Diagrama de Classe

Monitor: Paulo Gurgel Pinheiro

MC436 – Engenharia de Software - LAB
Professora: Ariadne Carvalho
Instituto de Computação
Universidade Estadual de Campinas – UNICAMP

29 de Abril de 2010

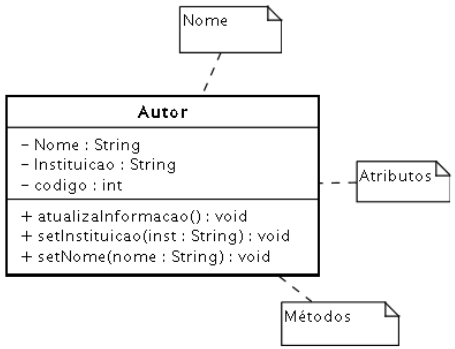
Introdução

Esta é uma apresentação sucinta sobre diagramas de classe. Seu proposito é servir como um rápido guia de consulta para os alunos da disciplina MC436, 1s2010.

Diagrama de Classe

Um diagrama de classe contém a representação da estrutura das classes do sistema e as relações entre elas.

"UMLmente"falando, do que estamos tratando mesmo?



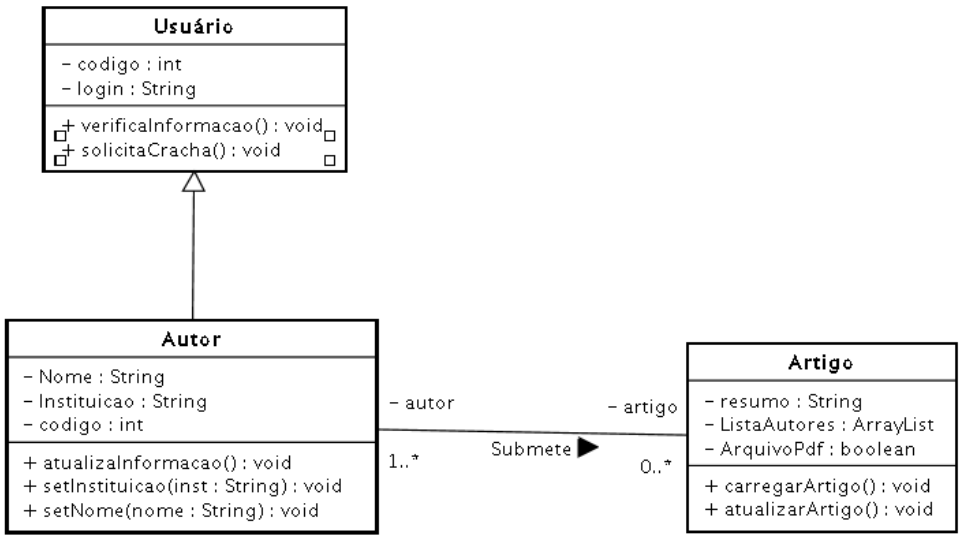


Diagrama de Classes

Existem **dois** grandes conceitos básicos: Classes e Relacionamentos.

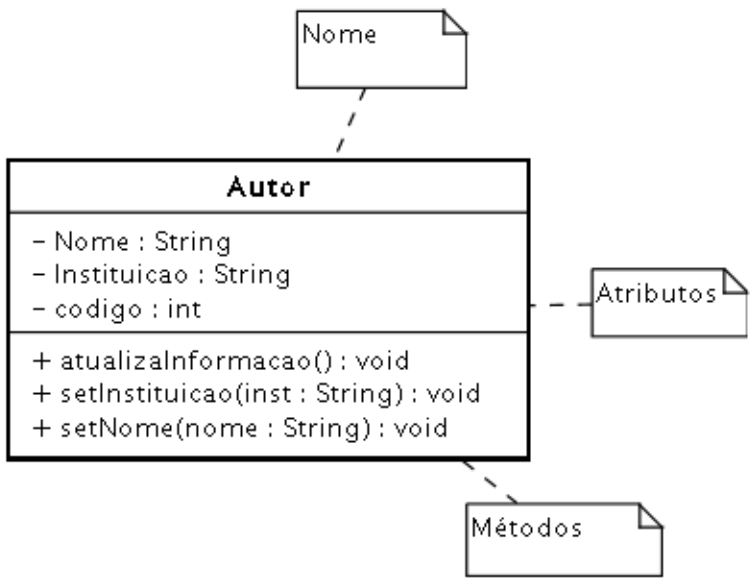
- **Classes**
  - Atributo:
    - Visibilidade: Pública, Privada, Protegida.
    - Nome
    - Tipo do dado
    - Valor inicial: Pode ser opcional
  - Operação:
    - Visibilidade: Pública, Privada, Protegida.
    - Nome
    - Parâmetros
- **Relacionamentos**
  - Associação
  - Dependência
  - Herança (Especialização - Generalização)
  - Agregação
  - Composição

