Análise Orientada à Objetos

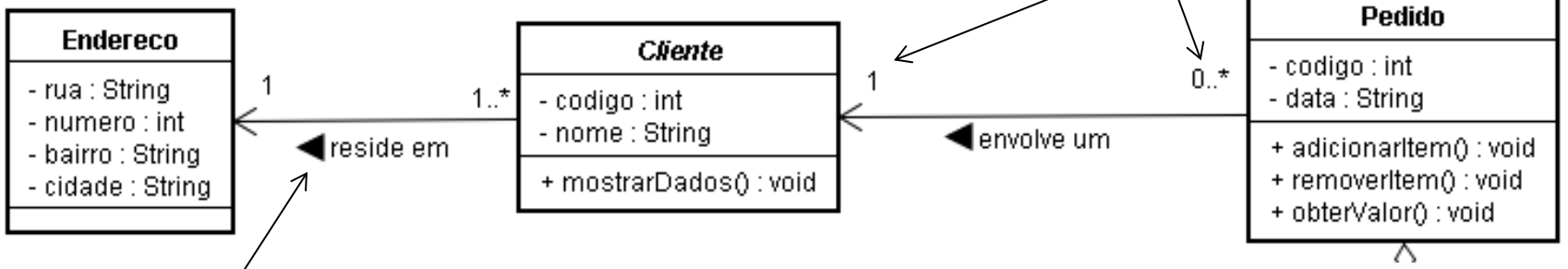
Associação e Dependência

Relacionamentos

- Como fornecer suporte ao envio de mensagens?
  - Por meio de "relacionamentos" entre as classes!
    - Associação
    - Dependência
    - Herança e Herança Múltipla
    - Todo-parte (agregação)
- Entenda bem a "semântica" dos relacionamentos para não cometer erros de modelagem !
- Para diagramas conceituais, apenas:
  - Associações, herança e agregação

Relacionamento de Associação

- Associação é um relacionamento estrutural que ocorre entre classes;
- Cuidado ! Associações não tem o objetivo de denotar sequenciamento de ações !



Direção do nome

```
class Cliente {

    end Endereco;
    ...
}
```

```
class Pedido {

    cli Cliente;
    ...
}
```

Relacionamento de Associação


```
class Turma {

    alunos Aluno[ ..];
    …
}
```

