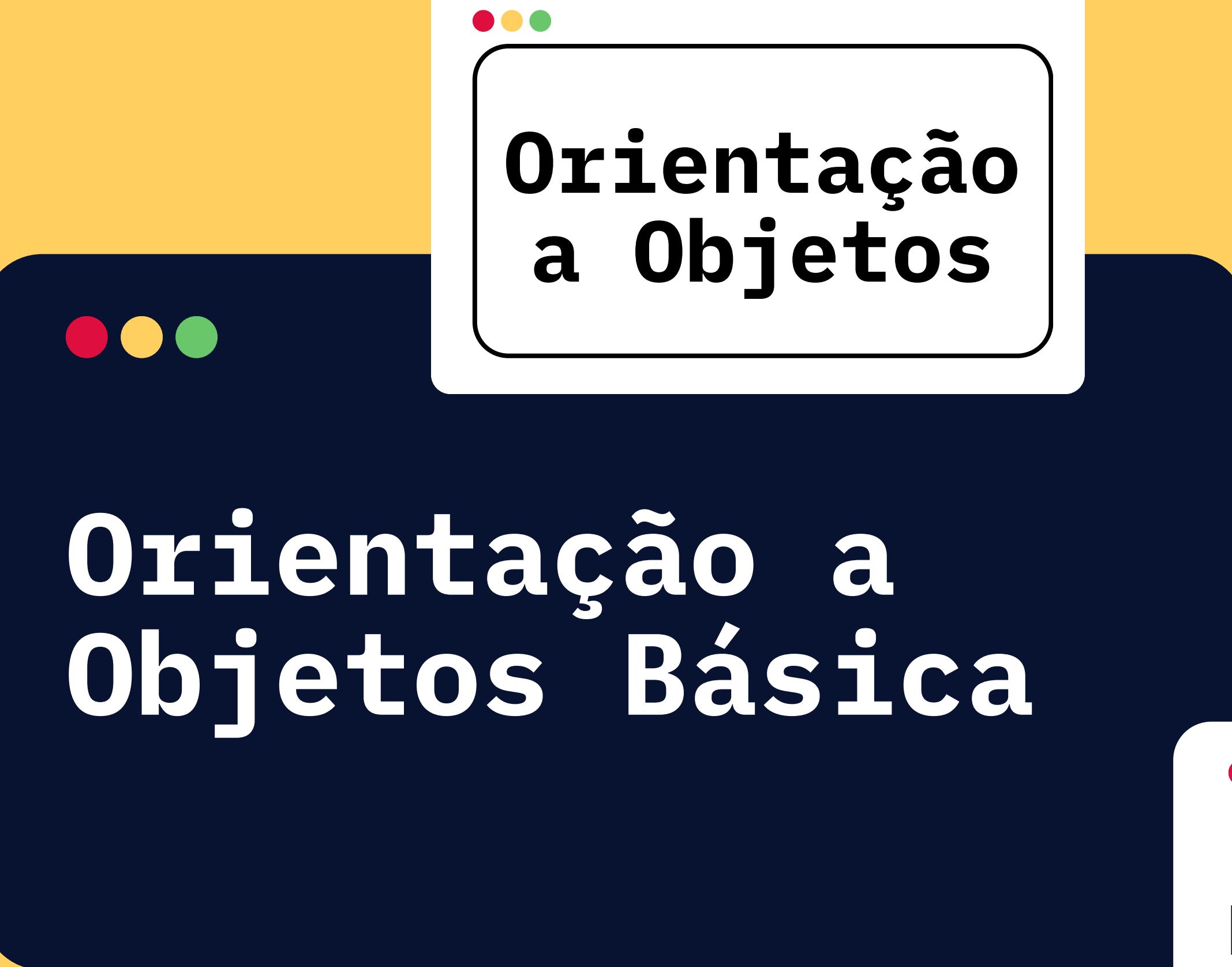
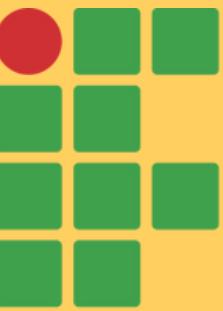


# Orientação a Objetos Básica

Orientação  
a Objetos



Professor  
**Me. Jonas Pontes**

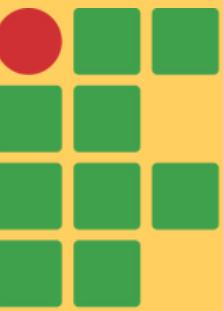


# Conceitos fundamentais



- > Classe
- > Objeto e instância
- > Atributo e estado
- > Método
- > Parâmetro
- > Tipo de dados



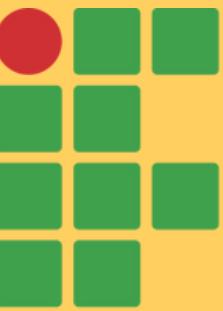


# Perguntas



- O que é programação orientada a objetos?
- O que são classes?
- O que são objetos?
- O que são atributos?
- O que são métodos?



