Metodologia de modelagem
Etapa 8

Prof. Murillo G. Carneiro
FACOM/UFU

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

Objetivos

- Apresentar a oitava etapa da metodologia de modelagem
  - Etapa 8 – Geração de código
- Discutir o critério de parada do desenvolvimento iterativo

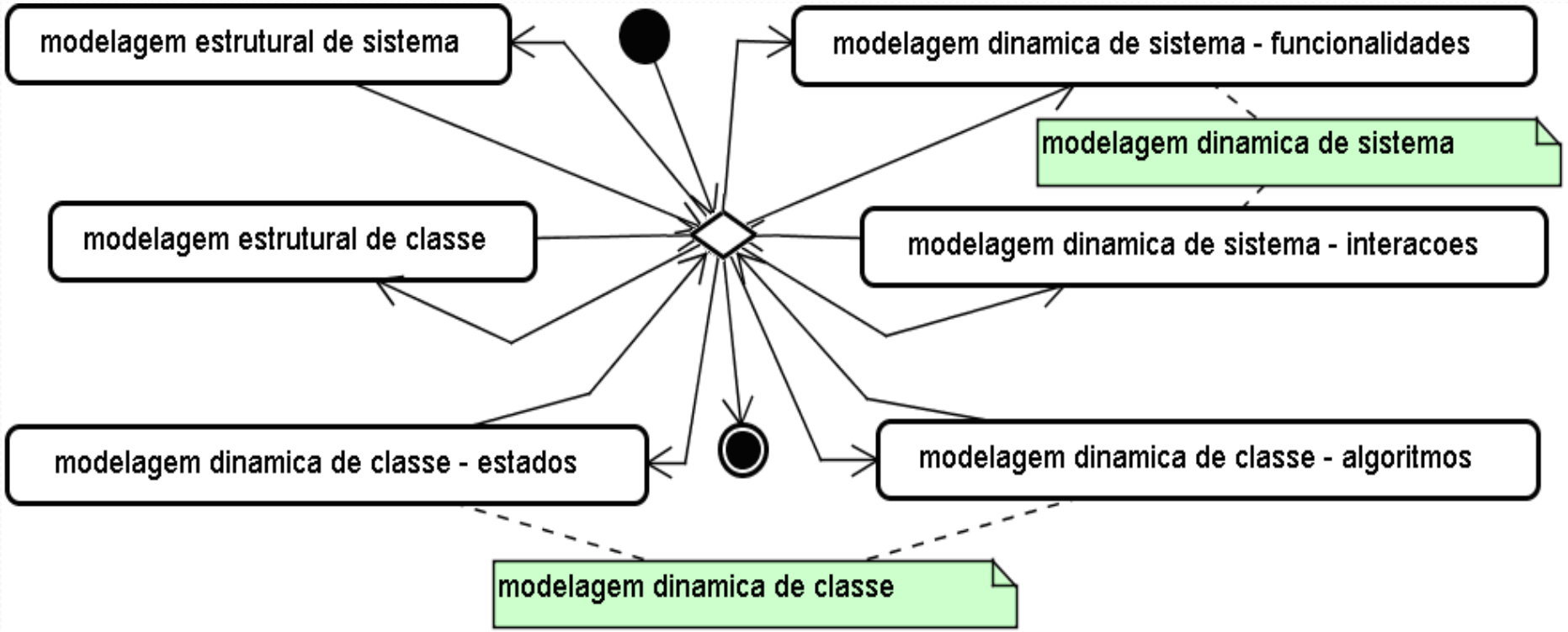
Etapa 8 – Geração de código

- A geração de código → integra-se ao procedimento iterativo
  - Resultado final → projeto e código
- De nada adiantam belos diagramas se eles não levam a um código
  - Que opere sem panes
  - Que cumpra os requisitos estabelecidos

As sete etapas anteriores – desenvolvimento iterativo

- ESTRATÉGIA → Ao constatar que a exploração de um ponto de vista não está gerando resultados satisfatórios, passar imediatamente ao tratamento de outro ponto de vista

