

DIAGRAMA DE CLASSES

OS DIAGRAMAS E MODELOS UML



NAS AULAS ANTERIORES...

- Modelo de Casos de Uso:
 - O modelo de casos de uso fornece uma perspectiva do sistema a partir de um ponto de vista **externo**.

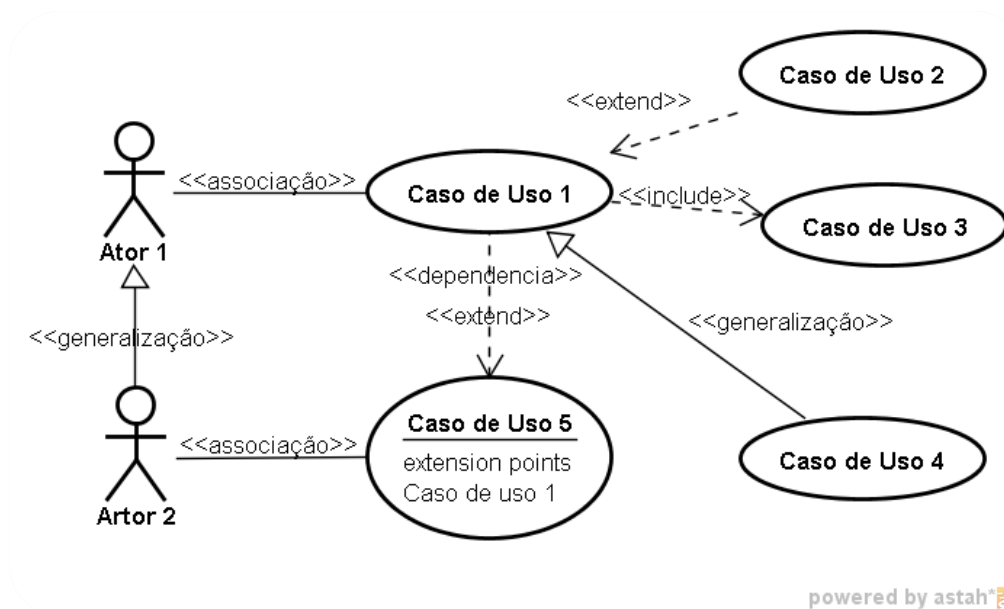


DIAGRAMA DE CLASSES

- Diagrama mais utilizado da UML.
- Permite a visualização das classes utilizadas pelo sistema e como elas se relacionam.
- Apresenta uma visão estática de como as classes estão organizadas a fim de definir sua estrutura lógica.
- Foi projetado para ser uma evolução (e não substituição) do Modelo Entidade-Relacionamento do Banco de Dados.



DIAGRAMA DE CLASSES

- A ferramenta da UML usada para representar o aspecto **estrutural estático** de um sistema é o Diagrama de Classes.
- Um **diagrama de classes** é uma representação da estrutura e relações das *classes* que servem de modelo para *objetos*.



DIAGRAMA DE CLASSES - OBJETOS

- O mundo real é formado de “coisas” que são úteis para realização de determinadas tarefas.
- Na orientação a objetos, “coisas” do mundo real são denominadas **objetos**.



DIAGRAMA DE CLASSES – CLASSE DE OBJETOS

- Seres humanos costumam agrupar os objetos para entendê-los.
- A Orientação a Objetos utiliza-se de um componente chamado **classe**, que tem por objetivo agrupar objetos semelhantes necessários para realização das funcionalidades do sistema em estudo.

Classe dos equipamentos de Informática



Classe dos equipamentos de Comunicação



DIAGRAMA DE CLASSES – CLASSE DE OBJETOS

Classe de objetos

Juntar objetos em uma classe é o mesmo que classificá-los por possuírem a mesma estrutura de dados (atributos) e o mesmo comportamento (operações).

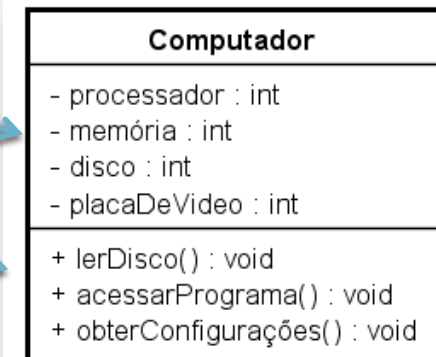


DIAGRAMA DE CLASSES – CLASSE DE OBJETOS

- Uma **classe** é representada por uma “caixa” com, no máximo, três compartimentos:
 - **Nome da classe:** por convenção esse nome é apresentado no singular e com as palavras componentes começando por maiúsculas;
 - **Atributos:** correspondem às informações que um objeto armazena;
 - **Operações:** correspondem às ações que um objeto sabe realizar.

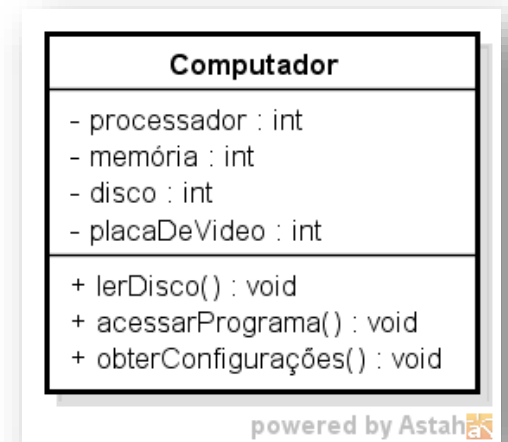


DIAGRAMA DE CLASSES – MENSAGENS

Mensagem

Representam aquilo que um objeto sabe fazer, portanto, representam as tarefas que um objeto sabe realizar.



DIAGRAMA DE CLASSES – MENSAGENS

Mensagem

- Para que um objeto realize alguma tarefa, deve haver um estímulo enviado a este objeto.
- Isto quer dizer que um objeto só irá executar a operação, que sabe realizar, se for solicitado por outro objeto.