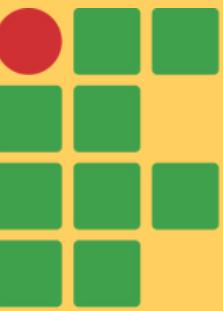


# Orientação a objetos



Paradigma de análise, modelagem e programação de sistemas de software que busca resolver problemas com base em conceitos do mundo real.

**E o que é paradigma?**

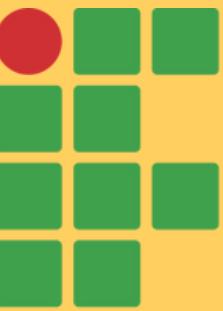


# Conceitos



- Suponha que você queira guiar um carro e fazê-lo andar mais rápido pisando no pedal acelerador.
- Antes de poder dirigir um carro, alguém tem de projetá-lo.
- Em geral, um carro inicia com os desenhos de engenharia, semelhantes às plantas utilizadas para projetar uma casa.
- Estes incluem o projeto de um pedal acelerador para aumentar a velocidade do carro.



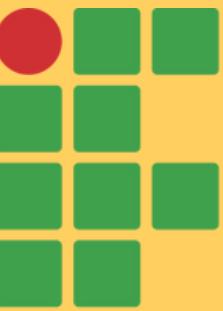


# Conceitos



- O pedal "oculta" do motorista os complexos mecanismos que realmente fazem o carro ir mais rápido;
- O pedal de freio "oculta" os mecanismos que diminuem a velocidade do carro;
- A direção "oculta" os mecanismos que mudam a direção do carro.

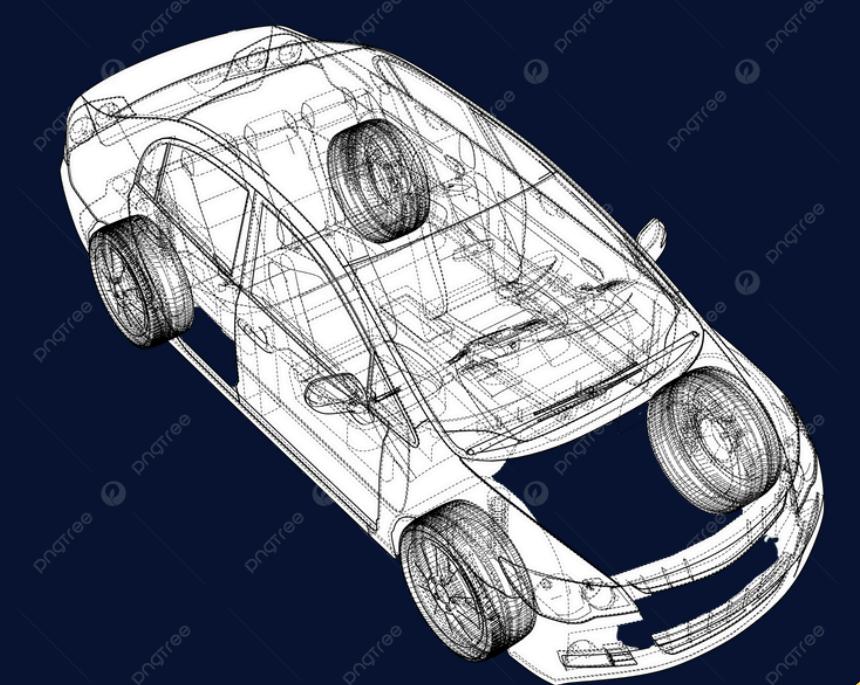
**Pessoas com pouco ou nenhum conhecimento de como os motores funcionam dirigem um carro facilmente**

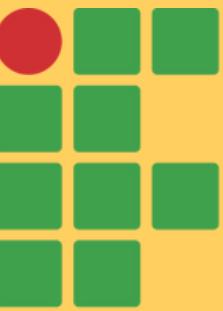


# Conceitos



- Antes de poder guiar um carro, ele deve ser construído a partir dos desenhos de engenharia que o descrevem;
- Um carro pronto tem um pedal de acelerador real para fazer o carro andar mais rápido, mas até isso não é suficiente – o carro não acelerará por conta própria, então o motorista deve pressionar o pedal do acelerador.

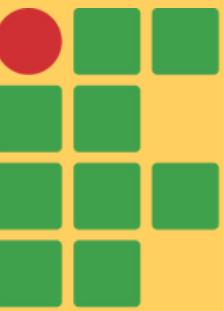




# Classe



- Abstração de entidades pertencentes a um domínio;
- Descrição de um molde que especifica as propriedades e um comportamento para um conjunto de objetos similares;
- No exemplo, o **projeto** com os desenhos de engenharia que especifica carro remete ao conceito de **classe**.

