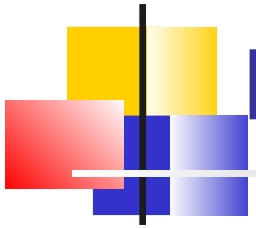
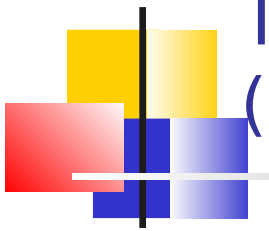


Diagrama de Classes

- Diagramas de classe são os diagramas mais comuns encontrados em modelagem de sistemas orientados a objetos.
- Um diagrama de classe mostra um conjunto de: **classes**, **interfaces** e seus **relacionamentos**.
- **Principal uso**: modelar a visão do projeto de um sistema de forma **estática**.
- São importantes para **visualização**, **especificação** e **documentação** dos modelos estruturais.



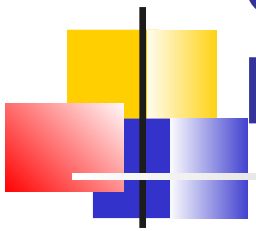
Termos e Conceitos de Diagramas de Classes (DC)

- Uma classe é uma descrição de um conjunto de objetos que compartilham: atributos, operações, relacionamentos e semântica.

- Graficamente, uma classe é desenhada como um **retângulo**.

- Uma classe em DC é composta de três partes:
  - Nome
  - Atributos
  - Métodos

O que são: Nome, Atributo e Método

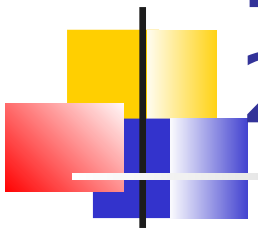


- **Nome**: Toda classe deve ter um nome que a distingue de outras.

- **Atributo**: é uma propriedade mencionada de uma classe que descreve uma variação de valores que instâncias da propriedade pode conter. A propriedade é compartilhada por todos os objetos desta classe.

- **Método**: é a implementação de um **serviço** que pode ser requerido a partir de qualquer objeto da classe para afetar seu estado.

Notações de Visibilidade em UML 2.0



- Encapsulamento:

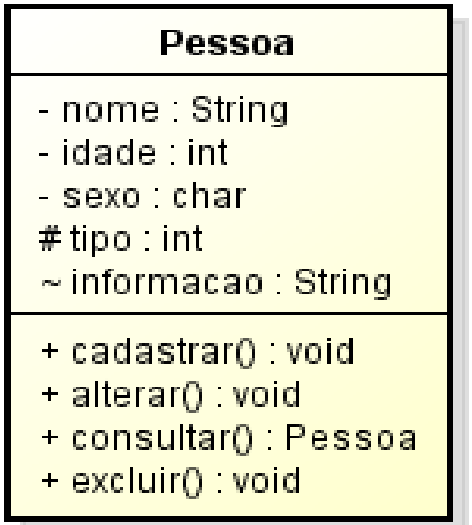
Público (+): Visível para qualquer elemento que possa ver a classe.

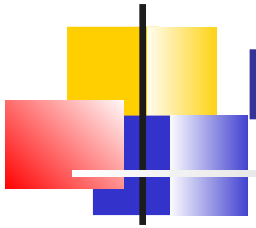
Protected (#): Visível a outros elementos dentro da classe, de subclasses e pacote.

Private (-): Visível a outros elementos dentro da classe.

