

## Conceitos Iniciais

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

Impresso por: Matheus Ferreira Santos  
Data: terça-feira, 15 ago. 2023, 18:14

## Índice

1. Introdução
2. Classes
3. Objetos
4. Propriedades
5. Métodos
6. Encapsulamento
7. Herança
8. Polimorfismo

## 1. Introdução



### CONCEITOS INICIAIS

Vamos adentrar mais um pouco no mundo da programação. Começaremos, nesse módulo a ver Orientação a Objetos, entendendo o que são esses objetos e como representá-los.

Orientação a Objetos (OO) é um paradigma utilizado para representar elementos do mundo real dentro das nossas aplicações; com ela é possível criar programas componentizados deixando partes com responsabilidades comuns agrupadas. Essas partes se comunicam por meio de mensagens e são chamadas de Objetos.

Conceitualmente, a Orientação a Objetos consiste na identificação dos objetos e de seus processamentos. O termo Objeto é utilizado para representar elementos do mundo real, onde tudo ao seu redor são exemplos de objetos: carro, mesa, janela, livro, pessoa, etc.

Veremos os conceitos de forma introdutória e depois aprofundaremos cada um deles. Dessa forma, vamos conhecer: Classes, Objetos, Propriedades, Métodos, Encapsulamento, Herança e Polimorfismo.

#### Dica

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



Caso persista, recarregue a página!

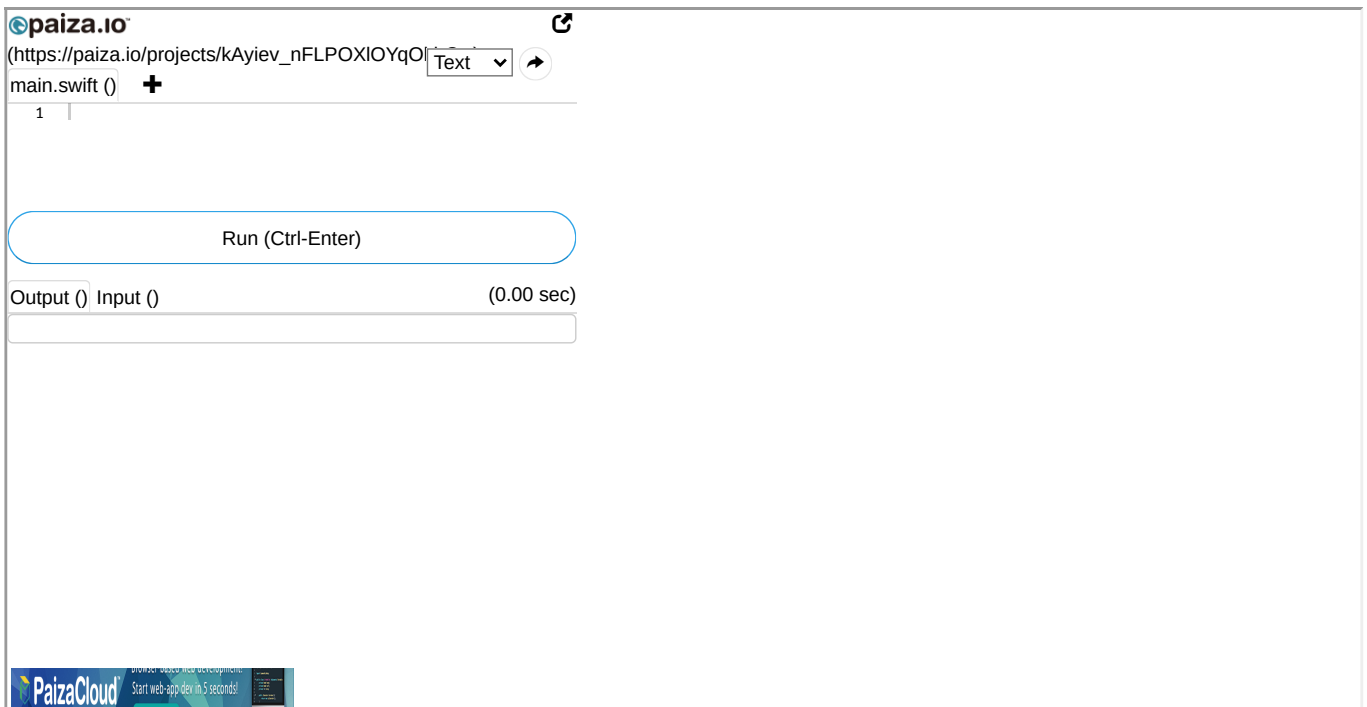
Outras dúvidas ou sugestões entre em contato com [contato@hackatruck.com.br](mailto:contato@hackatruck.com.br).

