

Programação Orientada à Objetos

Aula 09

Relacionamentos entre Objetos

Arrays

Henrique Poyatos

henrique.poyatos@bandtec.com.br

Mensagem

Interatividade entre Objetos



- É uma forma de relacionamento entre objetos;
- É através delas que os objetos interagem (se comunicam) em um programa.
- Tratam-se de pedidos enviados a um objeto, a fim de que este modifique seu estado ou retorne algum valor.
- Em um programa orientado a objeto, nenhuma outra parte do programa pode acessar diretamente os dados de um objeto. Toda a comunicação entre os objetos ocorre única e exclusivamente através de mensagens.

Mensagem

Interatividade entre Objetos



- Uma mensagem é uma solicitação feita por um objeto A a um objeto B;
- Como resultado desta solicitação, o objeto B irá modificar seu estado ou irá retornar algum valor;
- O conceito de mensagem está diretamente associado ao conceito de operação;
- A interação entre os objetos é feita através da troca de mensagens;
- São representados na UML através dos diagramas de iteração e de colaboração.

- Relações fornecem um caminho para a comunicação entre os objetos
- Diagramas de sequência e/ou colaboração são examinados para determinar quais ligações entre objetos precisam existir para se chegar ao comportamento desejado. -- Caso dois objetos precisem “conversar” deverá haver uma **ligação entre elas**

Tipos de Relacionamentos



.Tipos de Relacionamentos

-Composição

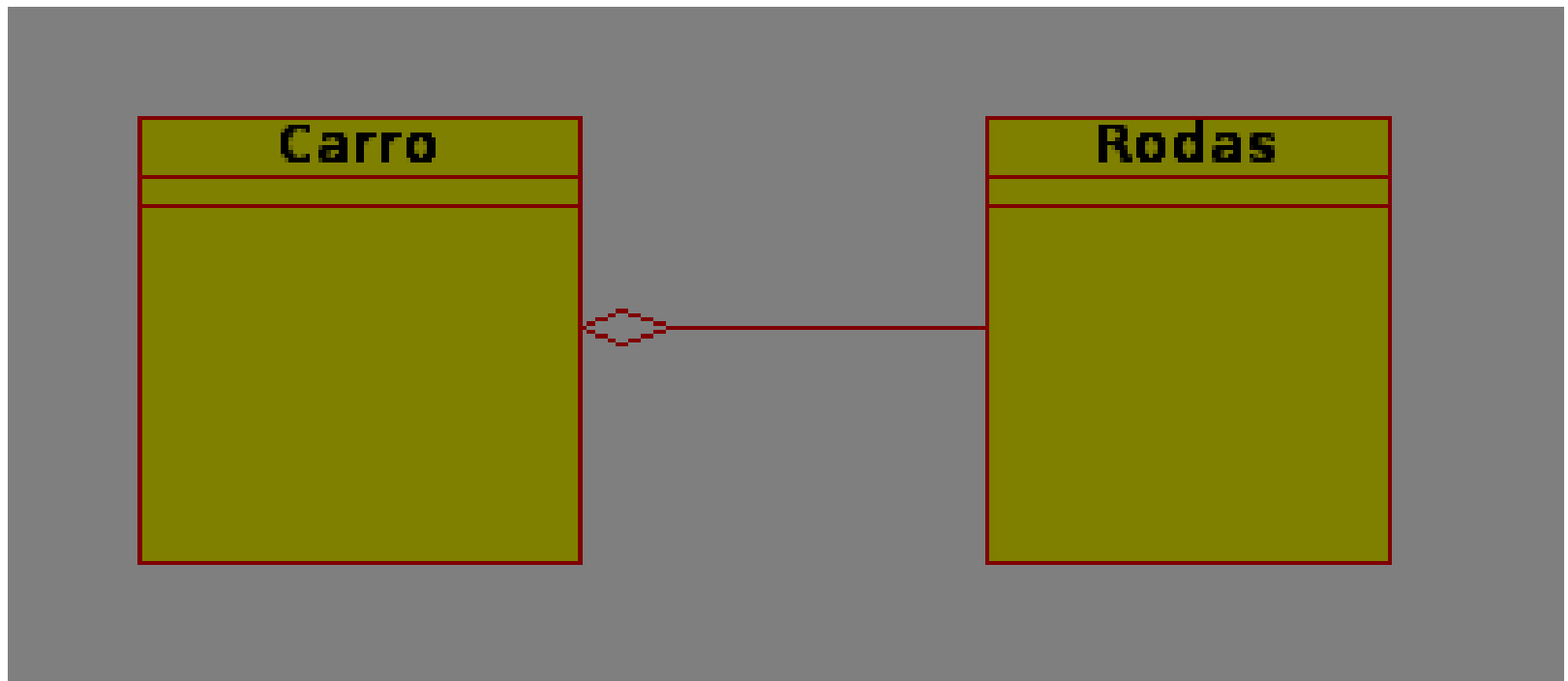
- Se a parte deixa de existir sem o todo