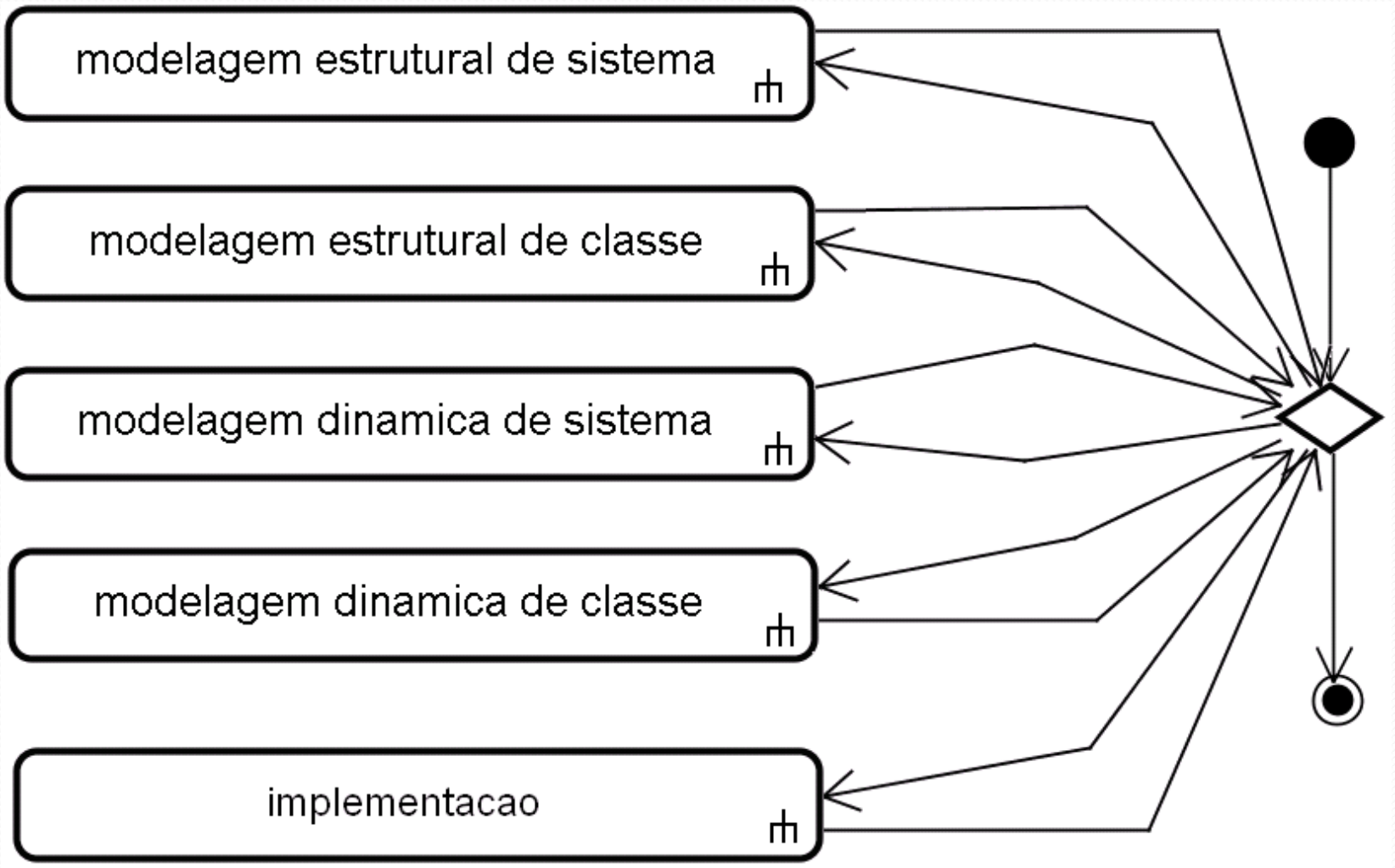
As sete etapas anteriores – desenvolvimento iterativo



