

Classes

Site: [HackaTruck MakerSpace](#)
Curso: Conceitos e Fundamentos: Algoritmos e Programação Orientada a
Objetos com Swift
Livro: Classes

Impresso por: Matheus Ferreira Santos
Data: terça-feira, 15 ago. 2023, 18:15

Índice

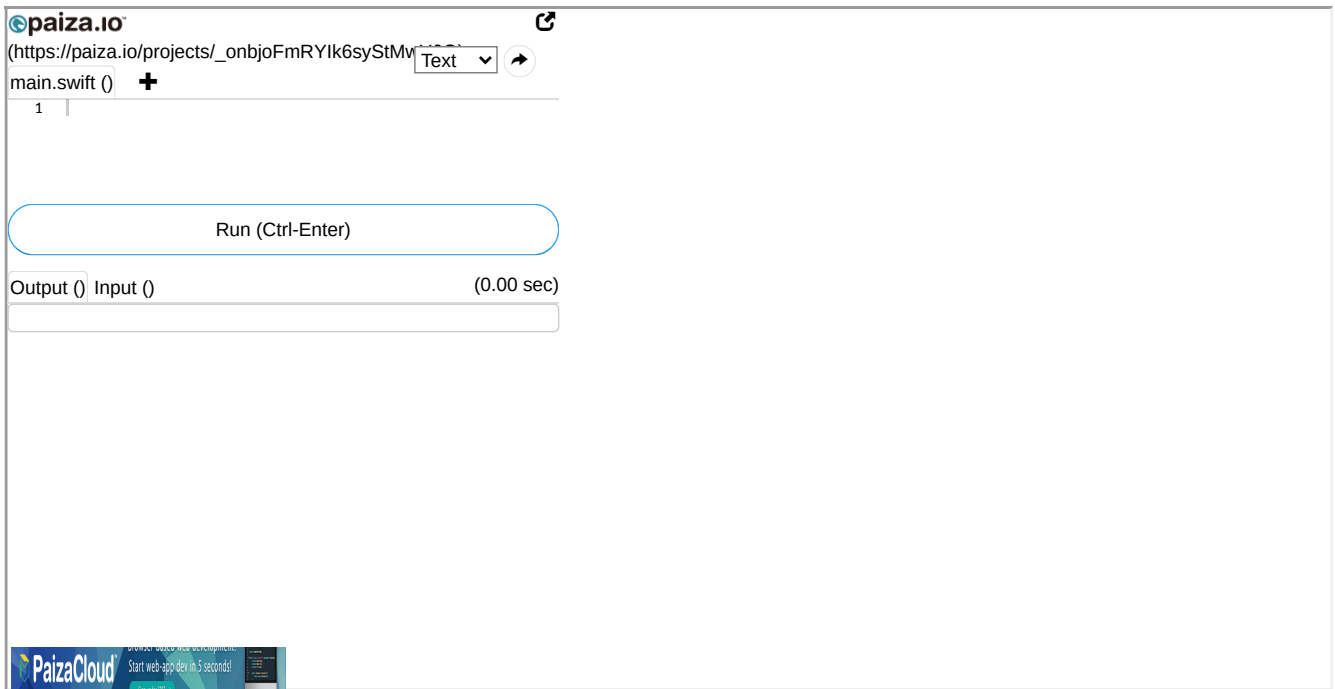
1. Introdução
2. Atributos
3. Métodos
4. Objetos
5. Mensagens

1. Introdução



Classe é um termo utilizado em Orientação a Objetos (OO) para designar um grupo de elementos com determinadas características. Uma classe pode ser considerada como uma base sobre a qual serão criados os objetos, com atributos e métodos, ou seja, ela descreve as características e comportamentos. A composição de uma classe se dá por 3 postos-chaves, o **NOME** da classe, o conjunto de **ATRIBUTOS** da classe e por fim o conjunto de **MÉTODOS** da classe. Vamos ver alguns exemplos de classe:





As classes, como podemos ver, são a implementação (descrição) do que virá a ser um objeto. Dessa forma, as classes possuem atributos e métodos. Deixaremos 2 descrições para irem se acostumando:

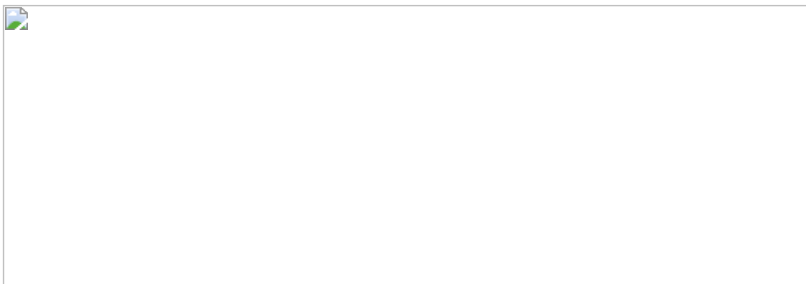
- **CLASSE:** é a descrição geral (implementação) de um objeto, com atributos e métodos;
- **OBJETO:** é uma instância de uma classe, com valores específicos para os atributos.

