



**Univ. Tecnológica Federal do Paraná
Campus Medianeira
Bacharelado em Ciência da
Computação**



CLASSES E OBJETOS

**Prof. Dr. Alan Gavioli
alan@utfpr.edu.br**

ALGUMAS LINGUAGENS E SEUS RESPECTIVOS PARADIGMAS

Orientado a Objetos

- C++
- Java
- C#
- Smalltalk

Imperativo

- C
- Pascal
- Cobol
- Basic
- Fortran

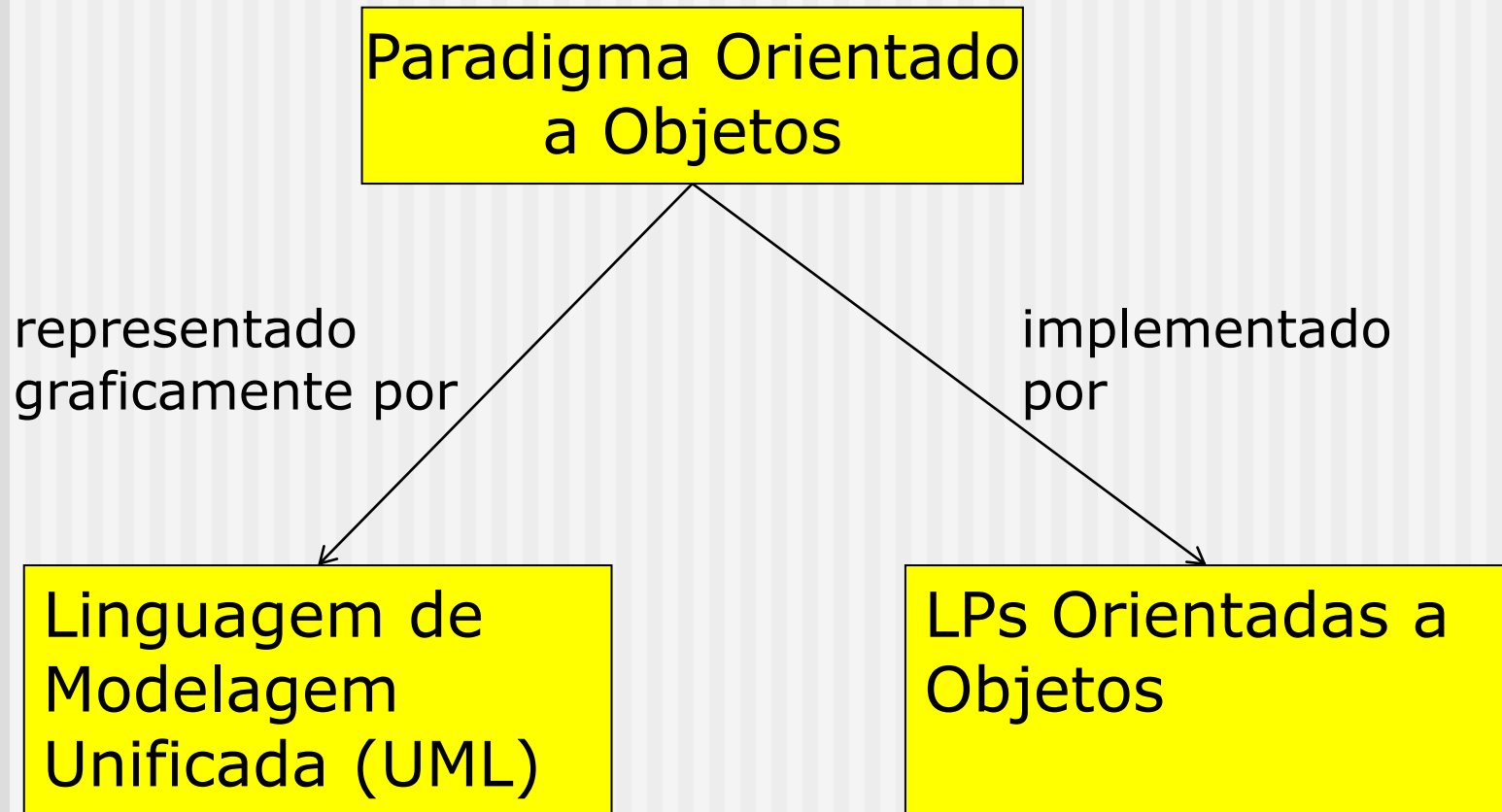
Funcional

- Lisp
- Haskell

Lógico

- Prolog

ORIENTAÇÃO A OBJETOS



PARA QUE SERVE A ORIENTAÇÃO A OBJETOS

- **Noção Intuitiva:**

- O mundo é povoado por objetos que interagem entre si.