Oitava etapa → quinto ponto de vista

- A geração de código junta-se aos quatro pontos de vista fundamentais como um quinto ponto de vista a explorar no desenvolvimento iterativo
  - Modelagem → subsídios para a geração de código
  - Código produzido → realimentação para a modelagem
    - identificação de oportunidades de aprimoramento do projeto

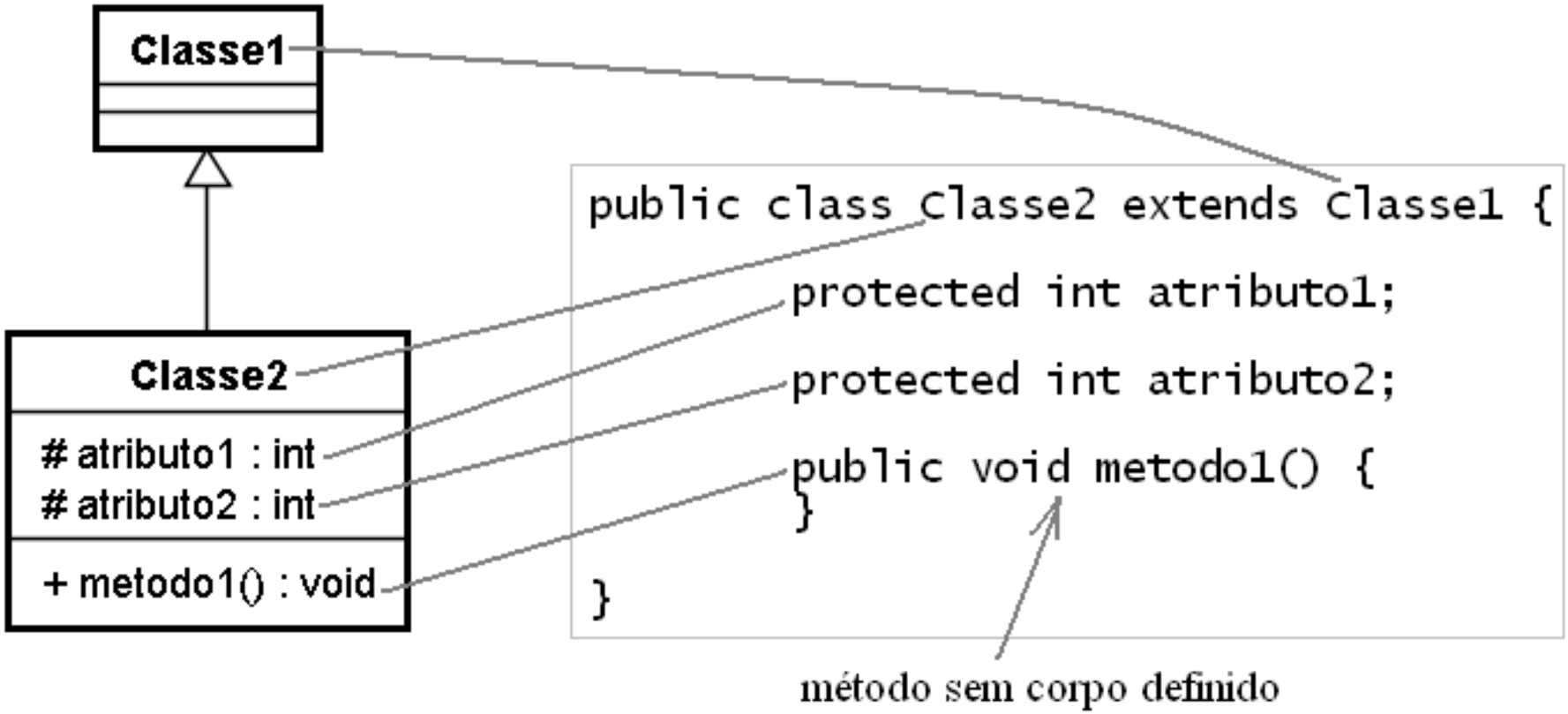
Oitava etapa → quinto ponto de vista



Tradução da especificação produzida

- Esqueleto do código → diagrama de classes
- É comum aos suportes ferramentais de edição terem geradores de esqueletos de código
  - Tradutores de diagramas de classe