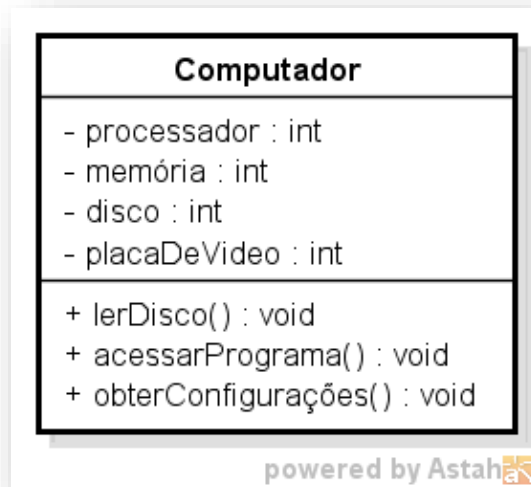


DIAGRAMA DE CLASSES

- É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.
- Representada por um retângulo que pode possuir até três divisões:
 - Nome da classe
 - Atributos da classe
 - Métodos da classe

Nome

Atributos
(características)

Métodos
(comportamento)

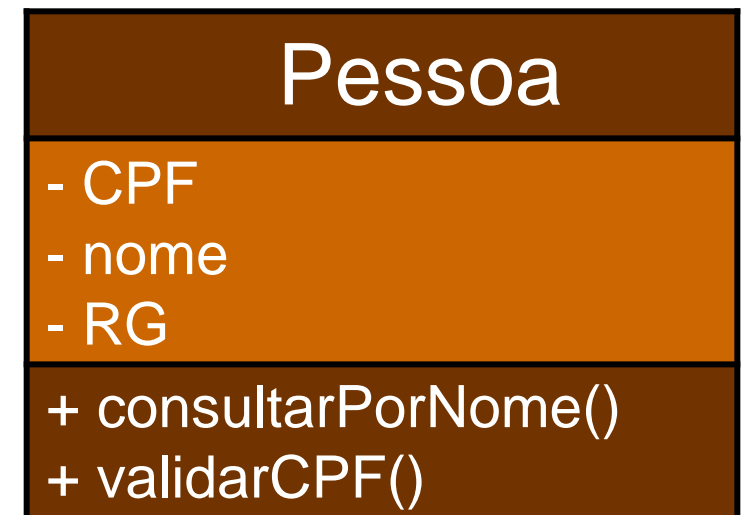


DIAGRAMA DE CLASSES – ATRIBUTOS

Atributos de uma Classe

- Um **atributo** é uma propriedade nomeada de uma classe que descreve um intervalo de valores que os objetos podem assumir.
- Uma classe pode ter qualquer numero de atributos ou mesmo nenhum atributo.
- Os atributos podem ser representados exibindo apenas o seu nome.
- O nome de um atributo começa com letra minúscula e quando o nome é composto o segundo nome começa com letra maiúscula.

NomeDaClasse
- atributoUm : int
- atributoDois : int
- atributoN : int

DIAGRAMA DE CLASSES – OPERAÇÃO

Operações de uma Classe

- Uma **operação** é a implementação de um serviço que pode ser solicitado por algum objeto.
- Uma operação é uma abstração do que os objetos da classe sabem fazer.
- Uma classe pode ter qualquer numero de operações ou até não ter nenhuma operação.
- O nome de uma operação é um verbo ou uma locução verbal breve, representa algum comportamento da classe correspondente.
- O nome de uma operação começa com letra minúscula e quando o nome é composto o segundo nome começa com letra maiúscula.

NomeDaClasse

+ operaçãoUm() : void
+ operaçãoDois() : void
+ operaçãoN() : void

DIAGRAMA DE CLASSES

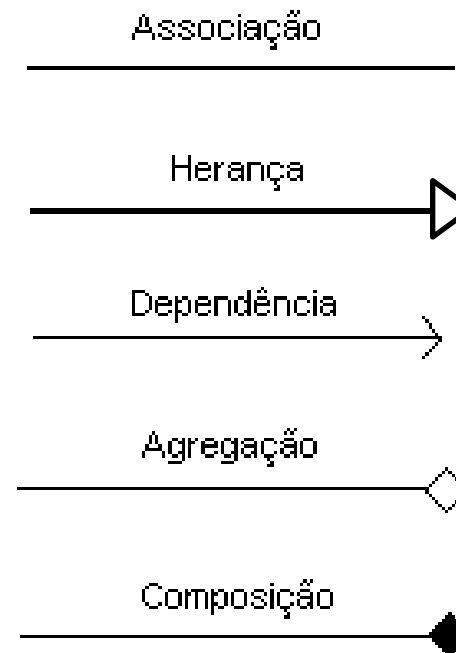
EXEMPLO

- **Atributo**
 - Representa características de uma classe.
 - Exemplo: Jogador (nome, sexo, idade etc.).
- **Método**
 - Representa atividades que um objeto de uma classe pode executar.
 - Exemplo: Jogador (correr, driblar, chutar).
- **Visibilidade**
 - Indica o nível de acessibilidade de um atributo ou método.
 - Tipos: Pública (+), Privada (-) e Protegida (#).



RELACIONAMENTO

- Permite compartilhar informações e colaborar com a execução dos processos do sistema.
- Descreve um vínculo que ocorre, normalmente, entre os objetos de uma ou mais classes.
- Os tipos de relacionamentos são:
 - Associação
 - Agregação
 - Composição
 - Especialização/Generalização
 - Dependência



ASSOCIAÇÃO

- Descreve um conjunto de vínculos entre elementos de modelo.
- Relacionamento estrutural que especifica objetos de um item conectados a objetos de outro item:
 - Associação binária – quando há duas classes envolvidas na associação de forma direta de uma para outra.
 - ✓ Relacionamento entre duas classes (tipo mais comum).
 - ✓ Podem possuir títulos para determinar o tipo de vínculo.
 - Associação unária – quando há um relacionamento de uma classe consigo mesma. Se comparada ao modelo ER, seria um auto-relacionamento.

