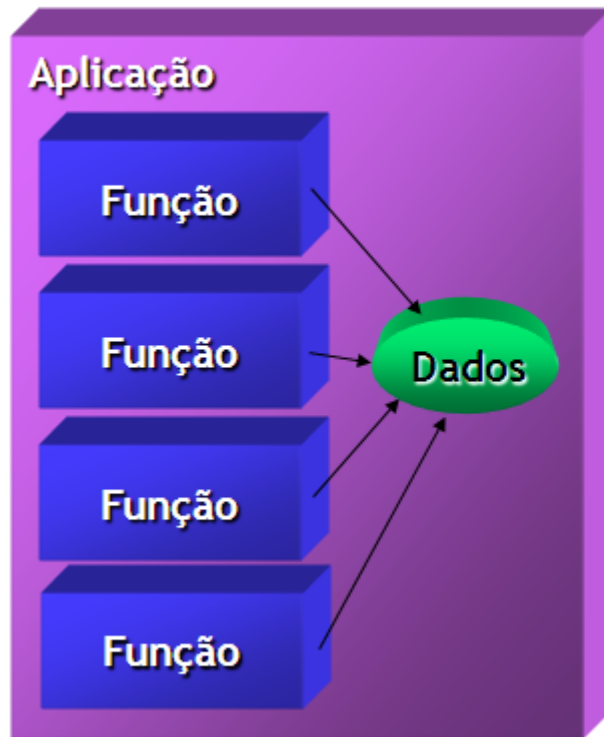
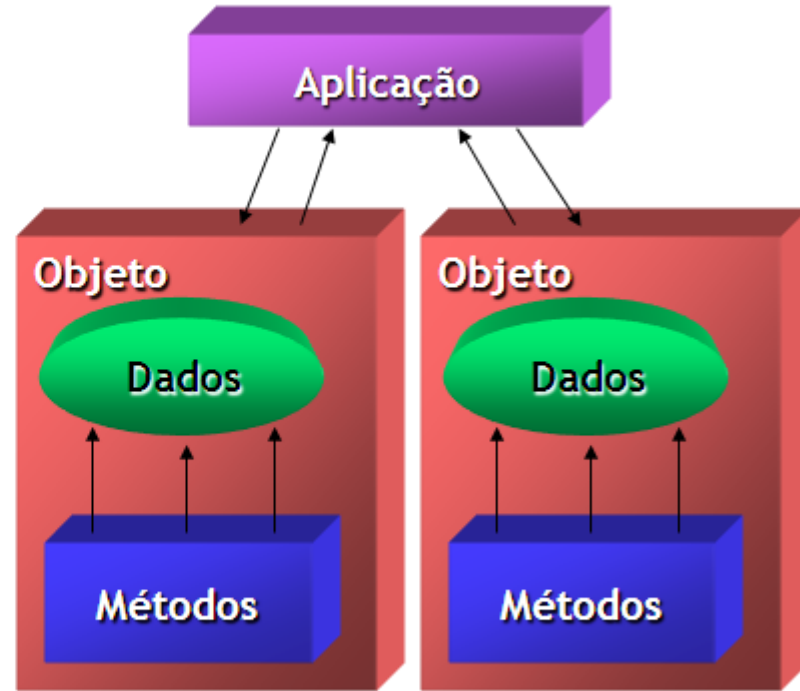


Orientação a objetos.

Estruturada



Orientação a Objetos



Python OpenCv

Transformar Python em um executavel, instalar pyinstaller:

```
pip install pyinstaller
```

Entre no diretorio e execute o comando:

Cria com arquivos de dependências o comando abaixo

```
pyinstaller arquivo.py
```

