



UNIVERSIDADE FEDERAL
DO ESPÍRITO SANTO

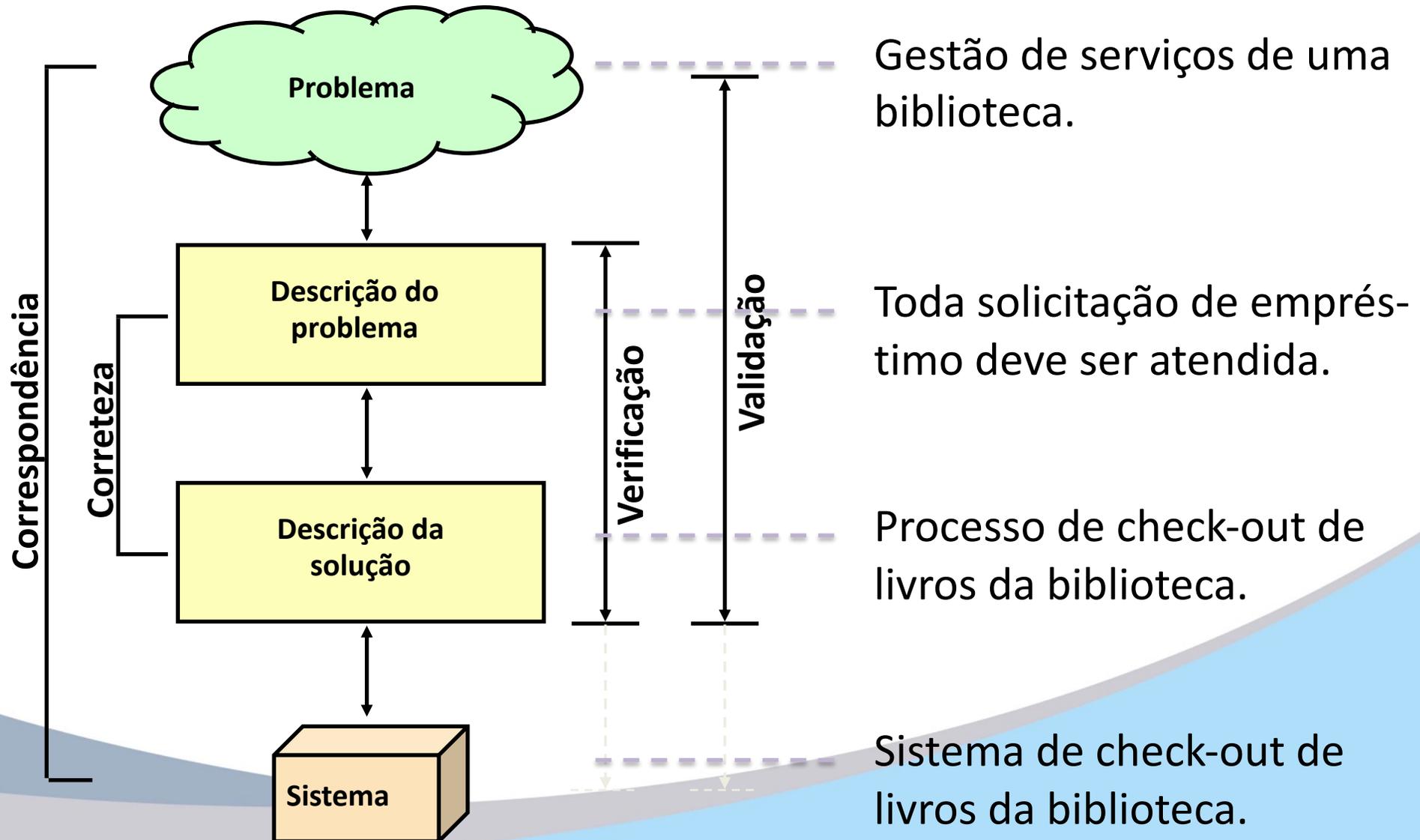
Centro Tecnológico
Departamento de Informática