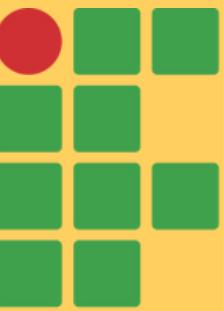


# Classe



**Outro exemplo:** Considere a abstração de uma lâmpada, independentemente de seu tipo, todas elas apresentam as propriedades de tensão, potência, cor e luminosidade. Todas podem acender, apagar, aquecer e iluminar.





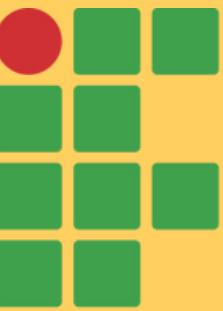
# Objetos



- Unidade geradas a partir do mesmo modelo;

A partir da classe Carro, pode-se criar infinitos carros (objetos), cada um com seu próprio conjunto de valores de propriedades, mas com o mesmo resultado de execução dos métodos.





# Objetos



A partir da classe Lampada, pode-se criar infinitas lâmpadas (objetos), cada uma com seu próprio conjunto de valores de propriedades, mas com o mesmo resultado de execução dos métodos.



“

“

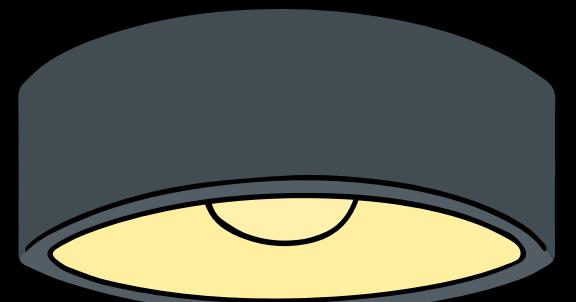
“

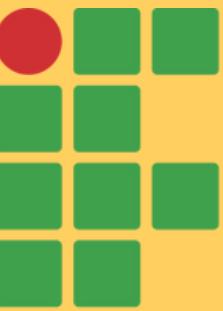
“

“

“

“



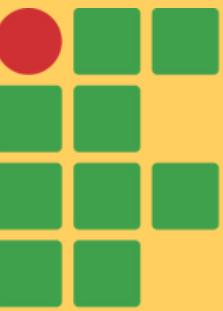


# Objetos



Tipo de lâmpada	Propriedade			
	Voltagem	Potência	Cor	Luminosidade
Incandescente	220V	60W	Laranja	864 lúmens
				
Fluorescente compacta	110V	60W	Branca	900 lúmens
				
Led	110V	18W	Amarela	932 lúmens
				

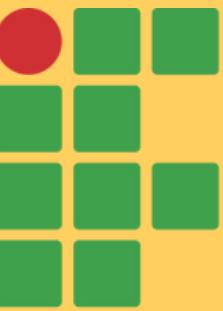




# Atributos



- Um carro tem muitos atributos
  - cor, o número de portas, a capacidade do tanque, a velocidade atual e a quilometragem.
- Atributos são representados como parte do projeto de um carro nos diagramas de engenharia.
- Cada carro mantém seus próprios atributos
  - sabe a quantidade de gasolina que há no seu tanque, mas não sabe quanto há no tanque de outros carros.



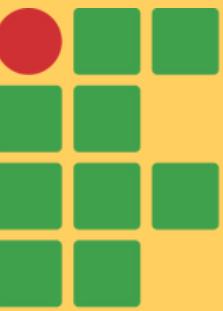
# Métodos



**Para realizar uma tarefa em um programa é necessário um método**

- O método descreve os mecanismos que realmente realizam suas tarefas.
- A função oculta de seu usuário as tarefas complexas que ele realiza, assim como o pedal acelerador de um carro oculta do motorista os complexos mecanismos que fazem o carro andar mais rápido.

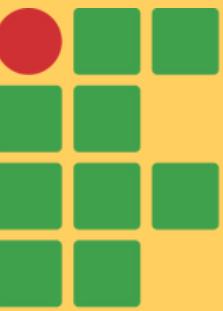
**O pedal do acelerador, o freio e a direção são interfaces simples que o motorista utiliza e que escondem mecanismos complexos**



# Métodos



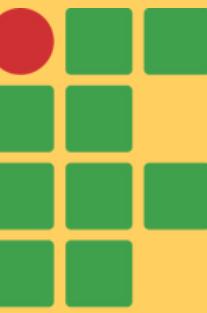
- Em POO, uma classe abriga um método, assim como os desenhos de engenharia do carro abrigam o projeto de um pedal acelerador;
- Em uma classe, você fornece um ou mais métodos que são projetados para realizar as tarefas da classe;
- Métodos podem ter parâmetros para passar informações adicionais necessárias para executar.