Não se preocupe se a definição de objeto não ficou clara ainda, debateremos mais à frente, ainda neste capítulo! Por ora, voltemos à definição de classe e suas observações.

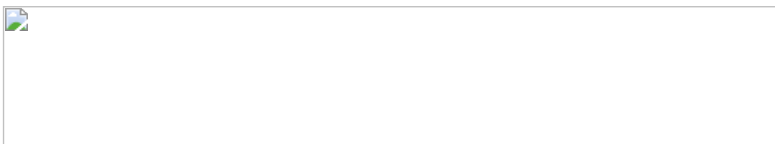
Para definirmos uma classe precisamos:

- O **nome** da classe;
- O nome e tipo das suas variáveis (**atributos**);
- O nome e especificação das suas funcionalidades (**métodos**).

Classe = Nome + Atributos + Métodos.



A classe acima pode ser instanciada, por exemplo, nos seguintes objetos:



Em termos de implementação, a definição da classe e a instanciação dos objetos do exemplo acima, podem ser feitos da seguinte forma:



É como se criássemos realmente o esqueleto de algo, e quando os atributos receberem os valores, por meio do objeto, ganharão vida.



O modelo (classe) estabelece as características e o comportamento que o objeto deve ter de forma genérica, propiciando que cada objeto, com seus respectivos valores, tenha seus atributos próprios. Outra característica das classes é que elas são conceitos facilmente identificadas na especificação dos sistemas, geralmente, descritas como substantivos. Assim, a classe é um conceito estático que, uma vez definido, permanece como está. Para o desenvolvimento de software são definidas as classes a serem utilizadas e seus inter-relacionamentos. Isso é feito na etapa de projeto, quando o software está sendo desenhado/modelado.

Uma vez com as classes definidas, são iniciados os procedimentos de definição para atributos e métodos.



Dica

Em nosso EAD utilizamos compiladores online, caso algum deles não carregue, basta clicar em **RUN**.

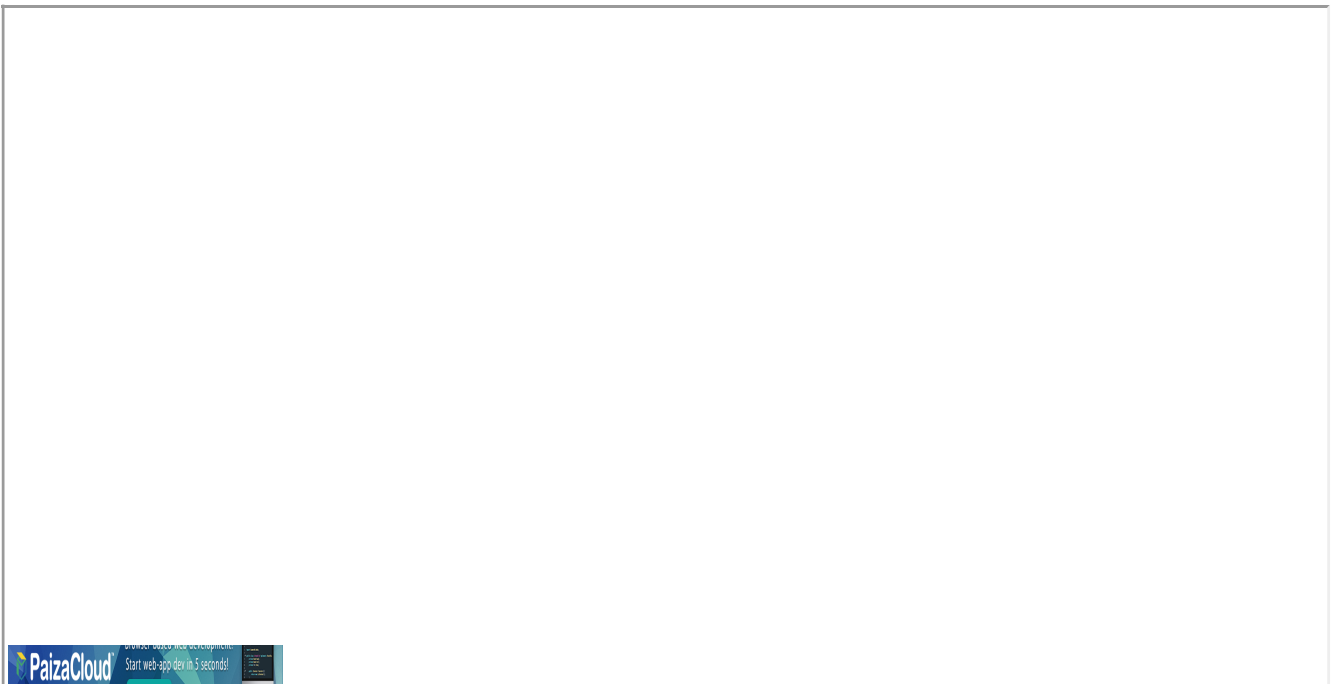
Caso persista, recarregue a página!