## 2. Classes

### CONCEITOS INICIAIS – ORIENTAÇÃO A OBJETOS

#### CLASSES

As classes podem ser exemplificadas por meio de agrupamentos que fazemos por semelhanças. Por exemplo, pense em um carro como um grupo, onde cada carro, seja ele utilitário ou passeio (diferentes categorias de carro), possui propriedades e atributos comuns a ambos. Exemplo de classe:



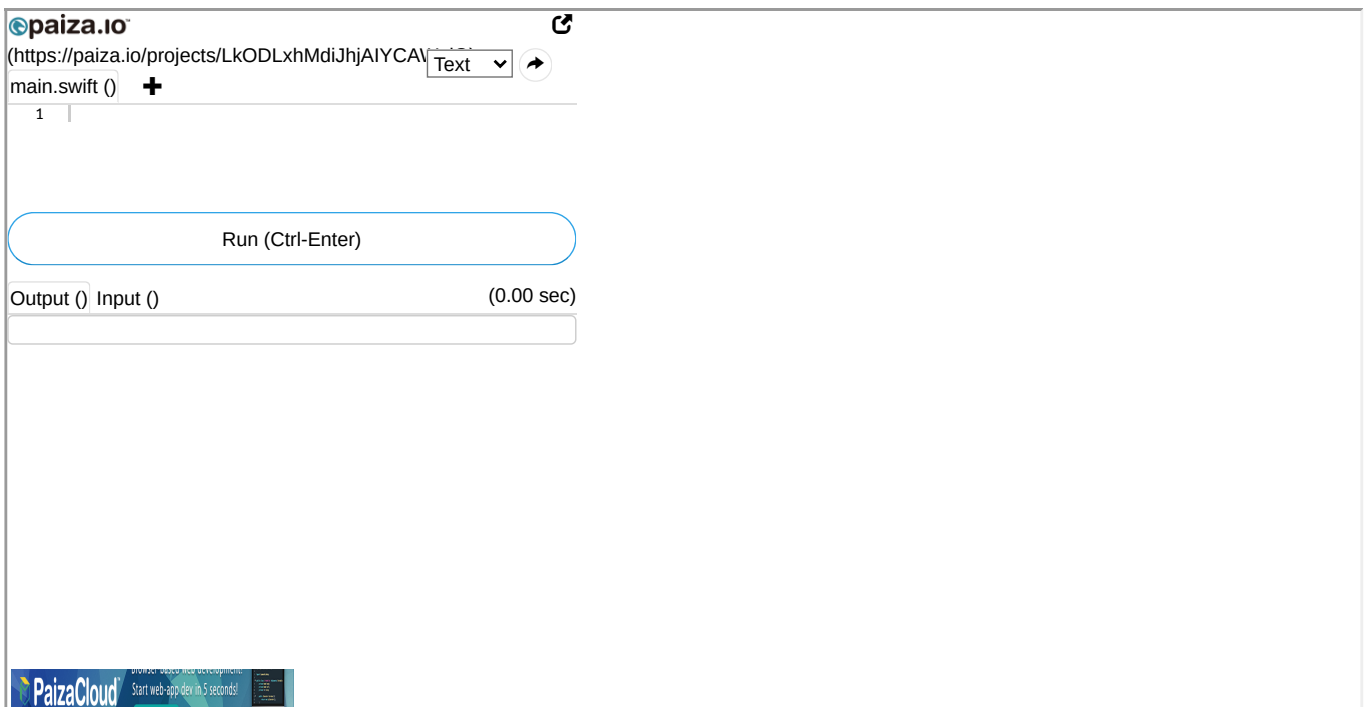
Dessa forma, declaramos uma classe chamada Carro.

