



# UML – Classes e Relacionamentos

Prof. Dr. Wilson M. Yonezawa  
Departamento de Computação  
UNESP – FC - Bauru

# De novo o “Hello World”

```
import java.awt.Graphics;  
  
class HelloWorld extends java.applet.Applet {  
  
    public void paint (Graphics g) {  
  
        g.drawString("Hello, World!", 10, 10);  
  
    }  
}
```

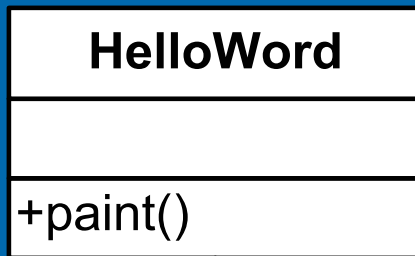
**Como utilizar UML na especificação?!?!?!?**

# Usando UML na especificação

- Qual o nome da classe?
- Quais as classes envolvidas?
- Quais os relacionamentos envolvidos?
- Qual o grau de abstração adequado?

# Primeira abstração do problema

## Classe



## Nota



`g.drawString ("Hello, World, 10, 10")`

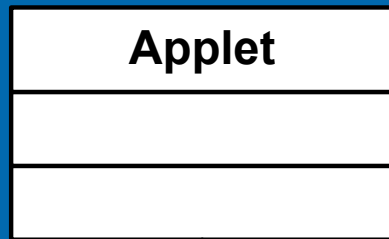
A white note box with a folded top-right corner, containing the code snippet `g.drawString ("Hello, World, 10, 10")`. A dashed line connects the bottom of the `+paint()` method in the class diagram to the left side of the note box.

# Um pouco mais de detalhe

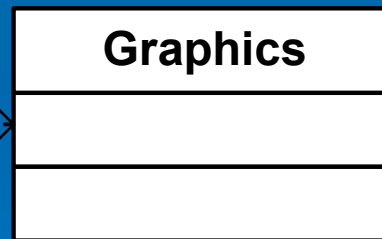
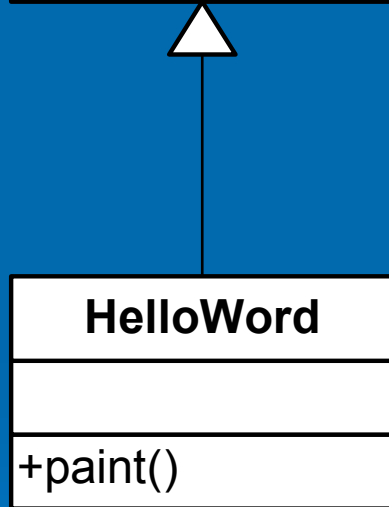
```
import java.awt.Graphics;  
class HelloWorld extends java.applet.Applet {  
    public void paint (Graphics g) {  
        g.drawString("Hello, World!", 10, 10);  
    }  
}
```

- **Classe: HelloWorld**
- **Classe pai: Applet**
- **Classe associada: Graphics**

# Melhorando a especificação

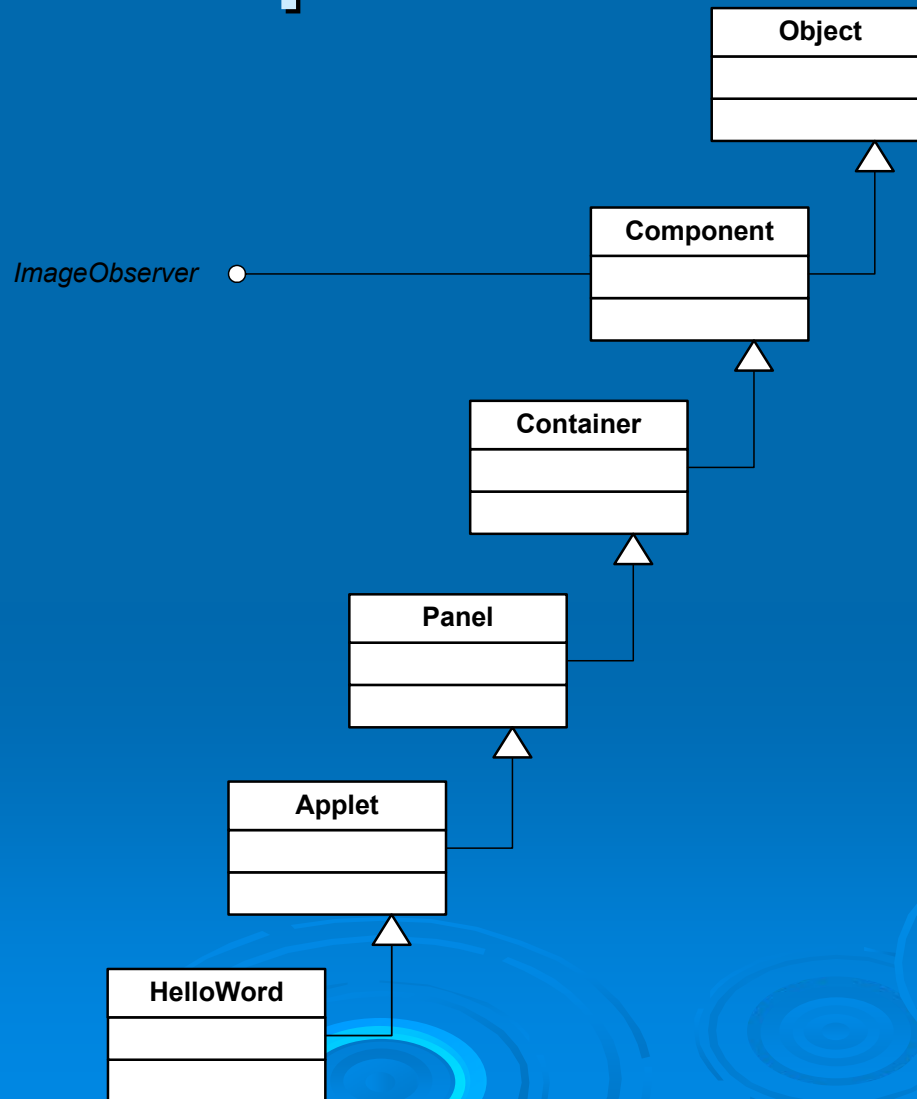


```
import java.awt.Graphics;
class HelloWorld extends java.applet.Applet {
    public void paint (Graphics g) {
        g.drawString("Hello, World!", 10, 10);
    }
}
```