ASSOCIAÇÃO UNÁRIA (OU REFLEXIVA)

- Ocorre quando há um relacionamento de um objeto de uma classe com objetos da mesma classe;
- No exemplo abaixo, percebe-se que um objeto da classe Funcionário pode (ou não) supervisionar outros objetos dessa mesma classe;
- Para o relacionamento ficar mais claro, pode-se informar a sua multiplicidade.

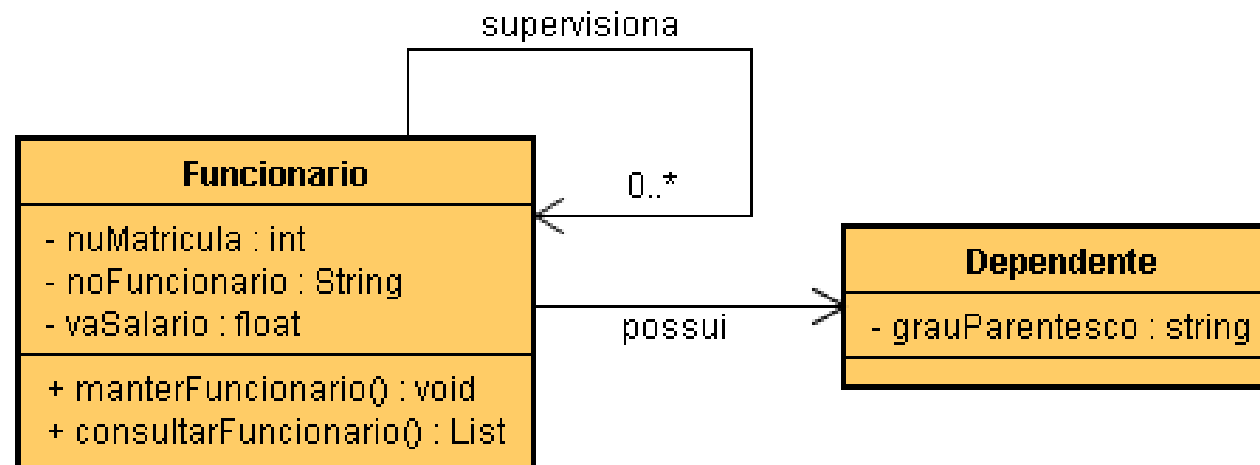
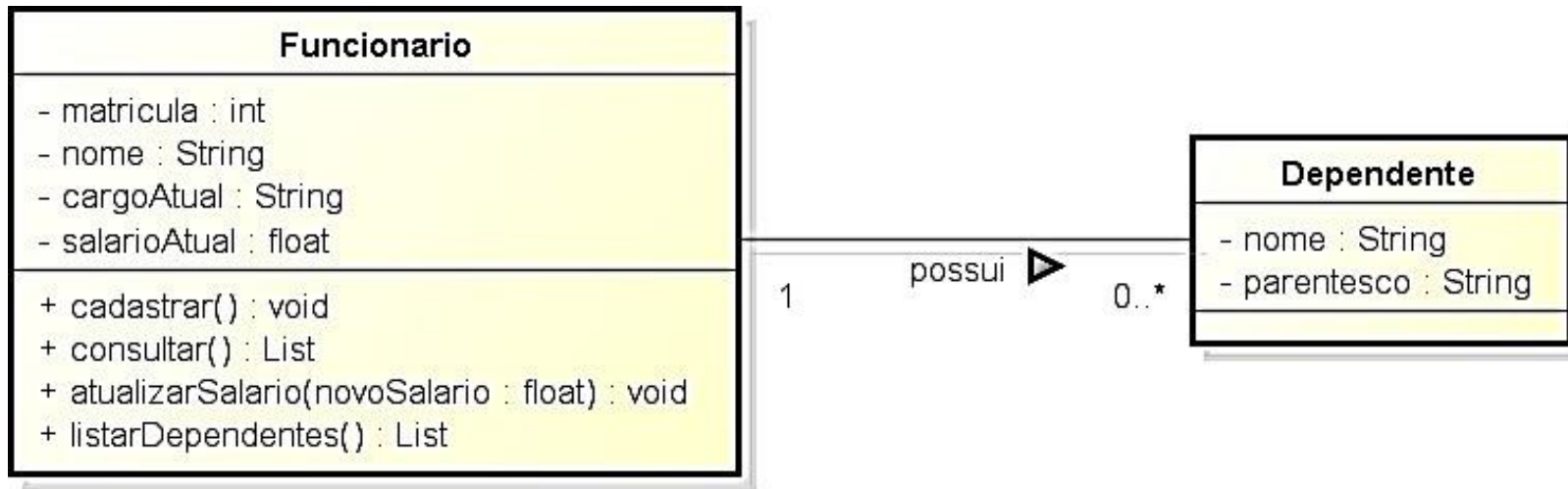


DIAGRAMA DE CLASSE - ASSOCIAÇÃO BINÁRIA



```
public class Funcionario {
    private int matricula;
    ...
    private Dependente[] dependentes;

    // métodos
    ...
}
```