### 3. Objetos



#### CONCEITOS INICIAIS - OBJETOS

Um objeto, de forma bem sucinta, pode ser definido como qualquer coisa que você vê e que "cabe" em uma classe específica. Exemplos: Gato, Cachorro, Baleia, Caneta, Lápis, Borracha, etc. A partir do nosso exemplo, a classe Carro, podemos exemplificar criando, saveiro e gol. Ambos serão objetos pertencentes à mesma classe, Carro. Código:



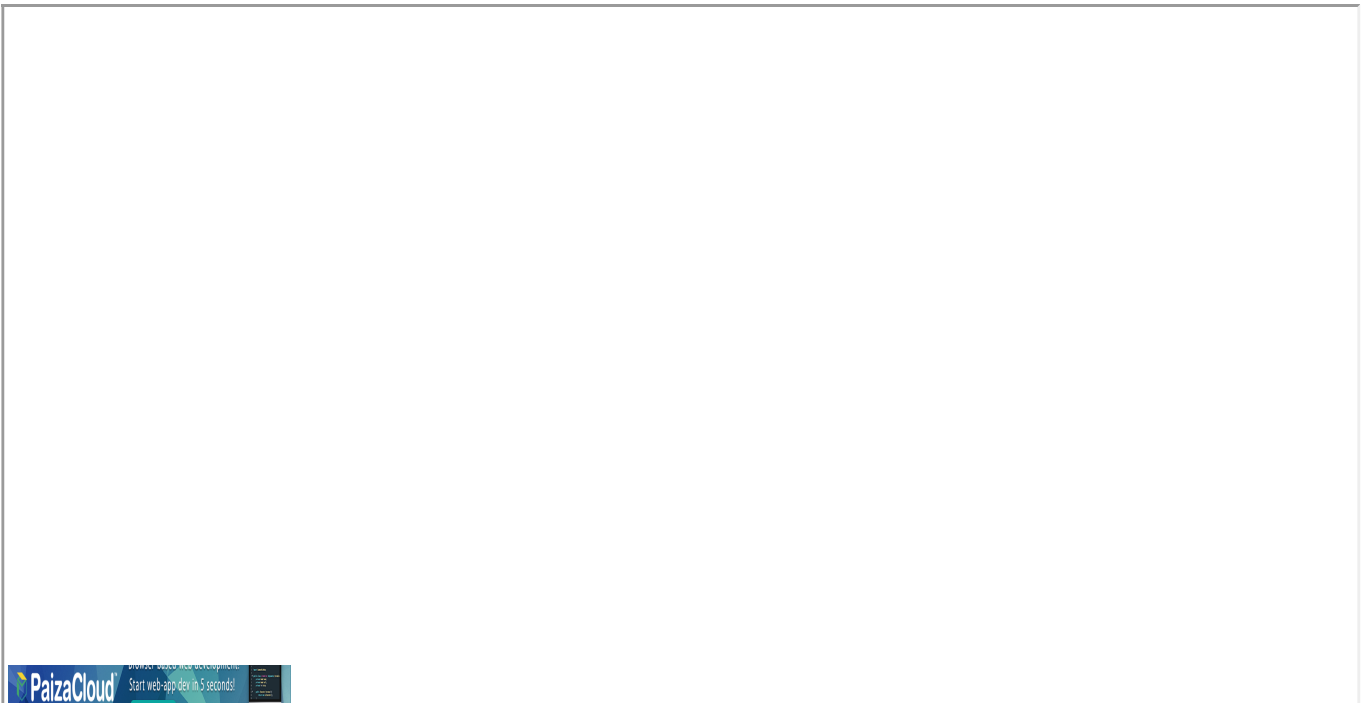
Dessa forma, declaramos dois objetos instanciando a classe Carro.

## 4. Propriedades



### CONCEITOS INICIAIS - PROPRIEDADES

Vamos voltar ao nosso exemplo da classe Carro. Como dito anteriormente, cada carro possui alguns atributos, propriedades e métodos comuns, ou seja, independentemente de ter uma determinada categoria, marca e modelo, todos possuem ano, cor, etc. Todas as informações podem ser chamadas de propriedades da classe Carro. Então, chegamos ao seguinte cenário: as propriedades são os atributos comuns dessa classe que serão compartilhados para cada objeto criado a partir dela. Vamos ver isso na prática:



Nessa primeira etapa da nossa classe, criamos algumas propriedades de carro.

💡 Repararam que utilizamos um “?” (sinal de interrogação) após o nome das variáveis? Veremos o porquê mais adiante, mas já adiantando, eles são chamados de “Opcionais”, ou seja, em algum momento estas variáveis podem não ter nenhum valor.