- **HelloWord é filha (deriva) de Applet**
- **HelloWord usa Graphics**

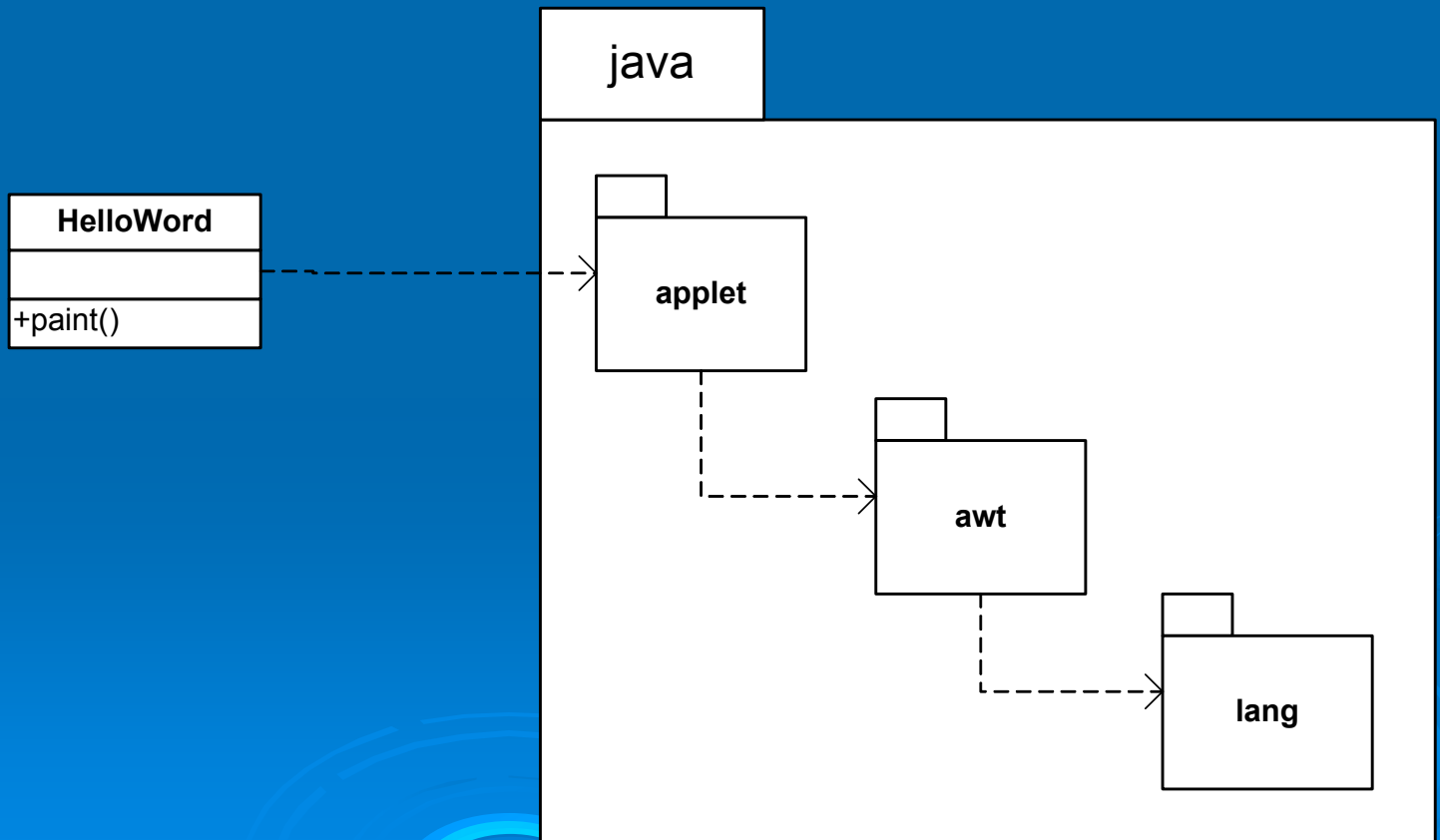
# Visualizando Herança na Hierarquia de classes