Classe

CLASSE

Classe

- Toda classe deve ter um nome.
- Classes contêm atributos, operações e suas respectivas visibilidades.



Relacionamentos

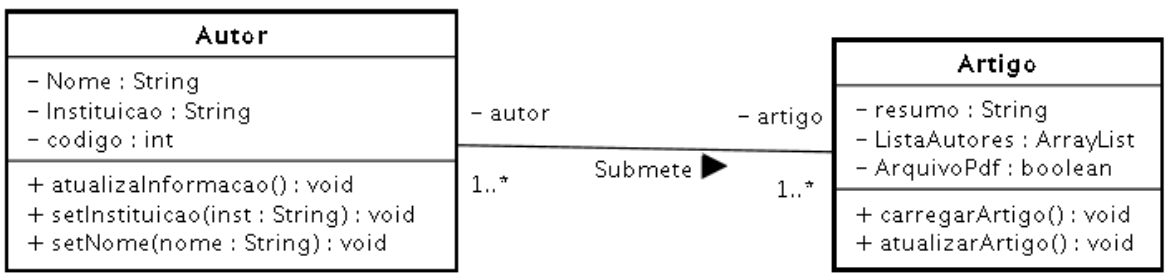
RELACIONAMENTOS

Relacionamentos

ASSOCIAÇÃO

Associação

- Relacionamento estrutural entre instâncias.
- Indicam que uma classe está relacionada à outra classe.
- Uma associação entre Artigo e Autor (um autor submete um artigo) é um exemplo.



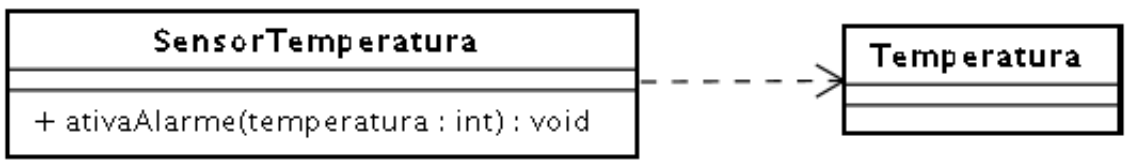
- Pode apresentar *multiplicidade*: Quantidade de elementos do conjunto = 1..*, *, etc.
- Pode ter um nome com uma direção.
- Classes podem ter papéis.
- Representada por um traço simples entre as classes.

Relacionamentos

DEPENDÊNCIA

Dependência

- Sempre que uma mudança em uma classe interferir em outra classe, mas não necessariamente o contrário.
- Sempre que um objeto de uma classe utilizar serviços do objeto de outra classe.
- Representada com uma linha pontilhada com seta aberta.

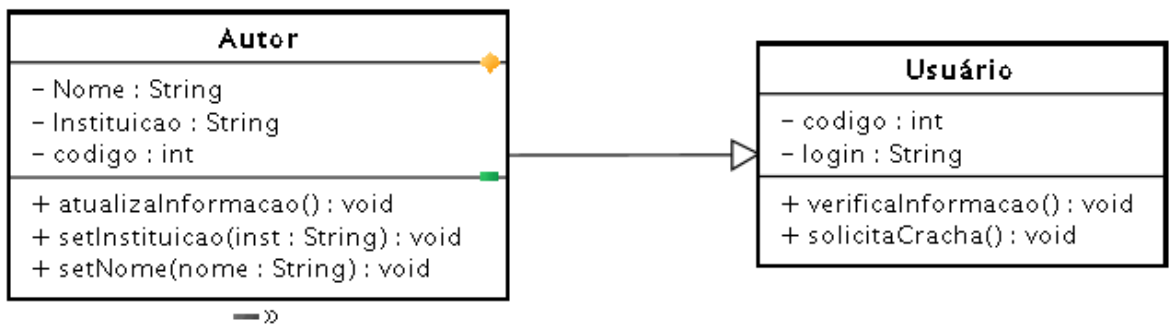


Relacionamentos

HERANÇA

Herança

- Em UML é indicada como *Generalização*.
- Define um relacionamento em que uma super classe é a generalização de uma ou mais sub classes.
- Uma sub classe herda os atributos e operações da super classe.
- Representação UML de herança.