Package (~): Visível aos elementos do pacote

Exemplo de Classe

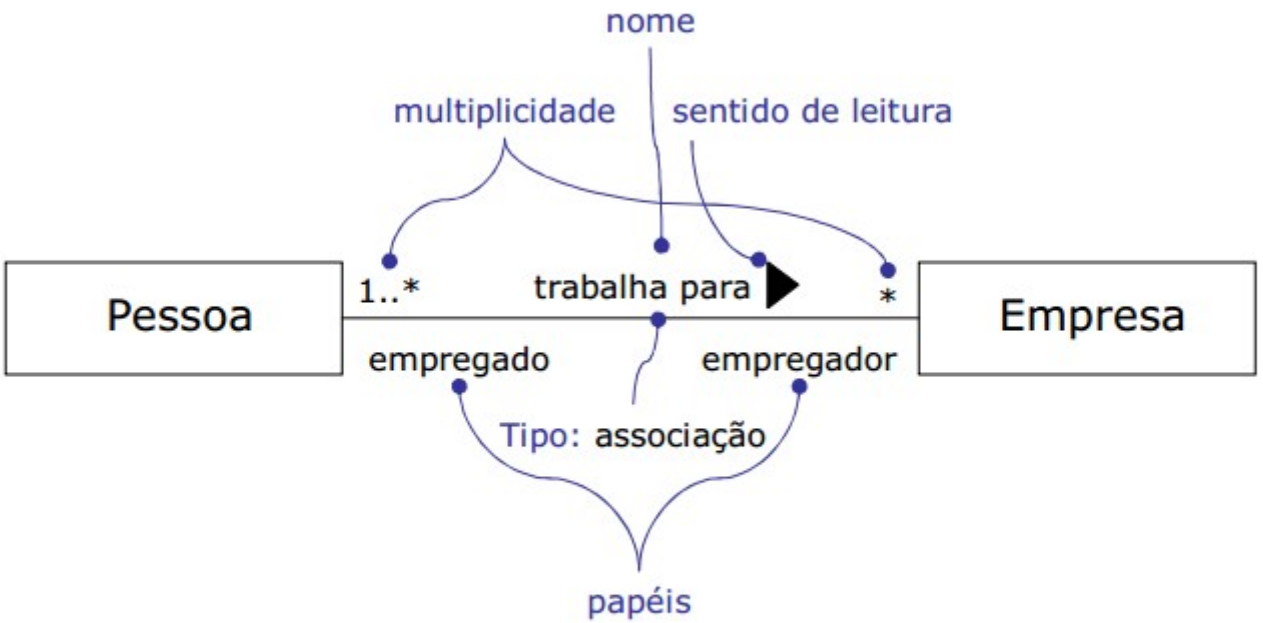




Relacionamento entre classes

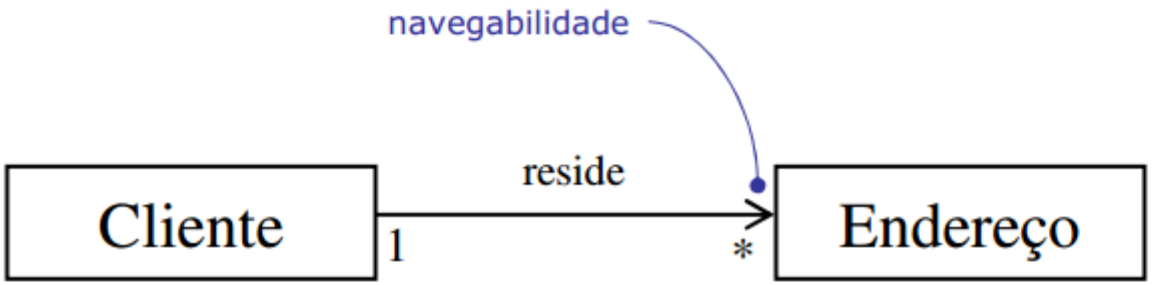
- Os relacionamentos possuem:
  - Nome: descrição dada ao relacionamento (faz, tem, possui,...)
  - Sentido de leitura
  - Navegabilidade: indicada por uma seta no fim do relacionamento
  - Multiplicidade: 0..1, 0..*, 1, 1..*, 2, 3..7
  - Tipo: associação (agregação, composição), generalização e dependência
  - Papéis: desempenhados por classes em um relacionamento

Relacionamento entre classes

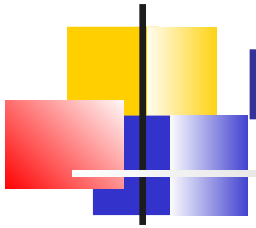


E a navegabilidade?