# Descrivendo pacotes

HelloWord depende de java.applet

java.applet depende de java.awt





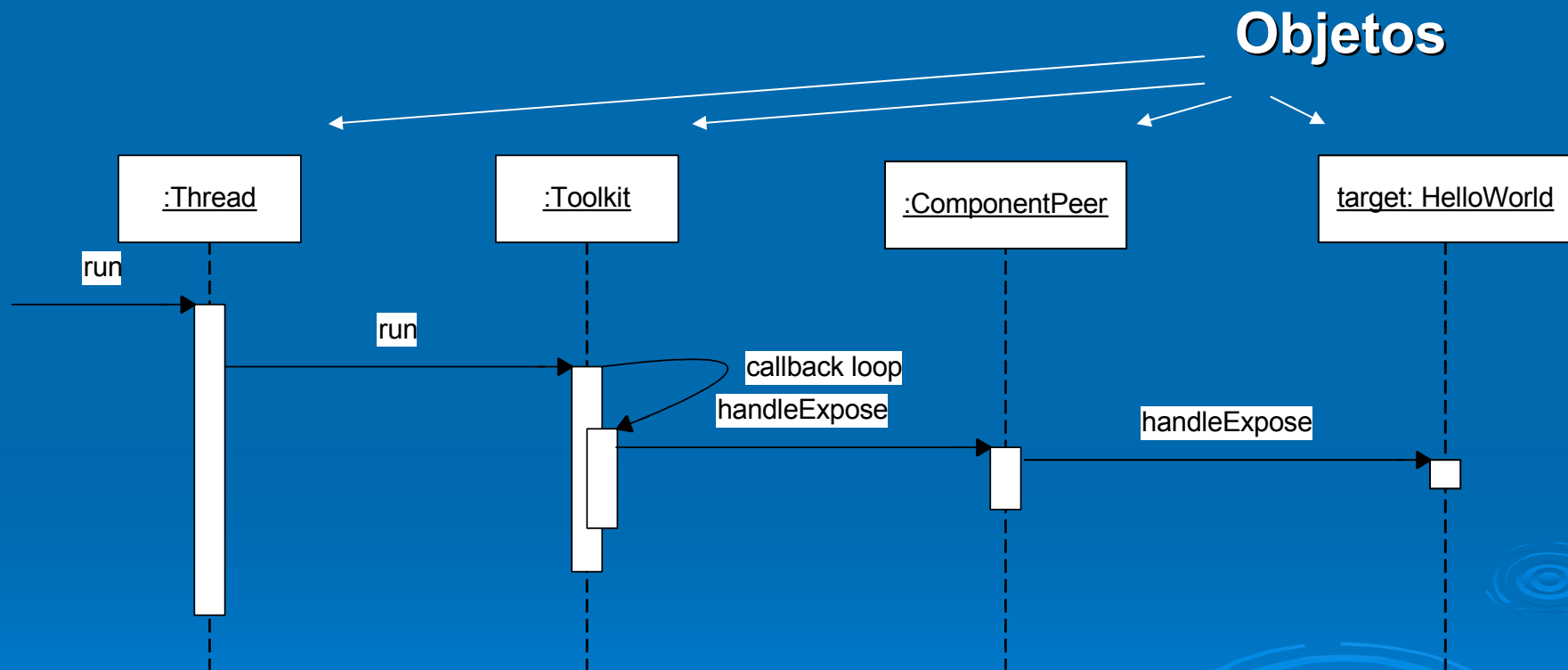
# E o método paint ???

- Como chamar o método paint() da classe HelloWorld ????
- O método paint() foi reescrito em HelloWorld, mas existe também na classe Component
- Mas como ele é executado??? Quando é chamado??

