

# Modelagem estrutural e modelagem dinâmica em orientação a objetos

**Prof. Murillo G. Carneiro**  
**FACOM/UFU**

Material baseado nos slides disponibilizados pelo Prof. Ricardo Pereira e Silva (UFSC)

# Objetivo

- Apresentar os requisitos para uma modelagem completa
- Discutir aspectos estáticos e dinâmicos relacionados ao processo

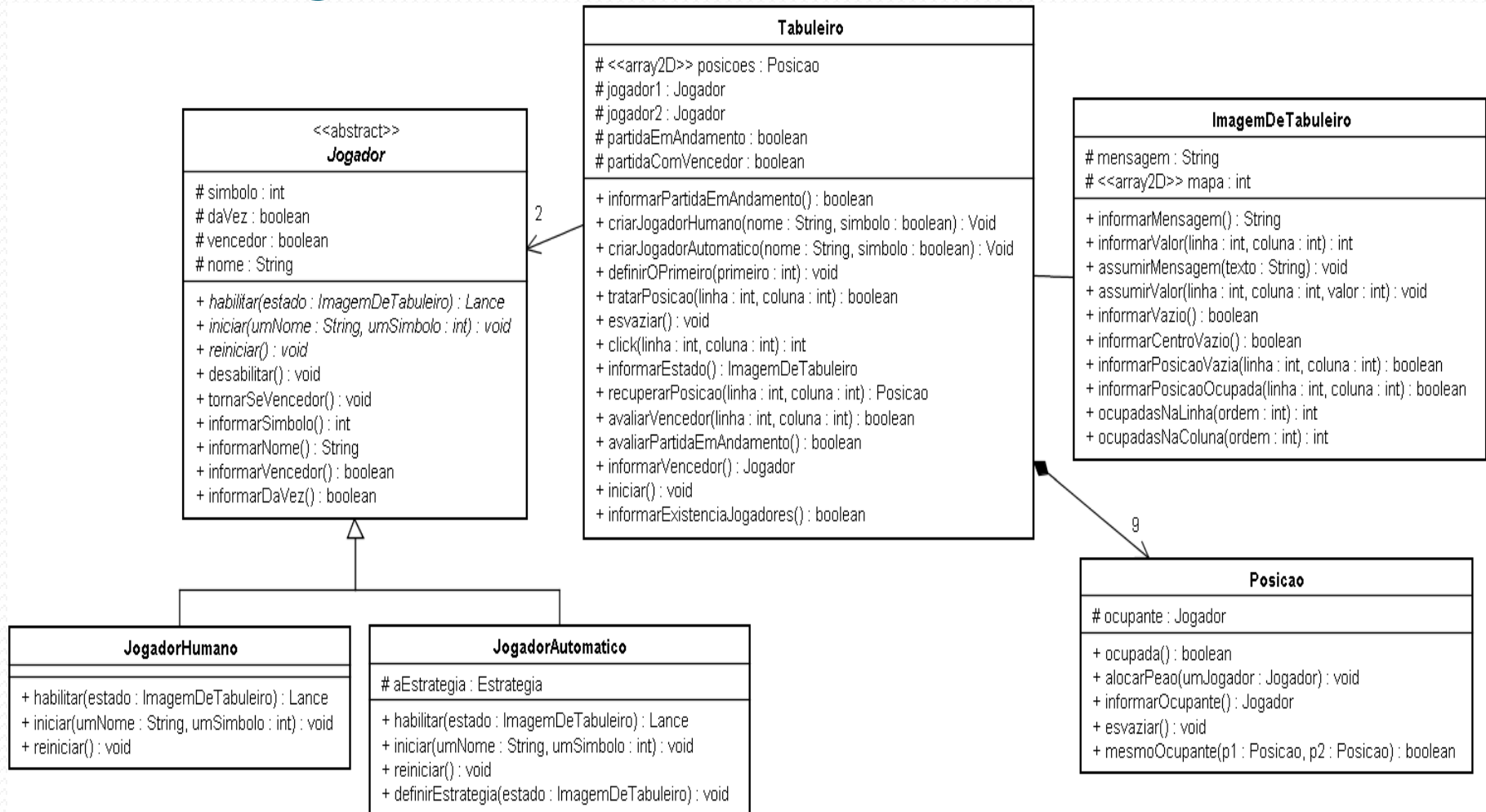
# Requisitos para uma modelagem completa

- Não é possível afirmar que um projeto está ou não completo, com precisão matemática
- Qual seria o mínimo de informação necessário a uma modelagem para ser considerada completa?
  - Uma modelagem incompleta não é útil
  - Uma modelagem com redundâncias
    - Consome mais esforço para produzir
    - Demanda mais esforço para manipulação e compreensão

# Requisitos para uma modelagem completa

- **Convenção adotada** → uma modelagem poderá ser considerada completa se
  - Possuir todas as informações para que possa executar com êxito a codificação
  - Possuir todas as informações para que se possa compreender o que está implementado