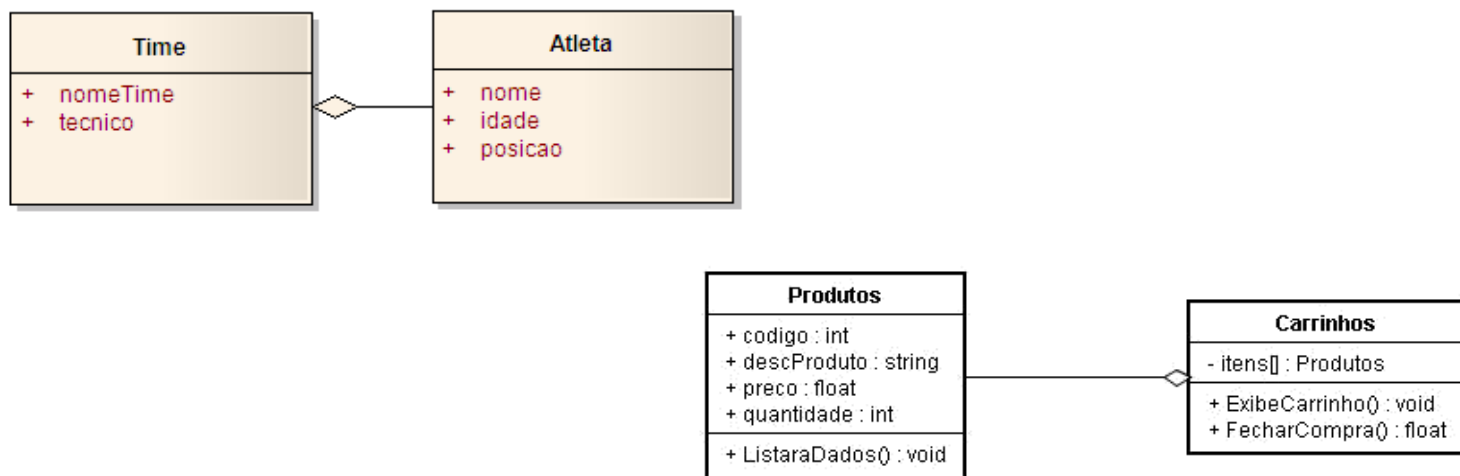
```
public class Dependente {
    private String nome;
    private String parentesco;
    private Funcionario funcionario;
}
```

MULTIPLICIDADE

Multiplicidade	Significado
0..1	No mínimo zero e no máximo um. Os objetos não precisam estar relacionados, porém se houver relacionamento deve ser de no máximo 1.
1..1	Um e somente um
0..*	No mínimo nenhum e no máximo muitos.
*	Muitos
1..*	No mínimo um e no máximo muitos.
3..5	No mínimo 3 e no máximo 5.

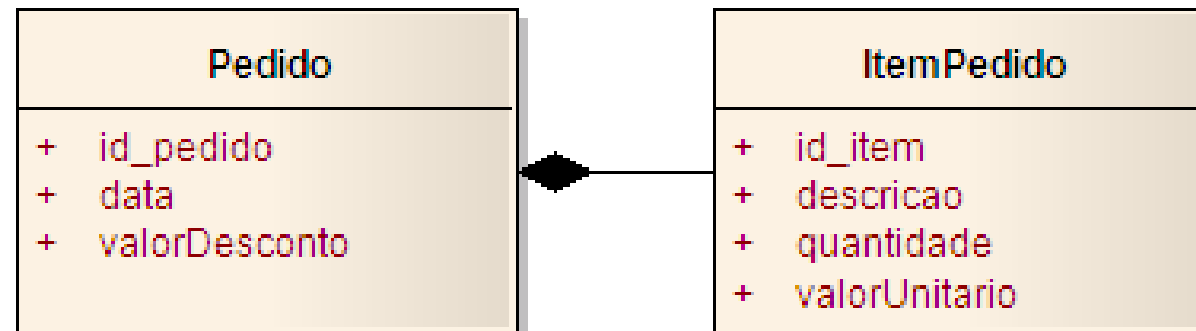
AGREGAÇÃO

- Tipo especial de associação que tenta demonstrar que as informações de um objeto-todo precisam ser complementadas pelas informações contidas em um (ou mais) objetos-parte.
- A existência do objeto-parte faz sentido mesmo não existindo o objeto-todo.
- A associação de agregação pode, em muitos casos, ser substituída por uma associação binária simples, dependendo da visão de quem faz a modelagem.



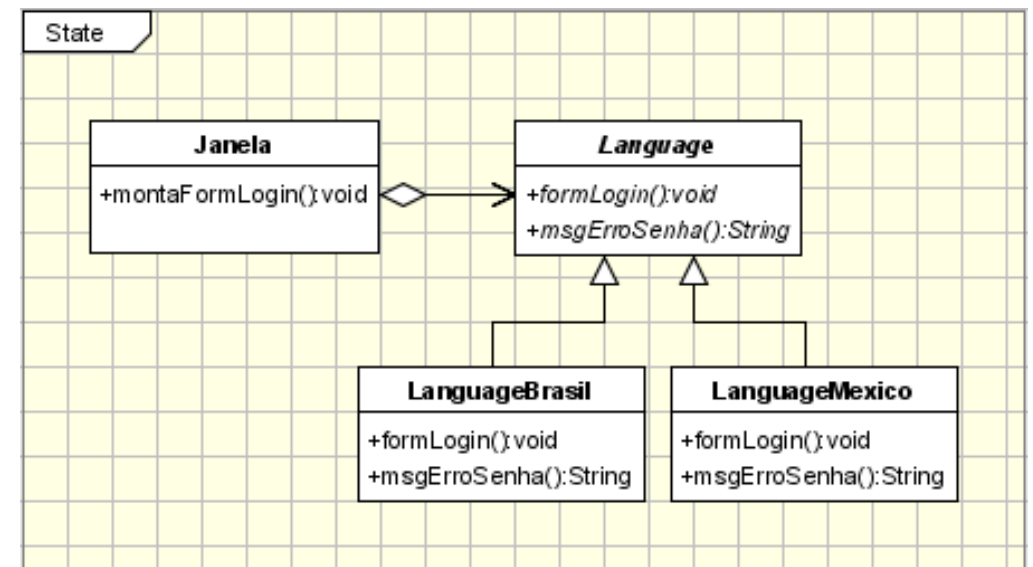
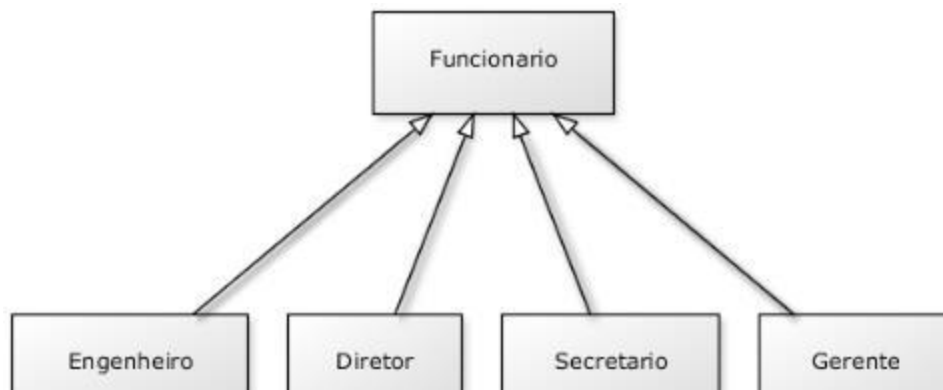
COMPOSIÇÃO