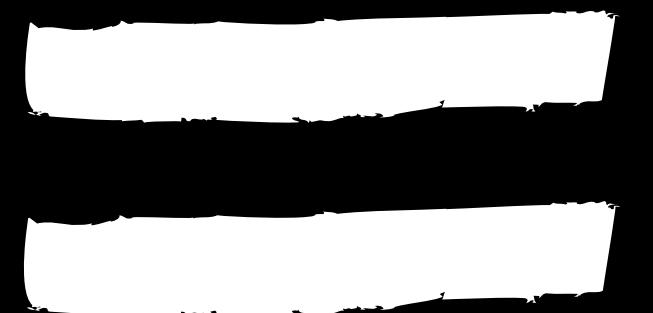
# Observações



- Muitas instâncias podem ser criadas a partir de uma única classe;
- Um objeto tem valores de atributos: valores armazenados em campos;
- A classe define quais campos um objeto tem, mas cada objeto armazena seu próprio conjunto de valores (o estado do objeto).

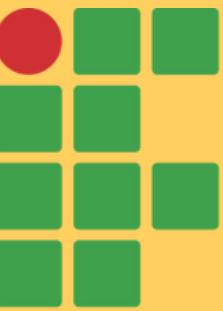


## OBJETO

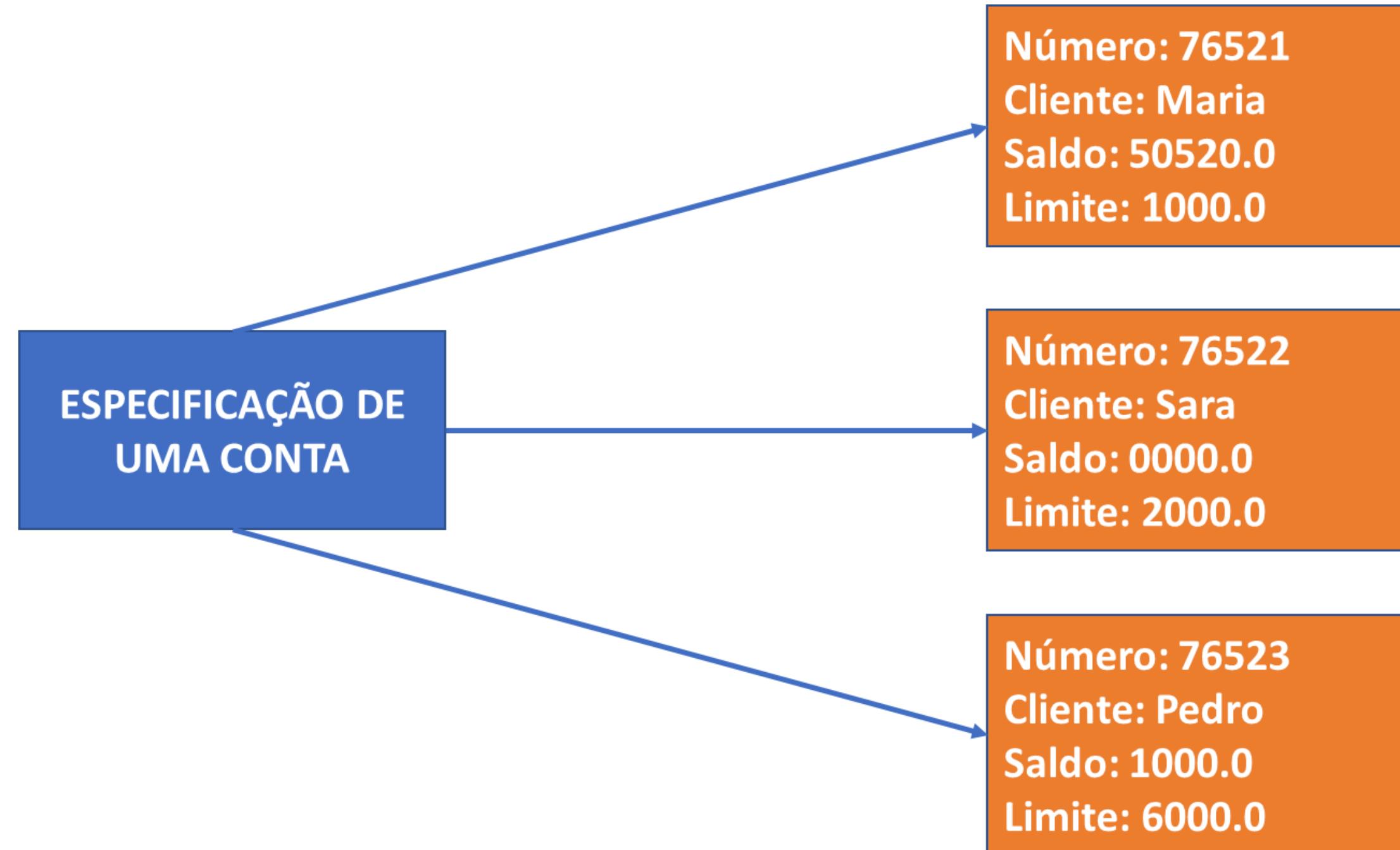


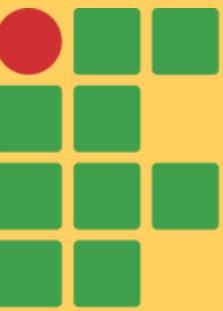
## INSTÂNCIA



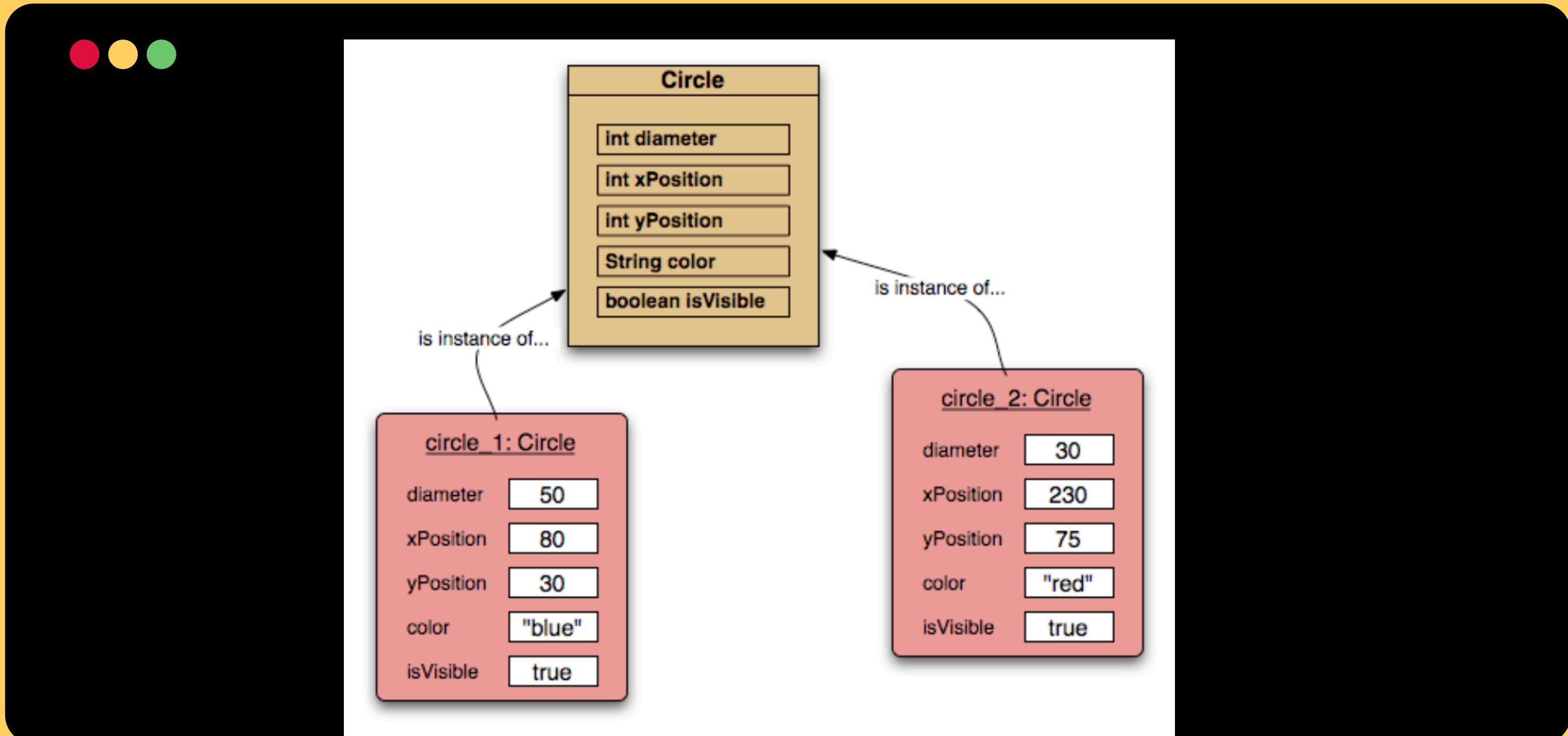


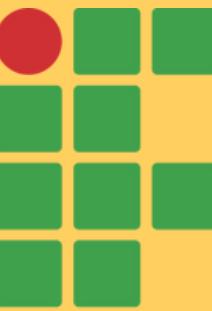
# Métodos





# Observações

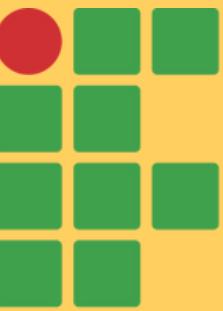




# POO



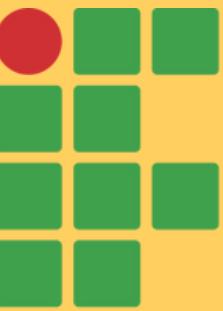
- A especificação de uma conta (papel) é uma conta?
- Deposita-se ou saca-se desse papel?
- Utiliza-se a especificação da **Conta** para poder criar instâncias que realmente são contas, nas quais pode-se realizar as operações criadas;
- Apesar de declarar-se que toda conta tem um saldo, um número e uma agência na classe, são nas instâncias desse projeto que realmente há espaço para armazenar esses valores.