# Diagrama de classes constituiria uma modelagem completa?



# Diagrama de classes suporta a codificação?

- No exemplo (Jogo-da-velha), na posição de um programador
  - Como deve ser implementado o procedimento de um lance de Jogo-da-velha?
  - Como ocorreria a identificação do vencedor?
  - E a definição da ordem dos jogadores?
- O diagrama de classes não contém as respostas

# Diagrama de classes suporta a compreensão de um código?

- No exemplo (Jogo-da-velha), na posição de quem vai dar manutenção
  - Como foi implementado o procedimento de um lance de Jogo-da-velha?
  - Como ocorre a identificação do vencedor?
  - E a definição da ordem dos jogadores?
- O diagrama de classes também não contém as respostas



# Diagrama de classes NÃO constitui modelagem completa

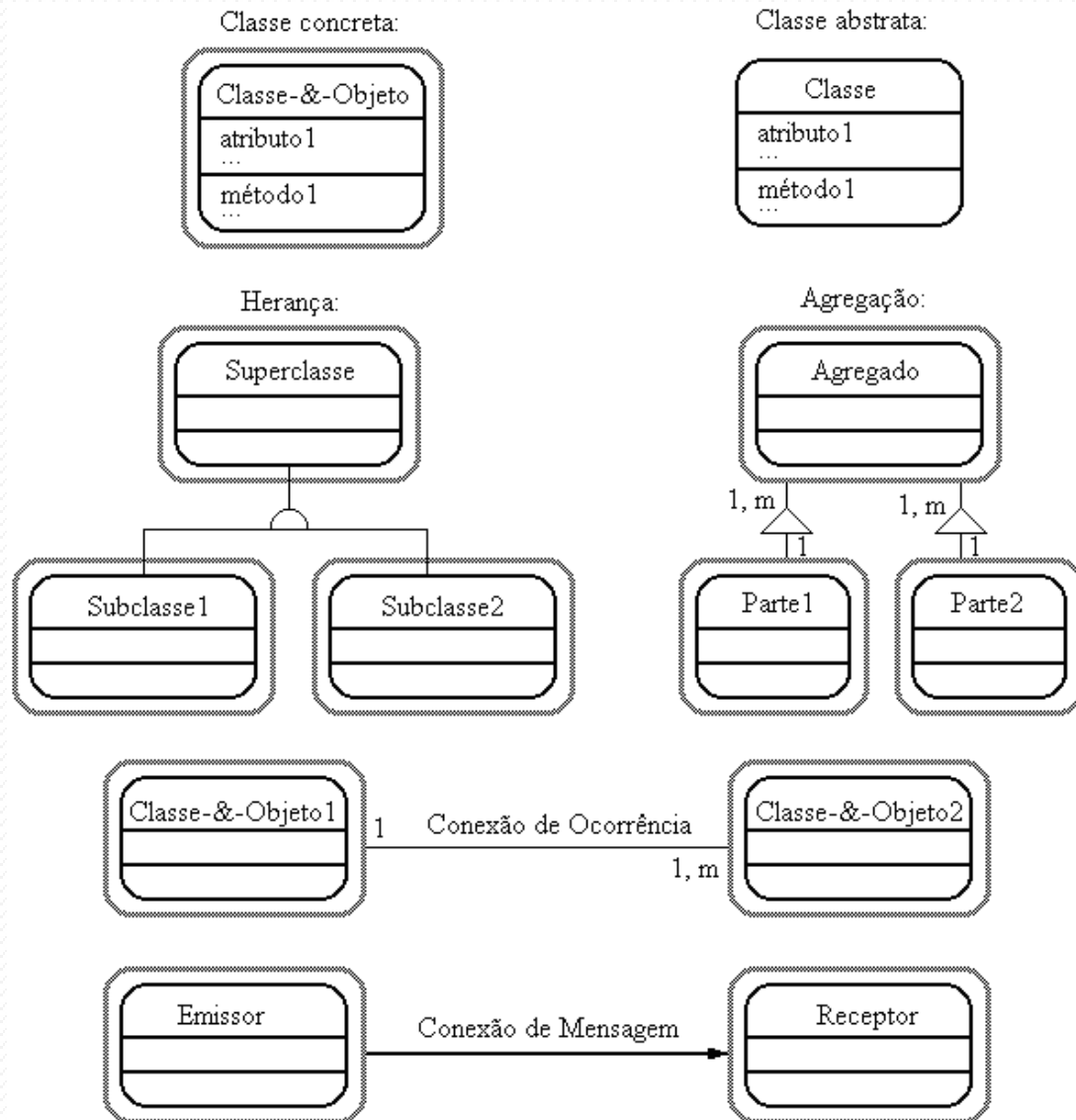
- Diagrama de classes tem informações NECESSÁRIAS mas não SUFICIENTES para uma modelagem
- Uma especificação precisa de informações adicionais para viabilizar
  - Geração de código
  - Compreensão de código existente



# Um exemplo de notação anterior a UML – modelagem completa?

- Metodologia de Coad e Yourdon
- Notação
  - Diagrama de classes
  - Especificação de classe-&-objeto
    - Descrição de classes, individualmente
    - Textual
  - Diagramas opcionais
    - Diagrama de estado de objeto
    - Diagrama de serviço (algoritmo)

