

LINGUAGEM UML

INTRODUÇÃO

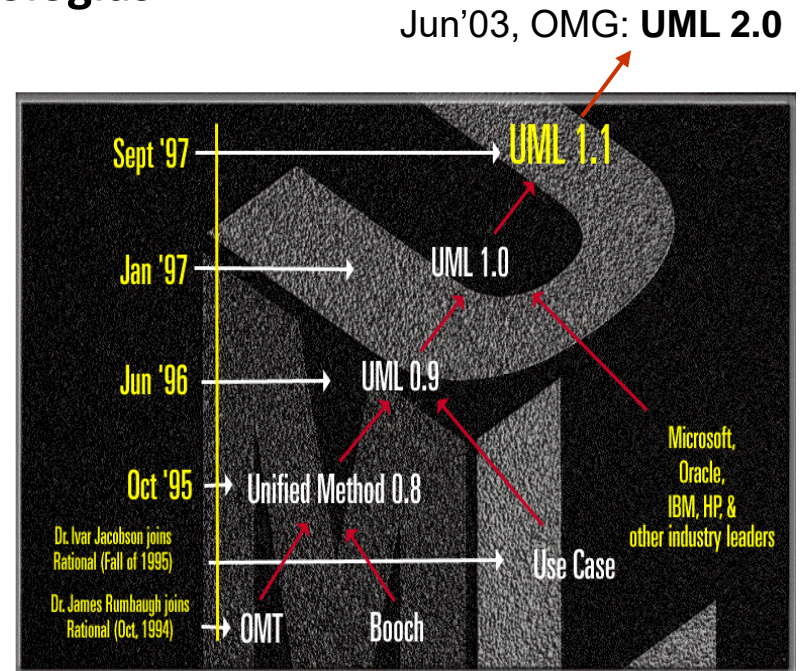
Luís Morgado

2021

LINGUAGEM UML

UML - *Unified Modeling Language*

- **Resulta da combinação de várias metodologias** de modelação orientadas a objectos
 - OMT (*Jim Rumbaugh*)
 - Booch Method (*Grady Booch*)
 - OOSE (*Ivar Jacobson*)
- **Linguagem para descrever conhecimento**
 - Acerca do domínio do problema
 - Acerca do domínio da solução
 - Linguagem gráfica
 - Elementos gráficos e textuais
- **Normalizada no âmbito do OMG** (*Object Management Group*)
 - Especificações formais
 - Adequada a diferentes âmbitos de modelação de sistemas