# POO



- O projeto da conta recebe o nome de **CLASSE**.
- Ao que se pode construir a partir desse projeto é dado o nome de **OBJETOS**.
- Cada classe tem o código-fonte (e.g., código Java) associado a ela que define seus detalhes (campos e métodos).

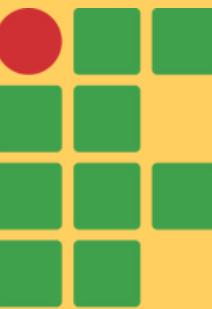


# Princípios da OO

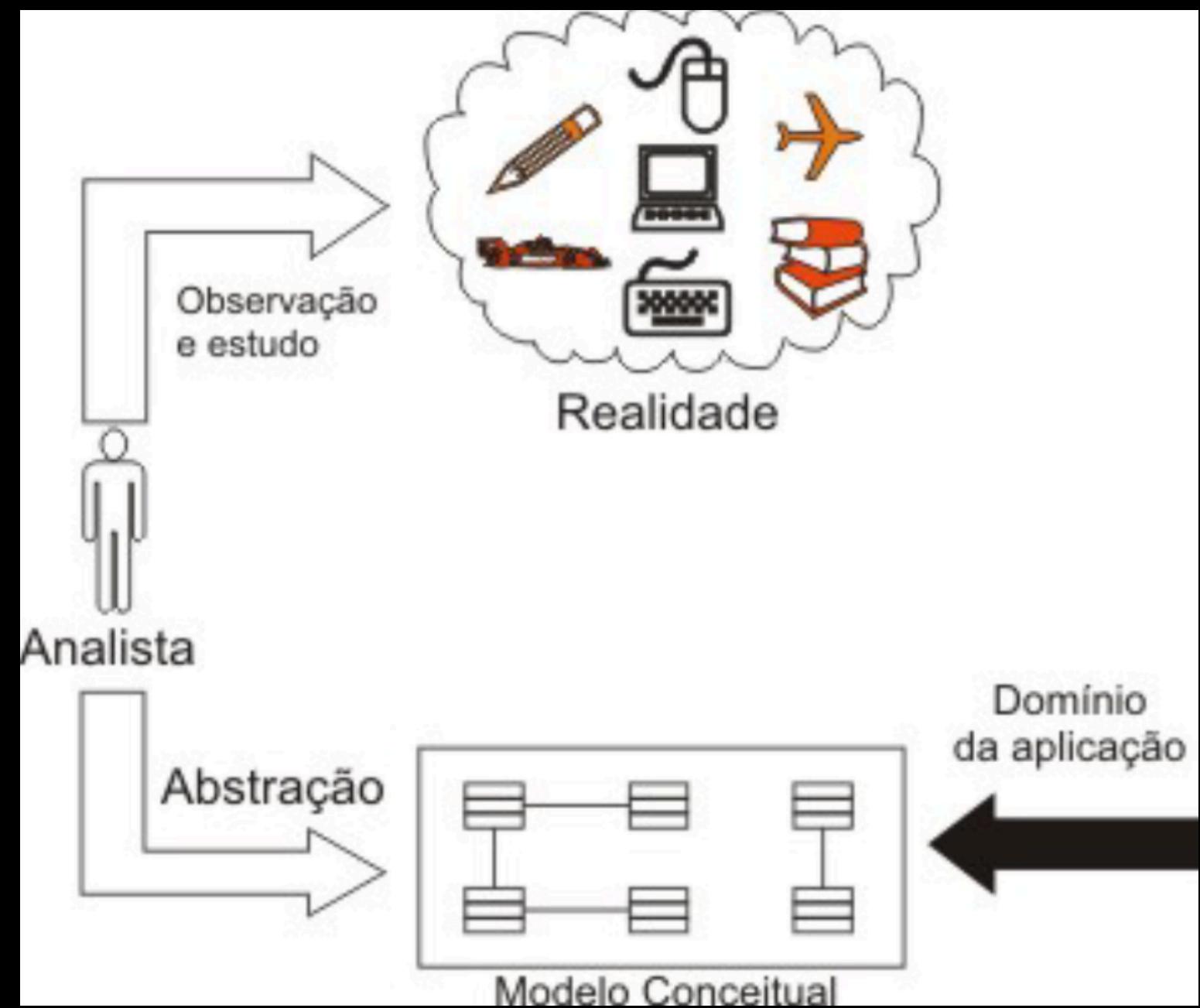


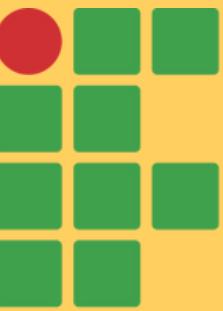
- Qualquer coisa é um objeto;
- Objetos realizam tarefas por meio da requisição de serviços a outros objetos;
- Cada objeto pertence a uma determinada classe;
- Uma classe agrupa objetos similares;
- A classe é um repositório para comportamento associado ao objeto;
- Classes são organizadas em hierarquias.