Tradução da especificação produzida



Tradução da especificação produzida

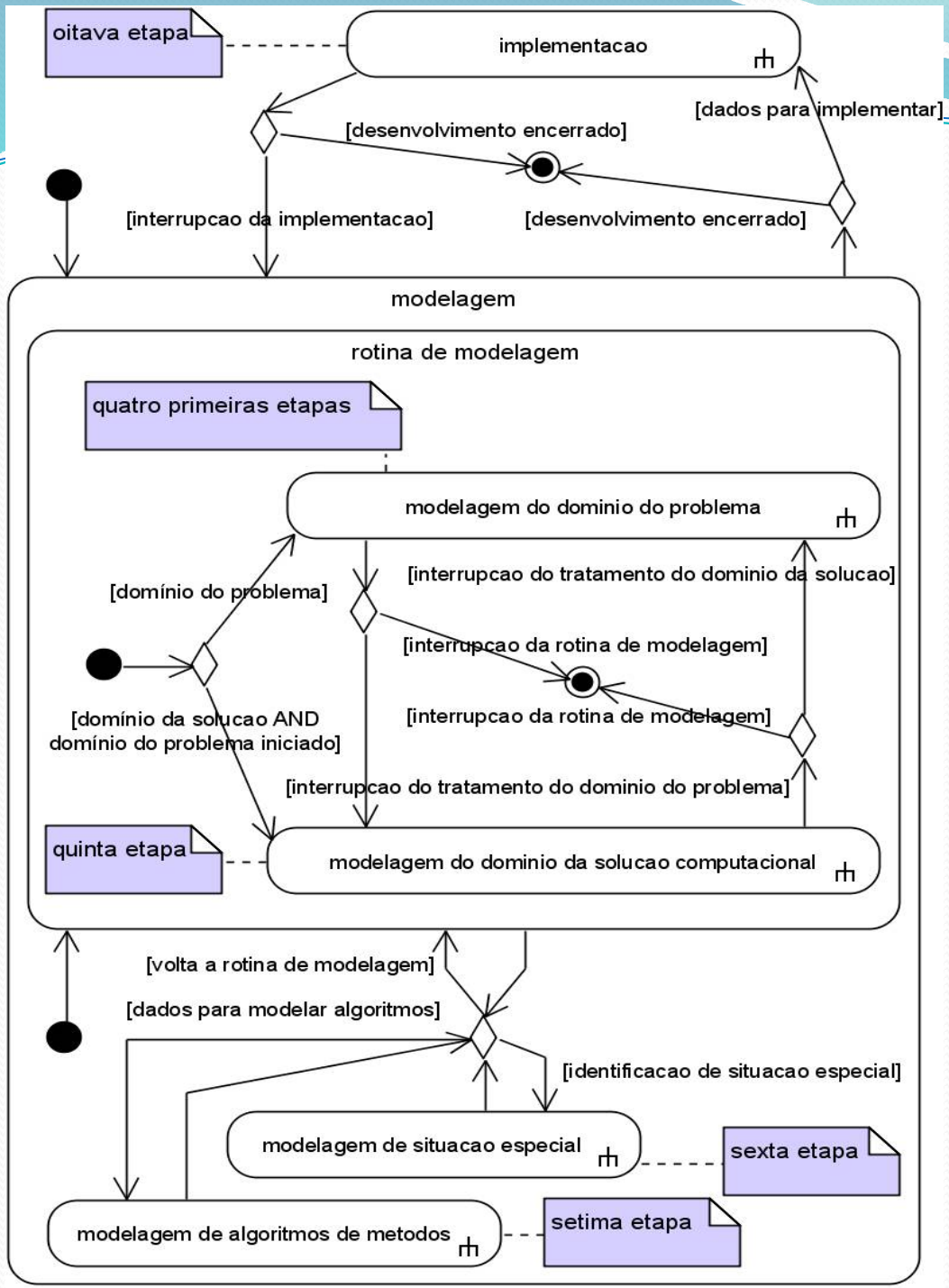
- Traduzir os algoritmos de métodos modelados em diagramas de atividades
  - Pode demandar ou não interpretação humana

