Engenharia de Requisitos

Modelo de Classes

Professor

Alberto Tavares da Silva

- ✓ Um diagrama de classes mostra a existência das classes e os seus relacionamentos numa visão lógica e estática do sistema
- ✓ A UML modela os elementos de um diagrama de classes:
  - ✓ Classes, sua estrutura e comportamentos
  - ✓ Associações, agregações, dependência e relacionamentos de herança
  - ✓ Multiplicidade e indicadores de navegação

Classe

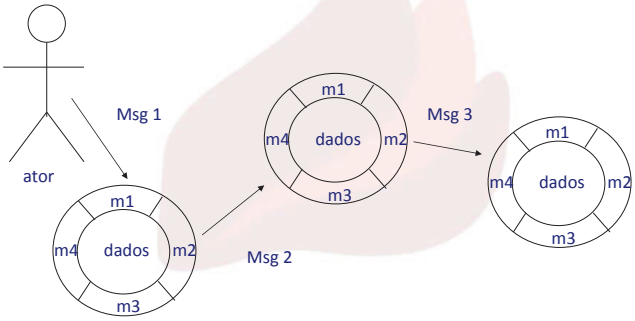
Uma **classe** descreve um conjunto de objetos com:

- ➢ propriedades semelhantes;
- ➢ comportamentos semelhantes;
- ➢ relacionamentos comuns com outros objetos.

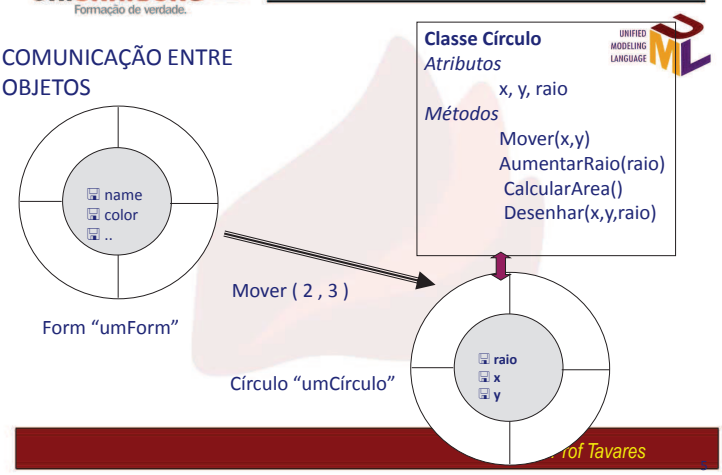
A classe é uma fábrica de objetos.

Modelagem de Objetos

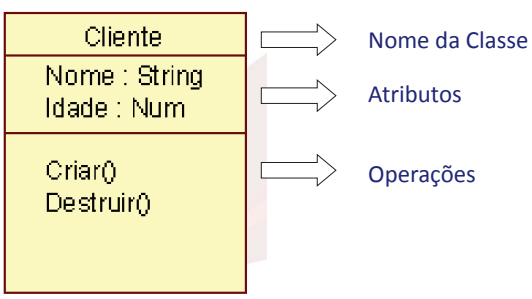
Paradigma objeto/mensagem/método



COMUNICAÇÃO ENTRE OBJETOS



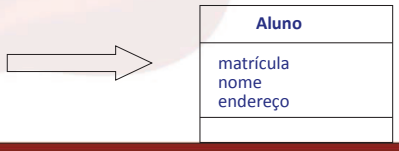
Classe - Diagramação



Atributos

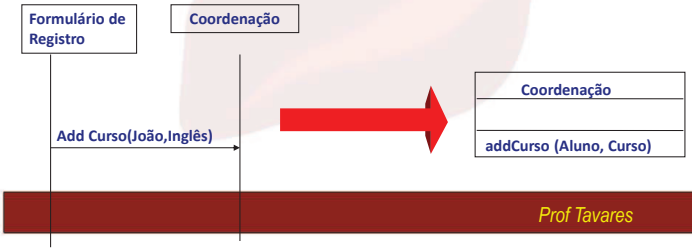
- ✓ A estrutura da classe é representada por seus atributos
- ✓ Os atributos podem ser encontrados examinando-se as definições das classes, as suas características e aplicando-se o conhecimento do domínio

Cada Aluno tem uma matrícula, nome, endereço



Operações

- ✓ O comportamento da classe é representado por operações
- ✓ As operações podem ser encontradas examinando-se os diagramas de interação



Relacionamentos

Os relacionamentos ligam as classes/objetos entre si criando relações lógicas entre estas entidades.