# Análise Orientada a Objetos



Consiste em definir quais objetos fazem parte de um sistema e a maneira como se comportam, isto é, modela o mundo real, estudando e criando classes a partir de suas características, como nome, cor, tamanho, etc.



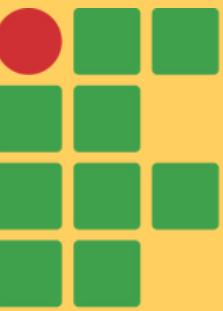


# Pilares da OO



No processo de análise, algumas características e operações não essenciais são abstraídas para a fase de programação.





# Benefícios em usar OO



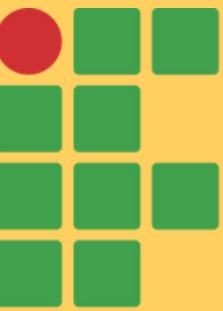
## PRODUTIVIDADE

- Reutilização de código;
- Agregação de módulos prontos;
- Servir de base à modelagem da base de dados;
- Menor custo de desenvolvimento;

## CICLO DE VIDA MAIOR

- Expansividade;
- Manutenção.

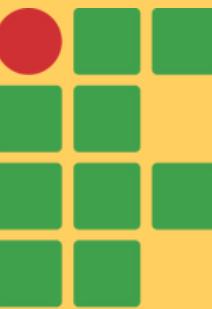




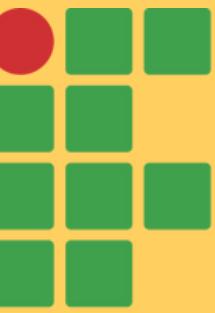
# UML

Unified Modeling Language

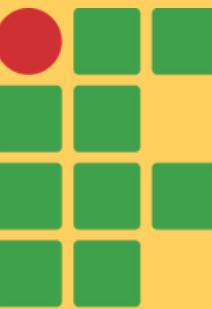




- Linguagem para modelagem de software orientado ao objetos;
- Permite criar modelos abstratos de qualquer software, permitindo flexibilidade e customização;
- É composta por diagramas, em que cada modelo representa uma visão ou perspectiva do software;
- A função da UML é representar os objetos de software e a comunicação entre eles.



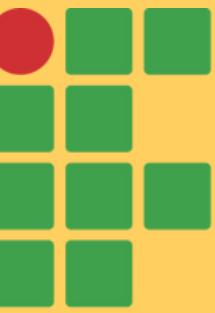
- Ao usar UML, constrói-se software sob uma perspectiva;
- Modelar software é uma tarefa indispensável no desenvolvimento de sistemas;
- O domínio da UML atenua as dificuldades quando do surgimento de novas ferramentas de desenvolvimento;
- Ferramentas de UML permitem gerar código-fonte a partir de modelos.



## Representa características relacionadas à:

- Requisitos de software
- Estado do software
- Estrutura lógica do software
- Requisitos de hardware
- Comunicação entre os objetos do software

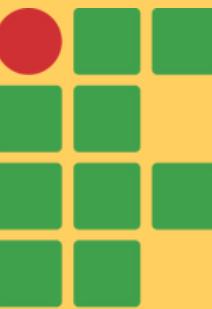
Tendo como elementos **classes, objetos e relacionamentos**



As classes utilizadas em UML representam as características de um objeto. Sua representação gráfica é dividida em três partes:

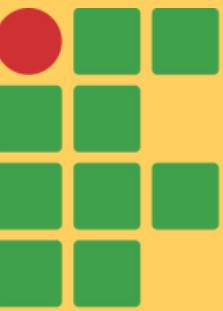
- Nome da classe
- Atributos
- Operações

Carro
+ligado: boolean = false +cor: String +modelo: String +fabricante: String +chassi: String
+ligar(): boolean +sair(): boolean +acelerar() +frenar() +desligar()