Tradução da especificação produzida

- Inferir a responsabilidade dos métodos a partir das situações em que é invocado e das consequências de sua execução
  - Informações buscadas nos diagramas da modelagem

Tradução da especificação produzida

- Inferir a responsabilidade dos métodos a partir das suas assinaturas
  - Quando não houver outras informações disponíveis



Introdução da implementação no processo de modelagem

O código realimentando o projeto – exemplos de situações

- O objeto que envia uma mensagem em um diagrama de sequência não possui a referência do objeto destinatário e isso não é percebido durante a modelagem
- Um fragmento combinado de um diagrama de sequência abrange menos (ou mais) mensagens do que realmente deveria e isso é percebido na tentativa de executar o código

O código realimentando o projeto – exemplos de situações

- O código de um método apresenta envios de mensagem que não estão presentes nos diagramas de sequência, por sua necessidade só ter sido percebida na codificação

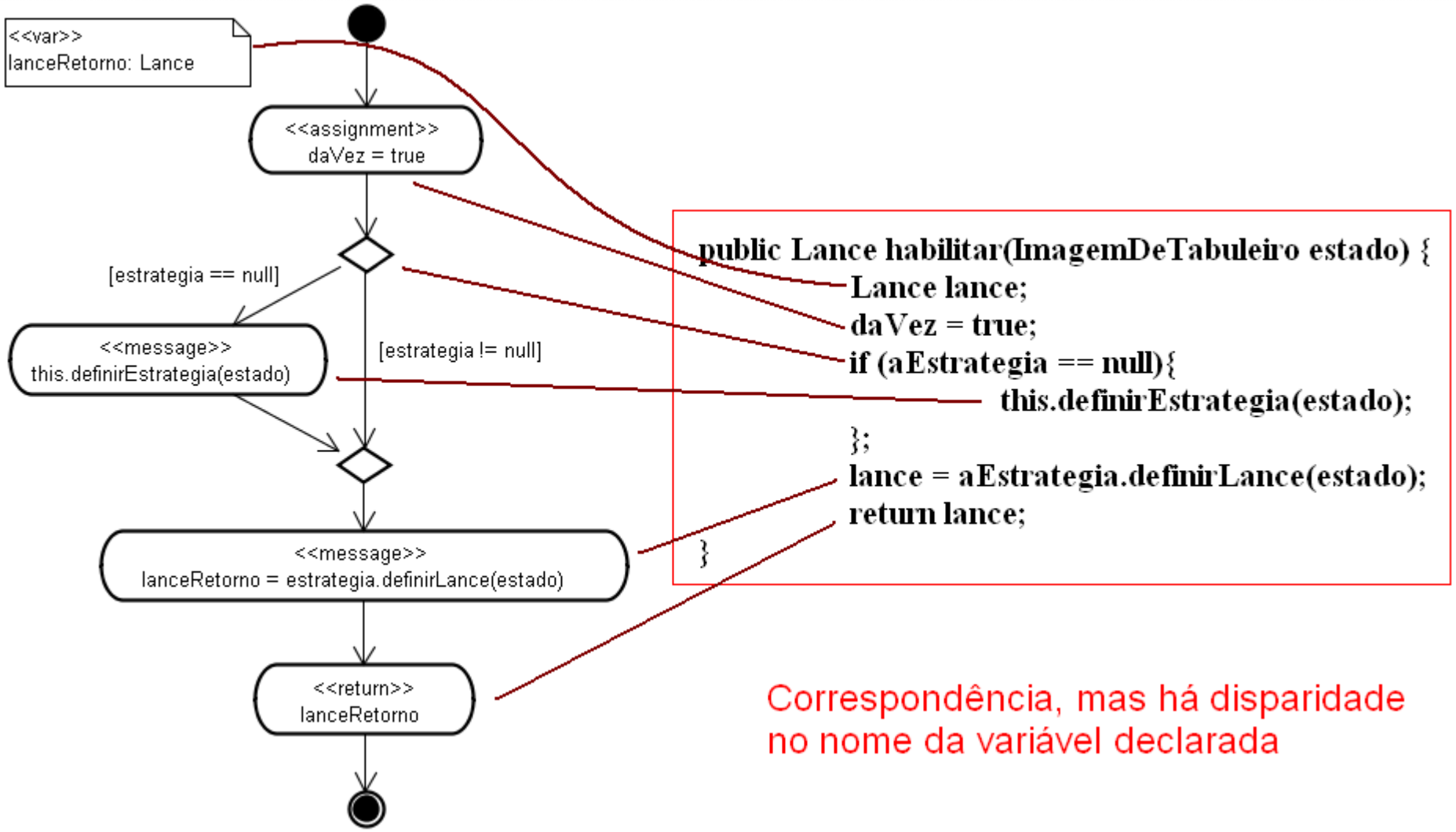
O código realimentando o projeto – exemplos de situações