Prof. Vítor E. Silva Souza

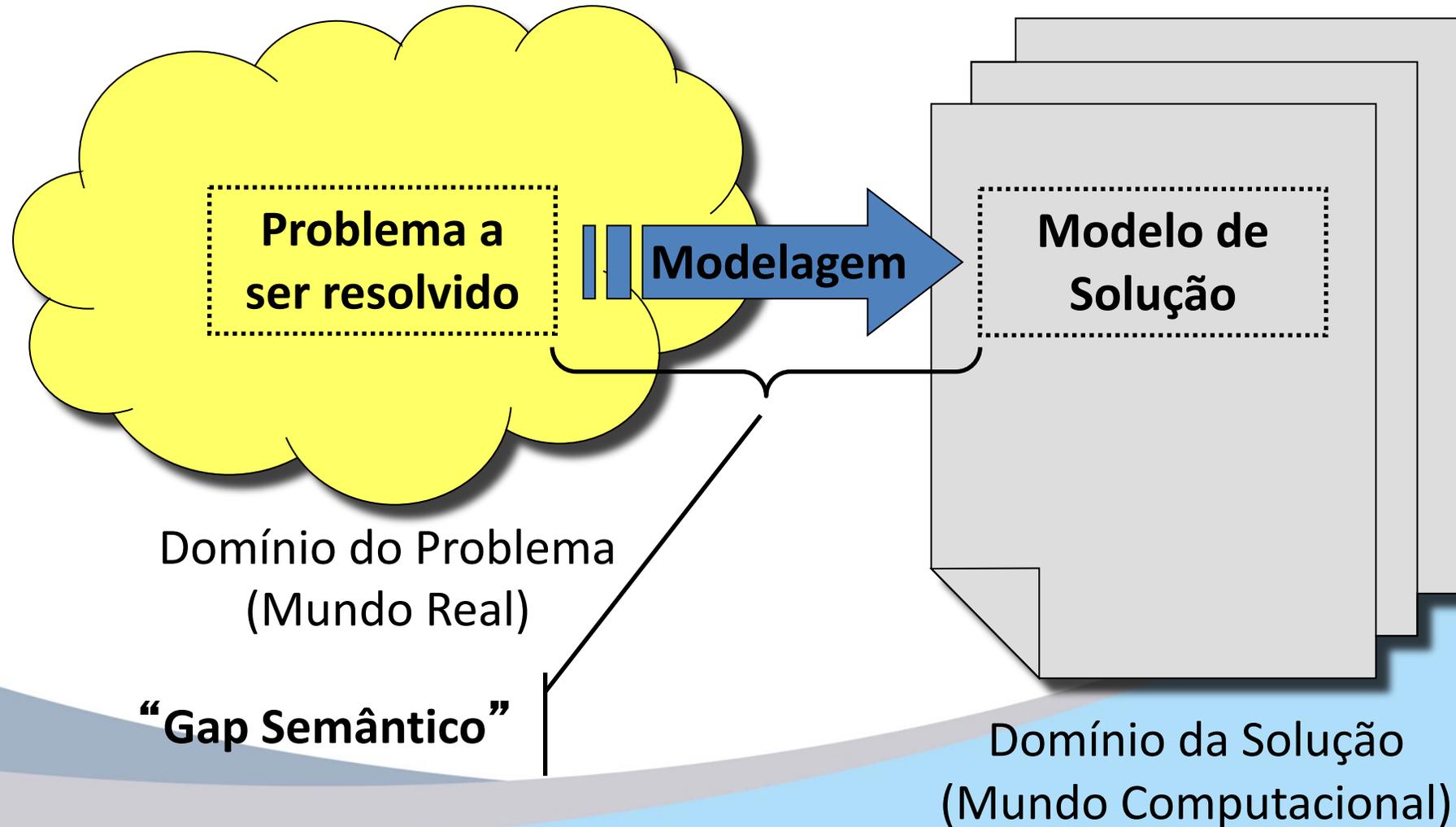
<http://www.inf.ufes.br/~vitorsouza>

Introdução à Programação Orientada a Objetos

Desenvolvimento de sistemas



Gap semântico



Paradigmas de programação

- Agrupamento de **métodos e técnicas** que seguem um mesmo conjunto de **princípios**;
- Estruturado:
 - *Modelo entrada – processamento – saída*;
 - *Dados separados das funções*;
 - *Visto na disciplina de Programação II / PBC.*
- Orientado a Objetos (OO):
 - *O mundo é composto por objetos*;
 - *Objetos combinam dados e funções*;
 - *Conceitos do problema são modelados como objetos que são associados e interagem entre si.*

Paradigmas de programação

Estruturado



efetuarEmprestimoLivro()