- **Objetivos:**

- Diminuir a distância entre o problema no mundo real e sua abstração.
- Trabalhar com noções intuitivas (objetos e ações) durante todo o processo de desenvolvimento de programas.

PRINCÍPIOS GERAIS DA ORIENTAÇÃO A OBJETOS

- **Abstração**
- **Classe e Objetos**
- **Encapsulamento**
- **Herança**
- **Todo-Parte**
- **Associação**
- **Polimorfismo**

OBJETO

- É uma abstração do mundo real, ou seja, é qualquer entidade do mundo real que apresente algum significado, mesmo que não se constitua em algo concreto ou palpável.
- Possui atributos (estado) e operações (comportamento).
- Representa um elemento de um conjunto.
- Exemplos de objetos: pessoa, calculadora, automóvel, curso, disciplina, aluguel, compra, etc.

ATRIBUTOS DE UM OBJETO

- Qualidades que permitem distinguir um objeto de outro objeto, seja da mesma classe ou de classe diferente;
- O conjunto de valores dos atributos de um objeto deve torná-lo único.
- Também chamados de propriedades do objeto.

ATRIBUTOS DE UM OBJETO



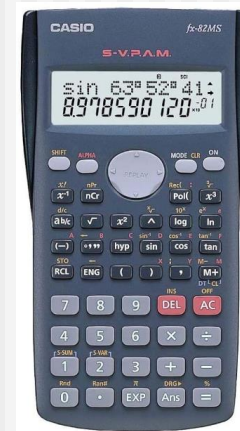
Atributos:

Nome

Cargo

Telefone

Salário



Atributos:

Tamanho

Peso

Velocidade

Número Instruções

ATRIBUTOS DE UM OBJETO



Atributos:

Modelo

Ano

Placa

Fabricante

COMPORTAMENTO DE UM OBJETO

- Ações que um objeto pode executar.
- **Operações** usadas para alterar o estado do próprio objeto ou para realizar serviços para outros objetos.
- Operações que, quando implementadas em uma LP orientada a objetos, passam a ser chamadas de **métodos**.
- Uma operação “especial” é o **construtor de classe**, responsável pela instanciação (criação) de objetos.

OPERAÇÕES



Atributos:

Nome

Cargo

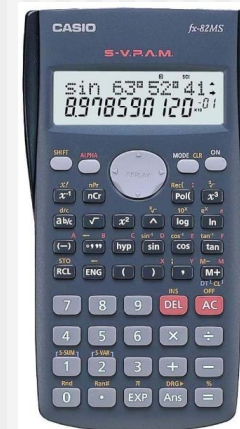
Telefone

Salário

Operações:

Promover

Pagar Comissão



Atributos:

Tamanho

Velocidade

Número Instruções

Operações:

+ * / - =

OPERAÇÕES



Atributos:

Modelo

Ano

Placa

Fabricante

Operações:

Acelerar

Acender Faróis

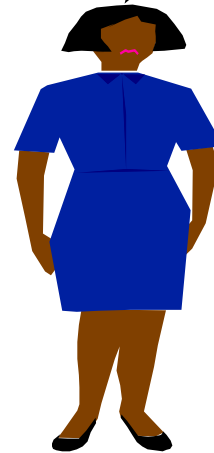
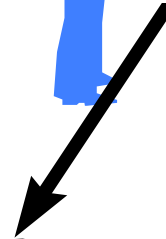
Virar à Direita