- Identificação de estados (de objetos) quando da execução do código, ausentes nas respectivas modelagens de estados
- Necessidade de modificação de parâmetros (inclusão, exclusão, troca de tipo) identificada na codificação

Comparação entre projeto e código

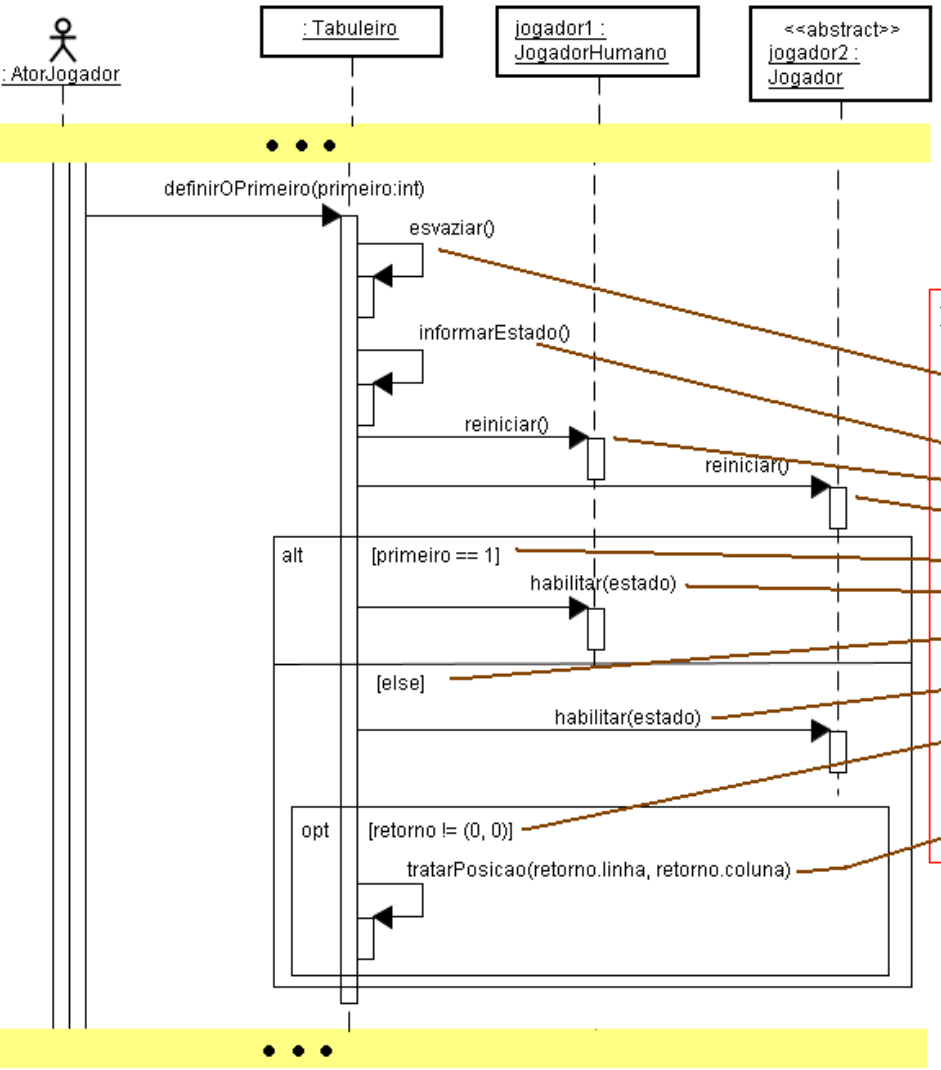
- Modelagem leva ao código, mas é preciso uma realimentação para avaliar se satisfatório
- Comparações necessárias
  - Código e algoritmo de método em diagrama de atividades
  - Código e interação de objetos em diagrama de sequência, de comunicação e de temporização
  - Código e diagrama de máquina de estados, quando ambos estão associados a uma mesma classe