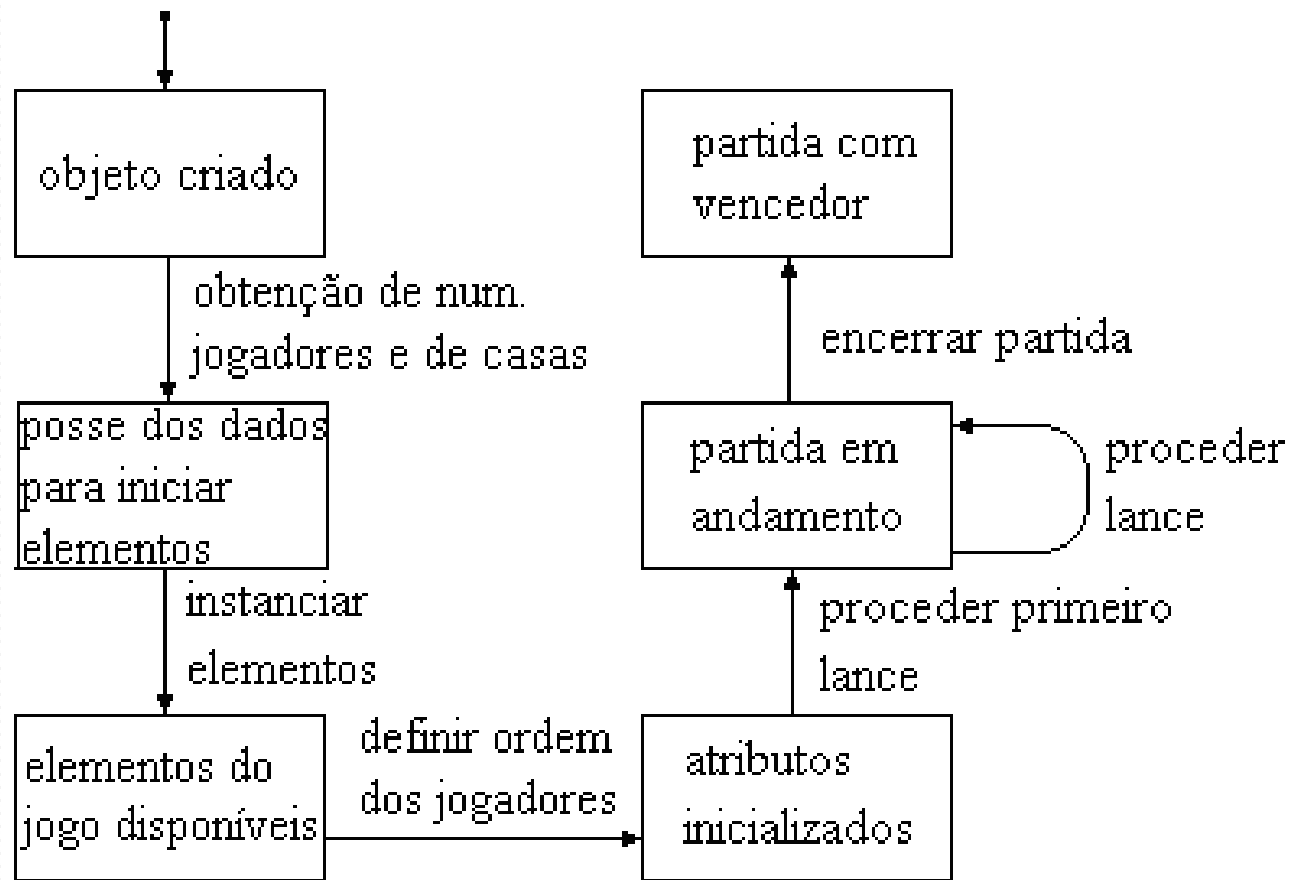
# Diagrama de classes de Coad e Yourdon



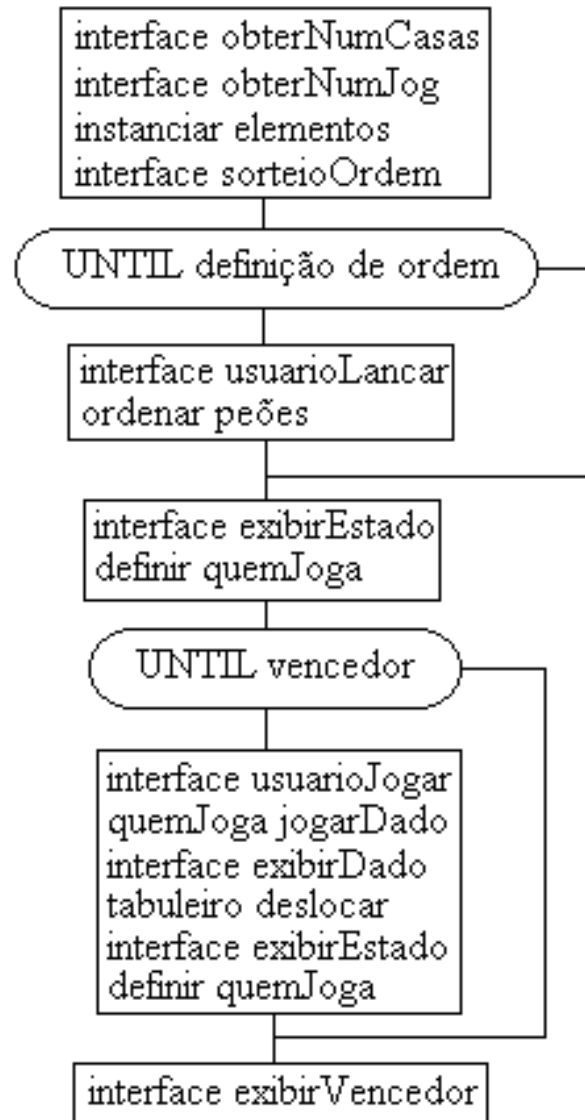
# Especificação de classe-&-objeto

- **Especificação** *<nome da classe>*
  - **atributo** *<nome do atributo>*
  - **atributo** *<nome do atributo >* ...
  - **entrada externa**
  - **saída externa**
  - **diagrama de estado de objeto** (opcional)
  - **especificações adicionais**
  - **notas**
  - **método** *<nome & Diagrama de serviço (opcional)>*
  - **método** *<nome & Diagrama de serviço (opcional)>*
  - ...