Relacionamento entre classes



- O cliente sabe quais são seus endereços, mas o endereço não sabe a quais clientes pertence



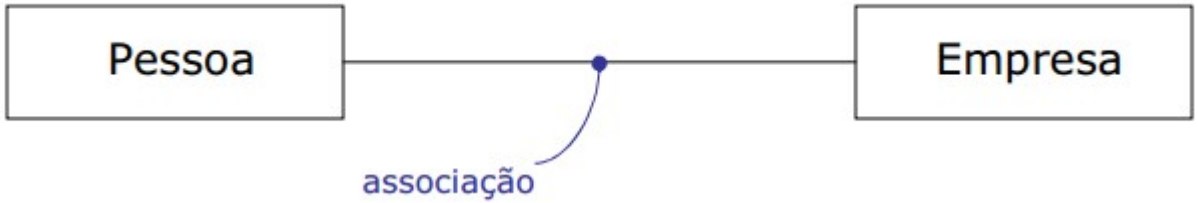
Relacionamento entre classes

- Tipos de Relacionamento:
  - Associação
    - Agregação
    - Composição
  - Dependência
  - Generalização

Relacionamento: Associação

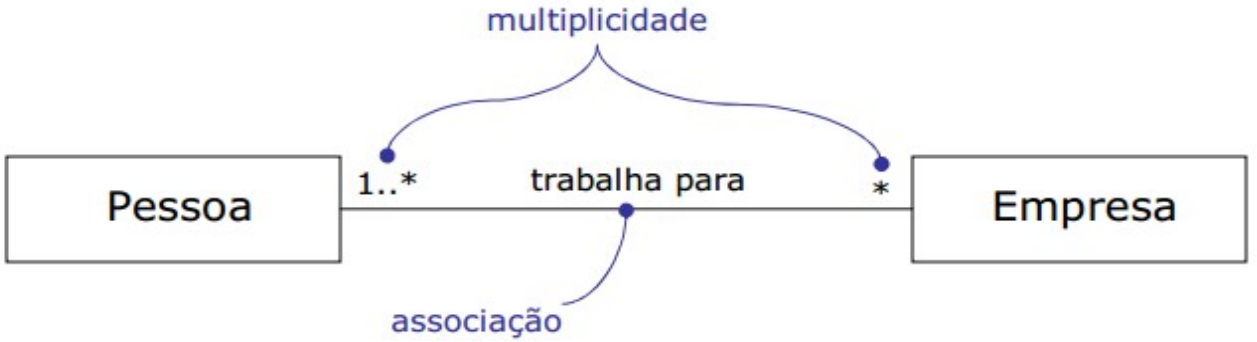
- **Associação** é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados a objetos de outra classe.

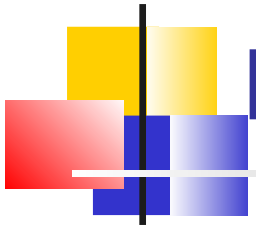
  Uma associação é representada por uma linha sólida conectando duas classes.



Relacionamento: Associação

- Indicadores de Multiplicidade
  - 1 Exatamente um
  - 1..* Um ou mais
  - 0..* Zero ou mais (muitos)
  - * Zero ou mais (muitos)
  - 0..1 Zero ou um
  - m..n Faixa de valores (por exemplo: 4..7)





Relacionamento: Associação

- Representação Java
  - Uma Pessoa trabalha em uma empresa

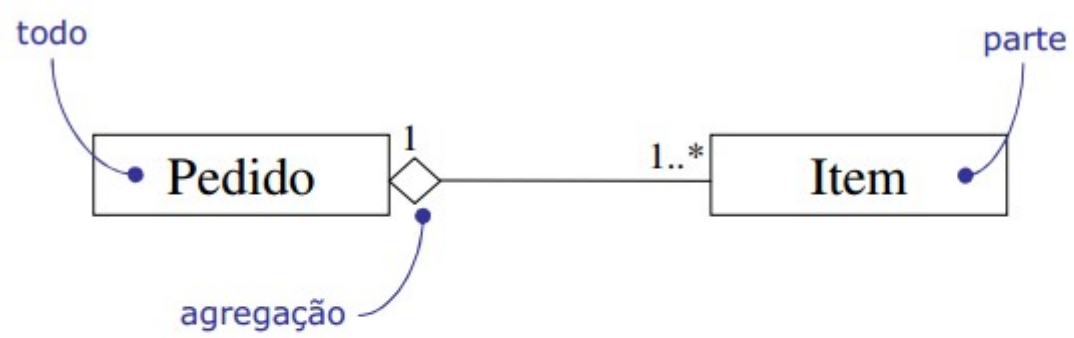
```
public class Pessoa {
    String nome;
    Empresa empresa;
}
```

  - Uma Pessoa pode trabalhar em várias empresas

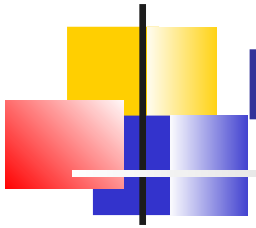
```
public class Pessoa {
    String nome;
    Empresa[] empresa;
}
```