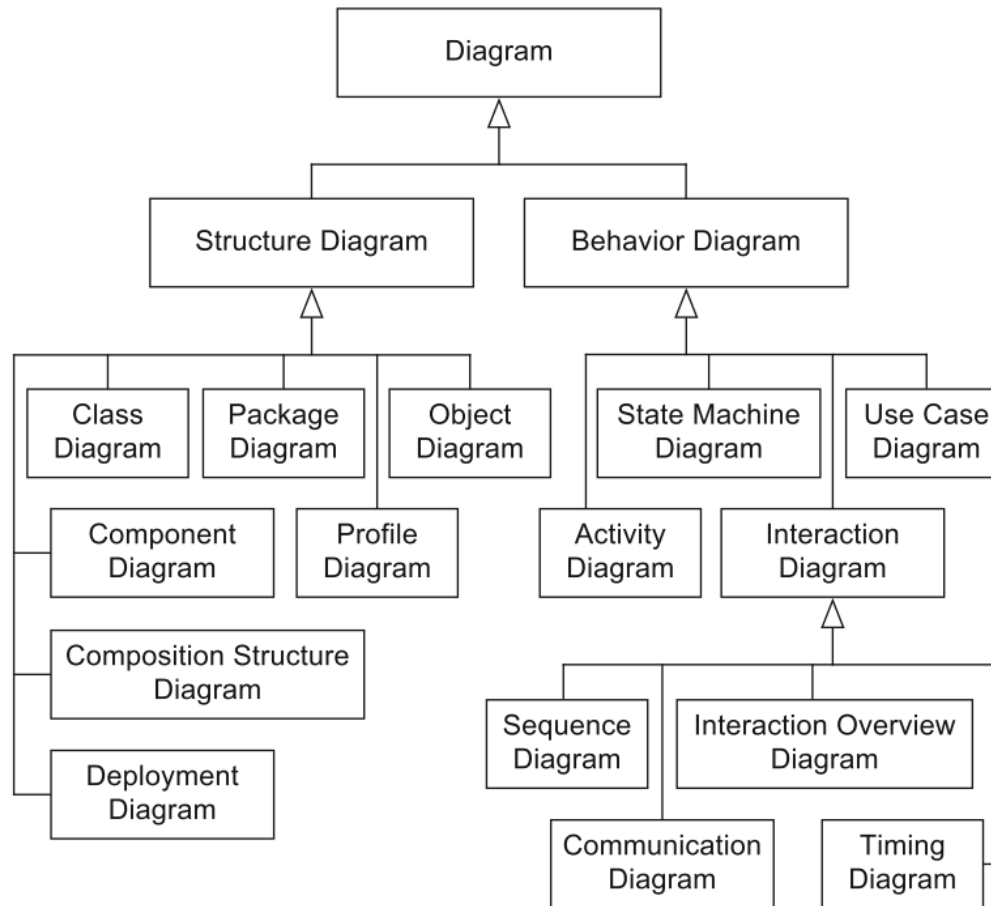


[Rational]

LINGUAGEM UML

PERSPECTIVAS DE MODELAÇÃO

Tipos de diagramas

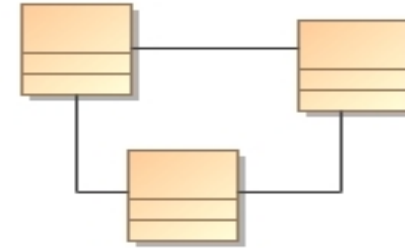


DIAGRAMAS DE CLASSES

REPRESENTAÇÃO DE ESTRUTURA

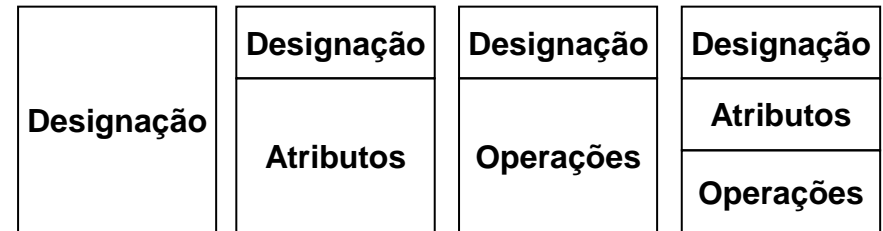
- Descrevem uma abstracção das partes e das relações entre partes de um sistema

- Organização estática do sistema
- Foco na estrutura



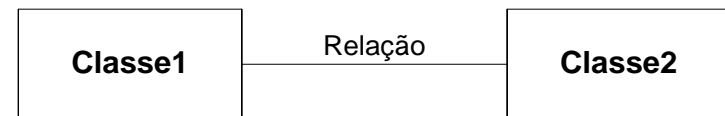
- **Classes**

- Atributos
 - Definição de estrutura
- Operações
 - Encapsulamento de comportamento

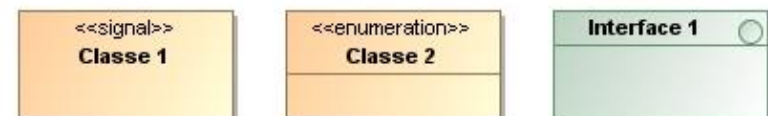


- **Relações**

- Representação de interdependências



- **Estereótipos de classes**



DIAGRAMAS DE CLASSES

- **Atributos**

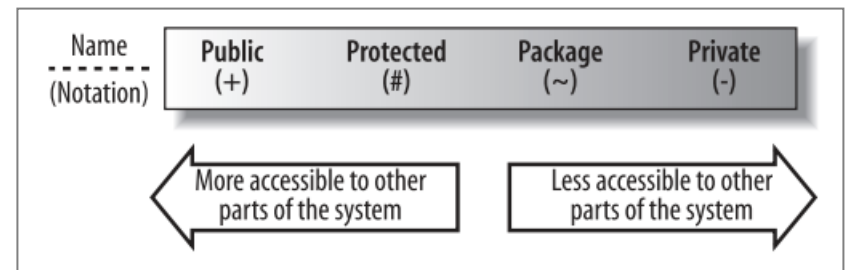
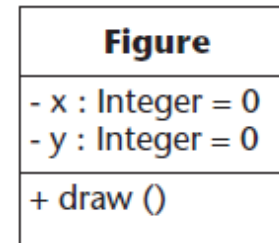
- Representação de estrutura
- Caracterizados por:
 - Designação
 - Tipo
 - Visibilidade
 - Público (+)
 - Privado (-)
 - Protegido (#)
 - Pacote (~)

- Sintaxe:

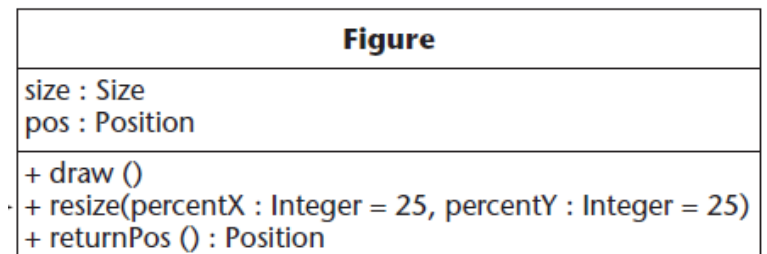
- visibility name:type = init_value
{property_string}

- **Operações**

- Representação de comportamento



[Miles & Hamilton, 2006]



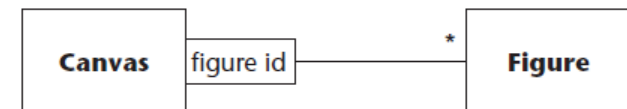
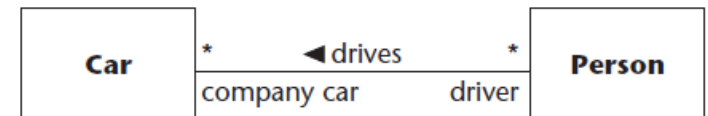
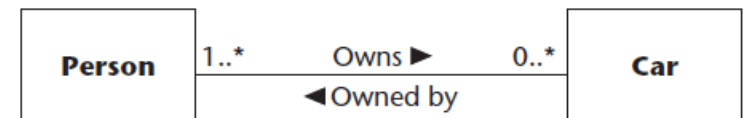
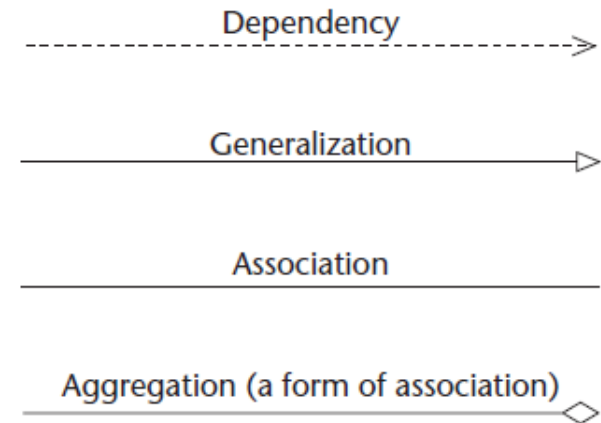
DIAGRAMAS DE CLASSES

- **Relações entre classes**

- Dependência
- Associação
- Agregação
- Composição
- Generalização

- **Propriedades das relações**

- Direcção
- Multiplicidade
- Papéis
- Qualificação
 - O qualificador de uma associação designa uma chave utilizada para obter um item da colecção respectiva

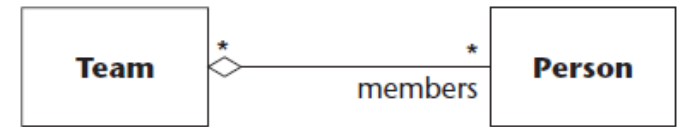


[Eriksson et al., 2004]

DIAGRAMAS DE CLASSES

• Agregação

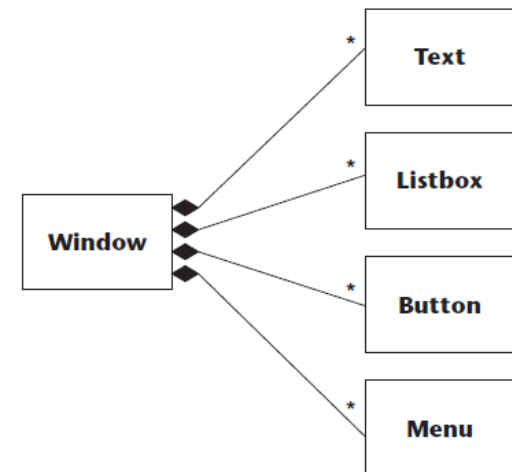
- Relação Parte - Todo
- Denota que uma parte contém outra em termos lógicos ou físicos
- Forma fraca de composição de partes
 - Não define restrições semânticas acerca de:
 - Pertença das partes
 - Criação e destruição das partes
 - Períodos de existência das partes