CLASSE

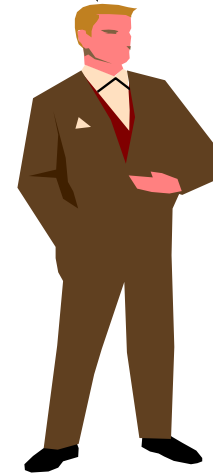
- Descreve um grupo de objetos com propriedades semelhantes (atributos), o mesmo comportamento (operações) e os mesmos relacionamentos com outros objetos.
- **Cada objeto é instância de uma classe.**

CLASSE E OBJETOS

classe Pessoa



objeto Maria



objeto Pedro

CLASSE E OBJETOS

Carro

Modelo

Ano

Placa

Fabricante

Carro()



Velar

2024

ABC 8X34

Land Rover



X6

2024

XYZ 7R56

BMW

UML - NOTAÇÕES PARA CLASSES

- Representada por um retângulo, que pode ser dividido no máximo em 3 compartimentos exibidos.
- A escolha da notação a ser exibida depende do nível de abstração desejado.

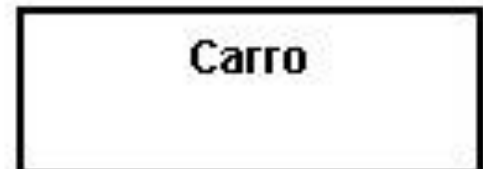
Nome da Classe

Nome da Classe
lista de atributos

Nome da Classe
lista de operações

Nome da Classe
lista de atributos
lista de operações

UML - NOTAÇÕES PARA CLASSES



ENCAPSULAMENTO

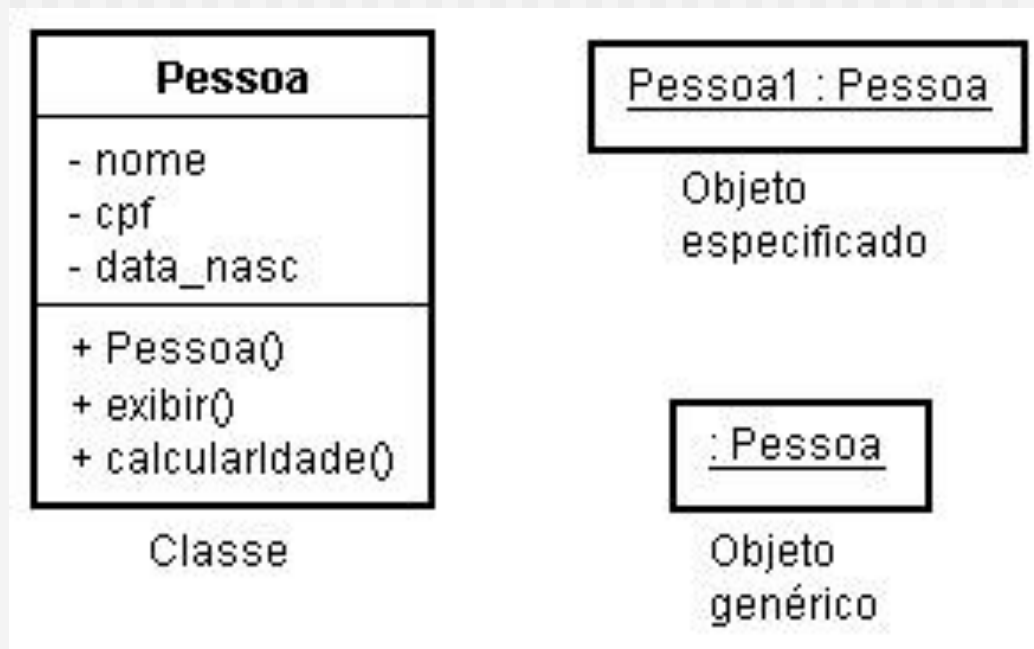
- Objetivo: ocultar a estrutura de cada objeto atrás de um conjunto de operações (implementadas por métodos) sobre ele.
- O objeto se torna acessível apenas por meio de suas operações.
- Proporciona segurança e facilita manutenções e extensões de um sistema orientado a objetos.
- Na **UML**: representado pelos caracteres **+** , **-** e **#**

VISIBILIDADE DE ATRIBUTOS E OPERAÇÕES

- **+ ou *public***: atributo ou operação será visto e usado dentro da classe e em qualquer elemento externo e classes descendentes.
- **# ou *protected***: atributo ou operação será visto e usado apenas dentro da classe e de classes descendentes.
- **- ou *private***: atributo ou operação será visto e usado apenas dentro da classe.