- É uma variação da agregação e considerada mais “forte”.
- O objeto-parte não pode existir sem o objeto-todo.
- Se o objeto-todo for destruído, o objeto-parte também será.



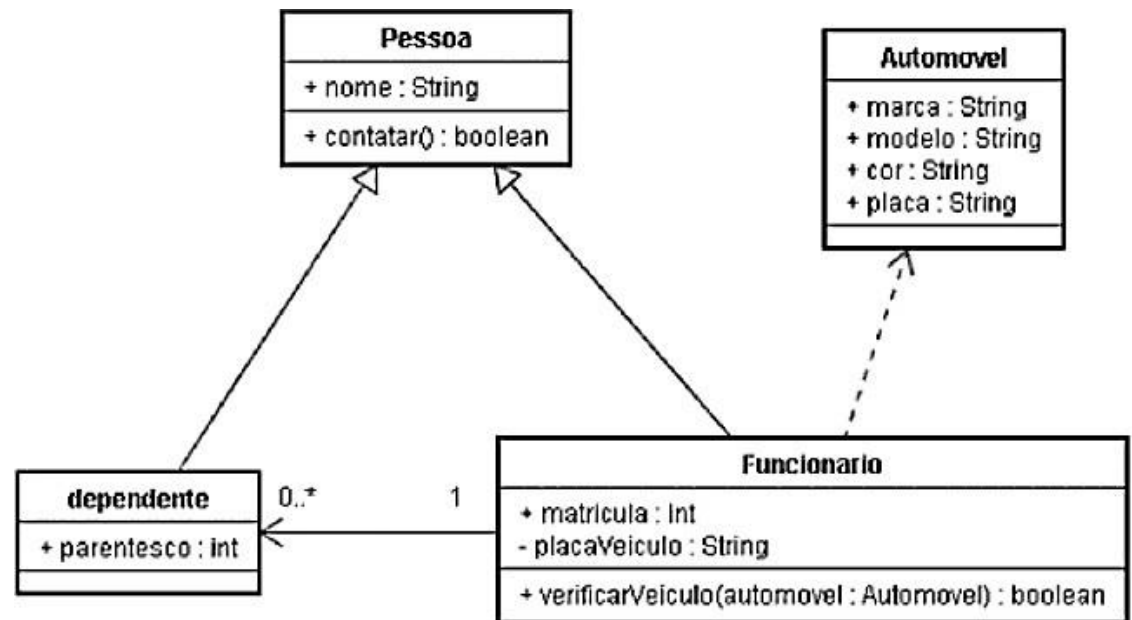
ESPECIALIZAÇÃO/GENERALIZAÇÃO

- Tem como objetivo identificar classes-mãe, denominadas de gerais, e classes-filha chamadas de especializadas;
- São chamados de relacionamentos “é um tipo de”.



DEPENDÊNCIA

- Como o nome sugere, indica um grau de dependência entre uma classe e outra.
- Uma dependência difere de uma associação porque a conexão entre as classes é temporária.
- Representada por uma seta tracejada entre duas classes.



Funcionário não instancia um Automóvel, apenas usa-o como parâmetro de um método.

CLASSE ASSOCIATIVA

- Utilizada quando ocorrem associações que possuem multiplicidade muitos para muitos em todas as suas extremidades;
- Armazena os atributos transmitidos pela associação;
- Pode possuir seus próprios atributos;
- Representada por uma reta tracejada partindo do meio da associação até uma classe.

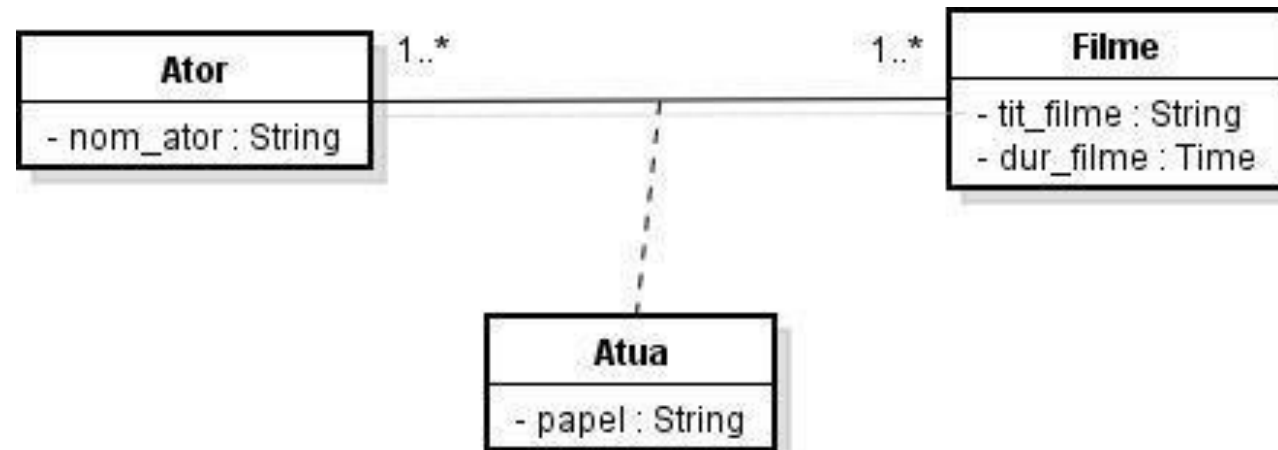


DIAGRAMA DE CLASSE

Classe intermediária

- Substitui as classes associativas;
- Apresenta, exatamente, a mesma função da classe associativa;.
- Pode possuir seus próprios atributos;



RESTRIÇÃO

- Informações extras que definem condições a serem validadas durante a implementação dos métodos de uma classe, das associações entre as classes ou mesmo de seus atributos;
- Representadas por textos limitados por chaves.