Relação *parte-todo*

• Composição

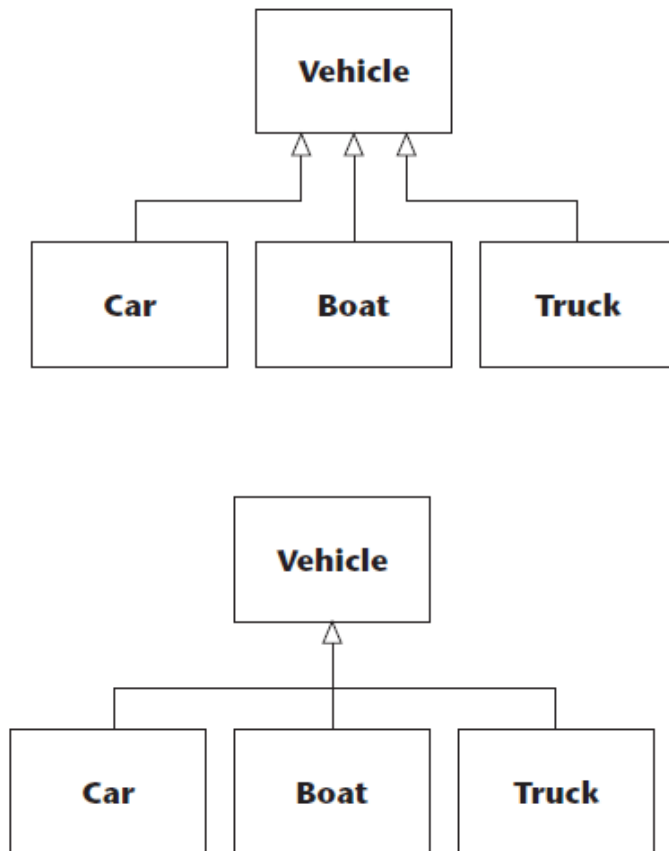
- Forma forte de composição de partes
 - Define restrições semânticas específicas
 - O todo cria e destrói as partes
 - Sobreposição de períodos de existência das partes
 - Implica exclusividade na composição das partes
- Organização hierárquica em árvore
 - Níveis de abstracção



