

EEEP Luiza de Teodoro Vieira

Classes e objetos

Equipe 2

Adriel Levi nº 01

João Hélió nº 15

Lucas de Sousa nº 23

Warley Pereira nº 40

Pacatuba – mar2023

## 1. Resumo

Um objeto é um elemento computacional que representa, no domínio da solução, alguma entidade (concreta ou abstrata) definida por atributos e ações dentro do domínio de interesse do problema em questão. Objetos similares entre si são agrupados em classes. Objetos são convertidos em função por valor, ou seja, uma cópia do objeto é feita quando ele é passado para a função. Pode-se dizer, portanto, que um novo objeto é criado. Da mesma forma, uma função pode ter como tipo de retorno uma classe, e assim retornar um objeto para a função anterior.

Classe pode ser definida como um agrupamento de objetos semelhantes entre si. Formalmente, é um conceito que empacota abstrações de dados e procedimentos que descrevem o conteúdo e o comportamento de entidades do mundo real, representadas por objetos. Que se torna um código da linguagem de programação orientada a objetos que define e implementa um novo tipo de objeto, que terão características/atributos que guardam dados ou valores e, também, funções específicas para manipular o objeto. A classe é um elemento primordial de um diagrama de classes; modelagem importante na programação orientada a objetos.

Objeto é uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Atributos são características de um objeto. Basicamente a estrutura de dados que vai representar a classe. Quando se quer ver de forma concreta uma classe, como por exemplo, float, chega-se nos objetos. Representa-se graficamente um objeto com o mesmo símbolo de classe, sublinhando-se o seu nome. Sabendo que uma classe de modo geral seria uma grande parte e o objeto seria uma pequena parte onde existem vários atributos/características e também as suas funções ou métodos onde fariam diversas atividades. Tal que essa atividade seja uma ação.

## 2. Introdução

Uma classe encapsula um conjunto de dados (variáveis) com um conglomerado de funções que operam sobre esses dados. O objetivo é alcançar um código mais modular através do agrupamento de dados e funções em unidades gerenciáveis (normalmente pequenas). A grande maioria dos cálculos matemáticos podem ser programados facilmente sem o uso de classes, mas em vários problemas, as classes tornam as soluções mais elegantes e/ou deixam o código mais fácil e prático de ser estendido a um novo estágio. No mundo não matemático,

existem conceitos e algoritmos a eles associados que ajudam a estruturar a solução de um problema. Classes podem ajudar no entendimento e resolução do problema e contribuir para simplificar a modelagem dos dados e as ações em programas. Classe e Objeto são um complemento um do outro, onde sem a classe não tem a função do objeto, e onde sem o objeto não a função a classe uma via dupla, de todo e atributos e métodos para determinado objeto.

### 3. Desenvolvimento

Os objetos têm individualidade e vários nomes (em vários escopos) e podem ser vinculados ao mesmo objeto. Isso é conhecido como aliasing em outros idiomas. Isso geralmente não é apreciado à primeira vista no Python e pode ser ignorado com segurança ao lidar com tipos básicos imutáveis (números, strings, tuplas). No entanto, o aliasing tem um efeito possivelmente surpreendente na semântica do código Python envolvendo objetos mutáveis, como listas, dicionários e a maioria dos outros tipos. Isso geralmente é usado em benefício do programa, pois os aliases se comportam como ponteiros em alguns aspectos.

As classes fornecem um meio de agrupar dados e funcionalidades. A criação de uma nova classe cria um novo *tipo* de objeto, permitindo que novas *instâncias* desse tipo sejam feitas. Cada instância de classe pode ter atributos anexados a ela para manter seu estado. As instâncias de classe também podem ter métodos (definidos por sua classe) para modificar seu estado.

Comparado com outras linguagens de programação, o mecanismo de classe do Python adiciona classes com um mínimo de nova sintaxe e semântica. É uma mistura dos mecanismos de classe encontrados em C++ e Modula-3. As classes fornecem todos os recursos padrão da Programação Orientada a Objetos: o mecanismo de herança de classe permite várias classes base, uma classe derivada pode substituir quaisquer métodos de sua classe ou classes base e um método pode chamar o método de uma classe base com o mesmo nome. Os objetos podem conter quantidades e tipos de dados arbitrários. Como acontece com os módulos, as classes participam da natureza dinâmica da programação: elas são criadas em tempo de execução e podem ser modificadas posteriormente após a criação.