# Diagrama de estado de objeto



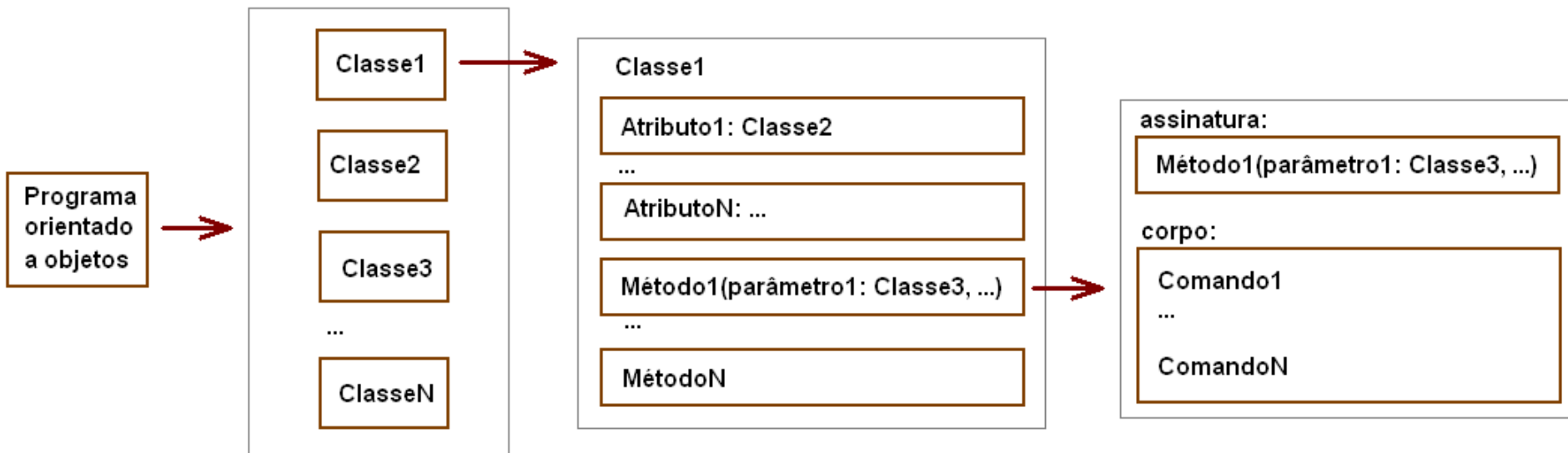
# Diagrama de serviço



# Uma especificação de Coad e Yourdon

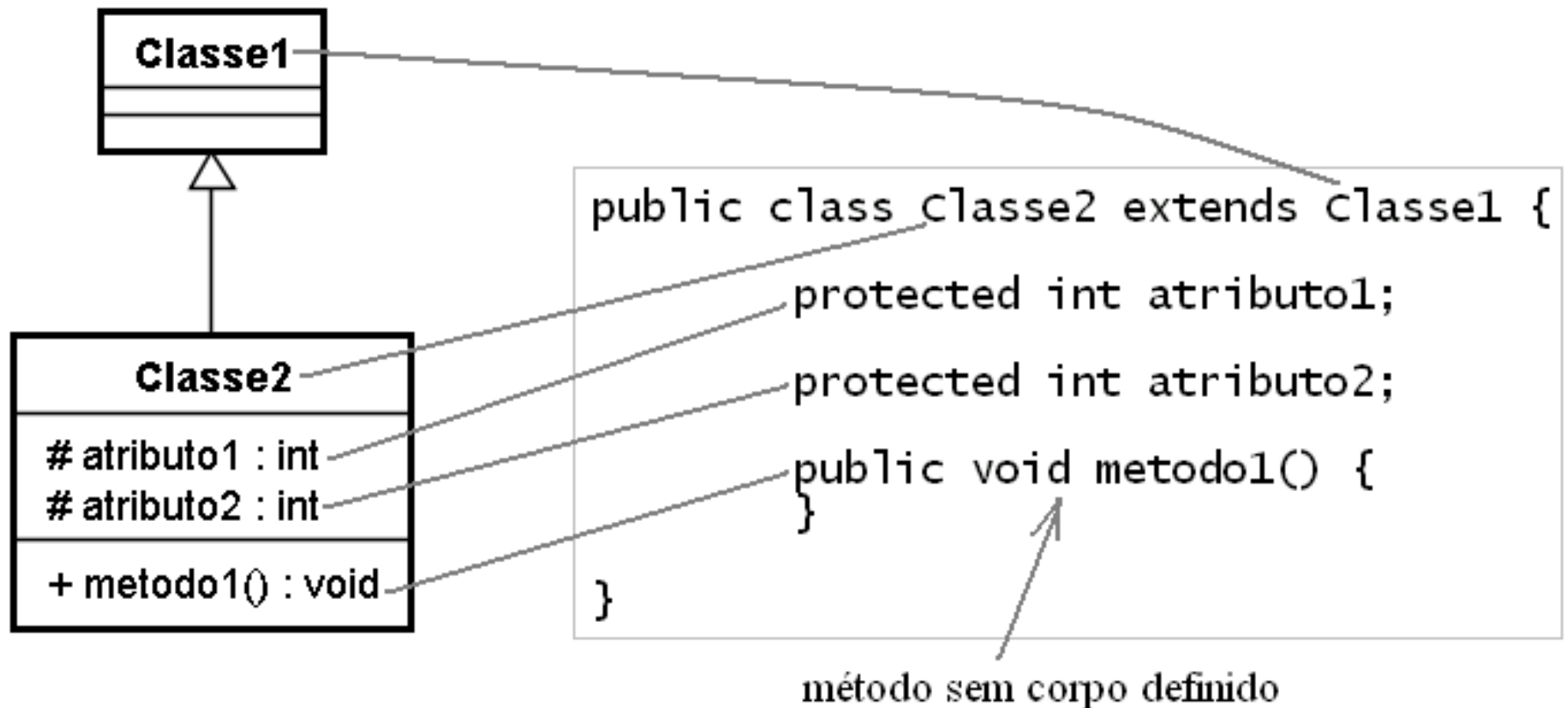
- Diagrama de classes
  - Classes, com atributos e métodos
  - Relacionamentos
- Especificações de classe-&-objeto
  - Uma para cada classe (individual)
  - Detalhamento dos atributos
  - Diagrama de transição de estados
  - Algoritmos de métodos descritos com diagrama de serviço ou de forma textual

# Necessidades para geração de código





# Necessidades para geração de código



# Necessidades para geração de código

- Diagrama de classes permite produzir esqueletos de classes
- Falta o detalhamento dos algoritmos de métodos

# Coad e Yourdon – geração de código

- Diagrama de classes
  - esqueleto das classes
- Especificação de classe-&-objeto
  - detalhes a respeito das classes
  - algoritmos
- Todas as informações → programador “feliz”