Cria somente o .exe

```
pyinstaller --onefile arquivo.py
```

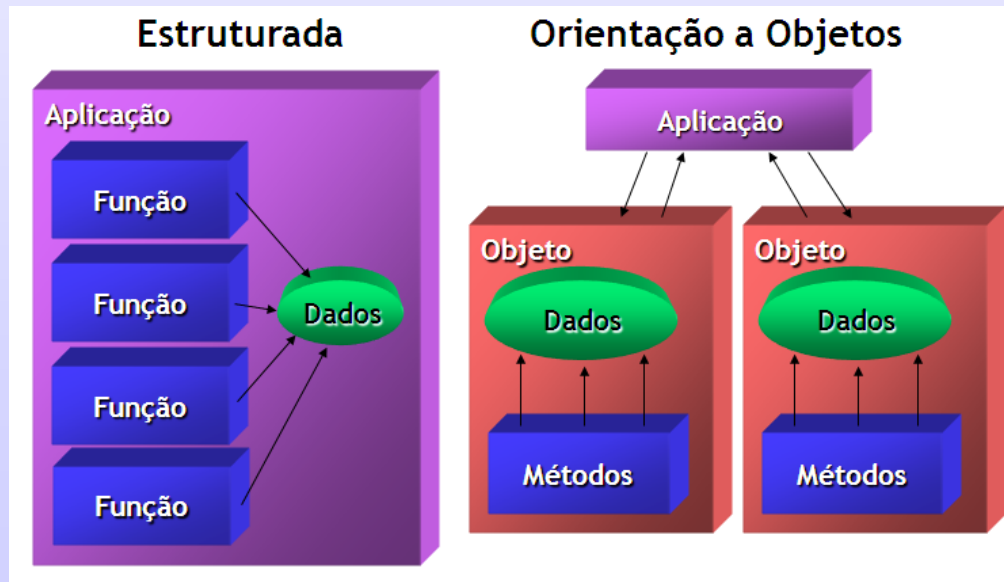
Cria somente o .exe e permite escrita

```
pyinstaller --onefile -w arquivo.py
```

Orientação a objetos.

O que é um objeto?

É uma coisa real ou abstrata que pode ser percebida pelos sentidos e descrita por meio das suas características, comportamento e estado atual.



Orientação a objetos.

O que é uma classe?

É uma espécie de molde para classificar o objeto.



Orientação a objetos.

Uma classe eu tenho que responder 3 perguntas:

- Coisas que tenho.
- Coisas que eu faço.
- Como estou agora.



Orientação a objetos.

Exemplo de uma classe de um objeto caneta:

- Que coisas tenho no objeto caneta?
Modelo , cor, ponta, carga, e tampada
- Que coisas que eu faço?
Escrever, rabiscar, pintar e tampar
- Como estou agora?



Orientação a objetos.

Exemplo de uma classe de um objeto caneta em POO:

- Atributo, para ficar mais fácil o entendimento pode ser comparado com uma variável na programação estruturada
Modelo , cor, ponta, carga, e tampada
- Método, para ficar mais fácil o entendimento pode ser comparado com as funções em programação estruturada
Escrever, rabiscar, pintar e tampar
- Estado.



Orientação a objetos.



The diagram shows a pen object on the left, represented by a vertical line with a triangular tip. A line points from the tip of the pen to the 'Classe Caneta' definition on the right.

```
Classe Caneta
modelo: Caractere
cor: Caractere
ponta: Real
carga: Inteiro
tampada: Logico
Metodo rabiscar()
    Se (tampada) entao
        Escreva("ERRO")
    senao
        Escreva("Rabisco")
    FimSe
FimMetodo
Metodo tampar()
    tampada = verdadeiro
FimMetodo
FimClasse
```