## 5. Métodos



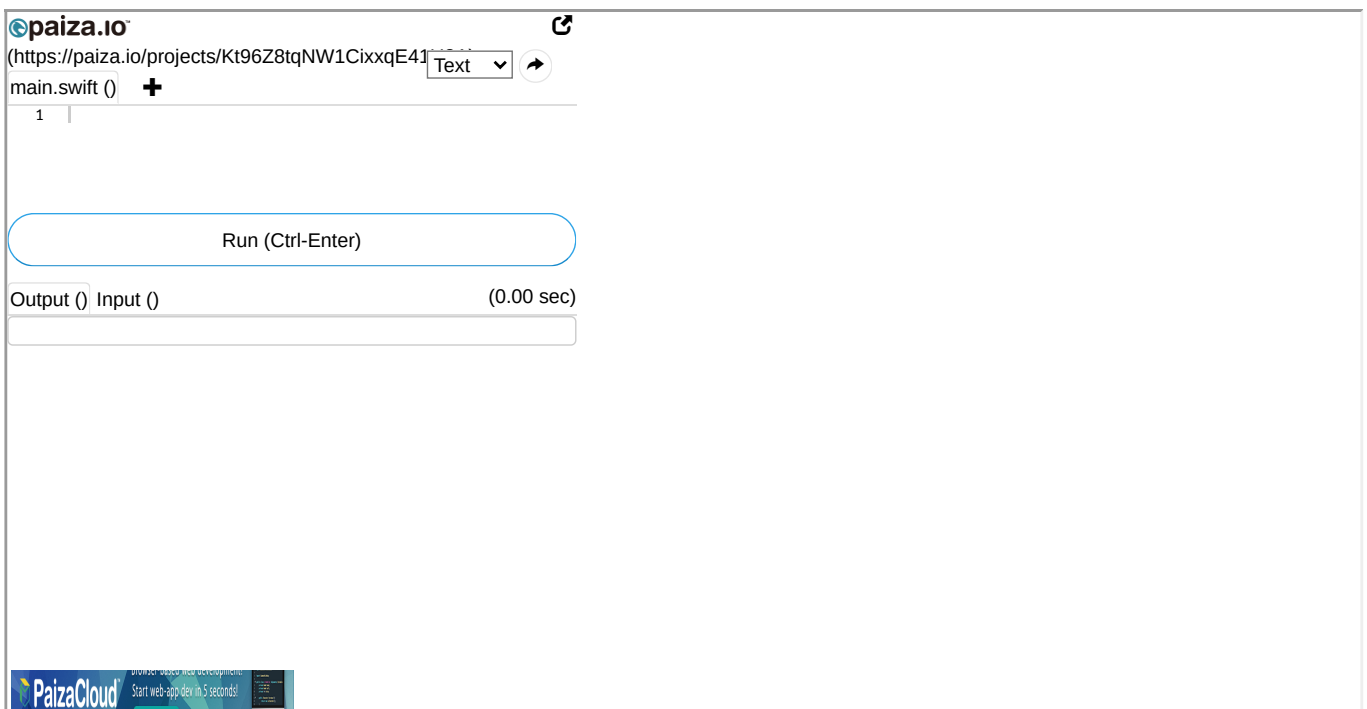
### CONCEITOS INICIAIS - MÉTODOS

Métodos/funções são os comportamentos dos objetos de uma Classe. Digamos que todo carro pode:

- Andar
- Dar ré
- Buzinar

Essas e outras ações ocorrem independentemente de qual dos objetos estivermos falando (Saveiro, Gol, etc.). Essas ações podem vir a ser os métodos/funções da classe.

Vamos adicionar à nossa classe exemplo Carro, estes métodos?