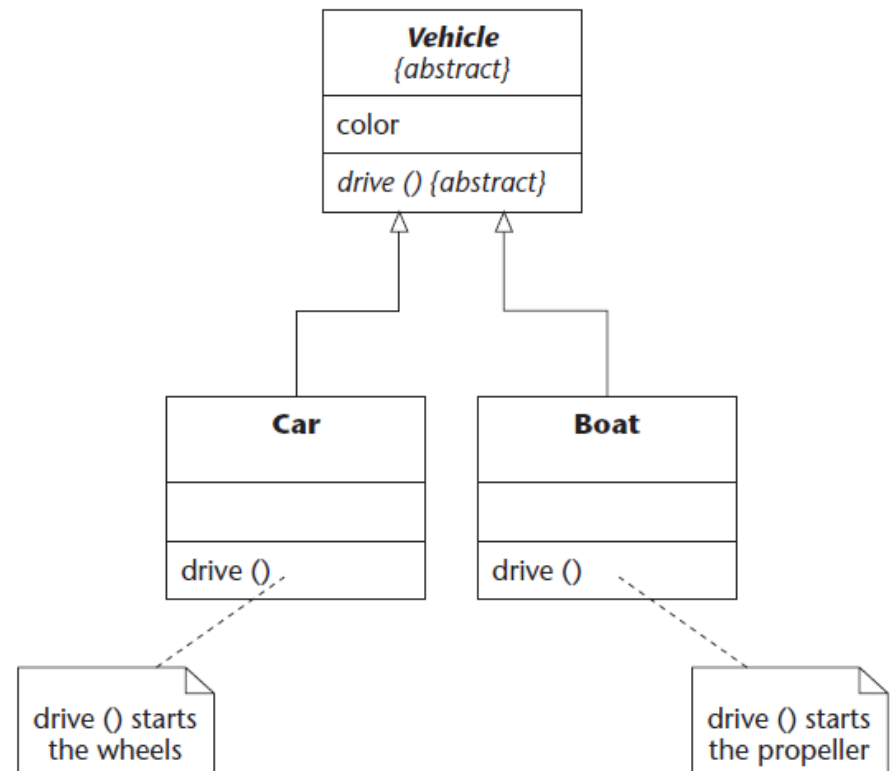
Regra da *não-partilha*

DIAGRAMAS DE CLASSES

- Generalização



- Polimorfismo

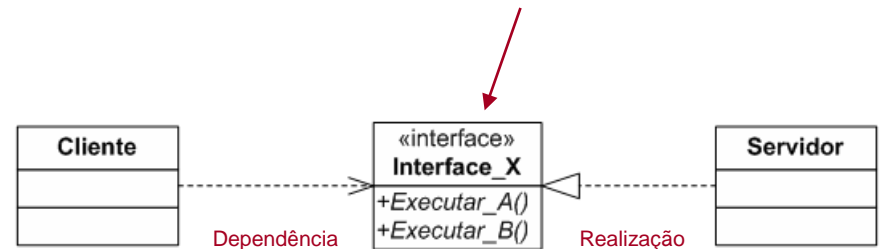


DIAGRAMAS DE CLASSES

• Interface

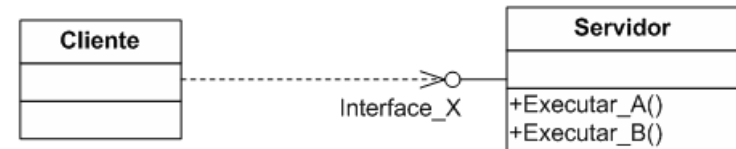
- Classificador que define as características visíveis de uma classe
- Conjunto **coeso** de características
- Define um contrato
 - Não implementa essas características
- **Encapsulamento**

A interface define um contrato de prestação de serviços independente da implementação



• Realização

- Representa implementação
- Relação entre uma interface e uma classe ou um componente
- Uma classe **realiza** as características de uma interface implementando-as
- Uma classe pode implementar mais que uma interface



DIAGRAMAS DE SEQUÊNCIA

REPRESENTAÇÃO DE COMPORTAMENTO

- **Modelo de interacção**

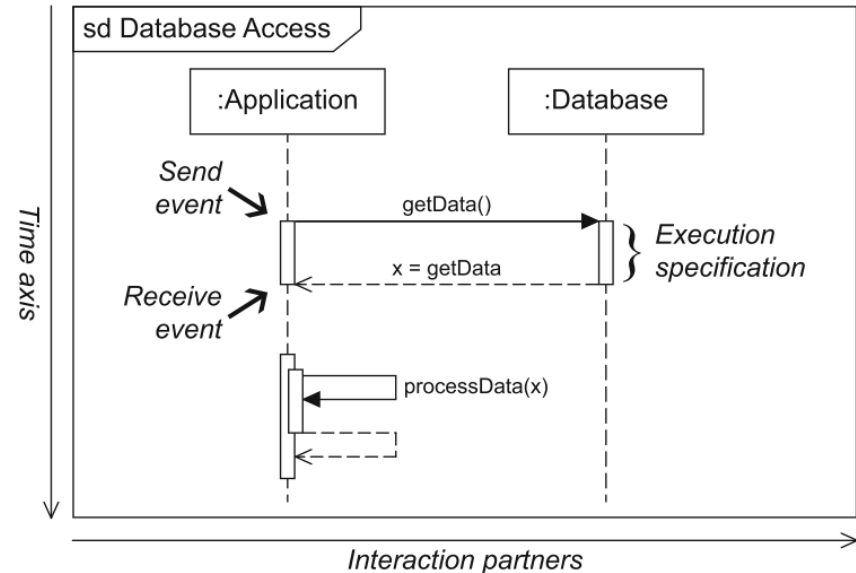
- Descrevem a comunicação entre partes do sistema e/ou com o exterior
- Ênfase na sequência temporal de interacção

- **Organização bidimensional**

- **Tempo** – vertical
- **Estrutura** (partes) – horizontal

- **Elementos de modelação**

- **Linha de vida** (*lifeline*)
 - Representa evolução temporal
- **Foco de activação** (*activation bar*)
 - Representam execução de operações
- **Mensagem**
 - Partes trocam mensagens
- **Operador**



[Seidl, 2012]