Relacionamento: Agregação

- Tipo especial de relacionamento de associação
- Usado para representar uma relação “todo - parte”



- um objeto “parte” pode fazer parte de vários objetos “todo”



Relacionamento: Agregação

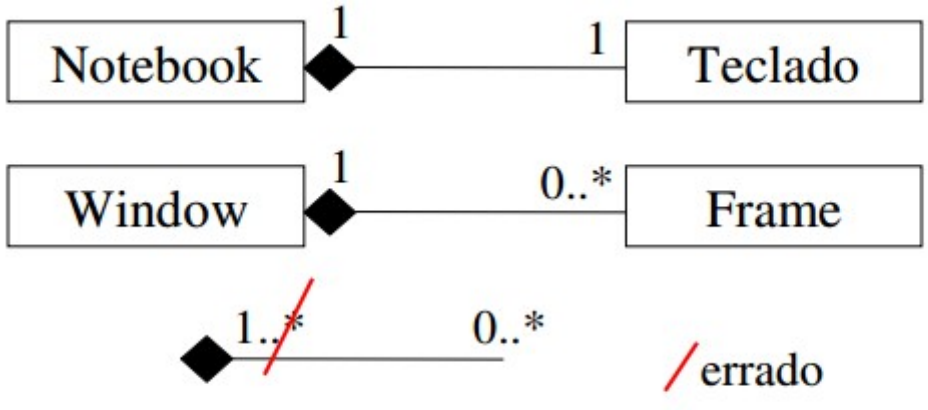
- Implementação em Java

```
public class Pedido {
   int codigo;
   ArrayList <Item> itens;

   Pedido(int codigo) {
      this.codigo = codigo;
   }
}

public class Item {
   int codigo;
   String descricao;

   Item(int codigo, String descricao) {
      this.codigo = codigo;
      this.descricao = descricao;
   }
}
```

```
public class Principal {
    public static void main(String[] args) {
       Pedido pedido = new Pedido(20);


       pedido.itens.add(new Item(1,"Pão"));
       pedido.itens.add(new Item(2,"Leite"));
       pedido.itens.add(new
Item(3,"Manteiga"));
    }
}
```

Relacionamento: Composição

- Tipo especial de relacionamento de associação
- Usado para representar uma relação “todo – parte” onde o objeto parte só pode pertencer a um objeto todo e tem o seu tempo de vida coincidente com o dele.