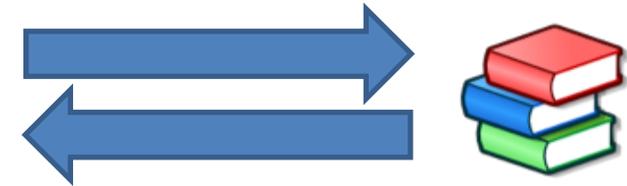


efetuarDevolucaoLivro()



efetuarReservaLivro()

Orientado a Objetos



emprestar()
devolver()
reservar()

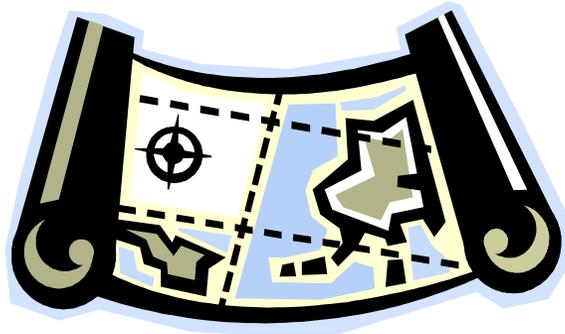
Menor gap semântico, mudanças em estrutura dos dados geram menos manutenção.

Vantagens da Orientação a Objetos

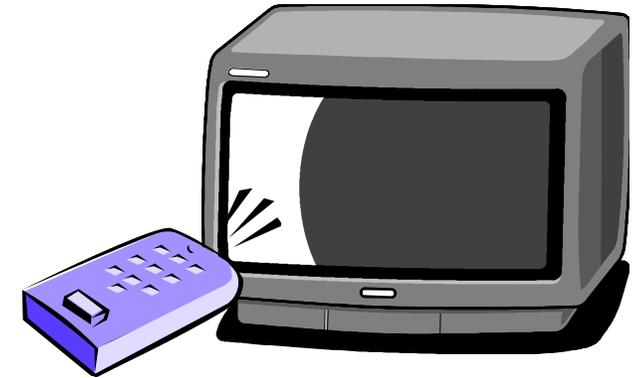
- Capacidade de enfrentar **novos domínios**;
- **Melhoria** da **interação** analistas x especialistas;
- Aumento da **consistência** interna da análise;
- **Alterabilidade, legibilidade e extensibilidade**;
- Apoio à **reutilização**.

Orientação a objetos não é mágica e nem a “tábua de salvação” do desenvolvimento. É preciso aplicá-la com disciplina e em conjunto com outras técnicas da Engenharia de Software.

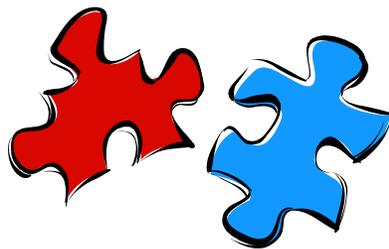
Princípios OO



Abstração



Encapsulamento



Modularização

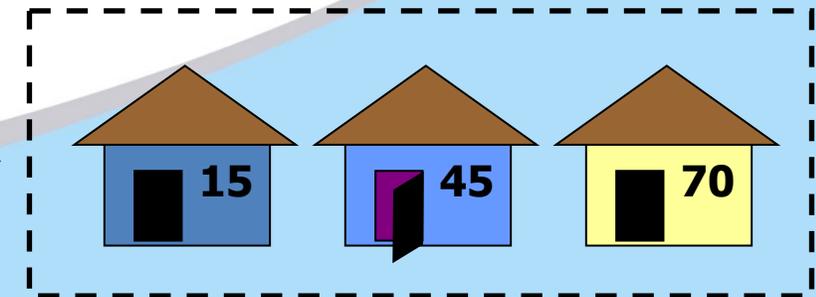
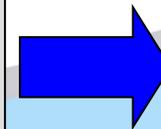


Herança

Classes e objetos (instâncias)

