

Aulas de Django e Python

Professor Anderson Henrique



PОО – Programação Orientada a objetos

- Herança é quando temos uma classe-pai, definida com suas propriedades e métodos e criamos uma classe-filha que herda todas as propriedades e métodos da classe-pai, assim, podemos ter objetos que aproveitam rotinas e variáveis sem que precisemos reescrevê-las
- Sabemos que uma classe é filha de outra quando olhamos na declaração da classe entre parêntesis o nome da classe-pai, vamos praticar:

- Definindo uma classe-pai (NPC) com construtor e método:

```
class NPC : #Base, Pai, Super
    #construtor
    def __init__(self, nome, time, forca, municao):
        self.nome = nome
        self.time = time
        self.forca = forca
        self.municao = municao
        self.vivo = True
        self.energia = 100
```

#método

```
def info(self) :
```

```
    print('Nome....: ' + self.nome)
```

```
    print('Time....: ' + str(self.time))
```

```
    print('Forca...: ' + str(self.forca))
```

```
    print('Municao.: ' + str(self.municao))
```

```
    print('Vivo....: ' + ('sim' if self.vivo else 'nao'))
```

```
    print('Energia.: ' + str(self.energia))
```

```
    print('-----')
```

#classe Soldado herda a classe NPC

```
class Soldado(NPC) :
```

```
    def __init__(self, nome, time):
```

```
        self.forca = 200
```

```
        self.municao = 200
```

```
        super().__init__(nome, time, self.forca, self.municao)
```

#classe Guarda herda a classe NPC

```
class Guarda(NPC) :
```

```
    def __init__(self, nome, time):
```

```
        self.forca = 100
```

```
        self.municao = 20
```

```
        super().__init__(nome, time, self.forca, self.municao)
```

#classe Elite herda a classe NPC

```
class Elite(NPC) :
```

```
    def __init__(self, nome, time):
```

```
        self.forca = 400
```

```
        self.municao = 500
```

```
        super().__init__(nome, time, self.forca, self.municao)
```

```
#método
```

```
def inf(self):
```

```
    super().info()
```

#instanciando os objetos

s1 = Guarda('Olho Vivo', 1)

s2 = Soldado('Bala na Agulha', 1)

s3 = Elite('Ninja', 1)

s4 = Guarda('Super Atento', 2)

s5 = Soldado('Tiro Certo', 2)

s6 = Elite('Samurai', 2)

s1.vivo = False

s6.vivo = False

#executando o método da classe-pai

s1.info()

s2.info()

s3.info()

s4.info()

s5.info()

#executando o método da própria classe-filha Elite

s6.inf()