXEMPLO DE TEXTO (LIGHT MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TEXTO (DARK MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TÓPICOS (LIGHT MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO DE TÓPICOS (DARK MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO COM IMAGENS (LIGHT MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM IMAGENS (DARK MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM TABELAS (LIGHT MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx

EXEMPLO COM TABELAS (DARK MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx