



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



# Engenharia de Software I

Diagramas UML – Parte 2  
Atividade Prática

Professora Pâmela Carvalho

28/02/23

# **Diagramas UML**

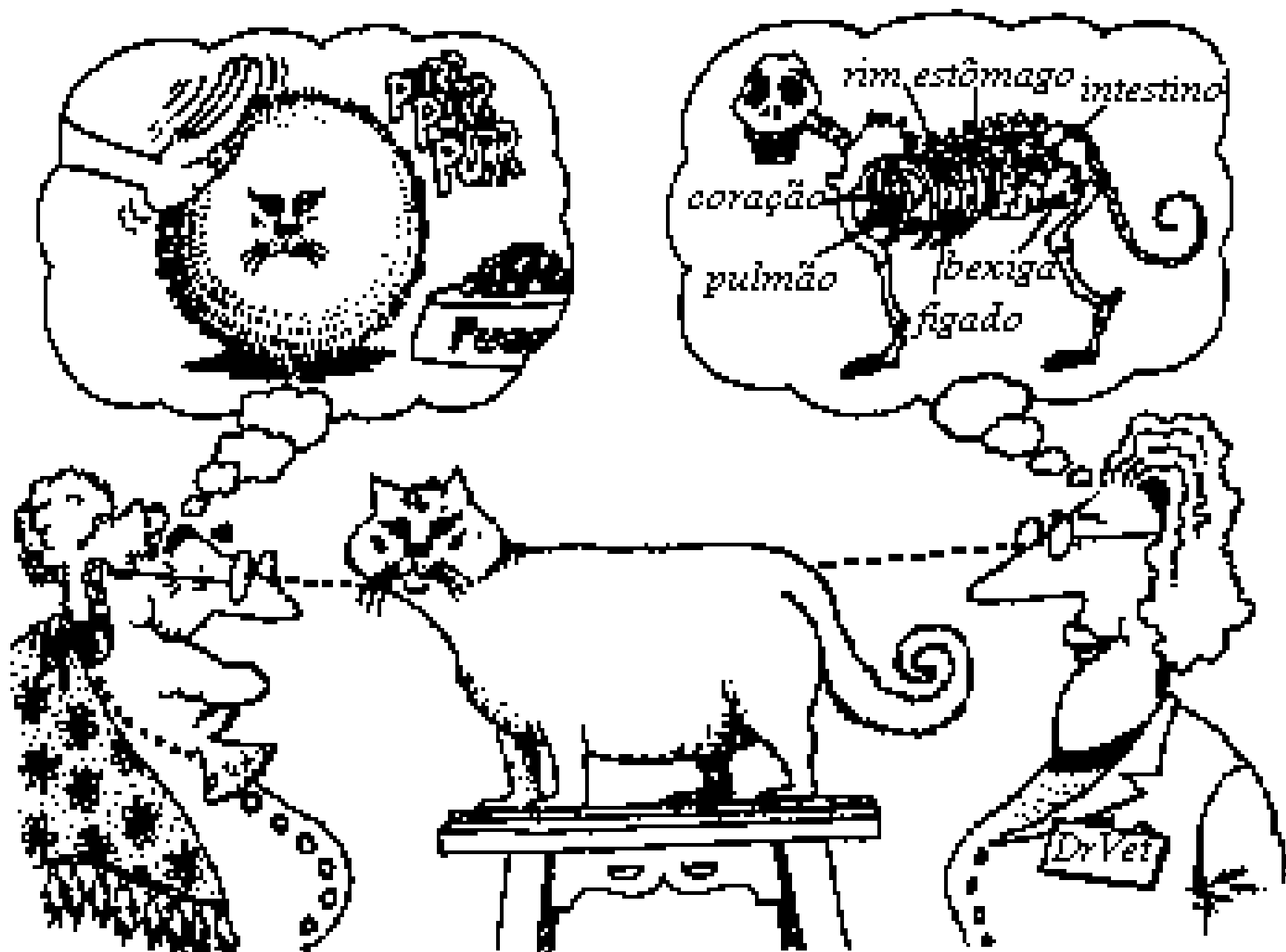
**Diagrama de Classes**

**Diagrama de Objetos**

# Diagrama de Classe

# Objeto

- É uma entidade que incorpora uma abstração relevante no contexto de uma aplicação.
- Podem ser coisas concretas ou abstratas (carro, empresa, componente, pessoa, reserva de passagem aérea, ...).
- Partes básicas:
  - **Estado**
  - **Componente**
  - **Identidade**

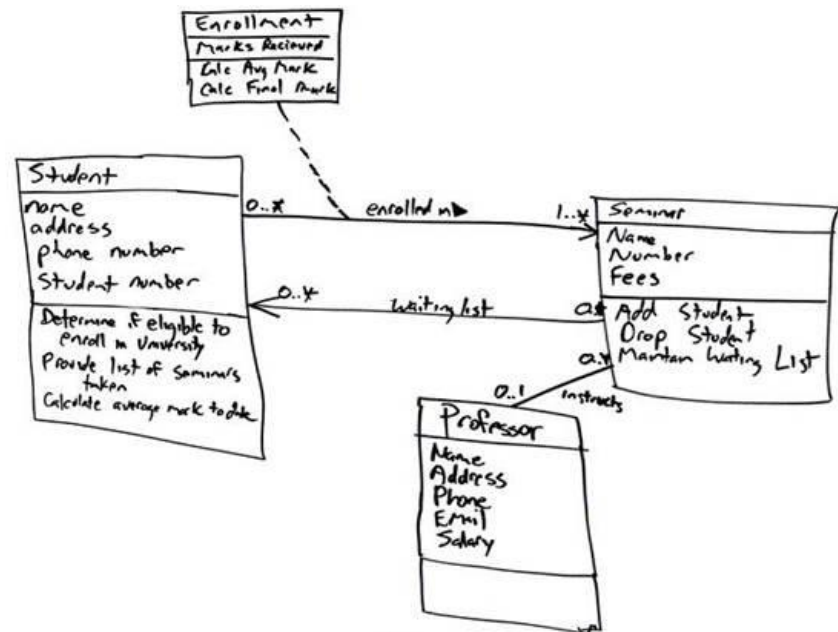


# Diagramas UML

## 2. Diagramas de Classe

A **UML** surgiu como uma proposta de ser uma linguagem para modelagem de dados que usa diversos artefatos para representar o modelo de negócio;

- um destes artefatos é o **diagrama de classes**.



# Diagramas UML

## 2. Diagramas de Classe

- Os diagramas de classe são um tipo de diagrama da **estrutura** porque descrevem o que deve estar presente no sistema a ser modelado.
- Normalmente são utilizados por engenheiros para documentar a arquitetura de software.

- Os diagramas estruturais são aspectos estáticos do sistema.



# Diagramas UML

## 2. Diagramas de Classe

- O diagrama de classe é o **principal diagrama estrutural da UML**.
- Diagramas de classe **descrevem a estrutura de um sistema**, modelando suas classes, atributos, operações e relações entre objetos.
- Os componentes de diagramação em um diagrama de classe **podem representar as classes que serão realmente programadas**, os principais objetos ou a interação entre classe e objeto.

UML	Banco de Dados	Estereótipo
Classe	Tabela	<<Table>>
Atributo	Coluna	<<Column>>, <<PK>>, <<FK>>
Tipos	Tipos de Dados	-
Associações	Relacionamentos	<<Identifying>>, <<Non-identifying>>

- Utilização de estereótipos para identificar os elementos



# Diagrama de Classe

## Resumindo...

- Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos.
- Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes.

# Diagramas UML

## 2. Diagramas de Classe

### COMPONENTES DE DIAGRAMAS DE CLASSE

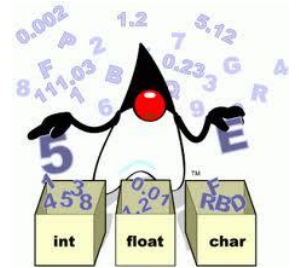
Para responder à pergunta “o que é um diagrama de classe em UML?”, você deve primeiro entender sua **composição básica**:

**Classes** - um template para a **criação de objetos e implementação de comportamento em um sistema**. Em UML, uma classe representa um objeto ou um conjunto de objetos que compartilham uma estrutura e comportamento comum.

**Sinais** - **símbolos que representam comunicação** unidirecional e assíncrona entre objetos ativos.

# Diagramas UML

## 2. Diagramas de Classe



### COMPONENTES DE DIAGRAMAS DE CLASSE

**Tipos de dados** - classificadores que definem valores de dados. Os tipos de dados podem modelar tipos primitivos e também enumerações.

**Pacotes** - esta **forma reservatória** é projetada para organizar classificadores relacionados em um diagrama.

**Interfaces** - semelhante a uma classe, exceto que uma classe pode ter uma instância do seu tipo, e uma interface **deve ter pelo menos uma classe para implementá-la**.

# Diagramas UML

Subtipo	Descrição
Empty	O Valor é zero para variáveis numéricas ou uma String de tamanho zero (" "), para variáveis de texto.
Null	A variável não contém dados válidos.
Boolean	Somente podem assumir dois valores: Verdadeiro ou Falso (True ou False).
Byte	Valor inteiro, na faixa de 0 até 255.
Integer	Valor inteiro, na faixa de -32768 até 32767.
Currency	Valores na faixa de -923.337.203.685.447,5808 até 922.337.203.685.447,5807.
Long	Valor inteiro, na faixa de -2.147.483.648 até 2.147.483.647.
Date(Time)	É um número que representa a data entre 01 de janeiro do ano 100, até 31 de dezembro de 9999 (olha o bug do ano 10000 chegando).
String	Texto de tamanho variável. Pode conter aproximadamente 2 bilhões de caracteres.
Object	Pode conter um objeto qualquer, como um Controle Activex ou um objeto COM+
Error	Pode conter um número de erro.

# Diagramas UML

## 2. Diagramas de Classe



Começando com as definições : "Um objeto é um termo que usamos para representar uma entidade do mundo real".

*(Fazemos isto através de um exercício de abstração.)*

- Vou usar como exemplo nosso primeiro cachorro **Chronos**. Posso descrever o **Chronos** em termos de seus **atributos físicos**: grande , sua cor principal é marron, olhos azuis, orelhas médias e levantadas, rabo grande e peludo, etc.
- Posso também descrever algumas **ações** que ele faz (temos aqui os **métodos**): balança o rabo quando chego em casa , foge e se deita se o mando sair debaixo da mesa, pula na cama quando estamos nela, atende e corre quando o chamo pelo seu nome. Temos então a representação do Chronos.

# Diagramas UML

## 2. Diagramas de Classe

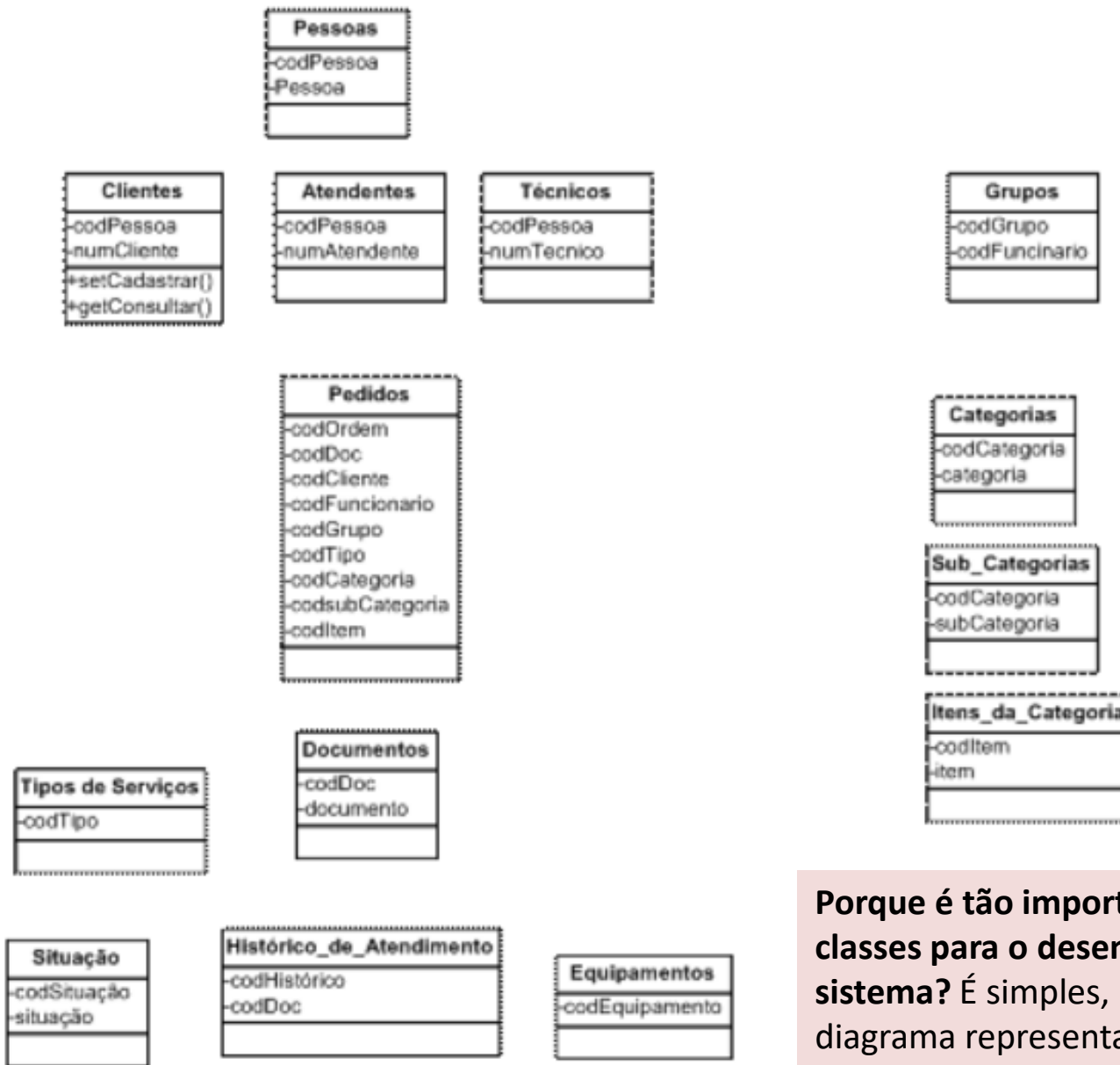
Uma classe descreve como certos tipos de objetos se parecem do ponto de vista da programação, pois quando definimos uma classe precisamos definir duas coisas:

**Propriedades** - Informações específicas relacionadas a uma classe de objeto. São as características dos objetos que as classes representam.

*Ex Cor , altura , tamanho , largura , etc...*

**Métodos:** São ações que os objetos de uma classe podem realizar.

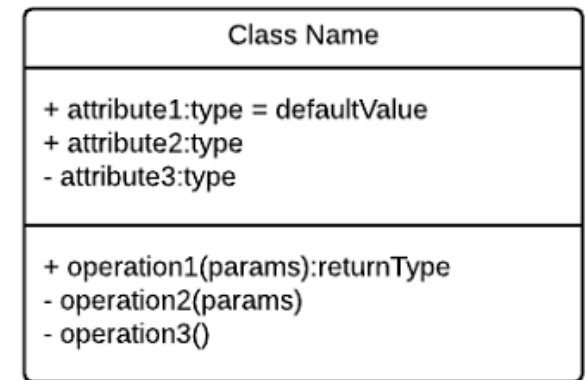
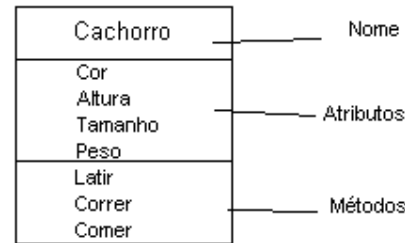
*Ex: Latir , correr , sentar , comer, etc.*



Porque é tão importante encontramos as classes para o desenvolvimento de um sistema? É simples, pois cada classe do diagrama representa uma tabela do banco de dados.

# Diagramas UML

## 2. Diagramas de Classe



O diagrama de classes é composto de **3 partes**:

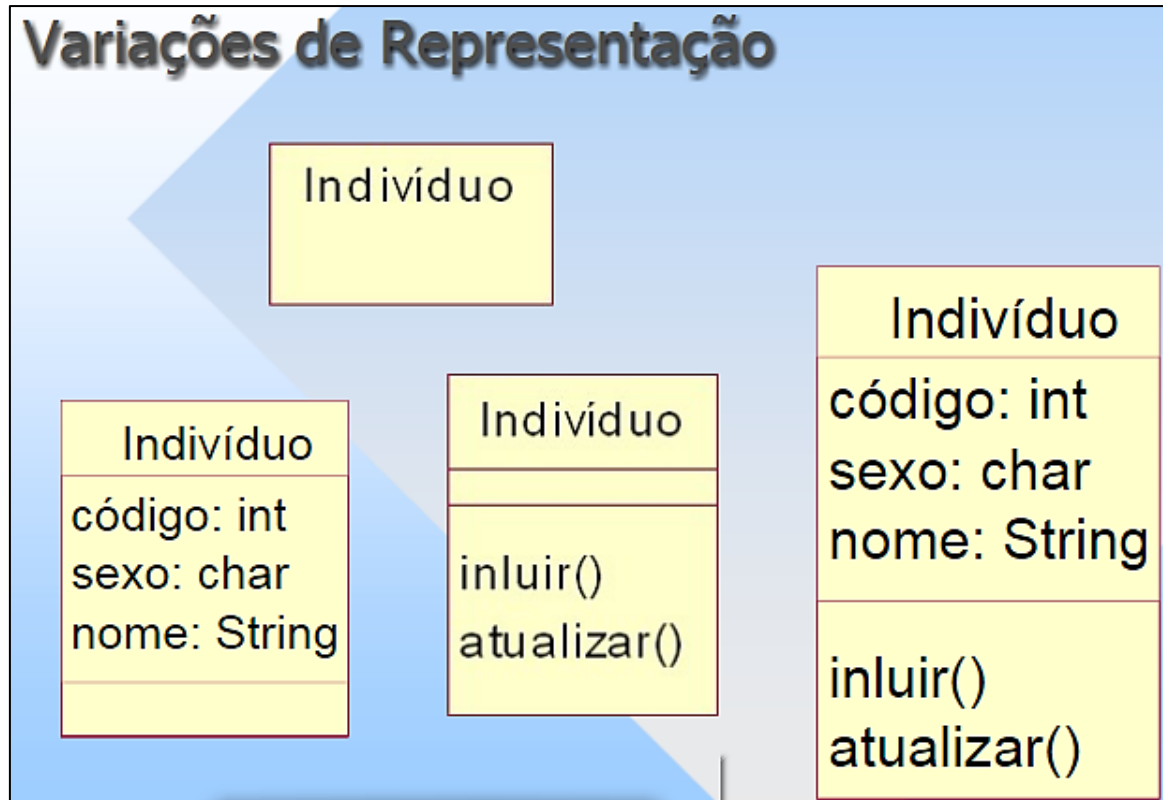
1. **Parte superior - Nome da classe** - Esta parte é sempre necessária, seja falando do classificador ou de um objeto.
2. **Parte do meio - Atributos da classe** - Descrevem as variáveis que descrevem as qualidades da classe. É necessária quando se descreve uma instância específica de uma classe.
3. **Parte inferior - Operações da classe (métodos)** - Formato de lista, cada operação ocupa sua própria linha. Operações descrevem como uma classe pode interagir com dados.



# Diagramas UML

## 2. Diagramas de Classe

O diagrama de classes é composto de 3 partes:

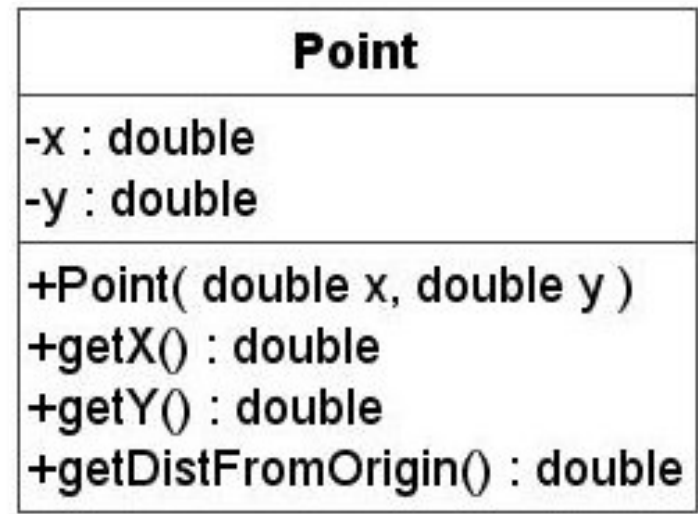


# UML

- Considere uma classe simples que representa um ponto no plano cartesiano:

```
class Point {  
    private double x, y;  
    Point(double x, double y) {  
        this.x = x; this.y = y; }  
    public double getX( ) { return  
        x; }  
    public double getY( ) { return  
        y; }  
    public double  
        getDistFromOrigin( ) { ... }  
}
```

- Seu diagrama de classe correspondente:



# Diagramas UML

## 2. Diagramas de Classe

### Modificador de acesso de membro

Todas as classes têm diferentes níveis de acesso, dependendo do **modificador de acesso (visibilidade)**.

Veja os níveis de acesso com seus símbolos correspondentes:

Público (+) - **visível em qualquer classe**

Privado (-) - **visível somente dentro da classe**

Protegido (#) - **qualquer descendente pode usar**

Pacote (~)

Derivado (/)

Estático (sublinhado)

# Diagramas UML

## 2. Diagramas de Classe

- **Relacionamento entre classes**

Os objetos tem relações entre eles: um professor ministra uma disciplina para alunos numa sala, um cliente faz uma reserva de alguns lugares para uma data, etc.

Essas relações são representadas também no diagrama de classe.

A UML reconhece três tipos importantes de relações:

**Associações : Agregação e composição**

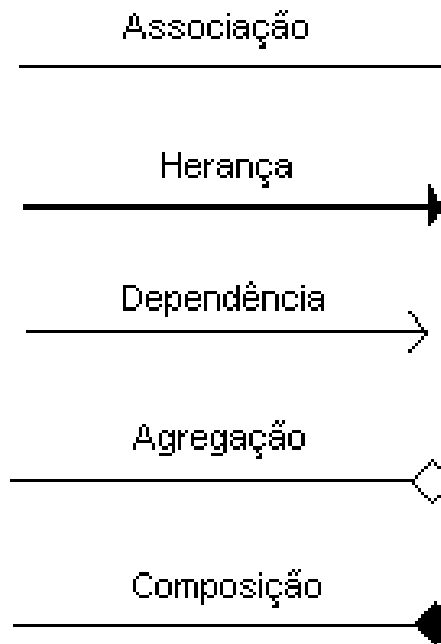
**Generalização (herança)**

**Dependências**

# Diagramas UML

## 2. Diagramas de Classe

- Relacionamento entre classes



**Associação** : São relacionamentos estruturais entre instâncias e especificam que objetos de uma classe estão ligados a objetos de outras classes.

**Agregação Regular** - tipo de associação ( é parte de , todo/parte) onde o objeto parte é um atributo do todo.

**Composição** - Relacionamento entre um elemento ( o todo) e outros elementos (as partes) onde as parte só podem pertencer ao todo e são criadas e destruídas com ele.

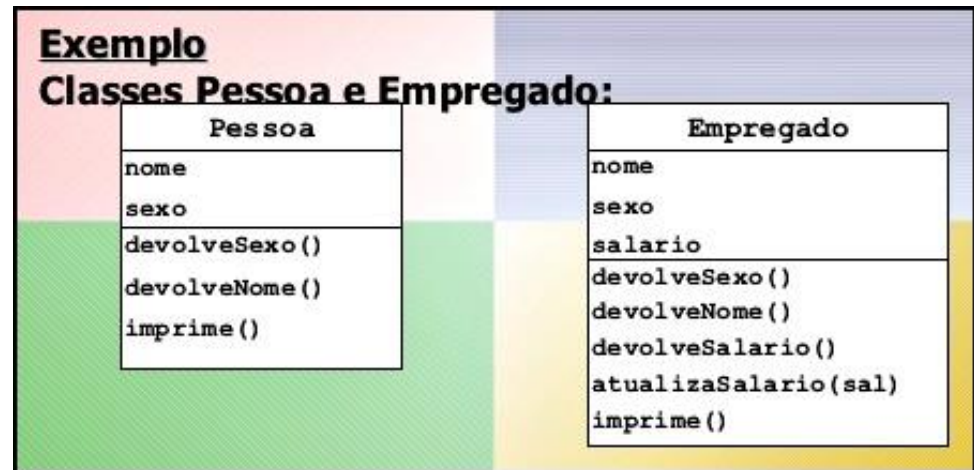
**Generalização** (herança : simples ou composta) - Relacionamento entre um elemento mais geral e um mais específico.

**Dependência** - São relacionamentos de utilização no qual uma mudança na especificação de um elemento pode alterar a especificação do elemento dependente.

# Diagramas UML

## 2. Diagramas de Classe

- HERANÇA



- A herança é quando um objeto filho assume todas as características de seu objeto pai.
  - Por exemplo, se tivéssemos o objeto veículo, uma classe carro filho herdaria todos os **atributos** (velocidade, número de passageiros, combustível) e **métodos** (andar(), parar(), mudar de direção()) da classe pai, além de atributos específicos (modelo e tipo, número de portas, fabricante do automóvel) e dos métodos de sua própria classe (rádio(), limpador de para-brisa(), ar condicionado/aquecedor()).
- A herança é exibida no diagrama de classe por meio de uma linha sólida com uma seta fechada e vazada.

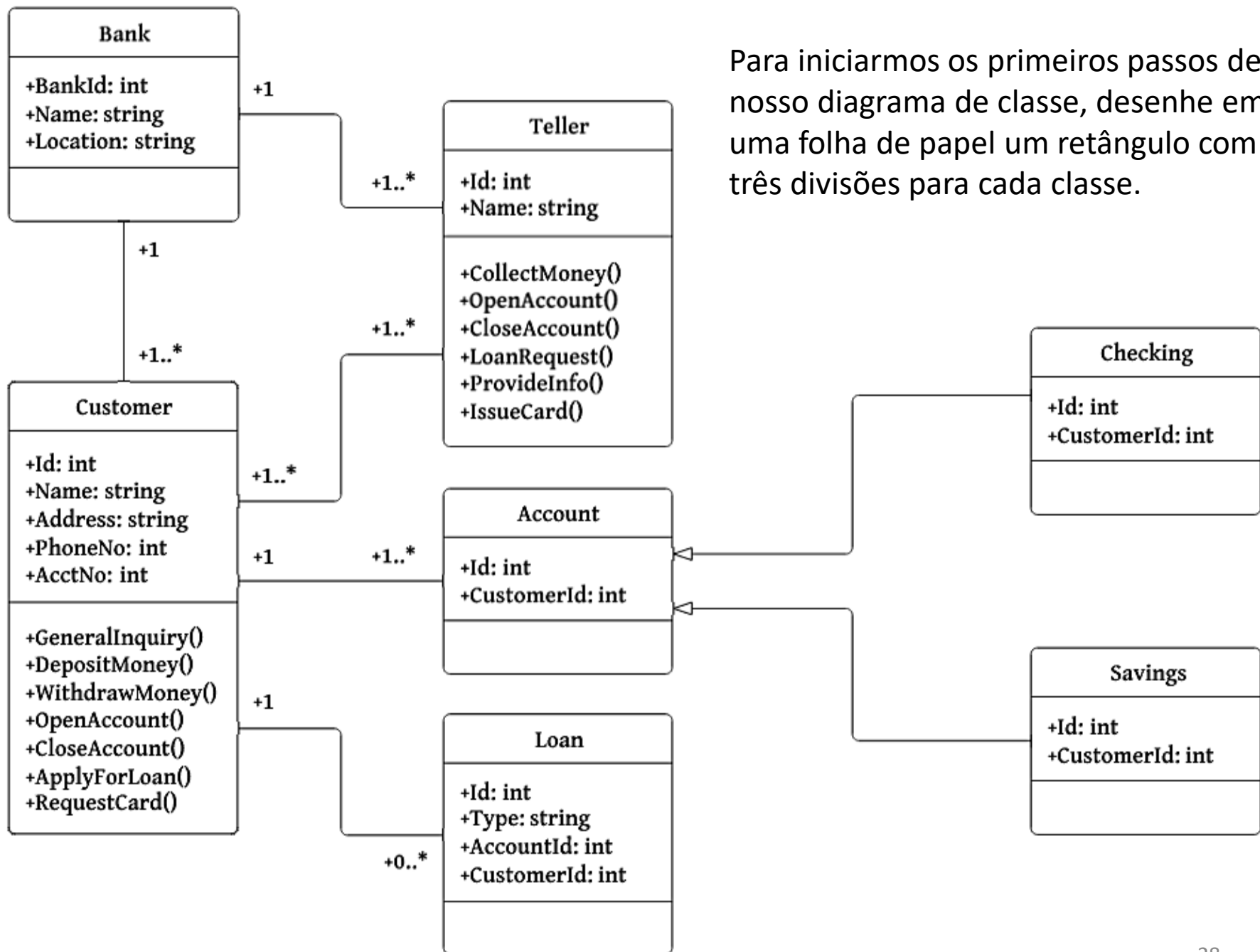
# Diagramas UML

## 2. Diagramas de Classe

### CARDINALIDADE

Uma propriedade importante dos relacionamentos é a especificação de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência de outra entidade





Para iniciarmos os primeiros passos de nosso diagrama de classe, desene em uma folha de papel um retângulo com três divisões para cada classe.



# Diagrama de Objetos

# Diagramas UML

## 3. Diagramas de Objetos

### Definição de Objetos

- Conceitual: representa uma entidade, “coisa”, processo ou conceito do mundo real e que possui:
  - Identidade – valor de uma característica que o identifica para reconhecimento
  - Atributos – qualidades, características
  - Comportamento – habilidade de processamento

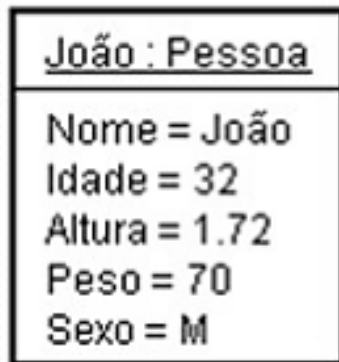
# Diagramas UML

## 3. Diagramas de Objetos

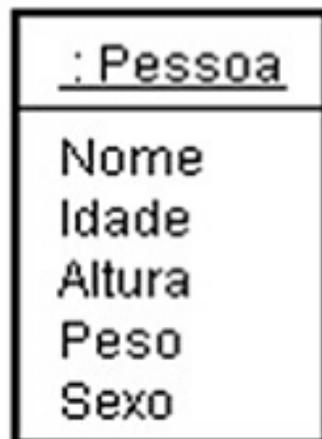
- O **diagrama de objetos** é uma variação do **diagrama de classes** e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os **objetos** que foram instanciados das **classes**.
- O diagrama de objetos é como se fosse o perfil do sistema em um certo momento de sua execução.
- Não são tão importantes como os diagramas de classes, mas eles são muito **úteis para exemplificar diagramas complexos** de classes ajudando muito em sua compreensão

# Diagramas UML

## 3. Diagramas de Objetos



Nesse primeiro caso temos um objeto específico do mundo real, representado pelo nome do objeto : nome da classe que define o objeto



Também podemos usar a forma abreviada :nome da classe sem o nome do objeto.

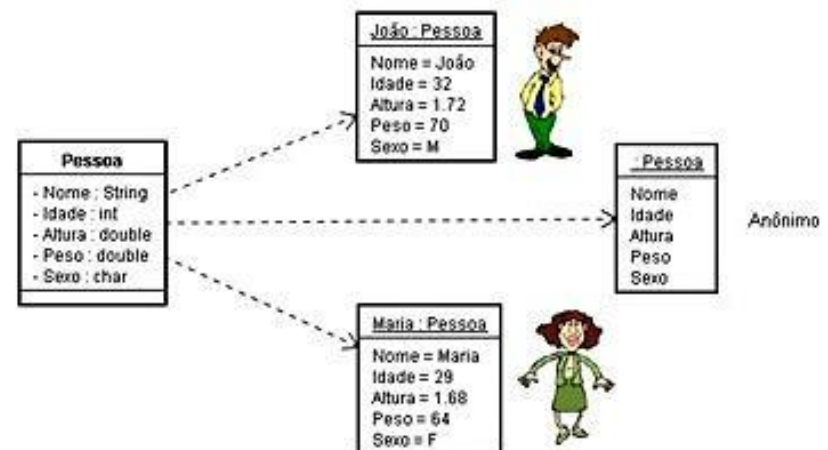
Esta forma é conhecida como objeto anônimo\*.

*\* Usamos esse tipo de objeto quando há necessidade de especificar o objeto que participa do diagrama, pois qualquer objeto dessa classe terá o mesmo comportamento.*

# Diagramas UML

## 3. Diagramas de Objetos

- Representam os objetos e as ligações entre eles;
- É uma instanciação do Diagrama de Classe;
- Utiliza a notação de objetos (nomes sublinhados) e todas as instâncias de um relacionamento;
- São também usados como parte do Diagrama de Colaboração.



# Diagramas UML

## 3. Diagramas de Objetos

**Diagramas de objetos são úteis nas seguintes situações:**

- Durante a fase de análise de um projeto, você pode criar um diagrama de classes para descrever a estrutura de um sistema e criar um conjunto de diagramas de objetos **como casos de teste para verificar a precisão e integridade do diagrama de classes.**
- Antes de criar um diagrama de classes, você pode criar um diagrama de objetos **para descobrir fatos sobre elementos de modelos específicos e seus links ou para ilustrar exemplos específicos de classificadores requeridos.**

# **Trabalho Prático**

# Trabalho Prático

Valendo 4,0 pontos para 3ª nota

- Equipe de até 3 pessoas.
- O trabalho consiste na documentação de um sistema que será escolhido pela equipe.
- O sistema escolhido deverá ser descrito para professora na reunião do dia 08/03.
- O foco da documentação será voltado para os requisitos do sistema e para elaboração de diagramas UML.
- Um modelo será disponibilizado para servir de base.
- O trabalho será enviado até o dia 14/03 e apresentado apenas para professora nos dias 14 e 15/03 com horário previamente agendado.



## Agendamento dos horários para as equipes (até 3 pessoas)

### TERÇA (14/03/23)

<b>1</b>	8	
<b>2</b>	8:30	
<b>3</b>	9	
<b>4</b>	9:30	

### QUARTA (15/03/23)

<b>5</b>	14	
<b>6</b>	14:30	
<b>7</b>	15	
<b>8</b>	15:30	

Amanhã (01/03) será nossa **2ª avaliação**

**Conteúdo:**

- Engenharia de Requisitos Parte 1 e 2;
- Introdução a UML
- Diagramas UML – Parte 1 (Diagrama de Caso de Uso).

**Até a próxima aula...**