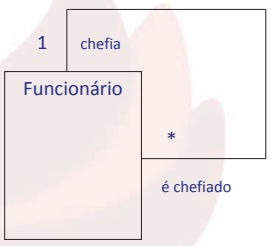
Podem ser:

- ➢ Associação
- ➢ Especialização/Generalização (Herança)
- ➢ Agregação/ composição
- ➢ Dependência

Associações

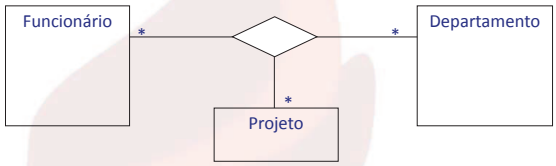
- Relação que descreve um conjunto de vínculos entre classes.
- Em relação ao número de classes envolvidas na associação, uma associação pode ser unária, binária ou n-ária.
- Uma associação entre classes é o equivalente ao relacionamento entre conjuntos de entidades, do DER.

Associação Unária ou Recursiva

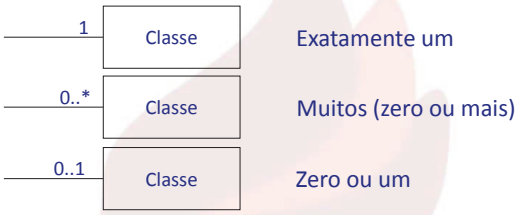


Associação Binária

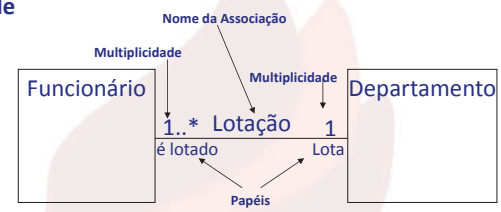
Associação Ternária



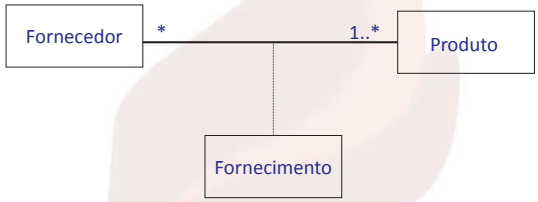
Associações - Multiplicidades



Multiplicidade
(cardinalidade)



Classe Associativa



Associações - Agregação

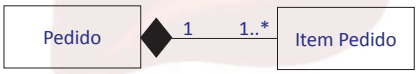
- ✓ Forma especial de Associação que serve para mostrar que uma determinada classe de objetos é composta por outra classe.
- ✓ Semanticamente indica que o objeto parte é um atributo do objeto todo (*é-parte-de*).
- ✓ A agregação oferece uma variação denominada COMPOSIÇÃO .

Associações - Agregação

A existência do objeto parte é independente do objeto todo, podendo, inclusive, fazer parte de mais de um objeto todo. A representação é feita colocando-se um losango vazio do lado do objeto todo.

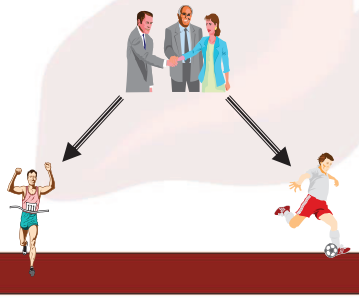
Associações - Composição

O objeto parte só tem existência se o objeto todo existir. A multiplicidade da associação é sempre 1 do lado do objeto todo. A representação é feita colocando-se um losango cheio do lado do objeto todo.



HERANÇA

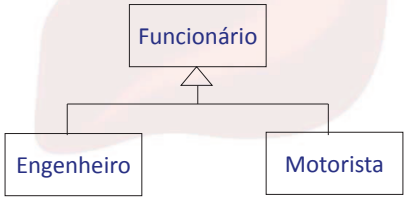
Generalização/Especialização é a abstração que permite compartilhar semelhanças, preservando diferenças



Associações - Generalização

É um relacionamento entre em elemento mais geral (superclasse) e um elemento mais específico (subclasse).
Indica que a superclasse tem atributos, operações e associações comuns que são compartilhados por todas as subclasses derivadas.

Um objeto de uma subclasse é um **tipo-de** objeto da superclasse.



HERANÇA

- Mecanismo para modelar similaridades entre classes, representando as abstrações de generalização e especialização.
- A herança possibilita:
  - Reutilização;
  - Captura explícita de características comuns;
  - Facilidade de manutenção;
  - Facilidade de criar novas classes.
- O desenvolvimento orientado a objetos é fortemente baseado na construção de hierarquias de classes.

HERANÇA

```
public class ClAluno extends ClPessoa  {

private String matricula;
private String email;

   //construtor
   public ClAluno() {
   super("");
   matricula = new String("");
   email = new String("");
}
   public ClAluno(String nome, String matricula, String email){
   super(nome);
   this.matricula = matricula;
   this.email = email;
}
   public void Imprime(){
      System.out.println("Aluno nome: " + super.nome + " matrícula: " + matricula + " E-mail: " + email);
   }
```


CLASSE ABSTRATA

- Uma classe abstrata é aquela no qual um ou mais métodos são declarados mas não definidos.
- Quando se deriva uma classe a partir de uma classe abstrata não há necessidade de definirem todos os métodos abstratos na subclasse; nesse caso, a subclasse também será abstrata.
- Não se pode instanciar um objeto a partir de uma classe abstrata, mas pode-se fazer referência a uma classe abstrata por meio de um objeto de uma sub-classe.

