

IN

dev full stack- Python

Aula 5 Python OO

# Revisão



## Encapsulamento



# Encapsulamento

- Proteger os atributos da nossa classe;
- Inserir regras nas atribuições de valores;

# Encapsulamento - Análise

Produto.py > ...

```
1 class Produto:
2     def __init__(self, nome, preco, descricao):
3         self.nome = nome
4         self.preco = preco
5         self.descricao = descricao
6     #preços podem ter até 10% de desconto
```

main.py > ...

```
1 from Produto import Produto
2
3 produto = Produto("Escova", 30.5, "Escova de cabelo")
4 produto.preco = 10.0
5
6 print("Produto: ", produto.nome, "\nValor: ", produto.preco)
```

# Encapsulamento

---

- Como criamos regras para classe???

**MÉTODOS!**

# GET

- Retorna valores dos atributos do objeto

```
@property  
def preco(self):  
    return self.__preco
```

# SET

- Atribui valores aos atributos do objeto com regras que desejarmos.

```
@preco.setter
def preco(self, preco):
    preco_min = self.__preco * 0.10;
    preco_min = self.__preco - preco_min;
    if preco_min >= self.__preco:
        self.__preco = preco
        return True
    else:
        return False
```

# Encapsulamento

```
from Produto import Produto
```

```
produto = Produto("Escova", 30.5, "Escova de cabelo")  
produto.preco = 15.0  
print(produto.preco)
```

**Resultado: 30.5**



# Atividade

- Desenvolva um sistema onde possa ser cadastrado dados de um funcionário que terá nome, salário, matrícula e função. Os valores serão cadastrados pelo RH no momento de criação do cadastro. Porém depois de um tempo, será possível alterar o salário do funcionário em até 20% a mais. E nunca será possível alterar o salário para um valor menor.

# Associação

---

# Associação

- Uma classe com tipo de dados de uma outra classe
- Lê-se “Tem um”
- Define como as classes interagem entre elas.



# Associação

```
class Endereco:
    def __init__(self, cep, rua, numero, cidade, estado, pais, complemento=None):
        self.cep = cep
        self.rua = rua
        self.numero = numero
        self.cidade = cidade
        self.estado = estado
        self.pais = pais
        self.complemento = complemento
```

# Associação

```
class Cliente:
    def __init__(self, nome, email, senha, endereco):
        self.__nome = nome
        self.__email = email
        self.__senha = senha
        self.endereco = endereco
```

```
@property
def nome(self):
    return self.__nome

@nome.setter
def nome(self, nome):
    self.__nome = nome

@property
def email(self):
    return self.__email

@email.setter
def email(self, email):
    self.__email = email
```

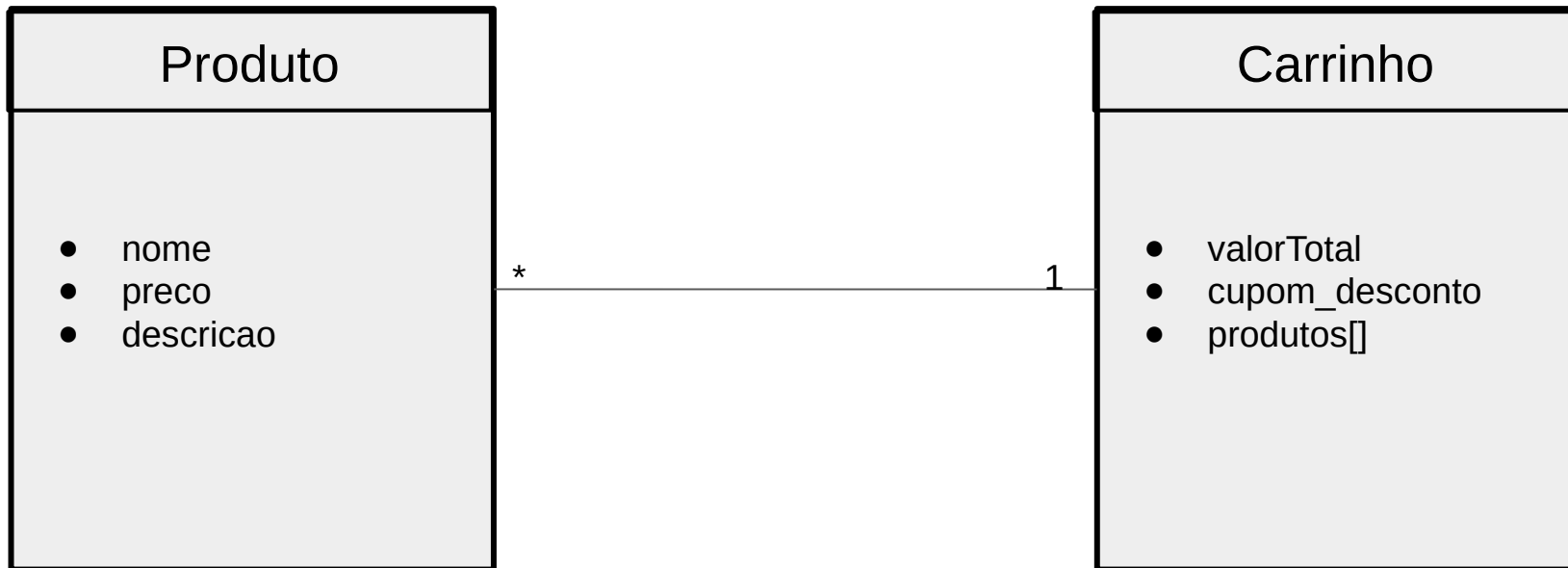
```
@property
def senha(self):
    return self.__senha

@senha.setter
def senha(self, senha):
    self.__senha = senha
```