Comparação entre projeto e código



Correspondência, mas há disparidade no nome da variável declarada

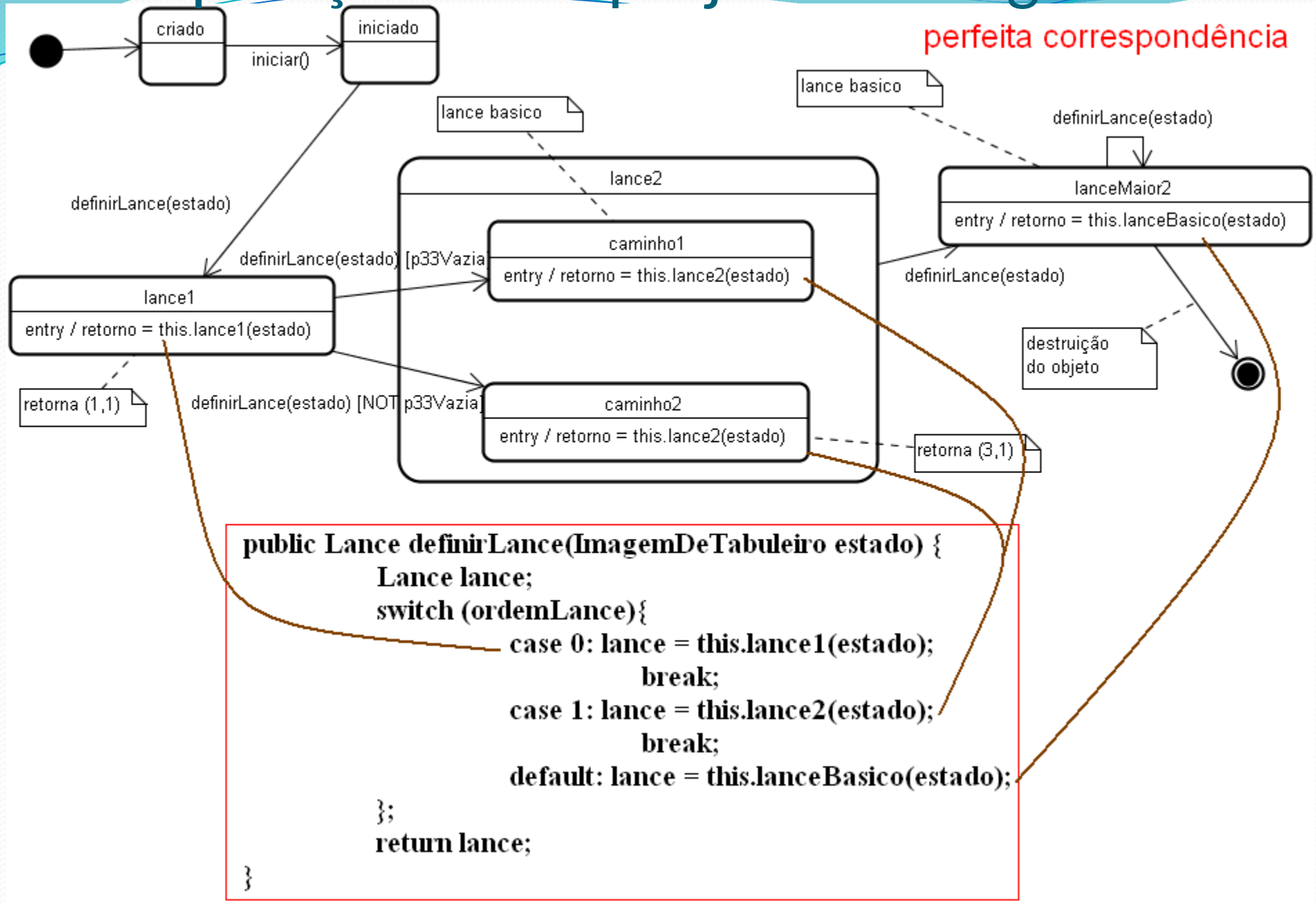
Comparação entre projeto e código




```
public void definirOPrimeiro(int primeiro) {
        ImagemDeTabuleiro estado;
        Lance j2Lance;
        this.esvaziar();
        partidaEmAndamento = true;
        estado = this.informarEstado();
        jogador1.reiniciar();
        jogador2.reiniciar();
        if (primeiro == 1) {
                jogador1.habilitar(estado);
        } else {
                j2Lance = jogador2.habilitar(estado);
                if (j2Lance.informarLinha() != 0) {
                        this.tratarPosicao((j2Lance.informarLinha()),
                                (j2Lance.informarColuna()));
                };
        };
}
```