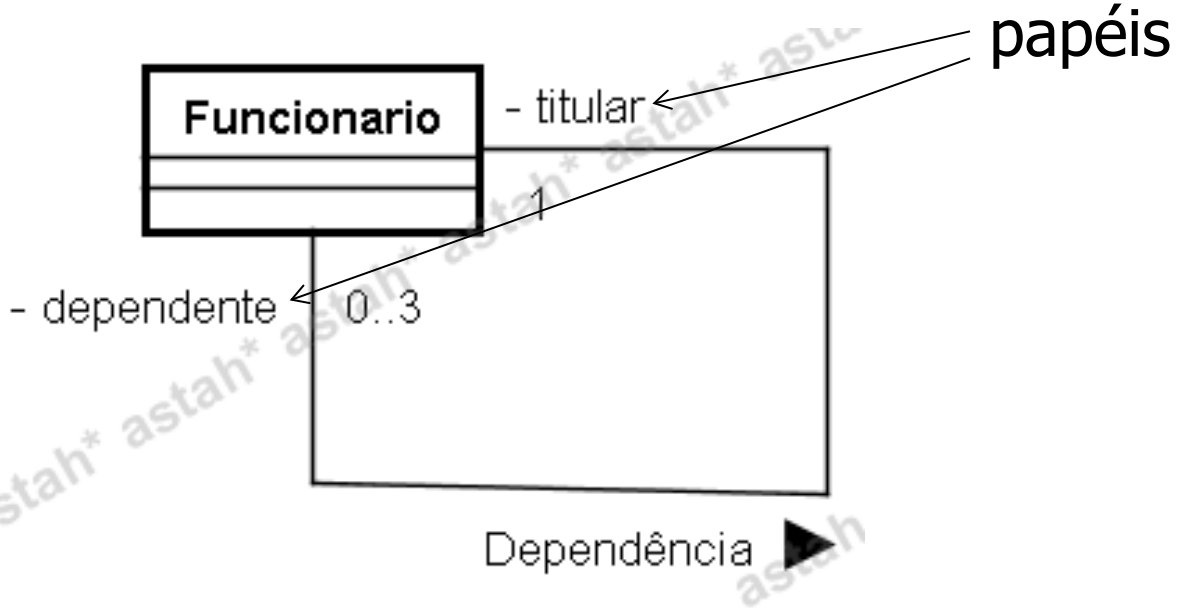
```
class Aluno {


    }
```

O relacionamento de associação se concretiza na forma de atributos em uma classe do tipo da outra.
No caso acima, o atributo que representa o relacionamento é Colocado apenas na classe Turma por causa da Navegabilidade

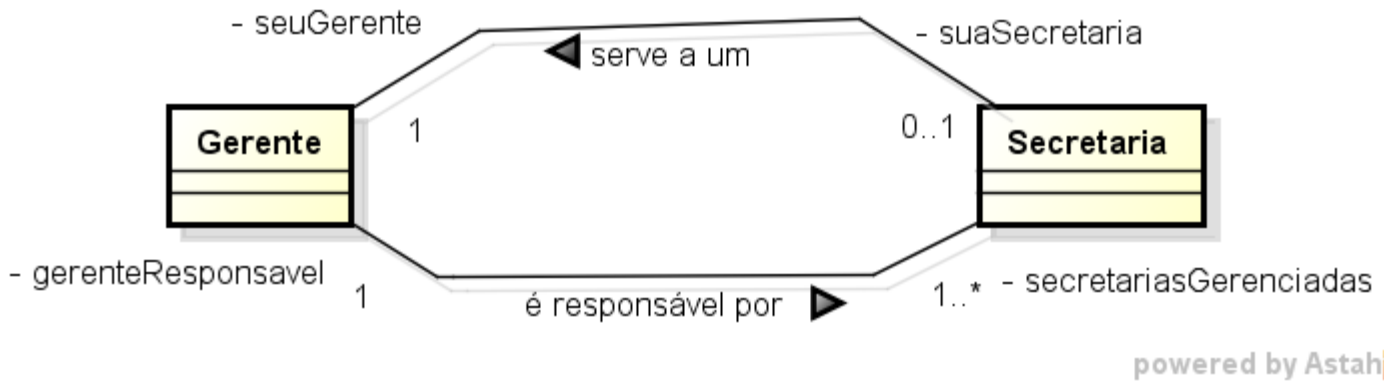
Conceitos da OO
Associação (em nível conceitual)

- Auto-relacionamento e papéis (roles)



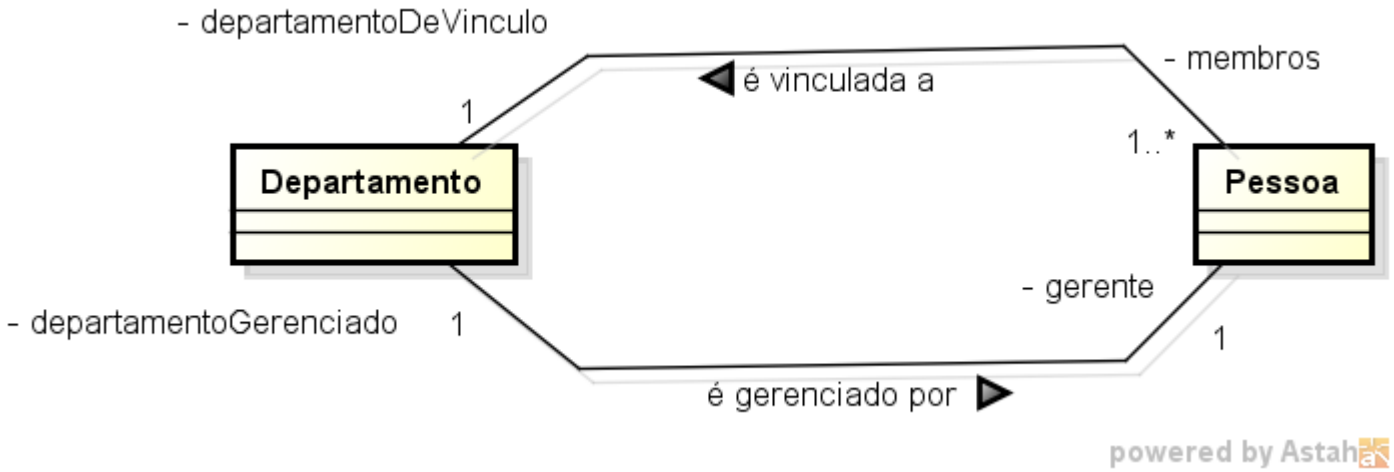
N relacionamentos entre as mesmas classes