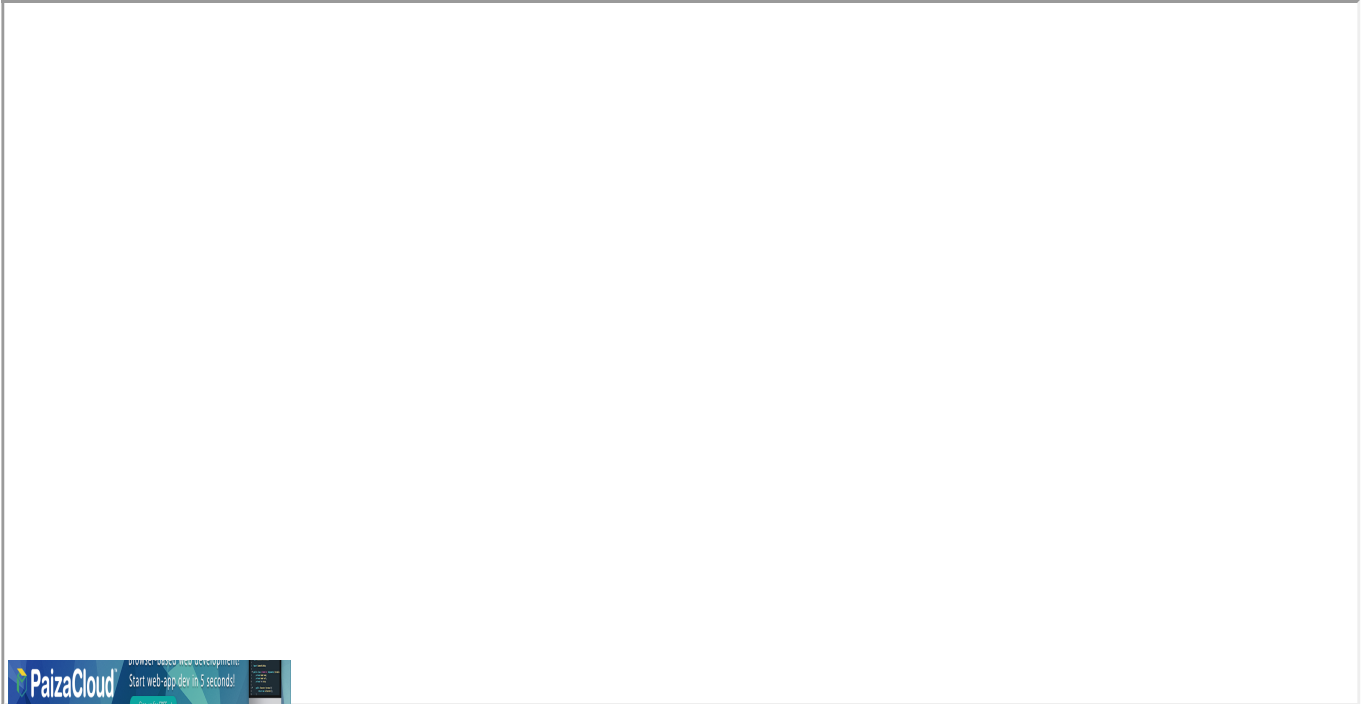
## 6. Encapsulamento



CONCEITOS INICIAIS - ENCAPSULAMENTO



Através do encapsulamento podemos definir diferentes níveis de acesso para as classes, propriedades e métodos. Utilizamos deste conceito quando queremos controlar como nossas classes, objetos e propriedades serão acessados por outras classes ou objetos dentro da aplicação. Por meio deste conceito, conseguimos ter mais segurança e controle. Existem alguns níveis que abordaremos mais à frente no capítulo específico. Um exemplo para facilitar o entendimento pode ser o saldo bancário de um cliente. Ele não pode ser de acesso público, pois isso definiria que qualquer outra parte do programa poderia alterar esse valor. Neste caso, definimos a variável como privada (palavra reservada ***private***), e assim ela será controlada apenas por métodos que irão conter mecanismos de segurança.



💡 É possível observar a mensagem de erro dizendo que não podemos alterar o valor da variável por ser inacessível e ser privada. Veremos no capítulo específico de “Encapsulamento” a forma correta de alterar, nesses casos. É válido lembrar que estamos trabalhando com um simulador. Encapsulamento, no Swift, funciona se e somente se estivermos trabalhando com a instância e a classe em arquivos separados!

## 7. Herança



A herança é um poderoso procedimento onde se herda as propriedades em comum de uma classe pai, também conhecida como superclasse, para serem utilizadas em classes filho. Por exemplo, vamos imaginar que uma fábrica de automóveis resolve fabricar novos estilos de veículos, como motocicletas e ônibus, os mesmos possuem diferentes processos para produção, mas todos possuem, motor, número de assentos, número de rodas, certo? Esse compartilhamento de propriedades pode se dar por meio de herança, onde todas as classes têm características em comum e herdam de uma superclasse. Veremos exemplos ilustrados no capítulo específico para facilitar o entendimento.

## 8. Polimorfismo



### CONCEITOS INICIAIS - POLIMORFISMO

A ordem dos nossos *insights* sobre POO (Programação Orientada a Objetos) é incremental, então, conseguimos ir construindo o conhecimento pouco a pouco. Para falar sobre polimorfismo utilizaremos dos conceitos de herança. Nos baseando no exemplo de criarmos uma montadora, conseguimos pensar e afirmar, por exemplo, que todos os veículos buzina, mas a buzina de um ônibus, de um carro e uma moto têm diferenças! Isso é polimorfismo, características em comum, mas com execuções diferentes. Aprofundaremos, no capítulo específico, com implementações e ilustrações.

Faça já os exercícios deste capítulo.

Exercícios Conceitos Iniciais ▶