Outras dúvidas ou sugestões entre em contato com contato@hackatruck.com.br.

2. Atributos



Os atributos são as propriedades da classe e servem para descrever o esqueleto da classe e representá-la. Se lembram das variáveis e tipos? São elas que serão utilizadas para definir nossos atributos. Vamos criar uma classe e treinar? Criem uma classe chamada Musica, e adicionem os atributos que vocês imaginam para música.



Possível implementação:



Tudo o que foi aprendido em “Variáveis e Constantes” pode e deve ser utilizado ao declarar atributos!

3. Métodos



As ações que um objeto pode executar são chamadas de métodos. Elas definem o que será possível executar a partir dos objetos dessa classe. Um método em uma classe é apenas uma predefinição, pois a execução se dá por meio do objeto.

Uma classe, normalmente, tem diversos métodos, e cada um deles possui sua assinatura composta por um nome, o tipo de dado do retorno e sua lista de argumentos, sendo estes identificados por tipo e nome. Como vimos em “Funções”, os métodos são iniciados com a palavra reservada **func**, nome e os parâmetros, e logo após o tipo de retorno (tipo que será retornado após a execução). Vamos ver a sintaxe seguida de exemplos:



Identificamos atributos e ações compartilhadas por qualquer item da categoria música. Embora cada item tenha seus valores, nossos métodos estão preparados para estas especificidades, ou seja, para cada música que chamarmos o método, ele fará o mesmo processamento, mas com valores distintos. Dessa forma, afetamos somente quem solicitou a ação. Veremos mais na sessão “Objetos”!

4. Objetos



Um programa orientado a objetos é composto por um grupo de objetos que se comunicam através de mensagens. Um objeto é capaz de armazenar atributos e executar ações como resposta a mensagens recebidas, tal como, enviar mensagens a outros objetos. Essa troca de mensagens pode ser acionada nos objetos por meio dos métodos.

Quando criamos um objeto, o mesmo requer um espaço em memória, para assim conseguir armazenar seus atributos e seus métodos (conjuntos de ações que são definidas para o objeto).

Exemplos de objetos do tipo Humano: José, Maria, Joaquim, etc.

💡 **Repararam que Humano foi escrito com letra maiúscula? Humano? Referencia a classe e o tipo, então, uma outra possibilidade seria dizer objetos da classe Humano.**

No mundo real observamos duas características marcantes: **ESTADO** e **COMPORTAMENTO**.

O estado de um objeto nos diz sobre as propriedades dele.

Por exemplo:

- Uma pessoa tem: idade, peso, altura, cor de cabelo, cor de pele.
- Um carro tem: rodas, cor, quantidade de lugares, ano de fabricação.

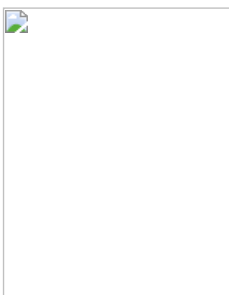
Em Orientação a Objetos chamamos essas propriedades de **Atributo**.

Já o comportamento nos diz sobre as ações que ele pode exercer/executar.

Por exemplo:

- Uma pessoa pode: andar(), falar(), ouvir(), pular().
- Um carro pode: acelerar(), frear(), buzinar().

Em Orientação a Objetos chamamos esses comportamentos de **Métodos**.



Os objetos podem representar conceitos reais (pessoa, animal, carro, pizza, etc.) ou abstratos (conta poupança, funcionário, pessoa física, etc.).

Cada objeto tem suas próprias “cópias” do que foi definido na classe, ou seja, cada um deles tem seus próprios atributos e métodos. Cada cópia representa uma instância daquela classe. Por exemplo: o carro 1, o carro 2 e o carro 3 são instâncias de Carro.



Vamos exemplificar e ver as definições do exemplo acima em código:



No exemplo, conseguimos ver que é possível criar N (diversos) objetos a partir de um só modelo (classe), cada qual com seus atributos e valores. Quando executamos a função *descricao()*, ela traz o estado atual com os valores atuais pertinentes ao objeto relativo à chamada, então, concluímos que cada objeto é capaz de executar suas próprias operações.

5. Mensagens



Um objeto isolado em um sistema não tem significado. Para que o mesmo seja útil, ele deve se relacionar com outros objetos e partes do sistema, viabilizando a troca de informações e o processamento dos dados. Essa comunicação se dá por meio de mensagens que trafegam a partir dos métodos. Veja as definições sobre mensagens:

- **Mensagens enviadas para as variáveis:** são as mensagens que resgatam ou alteram os valores de variáveis.
- **Mensagens enviadas para os métodos:** são mensagens que desencadeiam a realização dos métodos de cada objeto (chamadas dos métodos).

Uma mensagem enviada para um método deve ser composta por três partes:

- Um destino - objeto que receberá a mensagem;
- Nome do método a ser invocado pela mensagem;
- Parâmetros necessários para o método invocado.

Note que além desses três aspectos, a mensagem também pode receber uma resposta no formato do retorno do método invocado.

Faça já os exercícios desde capítulo.

Exercícios – Classes ▶