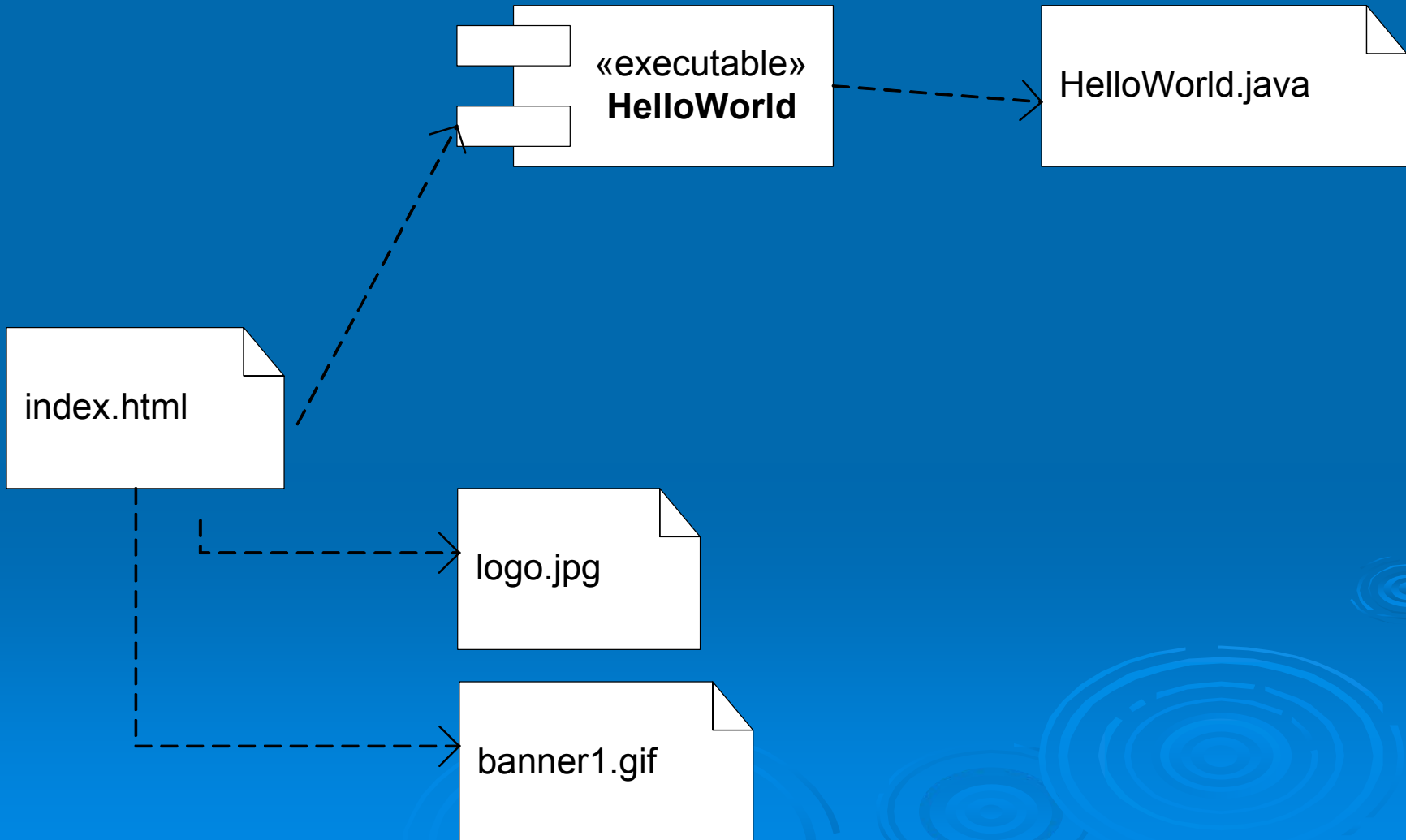
# Ainda sobre o paint

- **Paint é chamado como parte da execução da thread do applet**
- **O método paint está escrito no pai e no filho. O conceito de polimorfismo é utilizado neste caso**
- **Um diagrama de seqüência pode ser utilizado para demonstrar coisas deste tipo**

# Exemplo de diagrama de seqüência



# Componentes do HelloWorld



# Sobre “Classes”

- Classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica
- Termos e conceitos:
  - Nomes da classe
  - Atributos
  - Operações
  - Responsabilidade

# Sobre “Classes”

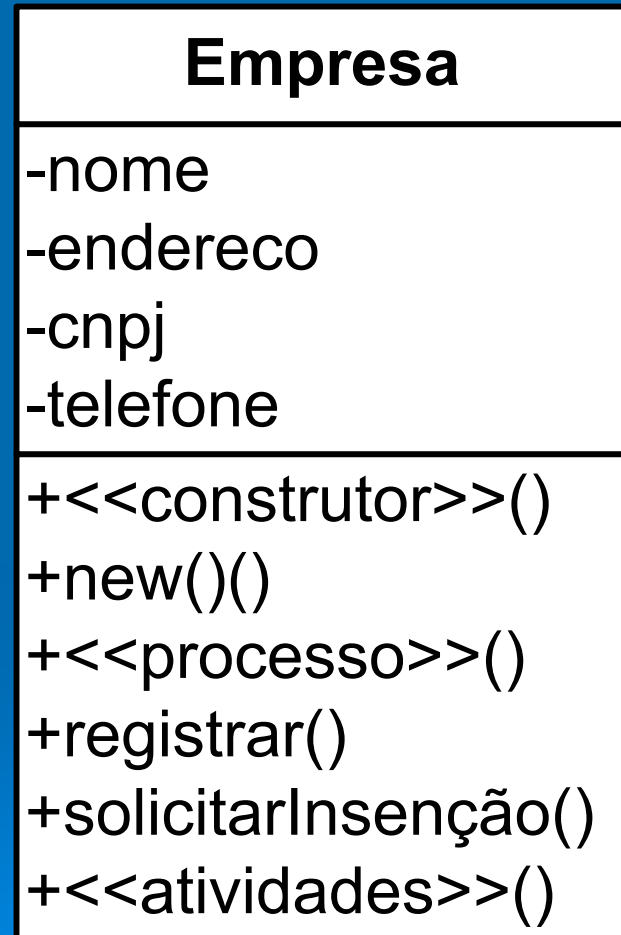
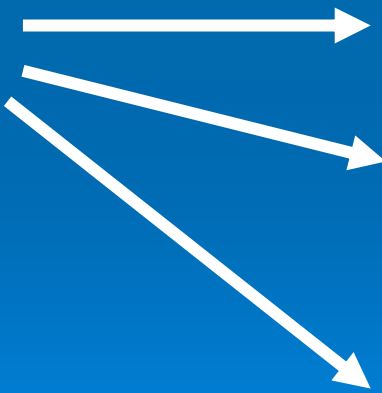
- Classes são descritas por substantivos (ex: Cliente, SensorTemperatura, Motor, Computador, Carro)
- Atributos são propriedades nomeadas de uma classe (ex: nome, endereço, dataNascimento)
- Operações são implementações de serviços que pode ser solicitado por algum objeto da classe para modificar o comportamento.