- Quando o “todo” *morre* todas as suas “partes” também *morrem*

Relacionamento: Agregação

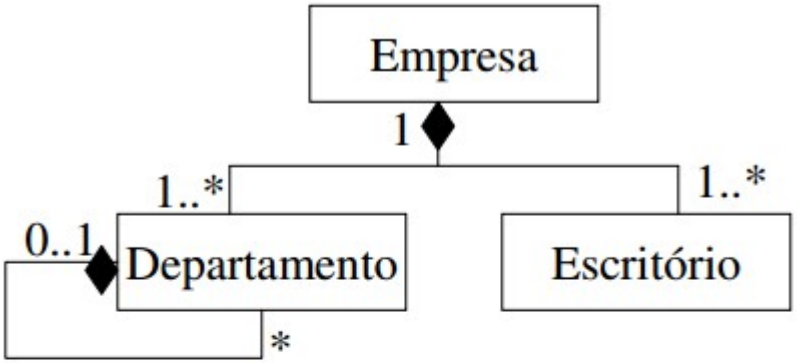
Implementação em Java

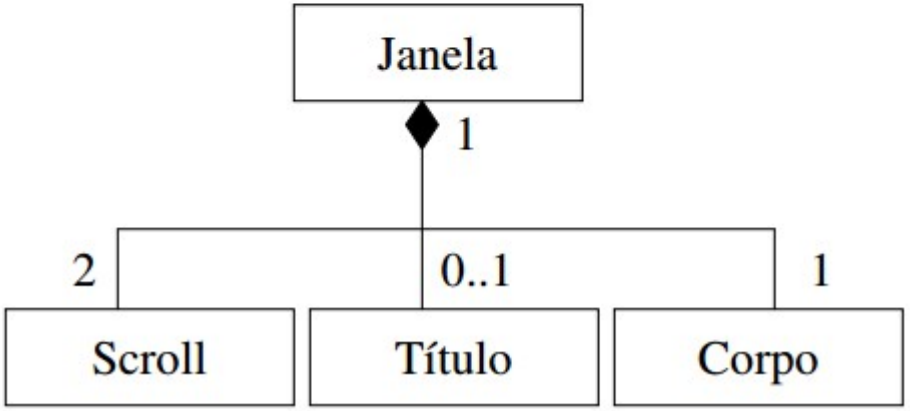
```
public class Window {
   ArrayList <Frame> frames;

   public void adicionarFrames(String descricao){
      frames.add(new Frame(descricao));
   }
}

public class Frame {
   String descricao;

   Frame(String descricao) {
      this.descricao = descricao;
   }
}
```

```
public class Principal {
    public static void main(String[] args)
{
       Window window = new Window();


       window.adicionarFrames("Titulo");
       window.adicionarFrames("Menu
Lateral”);
       window.adicionarFrames("Conteudo");
    }
}
```