—Agregação

- Se a parte existir sem o todo

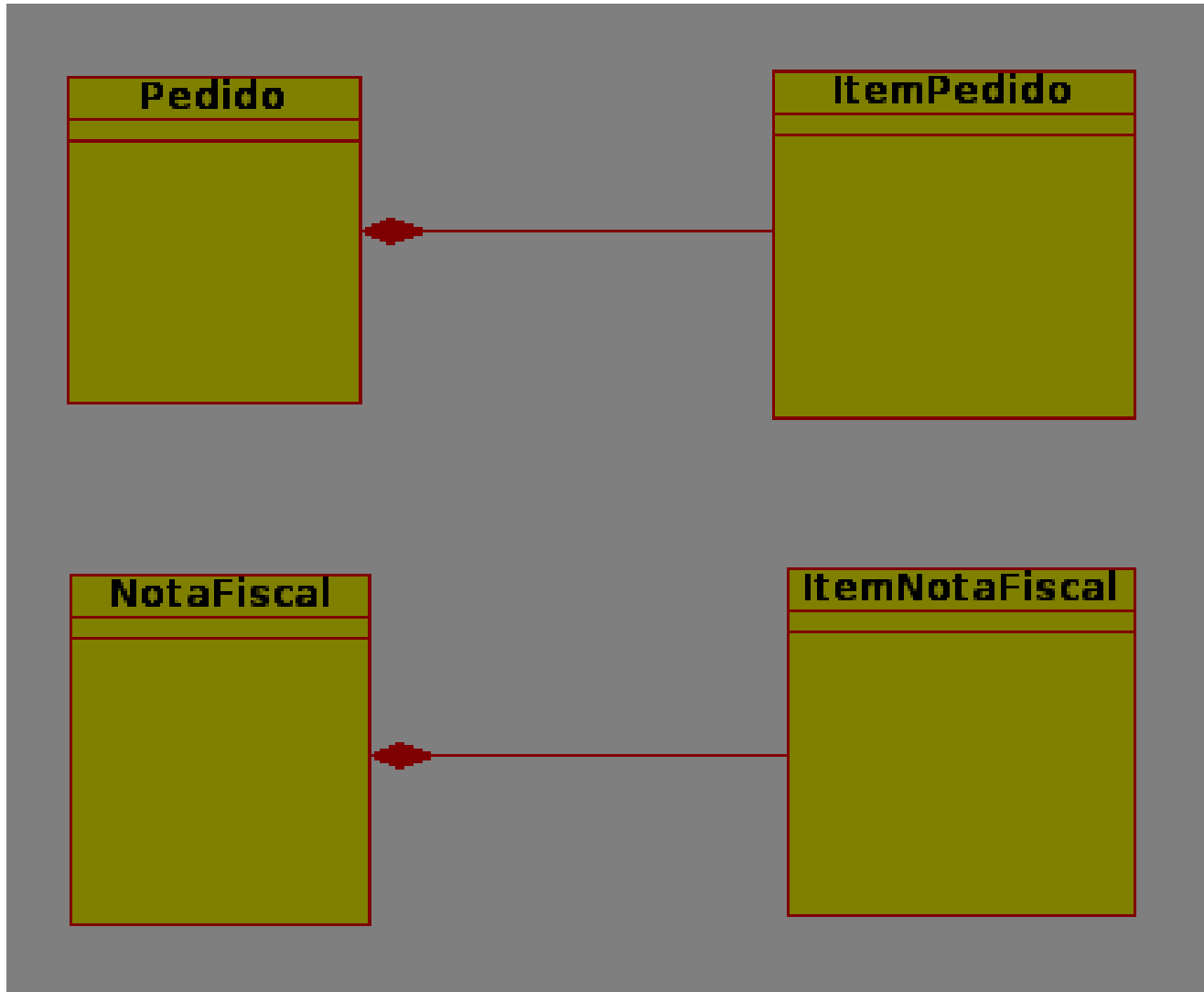
Aggregação

Exemplo