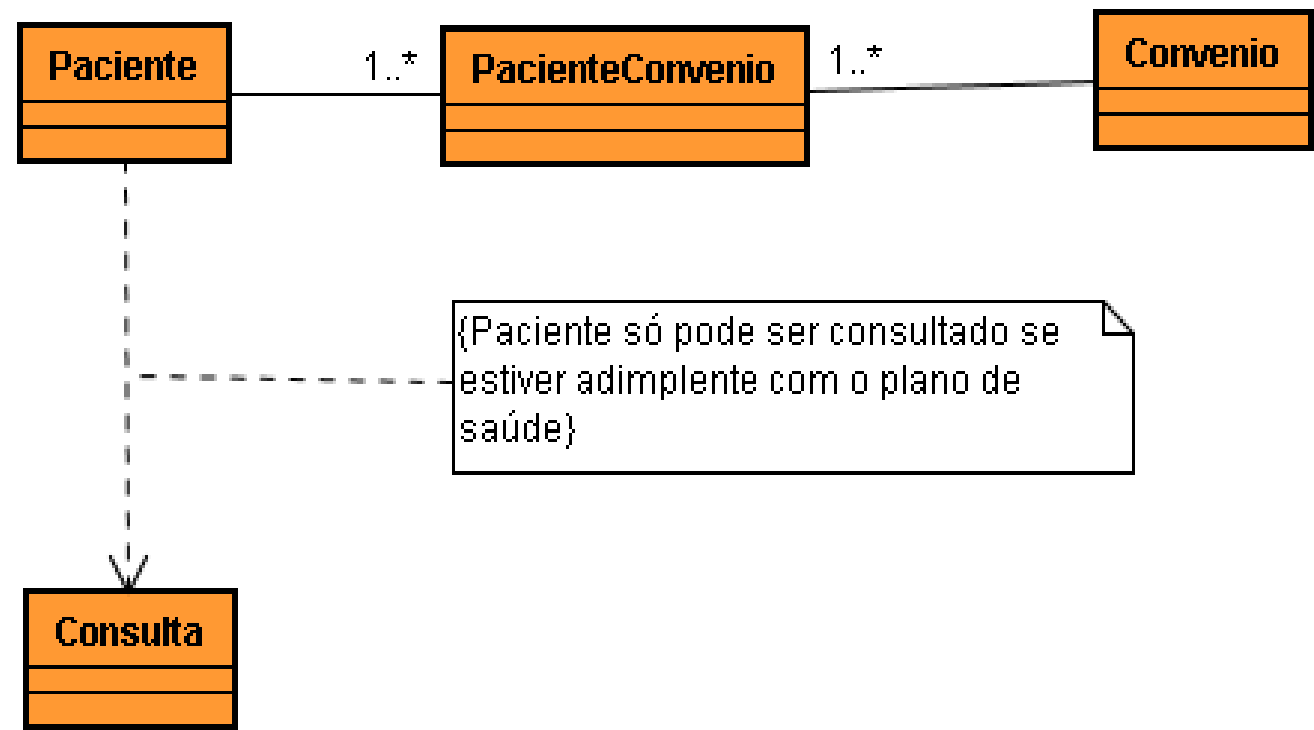


DIAGRAMA DE CLASSES

Processo para identificar classes

- Nas descrições dos casos de uso, as classes são identificadas através dos substantivos.
- Exemplos: aluno, curso, disciplina, cliente, pedido, produto, etc.
- Considerar que:
- Substantivo é uma palavra que denomina um ser ou um objeto, ou uma ação, qualidade ou estados;
- Substantivo também é uma palavra, que sem o auxilio de nenhuma outra, designa a substancia.

DIAGRAMA DE CLASSES - PROCESSO PARA IDENTIFICAR CLASSES

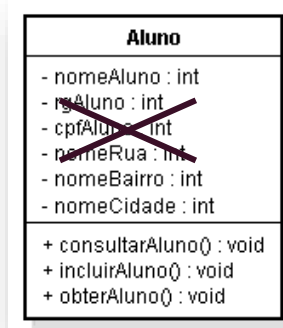
- **Método dirigido a responsabilidade**
- Esse método, para identificação de classes em casos de uso, parte do princípio que objetos possuem responsabilidades e necessitam, algumas vezes, de colaboradores para cumprirem com seus objetivos. Logo, esses colaboradores também devem ser considerados classes candidatas para o domínio da aplicação.

DIAGRAMA DE CLASSES - PROCESSO PARA IDENTIFICAR CLASSES

Método dirigido a responsabilidade

- Se um objeto tem uma responsabilidade com a qual não pode cumprir sozinho, ele deve requisitar colaborações de outros objetos.