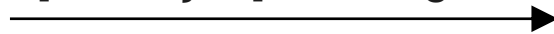
DIAGRAMAS DE SEQUÊNCIA

- Mensagens

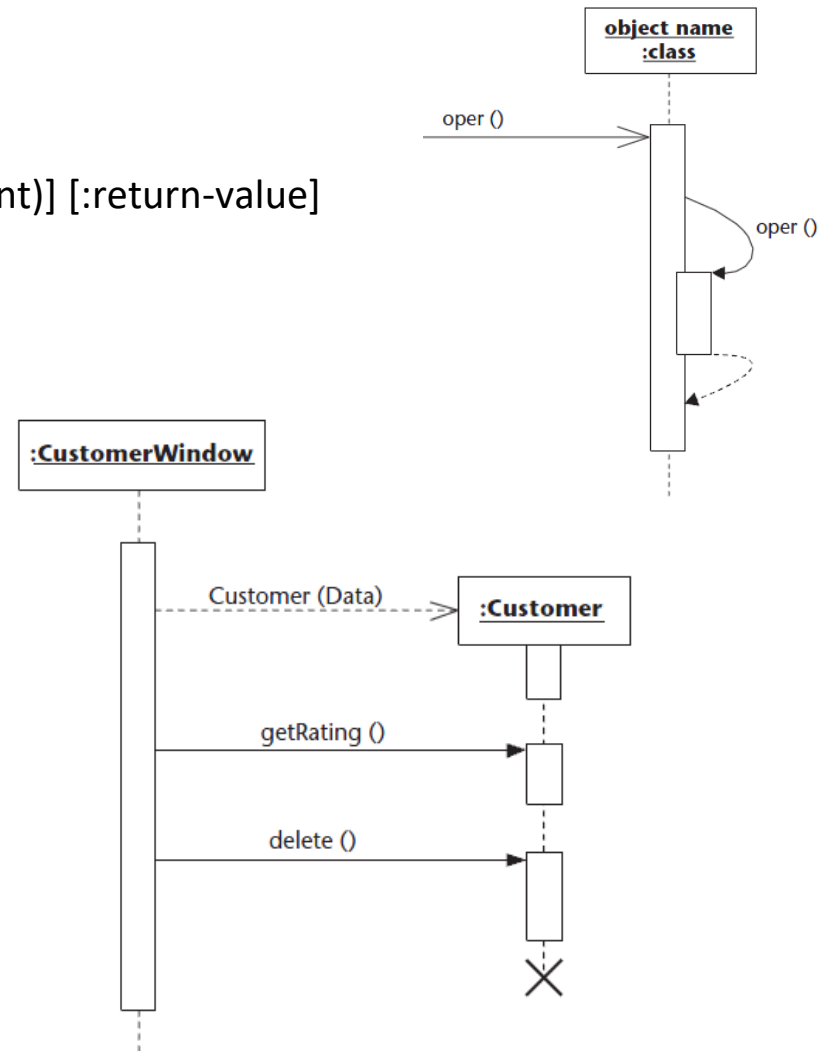
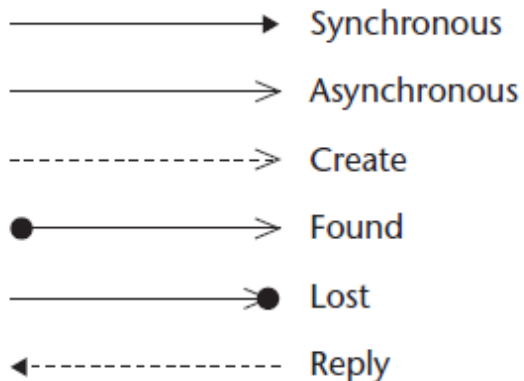
- Sintaxe

- [attribute=] message-name [(argument)] [:return-value]

[Condição] Mensagem



- Tipos de mensagens



DIAGRAMAS DE SEQUÊNCIA

Exemplo: Caso prático – iniciar jogo



O **jogo** consiste num **ambiente** onde a **personagem** tem por objectivo registar a presença de animais através de fotografias.