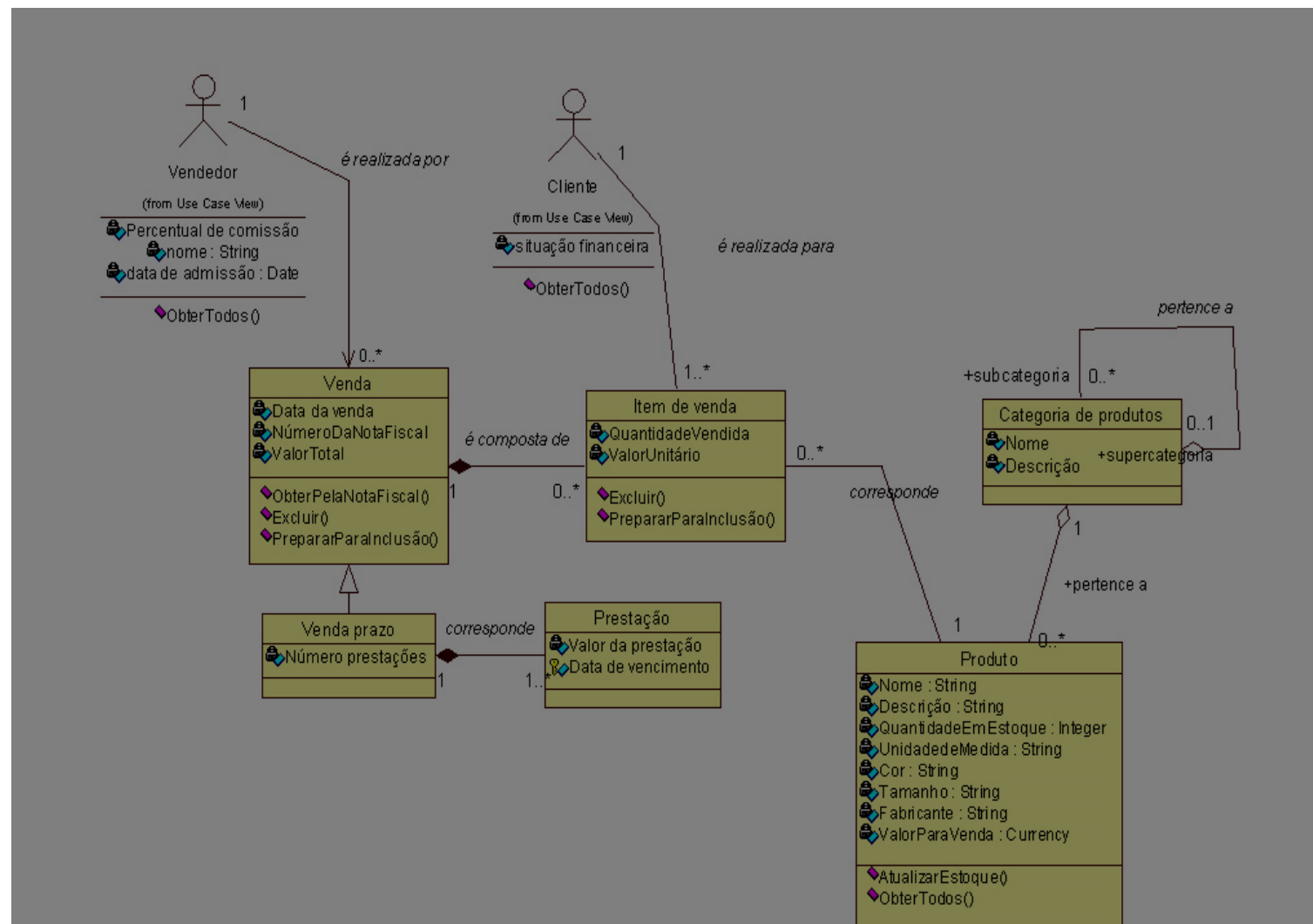


Composição

Exemplo



Relacionamentos



Tipos de Relacionamentos



.Associação:

- é uma conexão bi-direcional entre classes
- É mostrada como uma linha conectando as classes relacionadas

.Agregação:

- é um tipo mais forte de conexão, especialização de uma associação, aonde a relação é entre o todo e suas partes (relacionamento “parte-de” ou “contenção”)
- É mostrada como uma linha conectando as classes relacionadas com um losango perto da classes que representa o todo

Relacionamento entre Classes



.Herança:

-um dos princípios da OO, permite a reutilização, uma nova classe pode ser definida a partir de outra já existente

.Dependência:

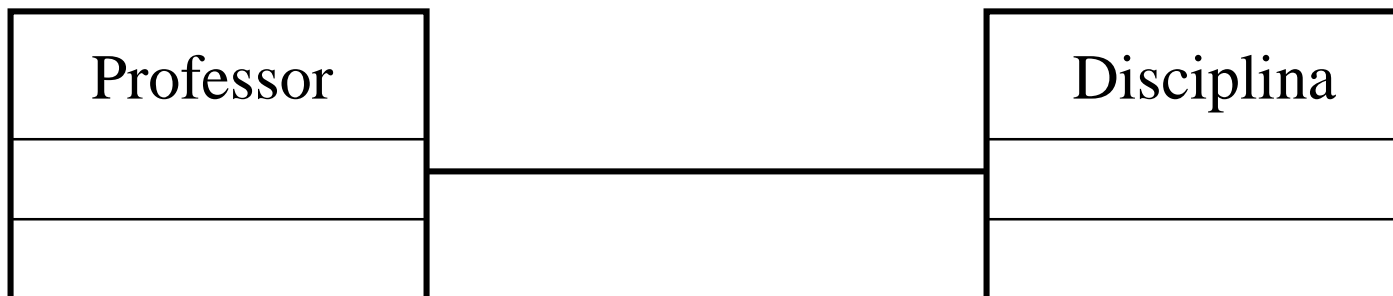
-é uma forma mais fraca de relação, mostrando uma relação entre um cliente e um fornecedor, aonde o cliente não tem conhecimento semântico do fornecedor

-é mostrada como uma linha pontilhada entre o cliente e o fornecedor

Associação



- Uma associação é uma conexão entre classes;
- Associações são representadas em um diagrama de classe através de uma linha conectando as classes associadas;
- Os dados podem fluir em uma ou em ambas as direções através do link.



Associação



- Para exemplificar será criada uma classe Disciplina que terá como função armazenar todas as disciplinas ministradas pelo professor ou cursadas pelo aluno.

```
/**
 * Implementação da classe Disciplina
 */
public class Disciplina {
    private String nomeDis;
    private int cargaHoraria;

    public Disciplina(String pnomeDis, int pcargaHoraria) {
        nomeDis = pnomeDis;
        cargaHoraria = pcargaHoraria;
    }

    public String getDescricao() {
        return " Disciplina: "+nomeDis+" Carga Horária: "+cargaHoraria;
    }
}
```

Classe Disciplina

Atributos

Construtor

Método de
recuperação de
informação
get...()

Associação



Por questões de simplificação o atributo aDisc faz parte da classe Pessoa. Isso é válido porque todas as subclasses do modelo terão referência a disciplinas ministradas (professor) ou cursadas (aluno). Entretanto, caso fosse implementado mais um nível na hierarquia, por exemplo: criar uma classe funcionário e as especializações professor e diretor, eventualmente deveríamos rever a hierarquia, pois somente professor e aluno teriam disciplinas.

```
/**
 * Implementação de uma classe abstrata
 */
public abstract class Pessoa {
    private String aNome = "", aEndereco = "", aSexo = "";
    private int aIdade = 0;
    private Disciplina aDisc[]; //Atributo da classe Disciplina

    //Conjunto de métodos já definidos

    public String getDisciplina(int pIndice) {
        return aDisc[pIndice].toString();
    }

    public String getDisciplinas() {
        String retorno = "";
        for (int i = 0; i < aDisc.length; i++) {
            retorno += aDisc[i].getDescricao()+"\n";
        }
        return retorno;
    }
}
```

Atributo da classe Disciplina

Método que retorna uma única disciplina utilizando um índice de acesso a **array** aDisc[];

Método que retorna todas as disciplinas constantes no **array** aDisc[];

Relacionamentos

Exemplo