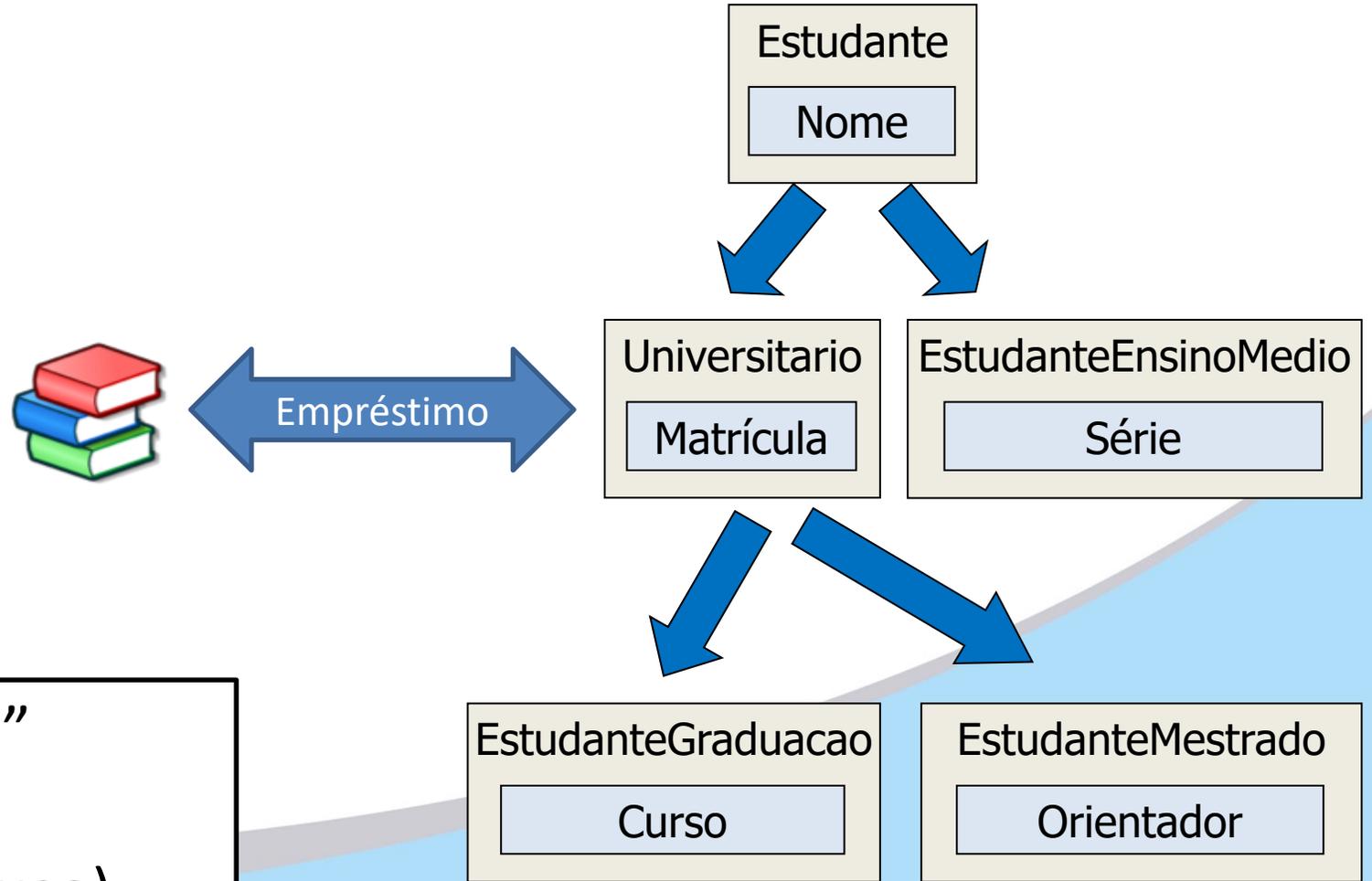
- “Um objeto é uma **entidade** que incorpora uma **abstração relevante** no **contexto** de uma aplicação”;
- Objetos possuem **estado**, **comportamento** e **identidade**;
- Uma classe descreve um conjunto de objetos com as mesmas **propriedades**, o mesmo **comportamento**, os mesmos **relacionamentos** com outros objetos e a mesma **semântica**;

Programação OO =
modelagem de classes



Mecanismos de estruturação

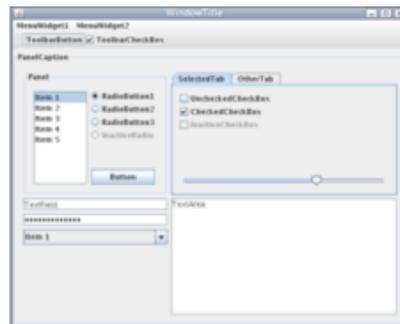
- Ligação / associação;
- Herança.