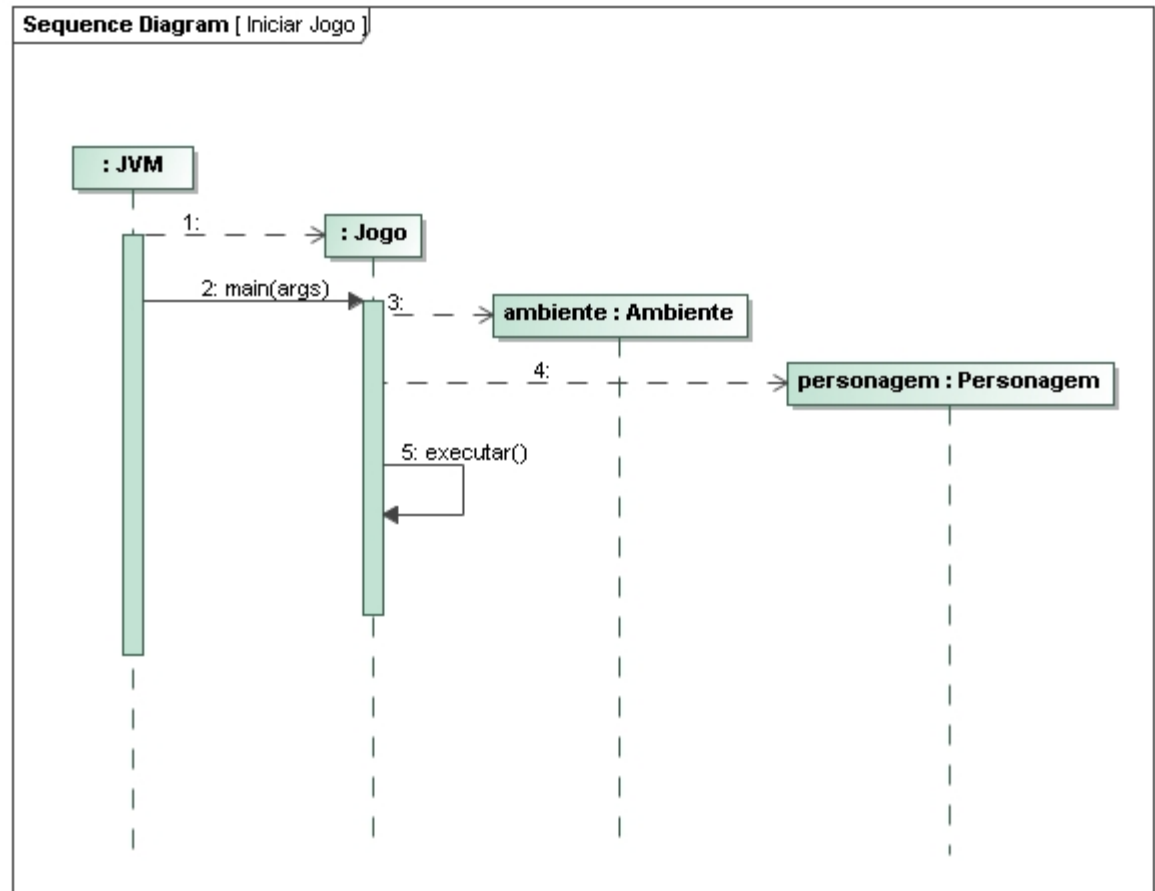
Conceitos do domínio do problema

Jogo

- Ambiente
- Personagem

Ambiente de execução

JVM: Java Virtual Machine



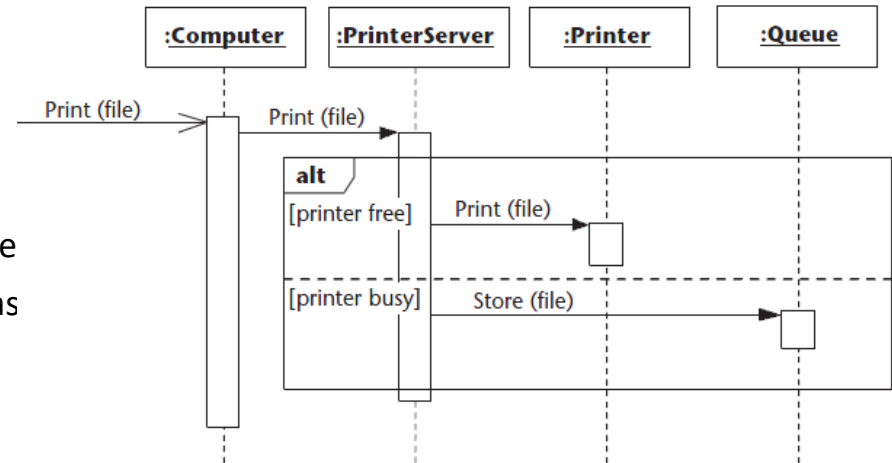
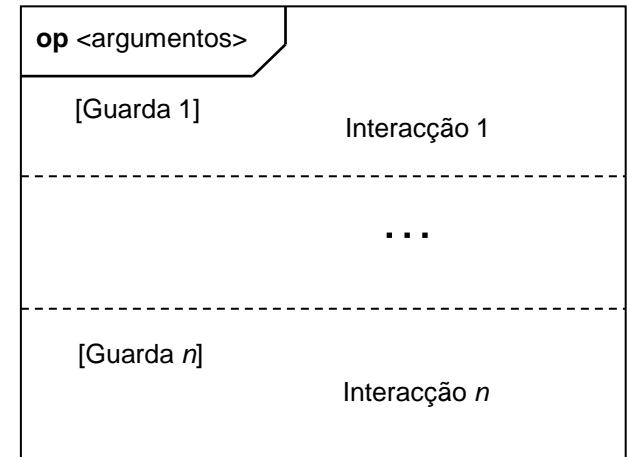
DIAGRAMAS DE SEQUÊNCIA