Relacionamento: Composição

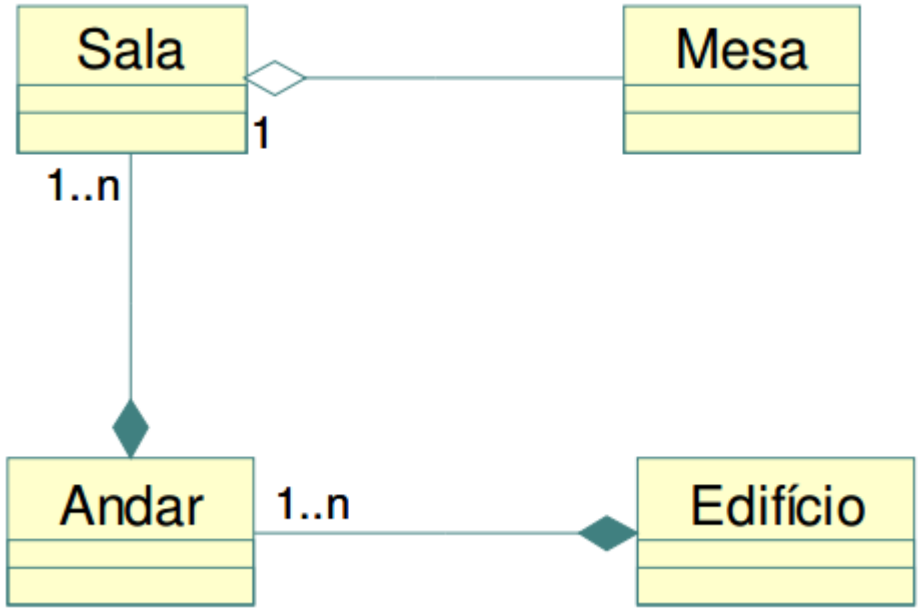
- Exemplos





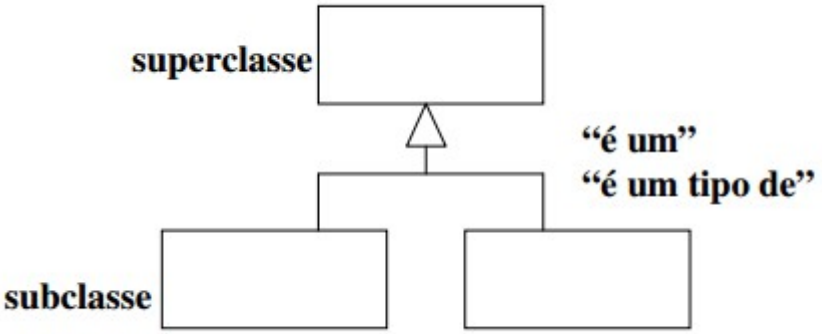
Relacionamento: Associação

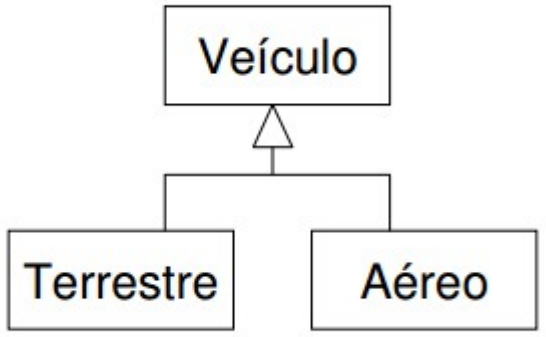
- Agregação X Composição



Relacionamento: Generalização

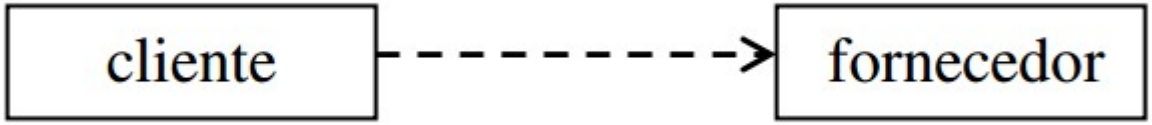
- É um relacionamento entre itens gerais (superclasses) e itens mais específicos (subclasses)





Relacionamento: Dependência

Representa que a alteração de um objeto (objeto independente) pode afetar outro objeto (objeto dependente)



Objeto dependente

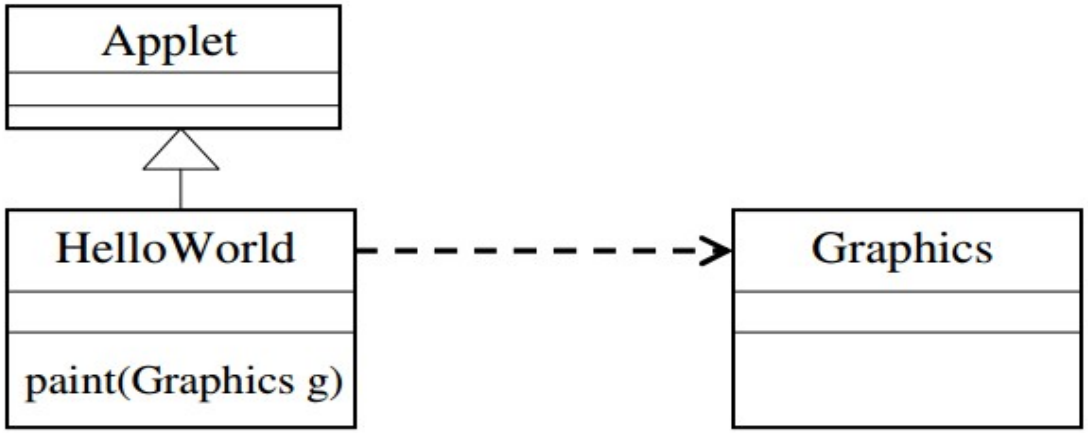
Objeto independente

- A classe cliente depende de algum serviço da classe fornecedor
- A mudança de estado do fornecedor afeta o objeto cliente
- A classe cliente não declara nos seus atributos um objeto do tipo fornecedor
- Fornecedor é recebido por parâmetro de método

Relacionamento: Dependência

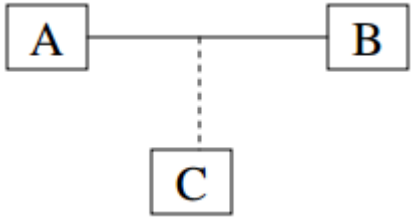
- Exemplo de Implementação

```
Import java.awt.Graphics;
class HelloWorld extends java.applet.Applet
{
    public void paint (Graphics g)
        g.drawString("Hello, world!", 10, 10);
}
```