Associação = “tem um”
Herança = “é um”
(Ambos possibilitam o reuso)

Métodos

- Função membro de uma classe;
- Chamar um método = enviar uma mensagem;
- Toda **funcionalidade** do sistema é realizada pela **troca de mensagem** entre objetos.



emprestar()
devolver()
reservar()

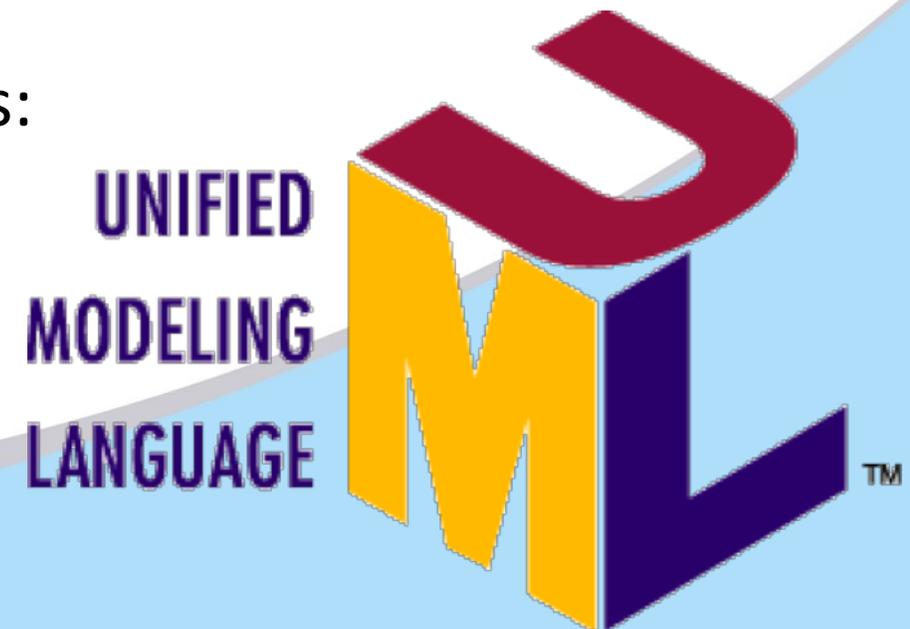
Conceitos avançados

- Classes e operações abstratas;
- Ligação dinâmica e polimorfismo;
- Sobrecarga e sobrescrita;
- ...

- Veremos estes conceitos na prática da programação OO.