# Sobre “Classes”

- O nome da operação é um verbo que representa algum comportamento da classe correspondente (ex: move, isEmpty)
- É possível especificar uma operação indicando sua **assinatura** com o nome, o tipo e valor padrão dos parâmetros, assim com o tipo a ser retornado
  - Ex: configurarSensor(limiteMinimo: int): boolean

# Sobre “Classes”

**Estereótipos**





# Sobre “Classes”

Aluno
-Id : int -nome -endereco -cidade -sexo
+EfetuarMatricula() : bool +CancelarDisciplina() +CancelarMatricula()

**Responsabilidade: É um contrato ou obrigação da classe.**

**Exemplo:**  
-----

- **Realizar a matricula todos semestre**
- **Cancelar disciplina sempre que desistir da mesma**

# Sobre “Classes” e responsabilidades

- Modelar classes é especificar responsabilidades dos itens
- Uma classe pode ter qualquer número de responsabilidades (pelo menos uma responsabilidade é necessária)
- As responsabilidades de uma classe podem ser traduzidas num conjunto de atributos e operações ao longo do processo de modelagem.

# Ainda sobre “Classes”

- Algumas vezes é preciso visualizar ou especificar características como visibilidade de atributos e operações (ex: polimorfismo)
- As vezes é preciso especificar Exceções que objetos produzem ou manipulam
- Modelagem de classes avançadas são utilizadas para isso

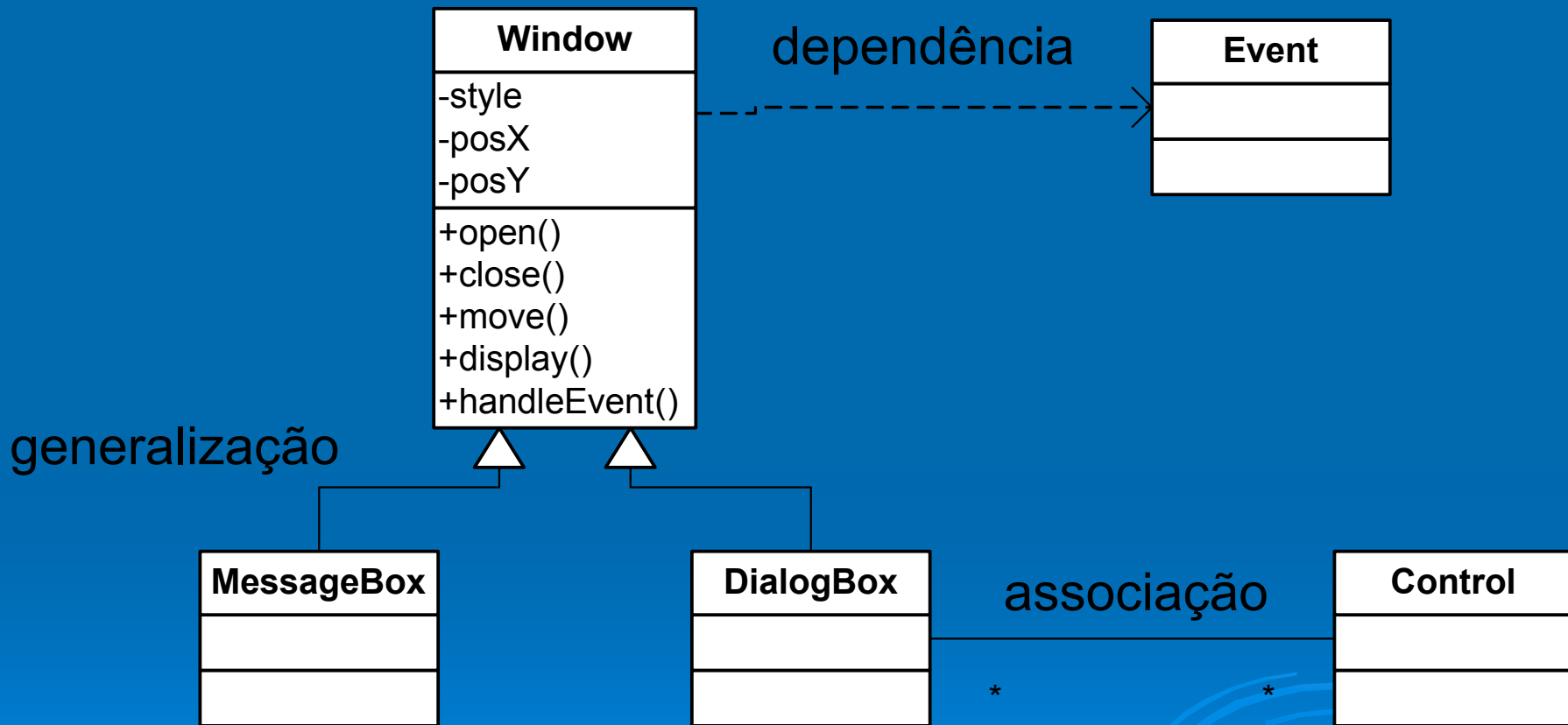
# Como identificar “Classes”?