Na terminologia C++, normalmente os membros de classe são públicos e todas as funções de membro são virtuais. Como no Modula-3, não há atalhos para referenciar os membros do objeto a partir de seus métodos: a função do método é declarada com um primeiro argumento explícito representando o objeto, que é fornecido implicitamente pela chamada. Como em Smalltalk, as próprias classes são objetos. Isso fornece semântica para importar e

renomear. Ao contrário de C++ e Modula-3, os tipos internos podem ser usados como classes base para extensão pelo usuário. Além disso, como em C++, a maioria dos operadores integrados com sintaxe especial (operadores aritméticos, subscritos, etc.) podem ser

redefinidos para instâncias de classe. A capacidade da Classe e de um Objeto de se misturar é a de um complementar o outro, tal que, a Classe seja o todo e o Objeto tem atributos métodos. São as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos tendo até uma união.

#Fazendo um programa que mostre a classe “Carro”, e criar um objeto chamado “meu carro” com seus atributos.

```
class Carro:
```

```
    def __init__(self, cor, marca, ano, velocidade):
```

```
        self.cor = cor
```

```
        self.marca = marca
```

```
        self.ano = ano
```

```
        self.velocidade = velocidade
```

```
# Cria um objeto Carro
```

```
meu_carro = Carro('Roxo', 'Onix', 2020, 180)
```

```
# Imprime as características e atributos do objeto
```

```
print("A cor do meu carro é:",meu_carro.cor)
```

```
print("A marca do meu carro é:",meu_carro.marca)
```

```
print("O ano do meu carro é:",meu_carro.ano)
```

```
print("A velocidade máxima do meu carro é:",meu_carro.velocidade)
```



Figura 1 - Classe e Objetos – Exemplo

#### 4. Conclusão

E como podemos ver a expressão "classe e objeto" se dar por uma classe ser um "geral" e o objeto ser algo específico dessa classe. Como por exemplo, temos a classe animal e temos os objetos que são gato, coelho e cachorro. Onde os Objetos podem ganhar características ou atributos que faz com que o objeto seja único, e se diferencie dos outros. Além de que a classe abrange uma grande parte de conteúdo dentro dela, voltando ao exemplo que usamos anteriormente vemos que a classe "animal" pode abranger uma espécie, ou várias espécies como no exemplo vimos 3 espécies diferentes. Essa é a função da classe, abranger um total um todo, onde os objetos irão servir com suas funções diferentes e suas características.

#### 5. Bibliografia

1. Disponível em SlidePlayer

<https://slideplayer.com.br/slide/397636/3/images/3/X+Classes+e+Objetos+Classe+Objeto+Abstra%C3%A7%C3%A3o+Concreto+Molde.jpg> acessado em:

08mar2023.

2. Disponível em Hashtag treinamentos

[https://www.hashtagtreinamentos.com/classes-no-python?gclid=Cj0KCQiA0oagBhDHARIsAI-BbgdDK8BAcyq65Y-1PsMw-QphF4H1rKRmfAyYh-sbf\\_nnT8QS1xFalIaAt2kEALw\\_wcB](https://www.hashtagtreinamentos.com/classes-no-python?gclid=Cj0KCQiA0oagBhDHARIsAI-BbgdDK8BAcyq65Y-1PsMw-QphF4H1rKRmfAyYh-sbf_nnT8QS1xFalIaAt2kEALw_wcB) acessado em: 03mar2023.

3. Disponível em Fundação Bradesco

<https://www.ev.org.br/cursos/desenvolvimento-orientado-a-objetos-utilizando-a-linguagem-python> acessado em: 03mar2023.

4. Disponível em DevMedia

<https://www.devmedia.com.br/principais-conceitos-da-programacao-orientada-a-objetos/32285> acessado em: 03mar2023.