# Coad e Yourdon – manutenção

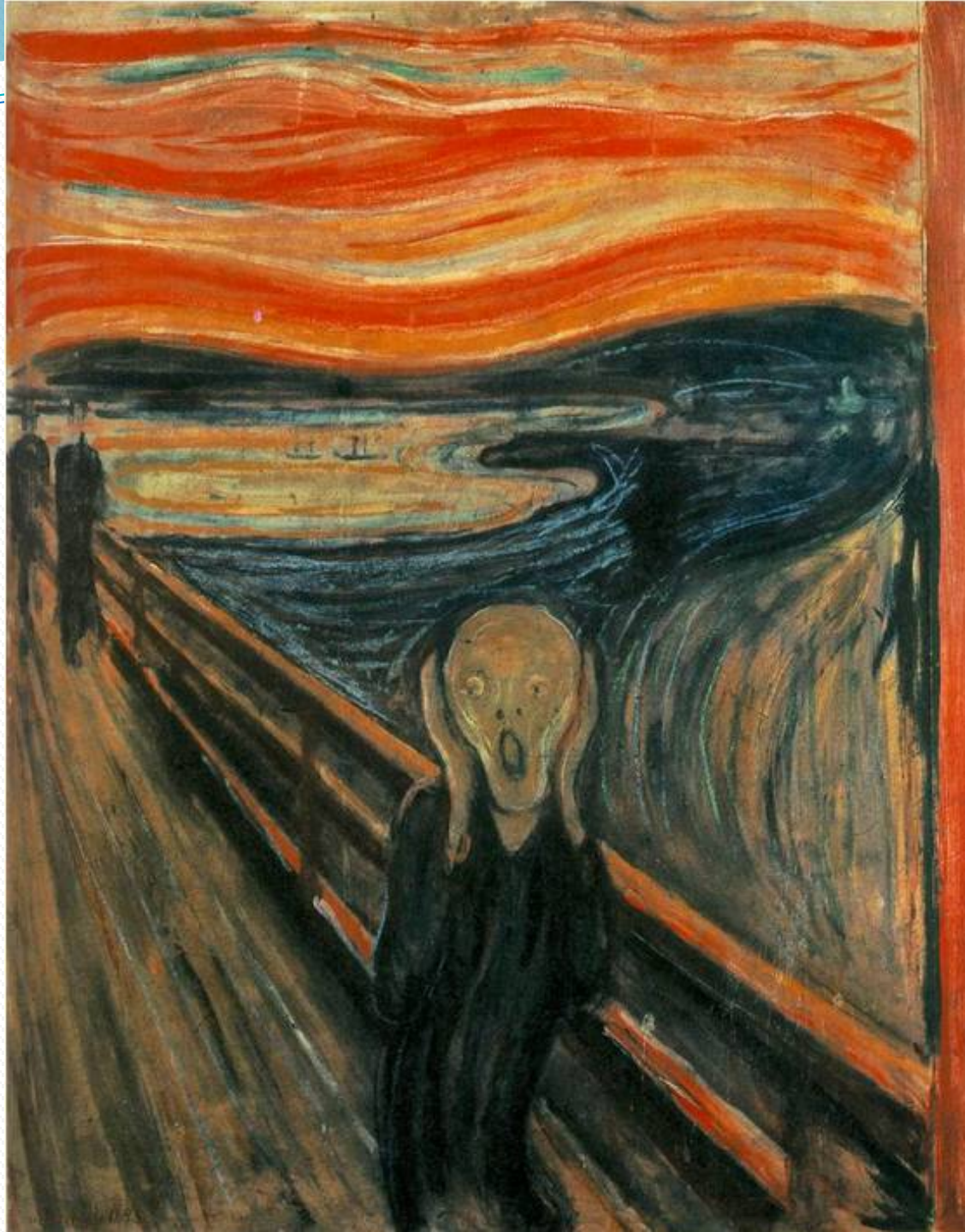
- Esforço de compreensão de um código, estudando o projeto
- Como ocorre um procedimento de lance?
- Difícil obter a resposta...
  - Não está no diagrama de classes
  - Não está no detalhamento individual das classes
  - Demanda vislumbrar o software em tempo de execução

# Requisitos para uma modelagem completa e a notação de Coad e Yourdon

- Uma especificação baseada na notação da metodologia de Coad e Yourdon
  - Apoia adequadamente a geração de código
  - NÃO apoia adequadamente a compreensão de software existente
- O que faltaria à notação de Coad e Yourdon ?
- Quais os requisitos mínimos para qualquer notação baseada no paradigma de orientação a objetos?

# Requisitos para modelagem de sistemas físicos e de software

- Duas modalidades de sistemas físicos
  - Estáticos
  - Dinâmicos
- Exemplos:







[https://www.youtube.com/watch?v=\\_w-m4esgiNo](https://www.youtube.com/watch?v=_w-m4esgiNo)

# Os exemplos

- O primeiro → uma pintura
  - Apresentada uma imagem
  - Um vídeo não traria acréscimo de informação
- O segundo → uma montanha russa
  - Apresentado um vídeo
  - Uma imagem seria uma representação menos expressiva
- O que diferencia os dois exemplos?

# O tempo como parte de uma descrição

- A diferença entre os dois exemplos é o tempo
- Uma pintura é uma estrutura estática que não apresenta modificações de estado ao longo do tempo
  - Observá-la em imagem ou vídeo não faz diferença
- Com a montanha russa, observar a evolução do seu estado ao longo do tempo é fundamental para compreendê-la
  - Percurso do carrinho, variações de velocidade
  - Uma imagem é incapaz de conter essas informações

# Sistemas físicos com ou sem característica dinâmica

- Sistemas físicos COM característica dinâmica devem ser descritos por meio de mecanismos que permitam observar seu comportamento ao longo do tempo
  - Necessidade de modelagem dinâmica
  - No caso da montanha russa, essa característica é clara porque o carrinho se move

# Sistemas físicos com ou sem característica dinâmica

- Sistemas físicos SEM característica dinâmica podem ser modelados exclusivamente por meio da descrição de suas partes e de como elas se interligam
  - Modelagem estrutural