Não normalizado →



↗ Normalizado

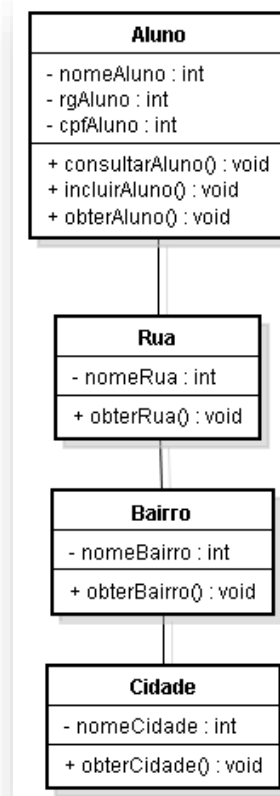


DIAGRAMA DE CLASSES - PROCESSO PARA IDENTIFICAR CLASSES

Método dirigido a responsabilidade

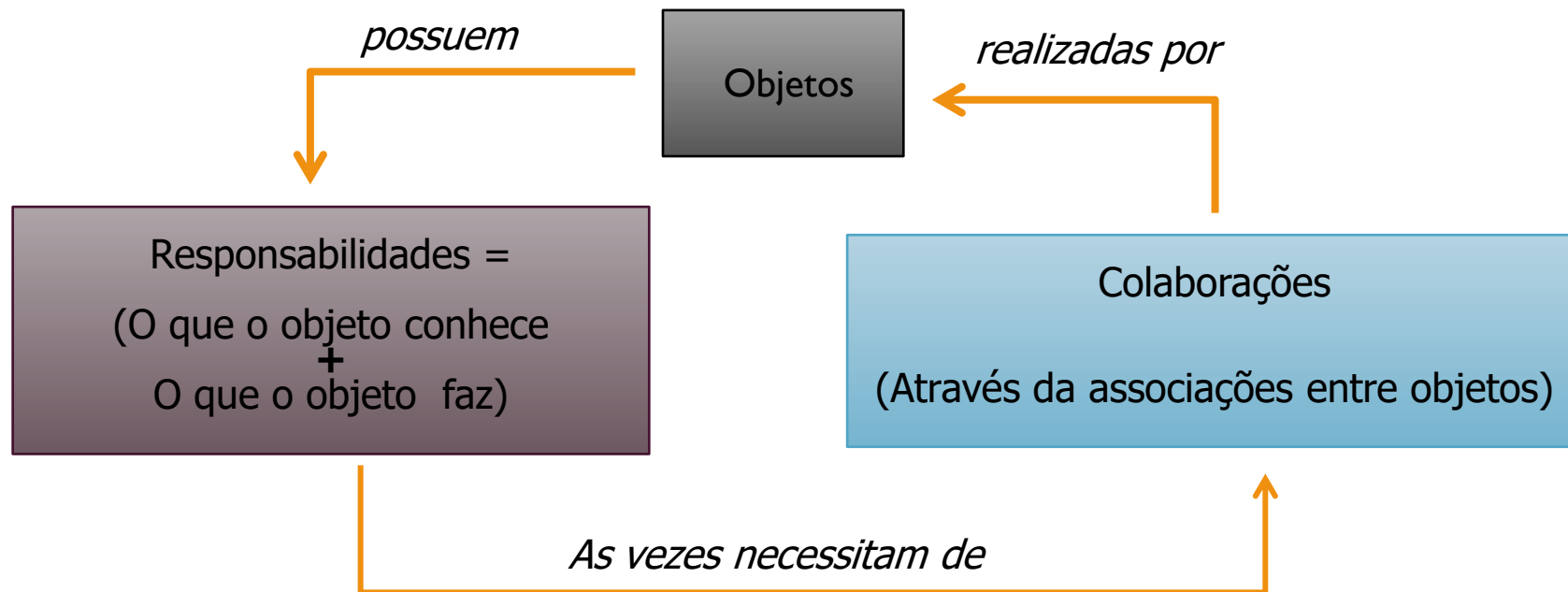


DIAGRAMA DE CLASSES

■ Categorias de responsabilidades

- Os objetos de um sistema sabem fazer alguma coisa. E o que sabem fazer pode ser categorizado de acordo com o tipo de responsabilidade a ele atribuída:
 - ❖ objetos de fronteira;
 - ❖ objetos de controle;
 - ❖ objetos de entidade;
 - ❖ objetos de interface.

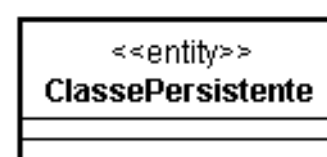
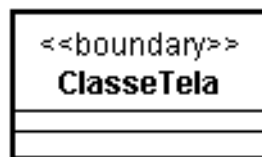


DIAGRAMA DE CLASSES

Objetos de Fronteira

- Um **objeto de fronteira** existe para que o sistema se comunique com o mundo exterior. São os objetos que conhecemos por “*tela do sistema*”.
- Traduzem os eventos gerados por um ator em eventos relevantes ao sistema.
- São responsáveis por apresentarem os resultados de uma interação dos objetos em algo inteligível pelo ator. Por consequência, estes objetos são altamente dependentes do ambiente tecnológico adotado.



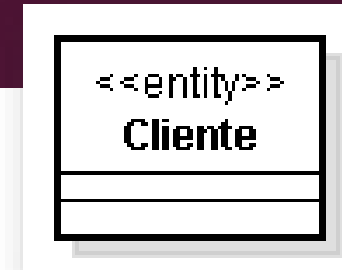
DIAGRAMA DE CLASSES

Objetos de Controle



- São a “ponte de comunicação” entre objetos de fronteira e objetos de entidade.
- Responsáveis por implementar a lógica de execução constantes em um caso de uso.
- Implementam o que conhecemos por programa de computador e são altamente dependente da linguagem de programação utilizada.
- Traduzem eventos externos em operações que devem ser realizadas pelos demais objetos que compõem um caso de uso.
- Os objetos de controle são tipicamente ativos, consultam informações e requisitam serviços de outros objetos.

DIAGRAMA DE CLASSES



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.
- Atores não têm acesso direto aos objetos da classe de entidade.
- Objetos de entidade normalmente participam de vários casos de uso e têm um ciclo de vida longo.
- Um objeto da classe Aluno pode participar dos casos de uso: Realizar Matricula, Registrar Avaliação, Registrar Pagamento, Emitir Boleto, etc.