- Levantar o vocabulário do sistema ou do problema
- Identifique as principais abstrações no problema
- Identifique as responsabilidades do sistema

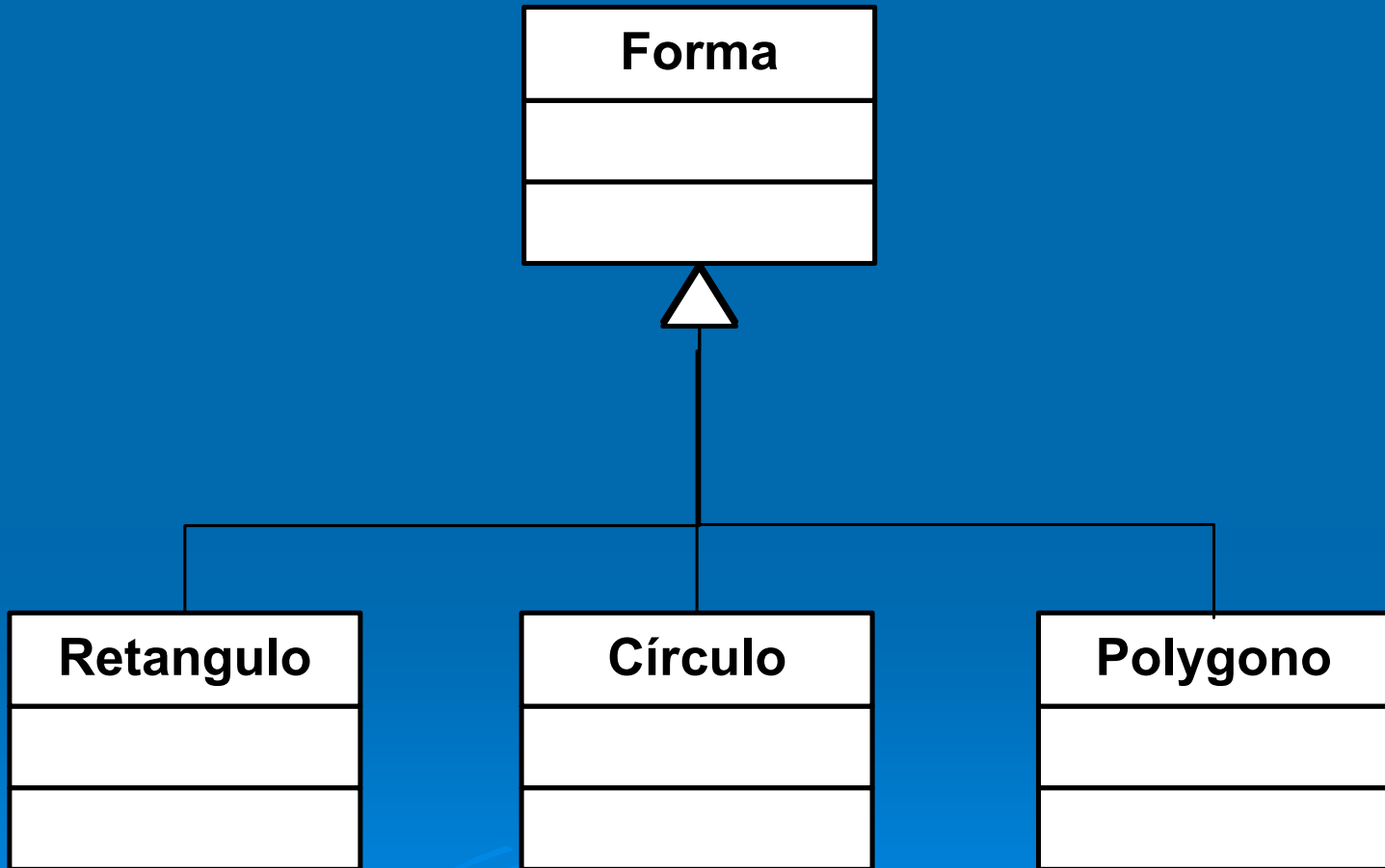
# Sobre os “Relacionamentos”

- Classes não trabalham sozinhas
- Classes **colaboram** umas com as outras
- Três tipos básicos de relacionamento:
  - Dependências
  - Generalizações
  - Associações
- Ex:
  - Construir uma casa, carro ou computador
  - Quais as dependências, generalizações e associações existentes nesta tarefa?