Diagrama de Classe de Análise

Classe Abstrata

•Uma classe que provê organização.

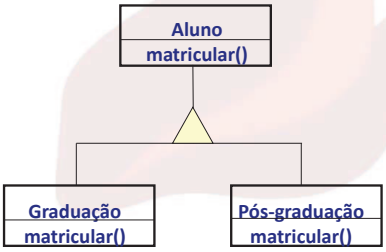


Diagrama de Classe Análise

CLASSE ABSTRATA

```
public abstract class ClPessoa {
    protected String nome;

public ClPessoa(String nome) {
    this.nome = nome;
}

public abstract void show();

}
```


Diagrama de Classe Análise

INTERFACE

- Uma interface é essencialmente uma coleção de atributos e métodos abstratos que podem ser implementados numa classe.
- Quando o que o programador deseja declarar um conjunto de métodos que serão implementados apropriadamente num conjunto de classes e que serão usados posteriormente, ele pode dispensar a definição de uma superclasse e obter o mesmo resultado usando **interfaces.**

Diagrama de Classe de Análise

HERANÇA MÚLTIPLA

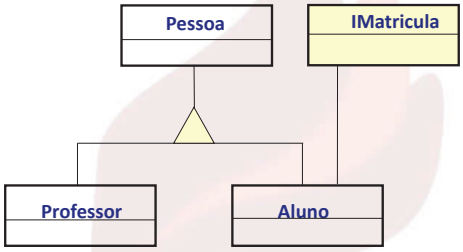


Diagrama de Classe Análise

INTERFACE

```
public interface IMatricula {

        void ImprimeMatricula();

}
```


Diagrama de Classe Análise

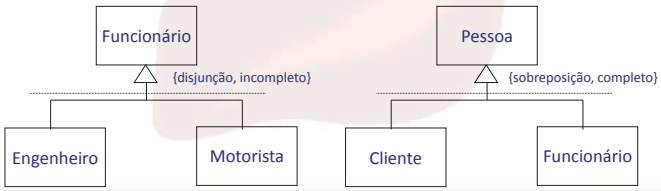
HERANÇA MÚLTIPLA

```
public class ClAluno extends ClPessoa implements IMatricula{
        private String matricula;
        private String email;
 public ClAluno() {
        super("");
        matricula = new String("");
        email = new String("");
 }
 public ClAluno(String nome, String matricula, String email){
        super(nome);
        this.matricula = matricula;
        this.email = email;
 }
 public void Imprime(){
    System.out.println("Aluno nome: " + super.nome + " matrícula: " + matricula + " E-mail: " + email);
  }

 public void ImprimeMatricula(){
    System.out.println(" Matrícula: " + matricula);
  }
}
```

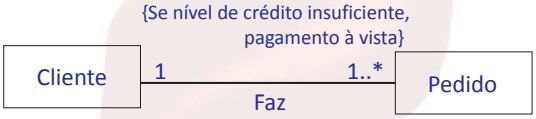
Associações - Generalização / Restrições

- **Sobreposição** - objetos podem ocorrer simultaneamente em mais de uma subclasse de uma mesma superclasse.
- **Disjunção** - objetos só podem pertencer a uma subclasse de uma mesma superclasse.
- **Completo** - todas as subclasses estão representadas.
- **Incompleto** - Nem todas as subclasses estão representadas.



Associações - Restrições

Especifica condições e proposições que devem ser satisfeitas nas associações. A restrição é colocada entre chaves, sobre a linha que representa a associação.



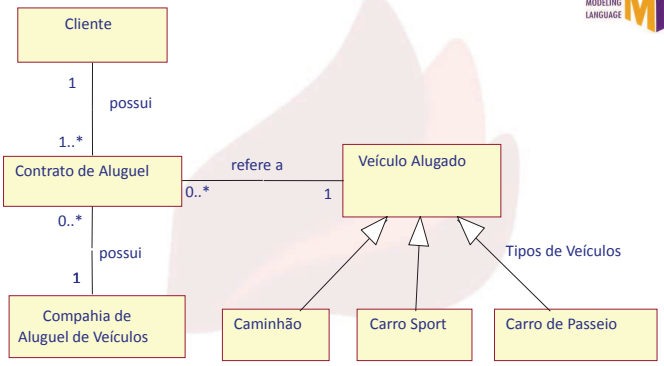


Diagrama de Objetos