- Não é incomum a existência de mais de um relacionamento entre as mesmas classes



N relacionamentos entre as mesmas classes

- Não é incomum a existência de mais de um relacionamento entre as mesmas classes



Relacionamentos de Dependência

Dependência

Três possibilidades na implementação:

- A classe 4 possui um ou mais métodos que instanciam Class 5
- A classe 4 possui um mais métodos que possuem o tipo Class 5 como parâmetro
- A classe 4 possui um ou mais métodos que possuem o tipo Class 5 como tipo de Retorno

Relacionamentos de Dependência

Dependência

É um relacionamento mais fraco do que uma associação. Isto é, o acoplamento causado por uma associação é mais difícil de ser removido, já que os dados da classe são do tipo de outra.

No caso da dependência, se ela existir por causa de um único método, esse acoplamento pode ser eliminado eliminando um único método da classe. Algo que (teoricamente) é mais fácil do que remover um atributo (já que muitos métodos podem usar aquele atributo)

Se a classe já possui um relacionamento de associação, não é necessário indicar também um de dependência (caso exista).

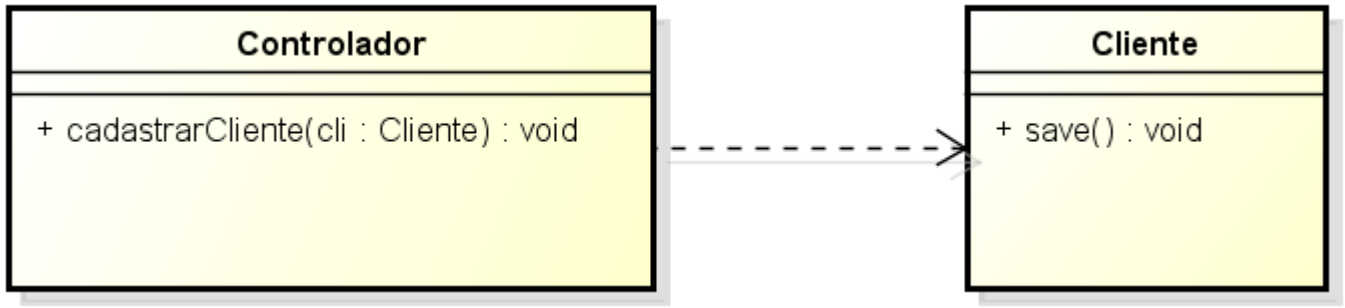
Relacionamentos de Dependência

Dependência

Embora cada uso da Class5 dentro da Class 4 seja uma dependência, não existe a necessidade de criar várias dependências. Isto é, um relacionamento de dependência significa que dentro da classe origem há 1 ou n menções à outra classe em seus métodos.

Por exemplo, pode ser que dentro de Class 4, todos os métodos instanciem Class5. Mesmo assim haverá apenas um relacionamento de dependência.