Herança Múltipla

As vezes é necessário utilizar herança múltipla, por isso cuidado:

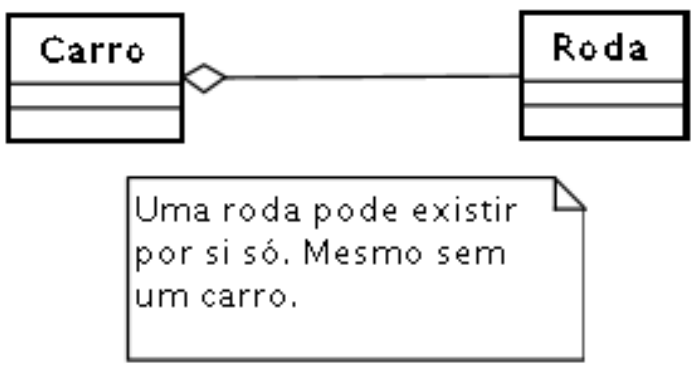
- Para não haver conflitos entre os atributos/métodos da subclasse e da superclasse.
- Verifique se é realmente necessário. Se for, use. Se não for, remodele. Não precisamos deixar mais o complexo o que não precisa ser.

Relacionamentos

AGREGAÇÃO

Agregação

- Informações de um objeto precisam ser complementadas por outro objeto.
- Tem-se um *objeto todo* e um *objeto parte*.
- Associação chamada de *"parte de"*.
- A parte pode existir sem o todo.
- A *classe todo* pode ser excluída independente da *classe parte*.
- Representada por uma linha com diamante vazio ao lado do *todo*.

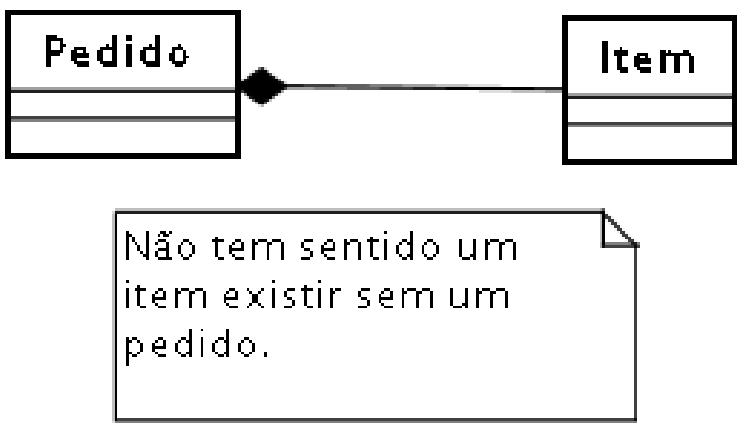


Relacionamentos

COMPOSIÇÃO

Composição

- Indica também uma associação *todo-parte* assim como a agregação
- Porém, a *parte* não pode existir sem o todo.
- Representada por uma linha com diamante cheio ao lado do *todo*.



Passos

Quais passos devo seguir para fazer um diagrama de classes?

Passos

1. **Classes:** Levante possíveis classes do seu sistema utilizando a descrição dos casos de uso que você fez.
2. **Classes:** Refine a lista de classes removendo as redundantes e irrelevantes.

PRONTO! AGORA VOCÊ TEM SUAS CLASSES

3. **Relacionamentos:** Defina os relacionamentos entre as classes.
4. **Multiplicidade e nome:** Defina a multiplicidade e nome das relações quando for conveniente.
5. **Atributos:** Identifique os atributos das classes.
6. **Métodos:** Identifique os Métodos das classes.

pinheiro@ic.unicamp.br