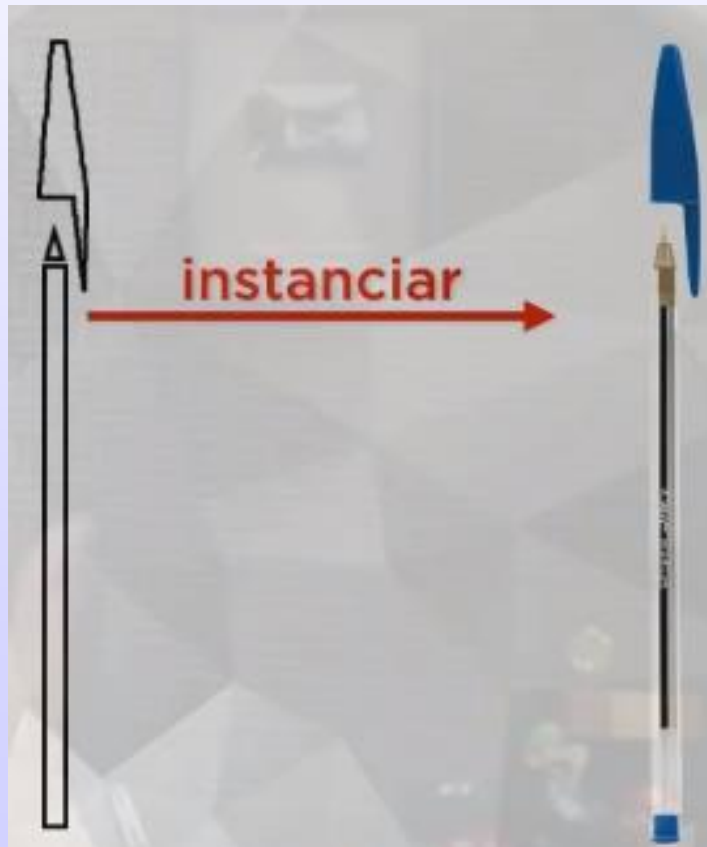

Orientação a objetos.



Orientação a objetos.

O que é instanciar?

Instanciar é pegar uma classe e transformar em um objeto



Orientação a objetos.



Orientação a objetos.

O que é uma classe?

Uma classe define os atributos e métodos comuns que serão compartilhados por um objeto.



Orientação a objetos.

O que é um objeto?

É a instancia de uma classe.



peessoa01 ---- Classe Pessoa.py

```
class Pessoa:
    #metodo construtor
    def __init__(self, nome, idade, comendo=False, falando=False):
        self.nome = nome
        self.idade = idade
        self.comendo = comendo
        self.falando = falando
    #metodo simples
    def comer(self, alimento):
        if self.comendo:
            print(f'{self.nome} já está comendo.')
            return
        print(f'{self.nome} está comendo {alimento}.')
        self.comendo = True
    #metodo simples
    def parar_comer(self):
        if not self.comendo:
            print(f'{self.nome} não esta comendo.')
            return
    #motodo simples
    def falar(self):
        print('Pessoa está falando....')
```

peessoa01 ---- Classe main.py

```
from pessoa import Pessoa
p1 = Pessoa('Eder ', 36)
p1.parar_comer()
p1.comer('Burger')
p1.falar()
```

peessoa02 ---- Classe Pessoa.py

```
class Pessoa:
    ano_atual = 2019

    def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade

    #metodo simples
    def get_ano_nascimento(self):
        print(self.ano_atual - self.idade)

    @classmethod
    def por_ano_nascimento(cls, nome, ano_nascimento):
        idade = cls.ano_atual - ano_nascimento
        return cls(nome, idade)
```


peessoa02 ---- Classe main.py

```
from pessoa import Pessoa
p1 = Pessoa('Eder ', 36)
print(p1)
print(p1.nome, p1.idade)
p1.get_ano_nascimento()
p2 = Pessoa.por_ano_nascimento('Carlos', 1983)
print(p2)
print(p2.nome, p2.idade)
p2.get_ano_nascimento()
```

peessoa03 ---- Classe Pessoa.py

```
from random import randint
class Pessoa:
    ano_atual = 2019

    def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade
    #metodo simples
    def get_ano_nascimento(self):
        print(self.ano_atual - self.idade)

    @classmethod
    def por_ano_nascimento(cls, nome, ano_nascimento):
        idade = cls.ano_atual - ano_nascimento
        return cls(nome, idade)
    @staticmethod
    def gera_id():
        rand = randint(10000, 19999)
        return rand
```

peessoa03 ---- Classe main.py

```
from pessoa import Pessoa
p1 = Pessoa('Eder ', 36)
print(p1)
print(p1.nome, p1.idade)
p1.get_ano_nascimento()
p2 = Pessoa.por_ano_nascimento('Carlos', 1983)
print(p2)
print(p2.nome, p2.idade)
p2.get_ano_nascimento()
print(Pessoa.gera_id())
print(p1.gera_id())
```