# Modelagem de sistemas físicos e de software

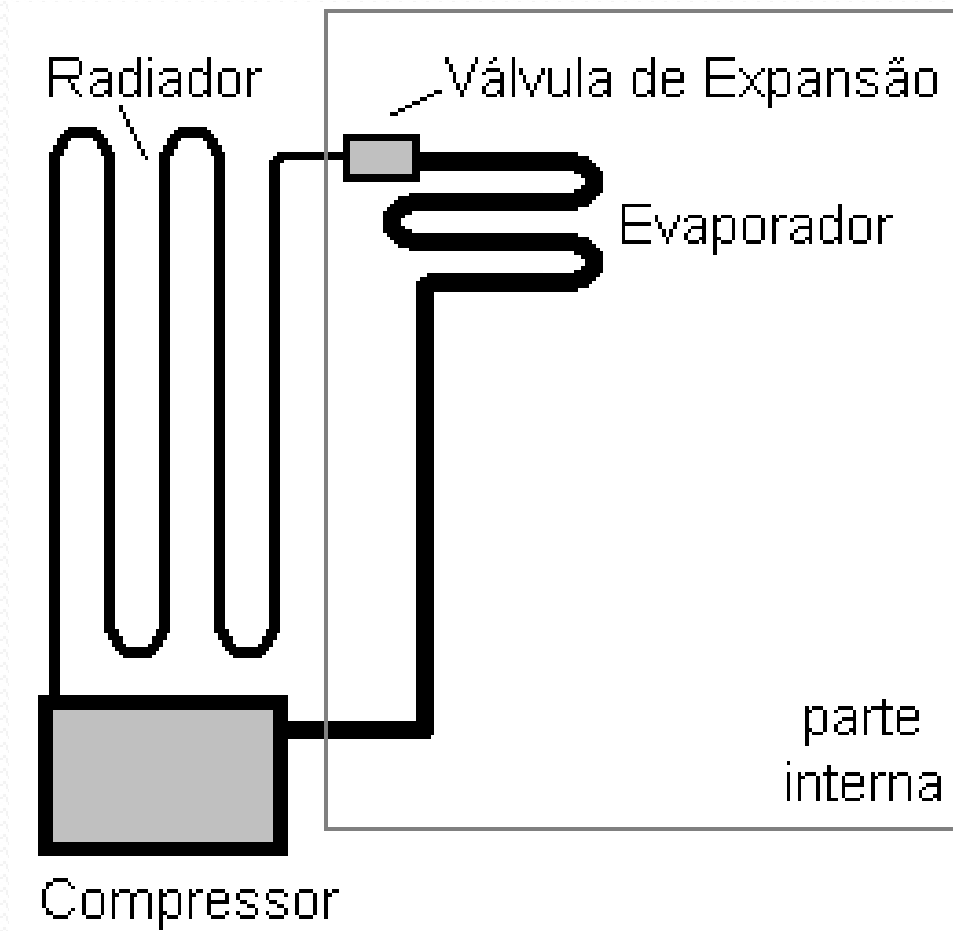
- Software é similar a sistemas físicos com característica dinâmica
  - Ao ser executado, observam-se eventos que ocorrem ao longo do tempo

# Requisitos para modelagem de sistemas físicos e de software

- Sistemas físicos SEM característica dinâmica
  - Modelagem estrutural (apenas)
- Sistemas físicos COM característica dinâmica
  - Modelagem estrutural E
  - Modelagem dinâmica
- Software demanda, portanto
  - Modelagem estrutural E
  - Modelagem dinâmica