 main.py > ...

```
1  from Endereco import Endereco
2  from Cliente import Cliente
3  #DADOS DO CLIENTE
4  nome = input("Informe o seu nome: ")
5  email = input("Informe o seu email: ")
6  senha = input("Informe a sua senha: ")
7  #DADOS DO ENDEREÇO DO CLIENTE
8  cep = input("Informe o cep da rua: ")
9  rua = input("Informe a rua: ")
10 numero = input("Informe o número: ")
11 cidade = input("Informe a cidade: ")
12 estado = input("Informe o estado: ")
13 pais = input("Informe o país: ")
14 #Criando endereço
15 endereco1 = Endereco(cep, rua, numero, cidade, estado, pais)
16 #Criando um cliente
17 cliente1 = Cliente(nome, email, senha, endereco1)
18 print("Nome: ",cliente1.nome)
19 print("Email: ",cliente1.email)
20 print("Endereço: ",cliente1.endereco.rua, ", ", cliente1.endereco.numero)
```

# Agregação



# Agregação

```
class Produto:
    def __init__(self, nome, preco, descricao):
        self.__nome = nome
        self.__preco = preco
        self.__descricao = descricao

    @property
    def preco(self):
        return self.__preco

    @preco.setter
    def preco(self, preco):
        preco_min = self.__preco * 0.10;
        preco_min = self.__preco - preco_min;
        if preco_min >= self.__preco:
            self.__preco = preco
            return True
        else:
            return False
```

Carrinho.py > ...

```
1 class Carrinho:
2     def __init__(self, produtos):
3         self.__produtos = produtos
4
5     def valorTotal(self):
6         total = 0.0
7         for p in self.__produtos:
8             total += p.preco
9         return total
0
```



# Agregação

main.py > ...

```
1  from Produto import Produto
2  from Carrinho import Carrinho
3
4  produto1 = Produto("Escova", 30.5, "Escova de cabelo")
5  produto2 = Produto("Garrafa de água", 3.5, "Água mineral")
6  produto3 = Produto("Carteira", 80.0, "Carteira de couro")
7  produtos = [produto1, produto2, produto3]
8  carrinho = Carrinho(produtos)
9
10 print(carrinho.valorTotal())
11
12
```

# Atividade

---

- Incremente na atividade da Folha de pagamento do RH (realizada na aula de encapsulamento), mostrando todos os valores da folha da empresa.

# Atividade

- Desenvolva um sistema capaz de cadastrar alunos com nome, semestre e matrícula e esses alunos terão notas. As notas poderão ser cadastradas com um nome (ex: nota da prova 1), disciplina (ex: Lógica de programação) e valor (ex: 8.9). O aluno terá várias notas e ao final poderemos saber o score do aluno (soma das notas de todas as disciplinas).

# Atividade

- Desenvolva um sistema para um processo seletivo. Serão capturados vários candidatos e eles farão uma prova que será a primeira etapa do processo. Todos os candidatos deverão ter informações de nome, endereço, tempo de experiência e descrição do candidato. A prova será feita em uma data específica e terá uma pontuação obtida pelo o candidato. Teremos uma lista de candidatos aprovados, que são apenas os candidatos que tiveram uma nota superior a 8.

The logo consists of the letters 'IN' in a white, serif font, centered within a solid red square. The background of the entire image is a solid red color with several overlapping, semi-transparent red rectangular shapes that create a sense of depth and movement, resembling architectural elements or stylized steps.

# IN

71 3901 1052 | 71 9 9204  
0134

@infinity.school

[www.infinityschool.com.br](http://www.infinityschool.com.br)

Salvador Shopping Business | Torre Europa Sala 310  
Caminho das Árvoreas, Salvador - BA CEP: 40301-155