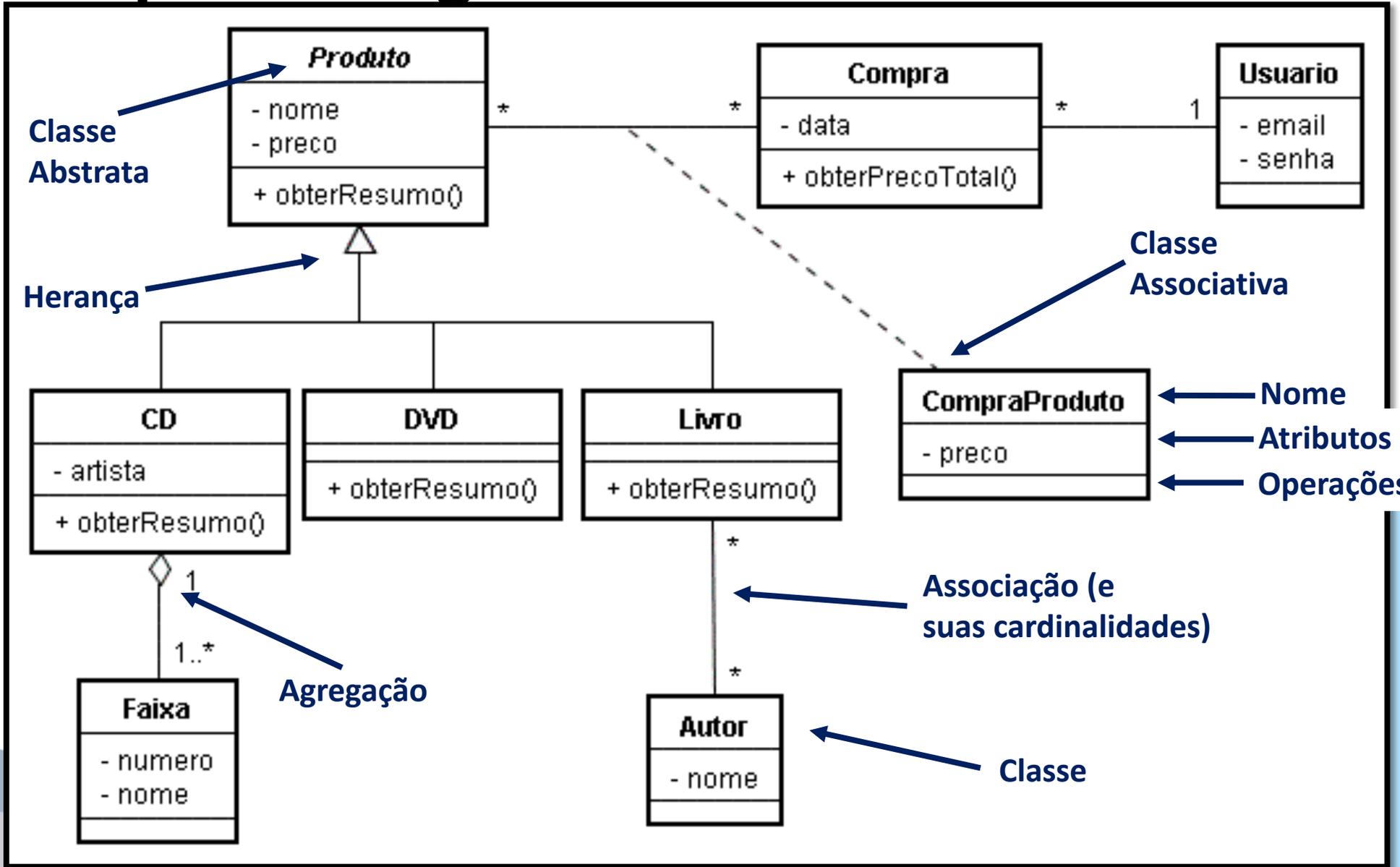
Unified Modeling Language

- Padrão “**de facto**” para especificar, visualizar, documentar e construir artefatos de um sistema desenvolvido sob o **paradigma Orientado a Objetos**;
- Nasceu na **Rational** Software, desde 1997 é um padrão da Object Management Group (**OMG**);
- Reconhecido pela **ISO** em 2000;
- Teve origem em três outros métodos:
 - *OMT (Rumbaugh et al., 1994);*
 - *Método de Booch (Booch, 1994);*
 - *Método OOSE (Jacobson, 1992).*



Exemplo: o Diagrama de Classes

Representa as classes relevantes (abstração!) para o domínio, problema ou solução.



Exercício: da programação estruturada à
programação orientada a objetos

Exercício

1) Escreva um programa em C que receba como argumento o nome de um arquivo e imprima seu conteúdo na tela. Caso não seja passado nenhum argumento, o programa deve pedir ao usuário que digite o nome do arquivo na entrada padrão.

2) Para armazenar os dados dos alunos (nome e nota), crie uma estrutura de dados (*struct*) `tAluno` e as seguintes operações:

- a) Dados um nome e uma nota, retorna um aluno;
- b) Obtém o nome de um aluno;
- c) Obtém a nota de um aluno.

A estrutura e as funções devem ser implementadas observando princípios de modularidade do código, ou seja, separadamente do arquivo de código do programa principal e divididos em arquivo cabeçalho (*header* ou `.h`) e código (`.c`).

3) Modifique o programa principal (desenvolvido no exercício 1) para que leia os dados do arquivo, monte um vetor de alunos para, em seguida, imprimir a frase "O aluno <nome> tirou a nota <nota>" para cada aluno no vetor, substituindo <nome> e <nota> pelos dados do aluno.

4) Modifique o programa principal para que calcule a média das notas dos alunos e imprima na tela apenas o nome dos alunos que possuem nota acima da média da turma.

5) Considerando que a média para passar direto é 7, faça com que o programa escreva num arquivo "saida.csv" o nome, a nota e a situação de cada aluno em formato CSV, como no exemplo abaixo:

```
Nome,Nota,Situação
Fulano,0.00,Prova Final
Ciclano,10.00,Aprovado
Beltrano,3.00,Prova Final
```