equivalência,mas discrepância entre a condição do opt e do respectivo if

Comparação entre projeto e código



Código → o quinto ponto de vista

- Explorar o código como o quinto ponto de vista → proceder ao percurso inverso à sua criação
  - Comparar o código gerado com o projeto que o originou
- A geração de código é um caminho de mão dupla

Quando termina o processo iterativo?

- A etapa de implementação se junta às etapas de análise e projeto em um ciclo iterativo de desenvolvimento
- A codificação tanto absorve informações da modelagem, quanto as fornece
  - O código que funciona pode implicar em atualizações do projeto
  - Essas atualizações podem exigir alterações no código e assim por diante
- Quando parar?

Requisito de parada

- Haver correspondência entre projeto e código
- Código operar sem pane, cumprindo requisitos

Critérios de parada (de detalhamento do projeto)

- Que a especificação de projeto tenha informações suficientes para que terceiros possam proceder à implementação
- Que seja possível compreender o software produzido a partir do projeto, isto é, sem a necessidade de “decifrar” o código

- ...avaliações subjetivas!

Resultados parciais após a oitava etapa do processo de modelagem

- Etapas 1 a 4 – modelagem do domínio do problema
- Etapa 5 – modelagem do domínio da solução
- Etapa 6 – Modelagem de destaques na especificação, quando julgado necessário
- Etapa 7 – Modelagem de algoritmos de métodos
- Etapa 8 – Geração de código (última etapa)

Sumário da oitava etapa do processo de modelagem

- Produzir a estrutura da implementação a partir do diagrama de classes e completar os corpos dos métodos a partir de:
  - Tradução dos algoritmos de métodos modelados em diagramas de atividades;
  - Inferência das responsabilidades dos métodos a partir das situações em que são invocados;
  - Inferência das responsabilidades dos métodos a partir das suas assinaturas;
- Buscar eventuais inconsistências entre projeto e código

Considerações sobre esta aula

- Etapa 8 do processo de modelagem → Geração de código
  - Produzir o código a partir da especificação de projeto
  - Buscar oportunidades de aperfeiçoamento do projeto no código
  - Buscar discrepâncias entre projeto e código
- Finalização do desenvolvimento iterativo
  - Correspondência entre projeto e código
  - Código operar sem pane, cumprindo requisitos

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.