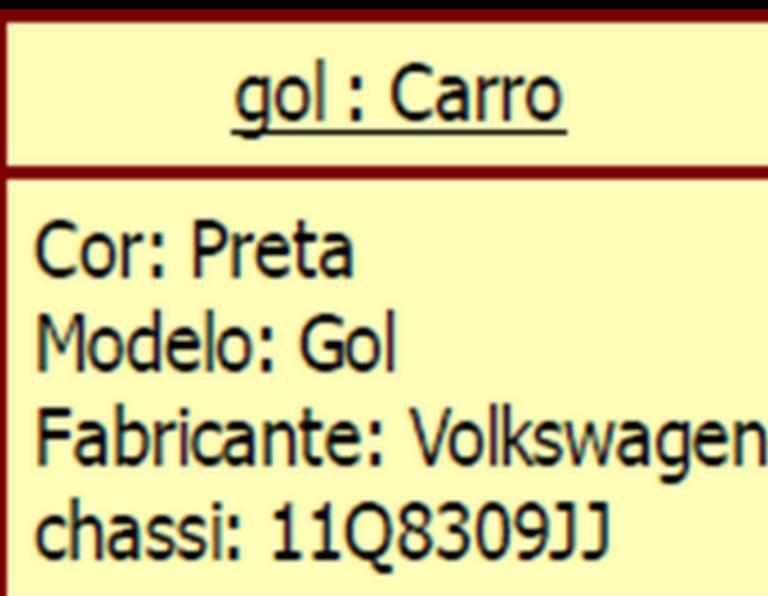
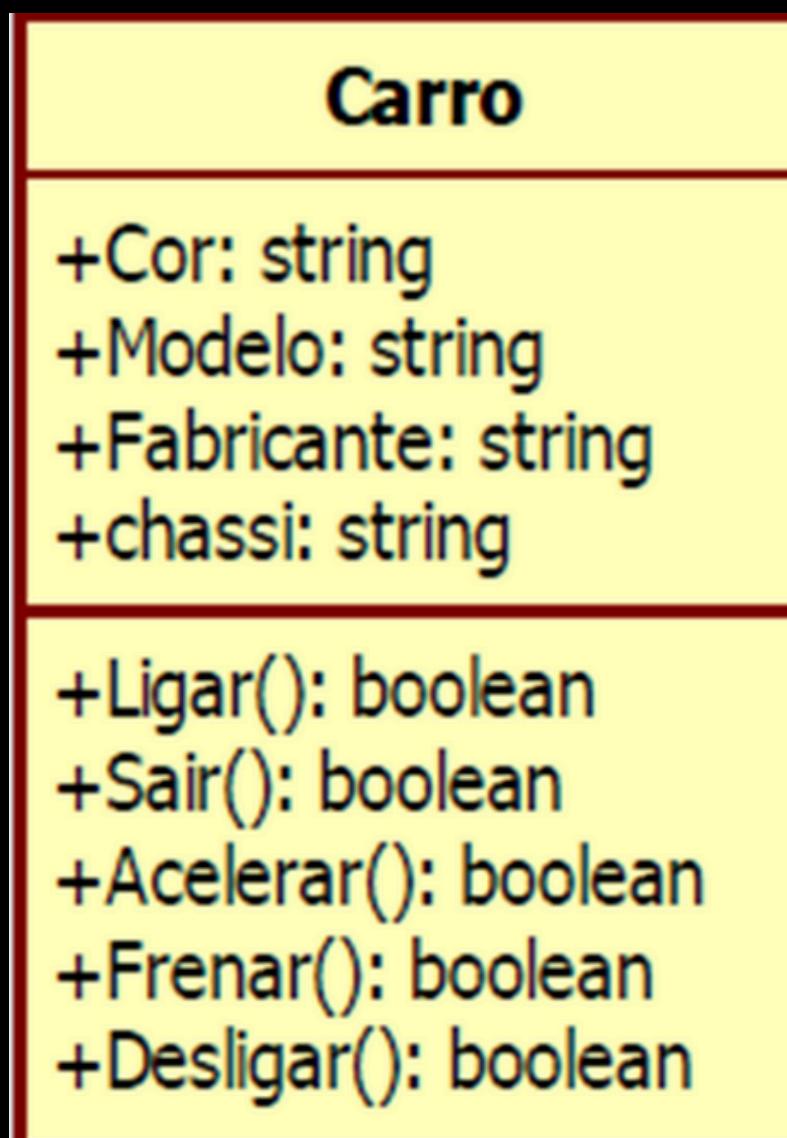


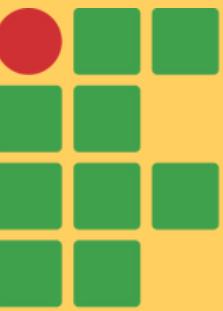
## Objetos:

- Em UML a representação de um objeto se parece com a de classes, mas apresentam valores reais para cada atributo;
  - Tem seus nomes sublinhados e possuem o nome da classe logo a seguir, separados por um sinal de dois pontos (:).



# UML





# Pratique



Crie uma porta. Para isso, responda as questões:

- Quais são os atributos necessário para a criação dessa classe?
- Quais são os métodos necessários para a criação dessa classe?
- Modele, em UML, essa classe.