VISIBILIDADE DE ATRIBUTOS E OPERAÇÕES

Note o uso dos caracteres de indicação de visibilidade. Note também as formas de representar objetos na UML.



IDENTIFICAÇÃO DAS CLASSES NECESSÁRIAS

- Quando se fala na identificação das classes, o objetivo é saber quais objetos **constituirão o sistema.**
- O comportamento de um objeto é definido de forma que ele possa cumprir com suas **responsabilidades.**

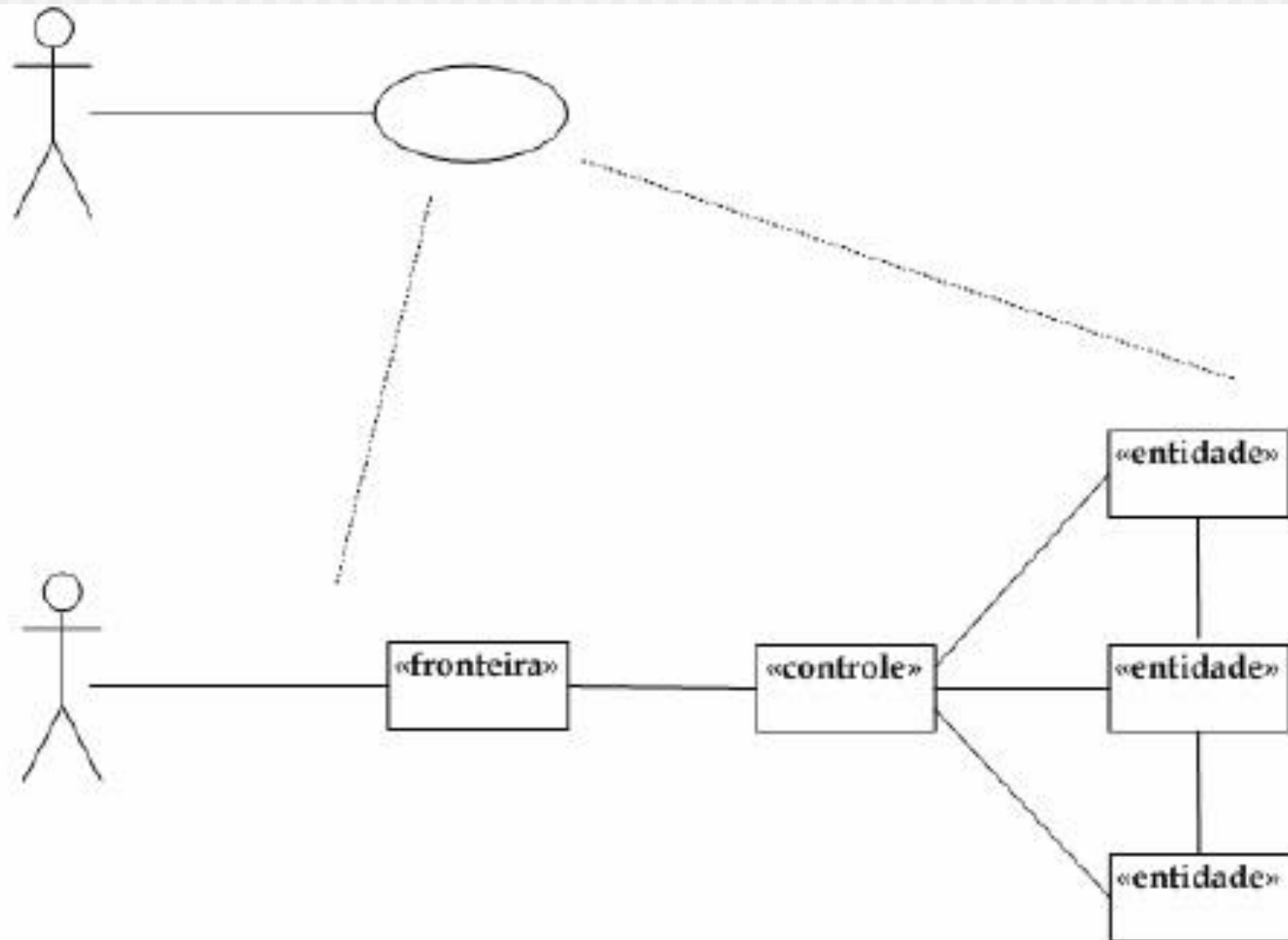
IDENTIFICAÇÃO DAS CLASSES NECESSÁRIAS

- Uma responsabilidade é uma obrigação que um objeto tem para com o sistema no qual ele está inserido. Através delas, um objeto colabora com outros para que os objetivos do sistema sejam alcançados.
- A existência de uma classe em um sistema só se justifica se ela participa de alguma forma para o comportamento externamente visível do sistema.

IDENTIFICAÇÃO DAS CLASSES NECESSÁRIAS

- **Análise dos casos de uso:**
 - Cada caso de uso é analisado para identificar classes candidatas.
- **Premissa: a partir da descrição textual dos casos de uso, podem-se derivar as classes:**
 - A existência de uma classe em um sistema só se justifica se ela participa de alguma forma para o comportamento externamente visível do sistema.
 - **Vantagem:** esta abordagem é bastante simples.
 - **Desvantagem:** depende de como os casos de uso foram escritos (as formas de expressar uma mesma ideia são bastante numerosas).

IDENTIFICAÇÃO DAS CLASSES NECESSÁRIAS



CATEGORIAS DE RESPONSABILIDADES

- **Costuma-se categorizar os objetos de um sistema de acordo com o tipo de responsabilidade atribuída:**
 - Objetos de entidade.
 - Objetos de controle.
 - Objetos de fronteira.

OBJETOS DE ENTIDADE

- Um objeto de entidade é um **repositório** para alguma **informação manipulada pelo sistema**.
- Esses objetos representam conceitos do domínio do negócio.
- Normalmente esses objetos armazenam informações **persistentes**.
- Há várias instâncias de uma mesma classe de entidade coexistindo no sistema.