Relacionamento: Associação

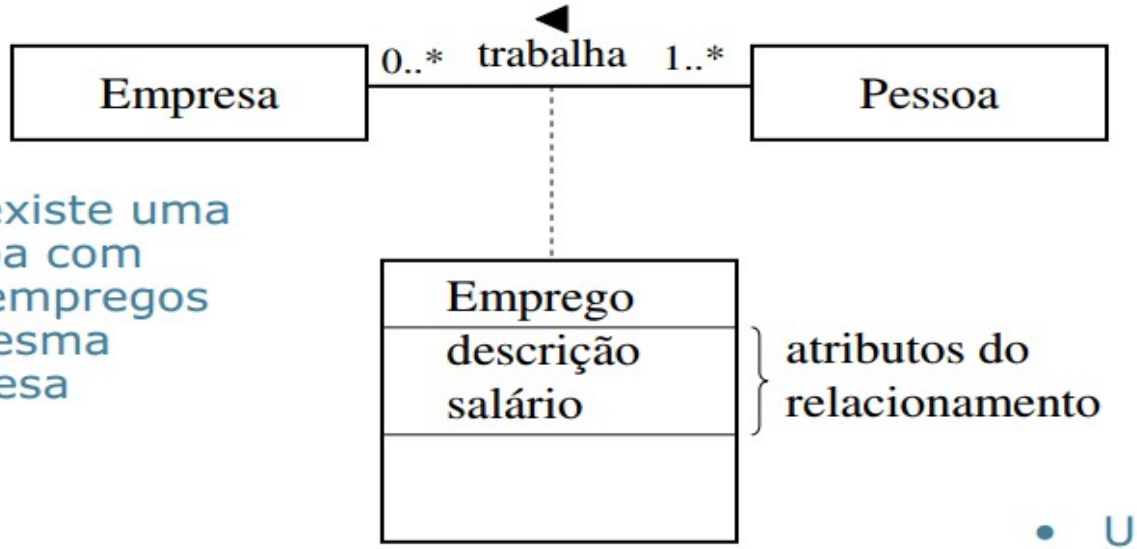
- Classe de Associação
  - Usada quando uma associação entre duas classes tiver atributos da associação
  - C existe para todo relacionamento de A com B