# Exemplo de sistema físico com característica dinâmica - geladeira



# Exemplo de sistema físico com característica dinâmica - geladeira

- Diagrama apresentado → modelagem estrutural
  - As partes do dispositivo
  - Como se interligam
- Compreender o funcionamento de uma geladeira demanda também modelagem dinâmica, isto é, a descrição do comportamento ao longo do tempo
  - O fluxo de fluido refrigerante pelas partes
  - As trocas de calor ao longo do percurso

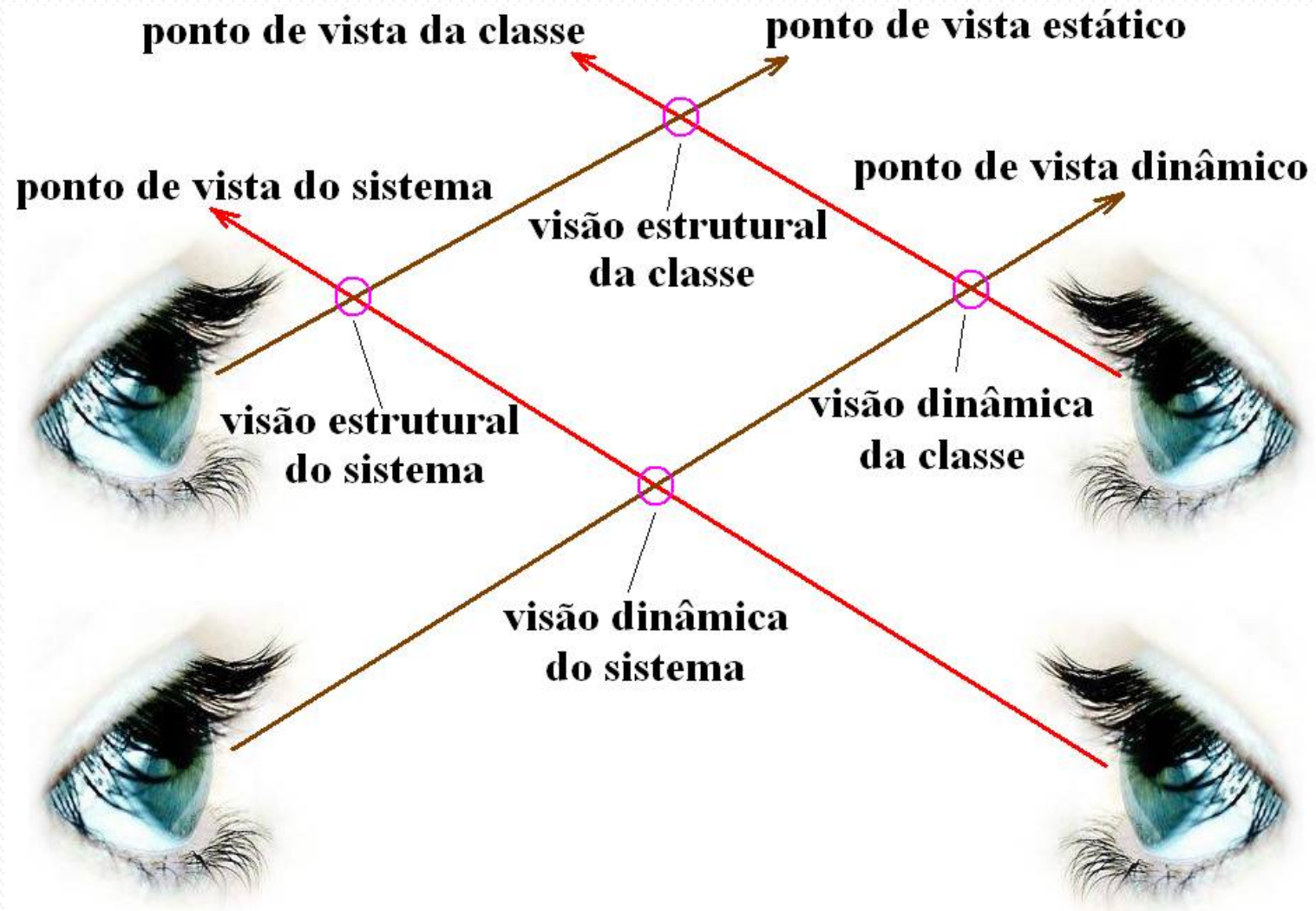
# Outra questão da modelagem de software