OBJETOS DE FRONTEIRA

- Classes de fronteira realizam a **comunicação do sistema com atores**, sejam eles outros sistemas, equipamentos ou seres humanos.
- Há três tipos principais de classes de fronteira:
 - As que se comunicam com usuários (atores humanos);
 - As que se comunicam com outros sistemas;
 - As que se comunicam com dispositivos atrelados ao sistema.

OBJETOS DE CONTROLE

- São a “**ponte de comunicação**” entre objetos de fronteira e objetos de entidade.
- Responsáveis por controlar a **lógica de execução correspondente a um caso de uso**.
- Decidem o que o sistema deve fazer quando um evento externo relevante ocorre.
 - Realizam o controle do processamento.
 - Agem como **coordenadores/controladores** dos outros objetos para a realização de casos de uso.

DIVISÃO DE RESPONSABILIDADES

- A categorização de responsabilidades implica que cada objeto é especialista em realizar um de três tipos de tarefas:
 - Se comunicar com atores (**fronteira**).
 - Manter as informações do sistema (**entidade**).
 - Coordenar a realização de um caso de uso (**controle**).
- A importância dessa categorização está relacionada à capacidade de adaptação do sistema a eventuais mudanças.

DIVISÃO DE RESPONSABILIDADES

Tipo de Mudança	Onde Mudar
Mudanças em relação à interface gráfica, ou em relação à comunicação com outros sistemas.	Fronteira
Mudanças nas informações manipuladas pelo sistema	Entidade
Mudanças em funcionalidades complexas (lógica do negócio)	Controle

ALGUMA DÚVIDA?



MOMENTO DE PRATICAR!

BIBLIOGRAFIA UTILIZADA

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: Guia do Usuário. 2 ed. Rio de Janeiro: Campus, 2005.
- LARMAN, C. Utilizando UML e Padrões: uma introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo. 3 ed. Porto Alegre: Bookman, 2007.
- PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 8 ed. Porto Alegre: AMGH Editora Ltda, 2016.
- SOMMERVILLE, I. Engenharia de Software. 10 ed. São Paulo: Pearson, 2019.