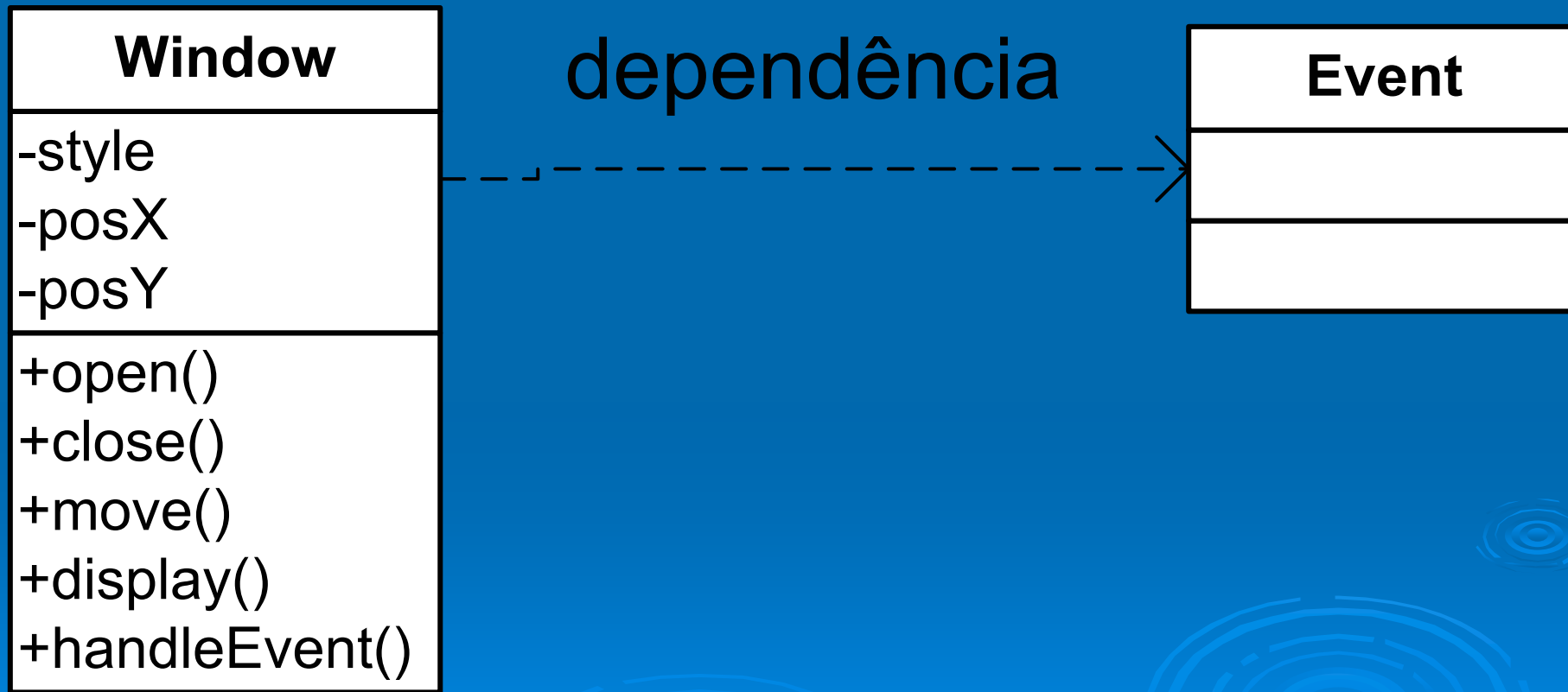
# Sobre os “Relacionamentos”