- Atenção pode ser voltada à totalidade
  - Envolvendo o conjunto de classes, no caso de software orientado a objetos
- Atenção pode ser voltada às partes
  - Detalhamento das classes, no caso de software orientado a objetos
- Nos dois casos, há demanda de modelagem estrutural e dinâmica

# Os quatro pontos de vista fundamentais

- Modelagem estrutural do sistema
- Modelagem dinâmica do sistema
- Modelagem estrutural da parte (da classe)
- Modelagem dinâmica da parte (da classe)

# Os quatro pontos de vista fundamentais



# Conteúdo da modelagem estrutural de sistema

- O conjunto de elementos que compõem um software orientado a objetos e seus relacionamentos
  - Considerando os elementos de composição como as classes, tem-se essa visão apresentada pelo diagrama de classes
    - O conjunto de classes e seus relacionamentos (herança, associação, composição, agregação)
  - Outros diagramas de UML exploram a modelagem estrutural do sistema em um nível de abstração mais elevado que o da organização de classes

# Conteúdo da modelagem estrutural de classe

- Os elementos que compõem uma classe
  - Seus métodos e atributos
  - Tipos (de parâmetro, de retorno) definem relacionamentos entre esses elementos
- Visão obtida com o diagrama de classes



# Conteúdo da modelagem dinâmica de sistema

- O conjunto de funcionalidades do software
- Detalhamento das funcionalidades
- Explicitação de como os objetos interagem em tempo de execução para efetuar cada funcionalidade

# Conteúdo da modelagem dinâmica de classe

- A parte da execução de um programa observável em uma instância de classe isolada
  - Evolução de estados dos objetos
  - Algoritmos dos métodos das classes

# Requisitos para uma modelagem completa

- Uma modelagem completa deve suportar
  - Geração de código
  - Compreensão de código existente
- Para uma modelagem ser considerada completa, os quatro pontos de vista fundamentais devem ser cobertos

# Os quatro pontos de vista fundamentais e a notação de Coad e Yourdon

- A notação de Coad e Yourdon
  - Satisfaz plenamente os requisitos de três dos quatro pontos de vista
  - Não tem mecanismos para modelagem dinâmica do sistema
    - Nem funcionalidades do sistema
    - Nem interação de objetos

# Avaliação de modelagens

- A avaliação do cumprimento dos requisitos dos quatro pontos de vista é um filtro para avaliar a qualidade de notações bem como de especificações de projeto existentes

# Avaliação de modelagens

- O cumprimento dos requisitos dos quatro pontos de vista também é um indicador da conclusão de um projeto orientado a objetos durante o desenvolvimento
  - Enquanto os requisitos dos quatro pontos de vista não forem cumpridos, o projeto ainda não estará concluído

# Considerações sobre esta aula

- Modelagem orientada a objetos exige
  - diferentes técnicas de modelagem
  - abrangência das diferentes visões
    - estrutural/ dinâmica
    - sistema / classe
- UML → notação que possibilita cobrir os quatro pontos de vista fundamentais

# Referências

Booch, G.; Jacobson, I. e Rumbauch, J. **UML: Guia do Usuário**. Campus, 2006.

Silva, R. P. **UML 2 em modelagem orientada a objetos**. Visual Books, 2007.