- **Operador**

- Fragmento de interação com semântica específica

- **Tipos de operadores**

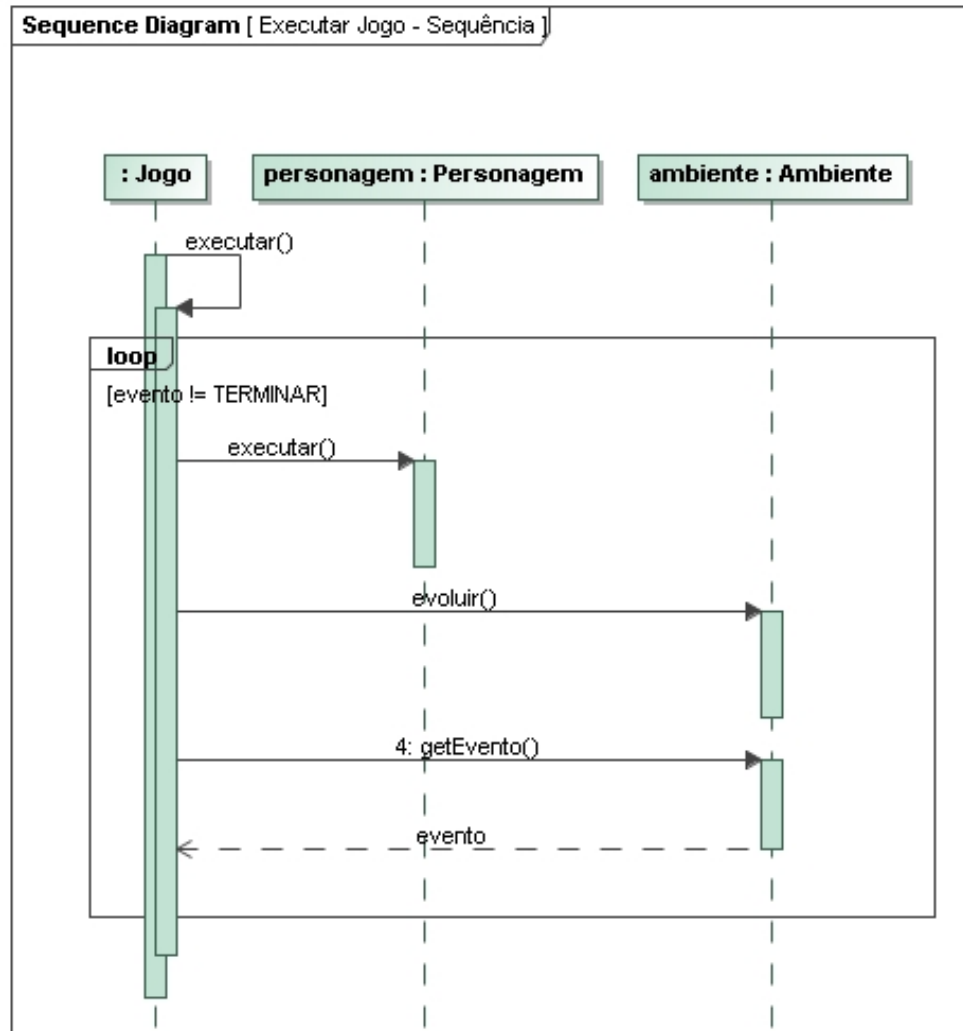
- **ref**: referência a fragmento de interação
- **loop**: repetição de fragmento de interação
- **break**: fim de repetição de fragmento de interação
- **alt**: selecção de fragmento de interação
- **par**: regiões concorrentes (paralelas)
- **assert**: fragmento de interação requerido
- **opt**: fragmento de interação opcional
- **neg**: especificação negativa (não pode acontecer)
- **region**: região crítica (não são permitidas outras mensagens)



DIAGRAMAS DE SEQUÊNCIA

Exemplo: Caso prático – executar jogo

Operador loop:
repetição de fragmento de
interacção



BIBLIOGRAFIA

[Watson, 2008]

Andrew Watson, *Visual Modeling: past, present and future*, OMG, 2008.

[Meyer, 1997]

B. Meyer, *UML: The Positive Spin*, American Programmer - Special UML issue, 1997.

[Yelland et al., 2002]

Yelland, M. J., B. I. Moat, R. W. Pascal and D. I. Berry, *CFD model estimates of the airflow over research ships and the impact on momentum flux measurements*, Journal of Atmospheric and Oceanic Technology, 19(10), 2002.

[Selic, 2003]

B. Selic, *Brass bubbles: An overview of UML 2.0*, Object Technology Slovakia, 2003.

[Graessle, 2005]

P. Graessle, H. Baumann, P. Baumann, *UML 2.0 in Action*, Packt Publishing, 2005.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[Seidl, 2012]

UML Classroom: An Introduction to Object-Oriented Modeling, M. Seidl et al., Springer, 2012

[Douglass, 2006]

B. Douglass, *Real-Time UML*, Telelogic, 2006.