  - C é único para o relacionamento de A e B

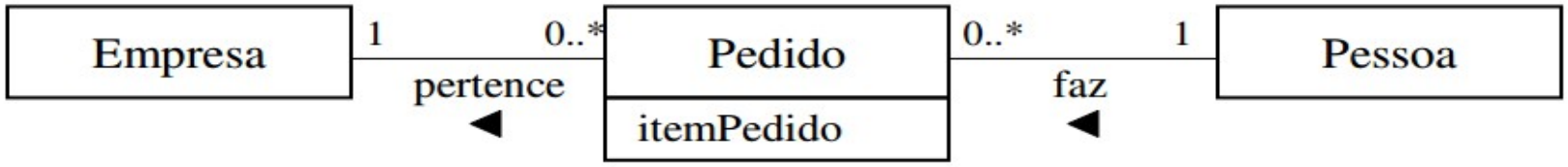
Relacionamento: Associação

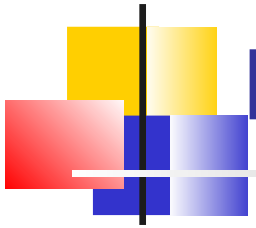
- Classe de Associação



- Não existe uma pessoa com dois empregos na mesma empresa

- Uma pessoa pode fazer mais de um pedido na mesma empresa



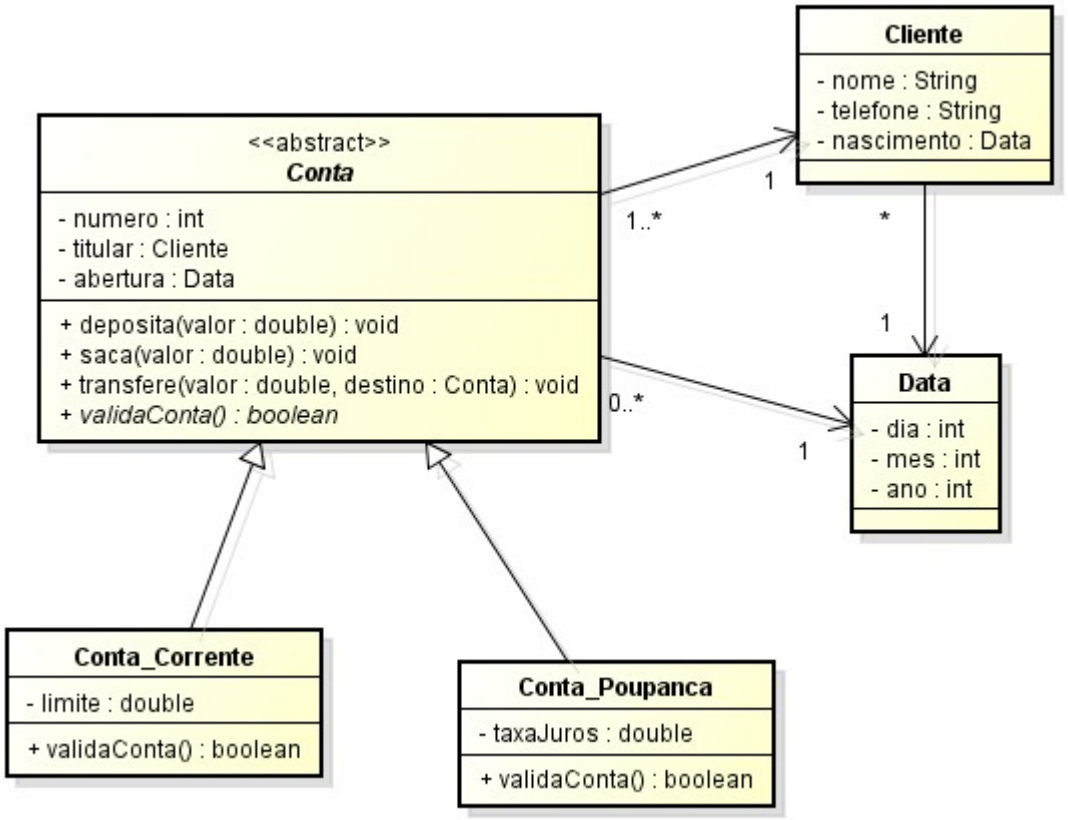


Relacionamento: Associação

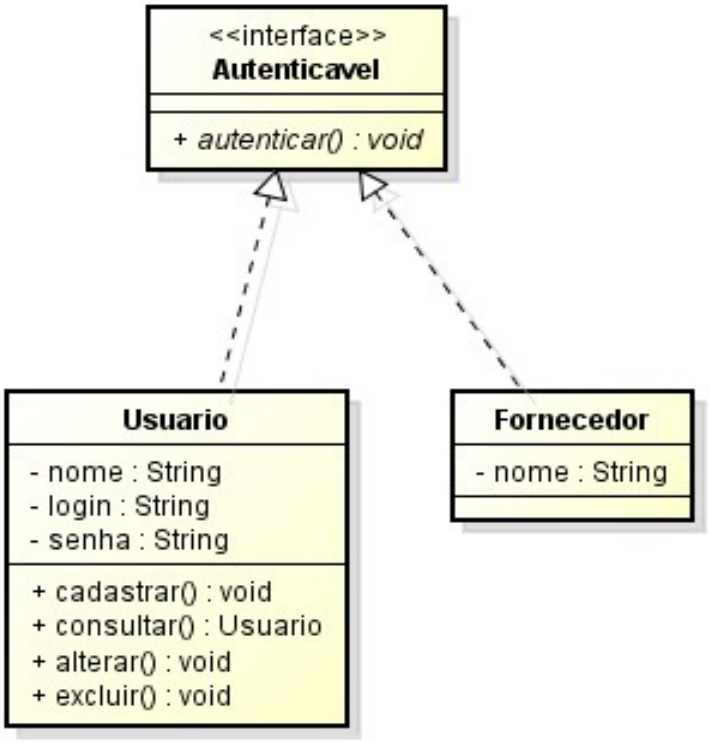
- Exemplo de Implementação

```
public class Pessoa {
    String nome;
}
public class Empresa {
    String nome;
}

public class Emprego {
    Pessoa pessoa;
    Empresa empresa;
    double salario;
    String descricao;
}
```

Classes e Métodos Abstratos - UML

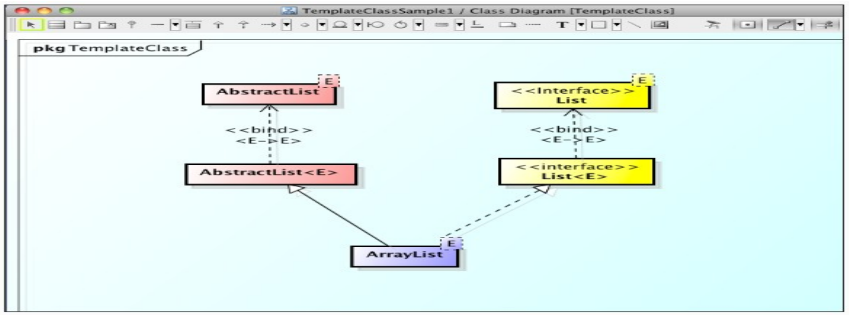


Interfaces - UML



Ferramentas para modelagem UML

- ASTAH http://astah.net/download



- Dia
- Umbrello
- ArgoUML