# Sobre os “Relacionamentos” – Generalização

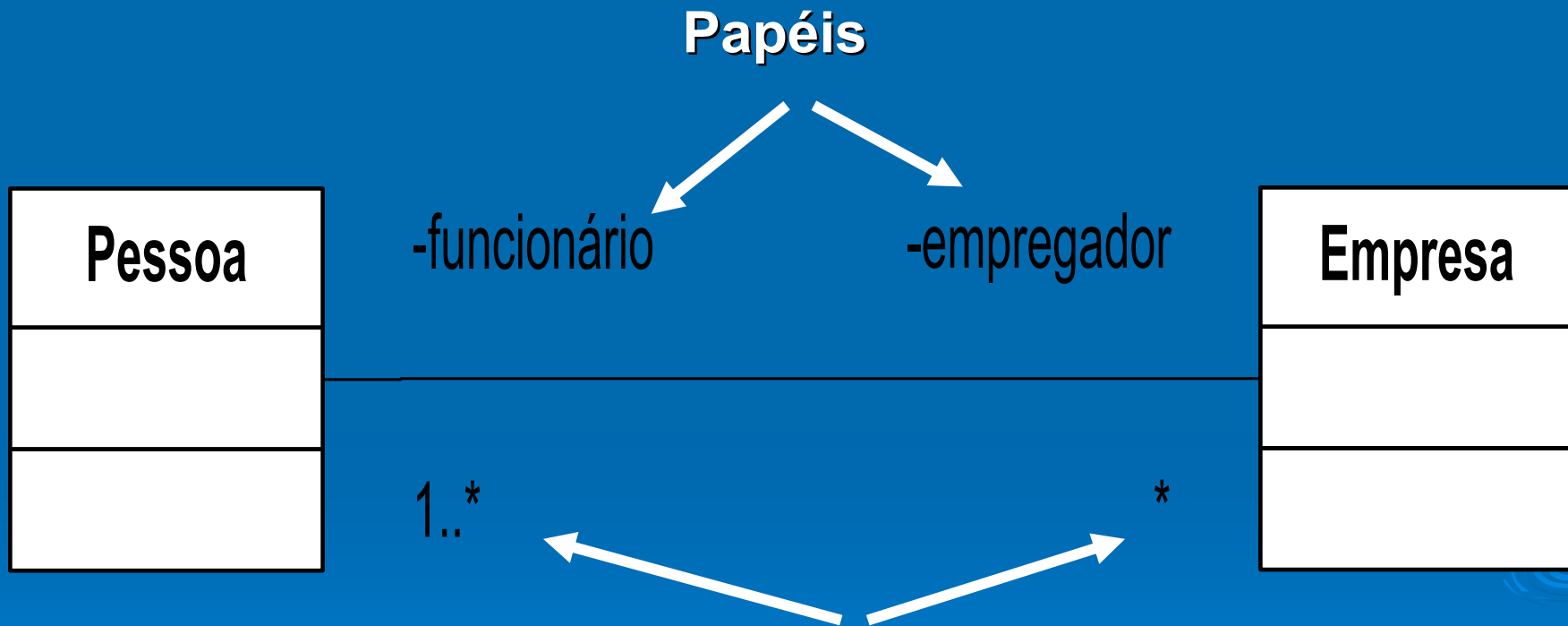


# Sobre os “Relacionamentos” – Dependência





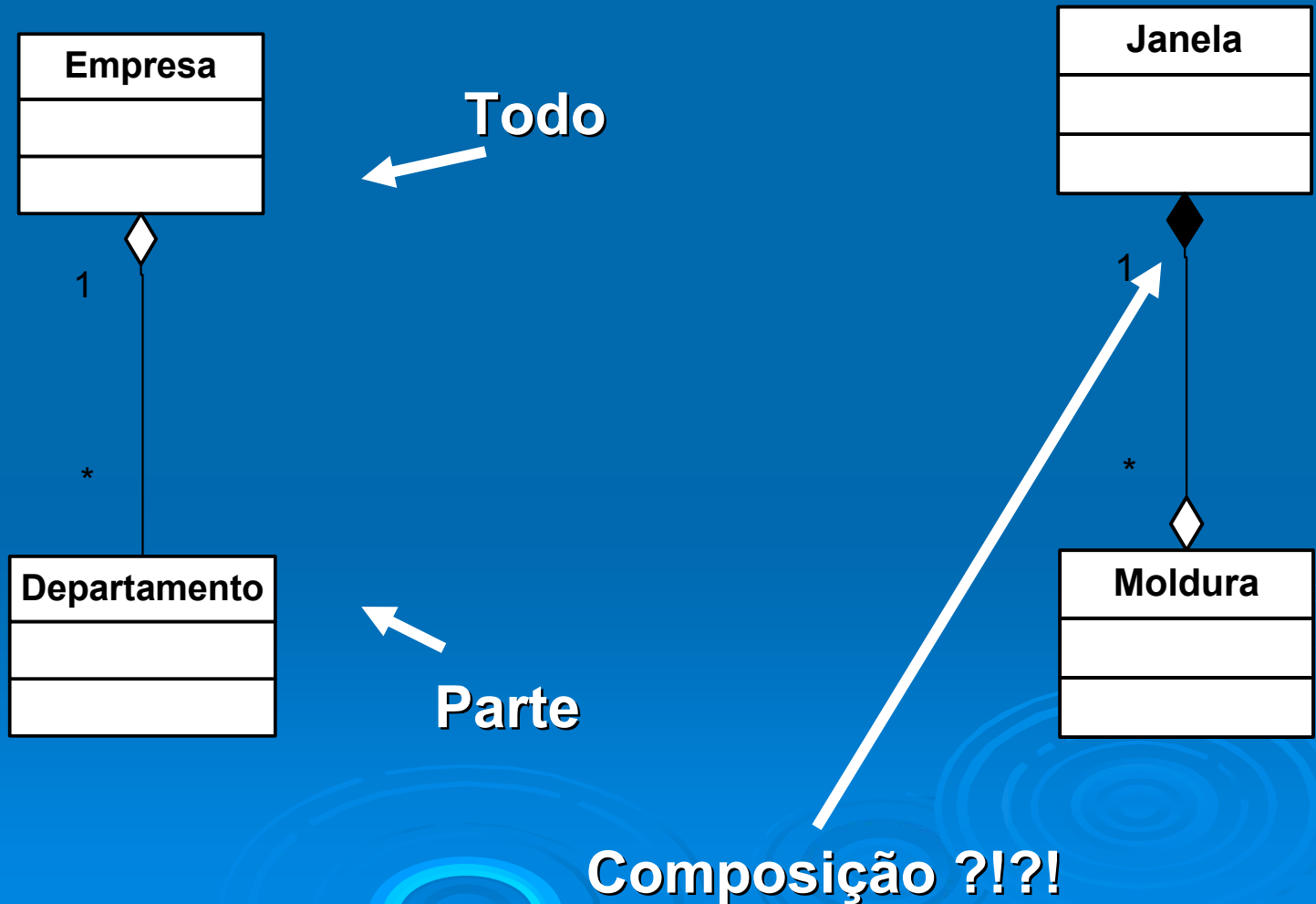
# Sobre os “Relacionamentos” – Associação



**Multiplicidade**

- Nome
- Papel
- Multiplicidade
- Agregação

# Sobre os “Relacionamentos” – Agregação



# Sobre os “Relacionamentos”

## ➤ Lembrar que:

- Relacionamentos de generalização conectam classes generalizadas para outras mais especializadas (hierarquia e herança)
- Relacionamentos de dependências representa utilização de classes
- Relacionamentos de associações são relacionamentos estruturais entre objetos

# Exercício No. 2 (adicional)

- Pense no problema de requisição de materiais e serviços (RMS) existentes na sua unidade
- Identifique as principais classes neste problema
- Especifique os atributos, operações e responsabilidades de pelo menos três classes no problema discutido

# Exercício No. 1

- Pense no problema para controle dos membros dos “Órgãos Colegiados” na UNEP
- Identifique as principais classes neste problema
- Especifique os atributos, operações e responsabilidades de pelo menos três classes no problema discutido

# Referências

- Booch, G., Rumbaugh, J., Jacobson, I. UML Guia do Usuário 11a Tiragem. Editora Campus, Rio de Janeiro, 2000.
- Pender, T. UML Bible, Wiley Publishing Inc. Indianapolis, 2003.
- Larman, C. Utilizando UML e Padrões, 1a edição. Bookman, São Palulo, 2002.

# Referências na Web

- <http://www.esm.jp/jude-web/index.html>
- [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/uml-basics.html](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-basics.html)