DIAGRAMA DE CLASSES

Objetos de Interface

- **Classes de interface** realizam a comunicação do sistema com outros sistemas ou equipamentos.
- Uma classe de interface realiza a comunicação com outros sistemas, representam propriedades de uma interface de comunicação.
- Tipicamente os objetos de interface têm a responsabilidade de notificar aos objetos de controle os eventos gerados externamente ao sistema.

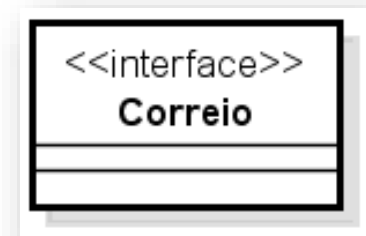
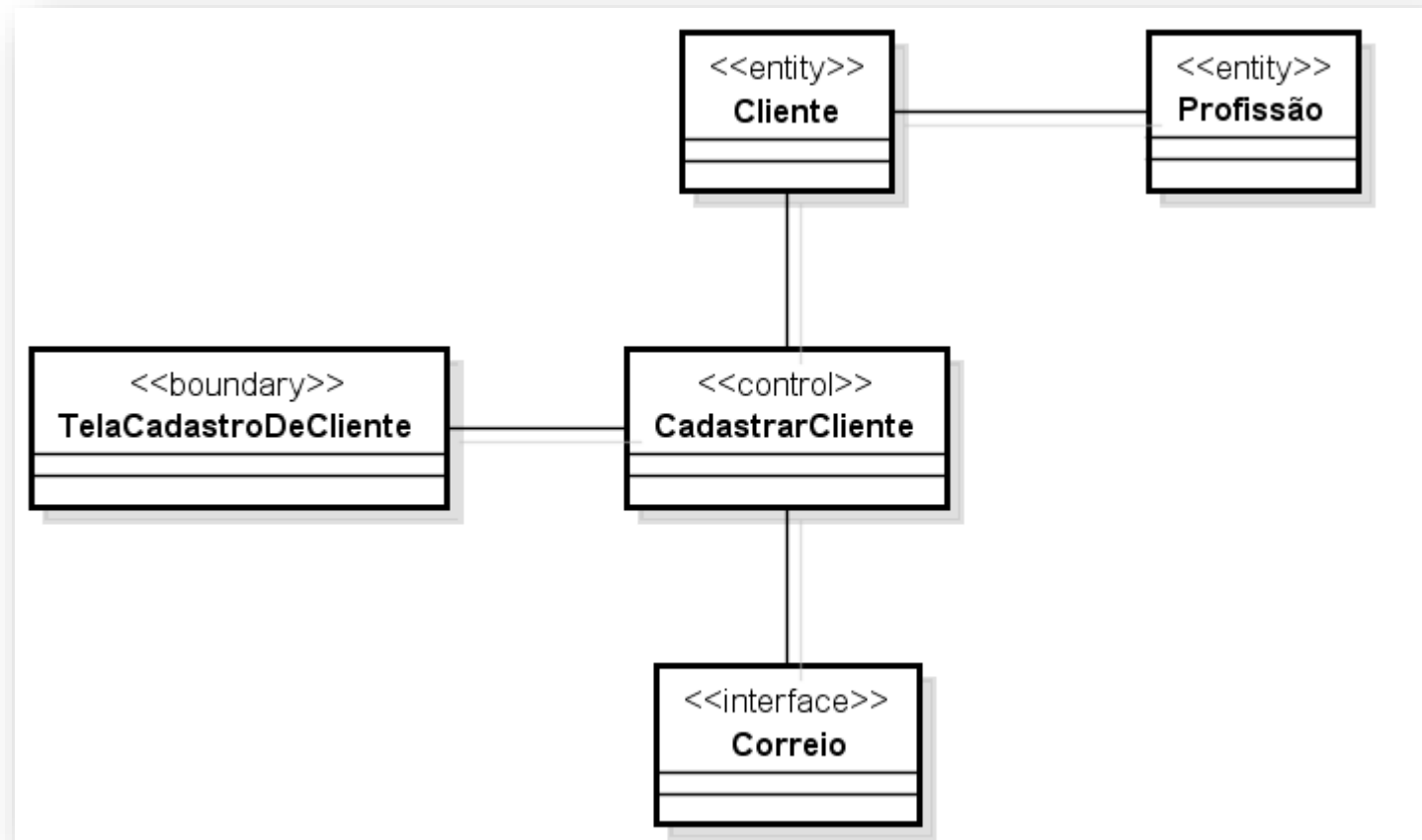


DIAGRAMA DE CLASSES

Exemplo:



EXERCÍCIO

Sistema de Controle de Cinema

Desenvolva o diagrama de classes para um sistema de controle de cinema sabendo que:

Um cinema pode ter muitas salas, sendo necessário, portanto, registrar informações a respeito de cada uma como sua capacidade, ou seja, o número de assentos disponíveis.

O cinema apresenta muitos filmes. Um filme tem informações como título e duração. Assim, sempre que um filme for apresentado, deve-se registrá-lo também.

Um filme tem um único gênero, mas um gênero pode se referir a muitos filmes.

Um filme pode ter muitos atores atuando nele, e um ator pode atuar em muitos filmes. Em cada filme, um ator interpretará um ou mais papéis. Por uma questão de propaganda, é útil anunciar os principais atores do filme e que papéis eles interpretam.

Um mesmo filme pode ser apresentado em diferentes salas e em horários diferentes. Cada Apresentação em uma determinada sala e horário é chamado de Sessão. Um filme sendo apresentado em uma sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala.

Os clientes do cinema podem comprar ou não ingressos para assistir uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informações do tipo de ingresso (meio ingresso ou ingresso inteiro). Além disso, um cliente só pode comprar ingressos para sessão ainda não encerradas.