Relacionamentos de Dependência
(exemplo)




```
public class Controlador {

    // não tem atributos do tipo Cliente

    public void cadastrarCliente(Cliente: cli)
    {
       ...
       cli.save;
       ...
     }
```

```
public class Cliente {


    public boolean save {
          // pega os dados próprios
          // monta String SQL
          // abre conexão e grava no BD
    }
```

Relacionamentos de Dependência

Dependência

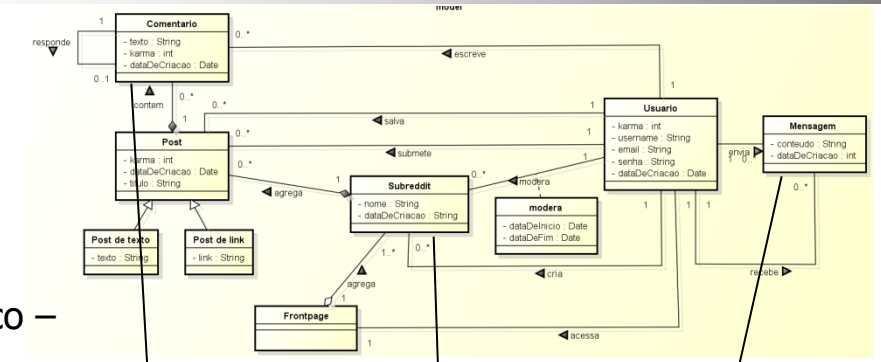
Visibilidade:

A não ser que explicitamente declarado, todas as classes dentro de um pacote podem instanciar/usar como parâmetro/ todas as outras em seus métodos.

Assim, é possível de criarmos relacionamentos de dependência entre quaisquer classes do sistema. Entretanto, essa liberdade é perigosa e deve ser controlada.

Uma prática interessante é, por exemplo, tentar não criar dependência alguma. Isto é, permitir a comunicação apenas entre classes que já estão associadas (associação).

Visão Estática e Dinâmica

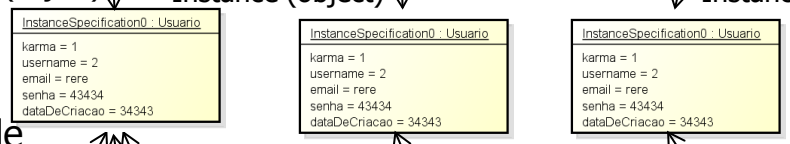


Classes (Visão Estática – disco – Tempo de compilação)

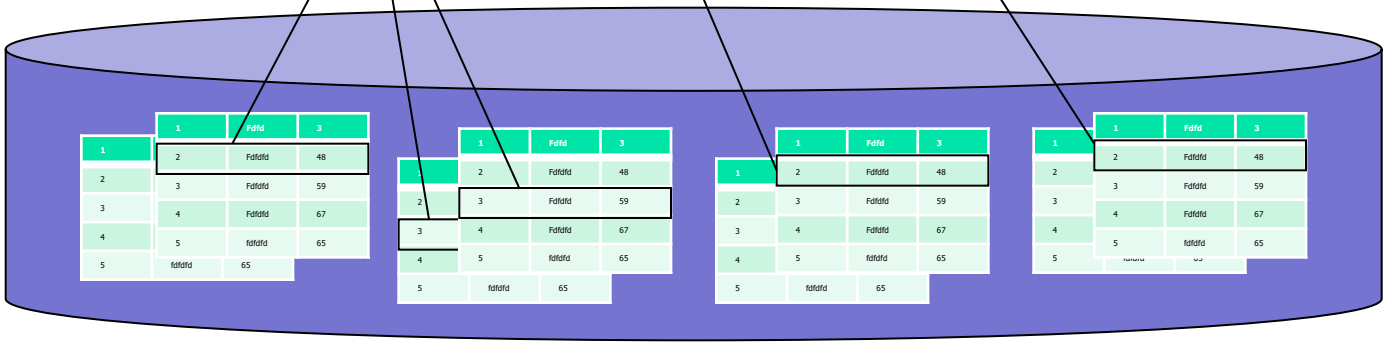
Instance (object)

Instance (object)

Instance (object)



Objetos (Visão Dinâmica memória volátil RAM – tempo de execução)



Tabelas (Visão Estática –Repositório de Dados (BD, arquivo texto, XML, etc)

Note que em geral tem-se uma Classe para cada tabela de BD. Entretanto, nada impede de existir uma classe que não Possui uma tabela correspondente.

Quando usamos Design Patters, por exemplo, haverão diversas classes sem tabelas correspondentes.

Também não é incomum a existência de classes que agrupam informações oriundas várias tabelas.