

SÃO JUDAS
UNIVERSIDADE

Modelagem de Sistemas Orientado a Objetos com UML.


Capítulo 2

**Fundamentos de Orientação
a Objetos com UML**

Ana Paula Gonçalves Serra, Dr.

2

Onde Estamos na Disciplina de Modelagem de Sistemas Orientado a Objetos com UML?



- 1 Conceitos fundamentais de orientação a objetos.
- 2 Estruturação e modelagem de sistemas.
- 3 Diagramas de classes.
- 4 Diagrama de sequência.
- 5 Realização de Casos de Uso.
- 6 Diagrama de estados.
- 7 Diagrama de atividades.
- 8 Diagramas de Implementação (Pacote, Componente e Implantação)

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

3

Objetivos do Capítulo

- Este capítulo tem por objetivo apresentar aos alunos os seguintes conceitos:

1. Porque Construir Modelos?
2. Paradigma Orientação a Objetos (OO)
3. Histórico da UML
4. Diagramas da UML
5. Ferramentas Case
6. Visões Arquiteturais
7. Fundamentos de Orientação a Objetos (OO)

Ana Paula G. Serra

Pós-Graduação em Eng. de Software
Orientado a Objetos com UML

Ana Paula G. Serra

1. Porque Construir Modelos?

A elaboração de um modelo atende a 4 objetivos:

- ❑ Ajuda a visualizar como o sistema será.
- ❑ Permite especificar a estrutura e comportamento do sistema modelado.
- ❑ Fornece um *template* a ser utilizado durante a construção do sistema.
- ❑ Documenta as decisões tomadas durante o desenvolvimento.

Construímos modelos de sistemas porque não compreendemos como ele funciona em sua totalidade.

Construímos modelos para entender melhor o sistema que estamos desenvolvendo.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

6

1. Porque Construir Modelos?

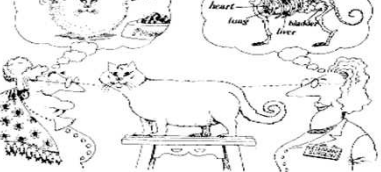
- ❑ O modelo desenvolvido por você irá refletir como o problema será tratado.
- ❑ Todo modelo pode ser expresso por diferentes níveis de precisão.
- ❑ Os melhores modelos são conectados a realidade.
- ❑ Nenhum modelo simples e único é suficiente.

Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

Modelagem de Sistemas
 Orientado a Objetos com UML

Ana Paula G. Serra

2. Paradigma Orientado a Objetos



Grady Booch; *Object-Oriented Analysis e Design with Applications 1994.*

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

8

2. Paradigma Orientado a Objetos

Um processo orientado a objetos busca:

- Primeiro objetos;
- Depois funções;
- Finalizando com implementação.

Um processo funcional busca:

- Primeiro funções;
- Depois os dados;
- Finalizando com a implementação.

Paradigma OO

Paradigma Funcional

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

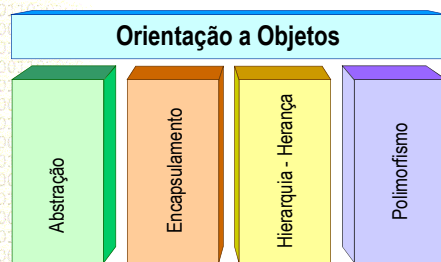
Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

2. Paradigma Orientado a Objetos

- ❑ O conceito de orientação a objetos surgiu no fim da década de 80, e significa que o sistema é organizado em uma coleção de objetos separados que incorporam tanto a estrutura quanto o comportamento dos dados.
- ❑ A Orientação Objeto pode ser aplicada na modelagem e no desenvolvimento de sistemas. Na modelagem, a orientação objeto pode ser aplicada através das notações para modelagem de sistemas orientados a objetos, essas notações são regras, conceitos e representações gráficas do sistema.

2. Paradigma Orientado a Objetos



Outros conceitos ...

Classe, Objeto, Relacionamento, Atributo, Operação, Método ...

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

3. Histórico da UML

Principais métodos utilizados:

- Booch;

- ✓ Um dos pioneiros em OO.
- ✓ Livro sobre Modelagem OO lançado em 1994.

❑ Jacobson – Objectory (OOSE)

- ✓ Modelagem OO baseada em Casos de Uso.

- ❑ Rumbaugh – OMT

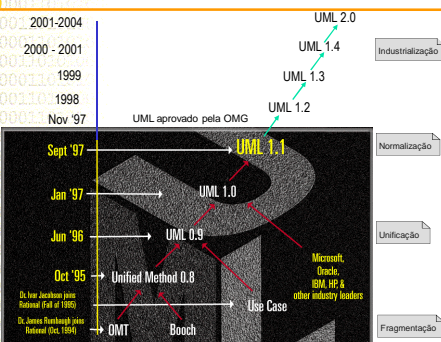
- ✓ Criação da OMT.
- ✓ Baseada na notações existentes (ER, DTE, DFD).
- ✓ Foi a mais popular.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

3. Histórico da UML



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Sargis

13

3. Histórico da UML

- UML significa “Unified Modeling Language” e é a padronização das metodologias de desenvolvimento de sistemas baseados na orientação a objetos.
- A UML incorpora as noções do desenvolvimento de software totalmente visual. Ela se baseia em diagramas que são modelados e classificados em visões de abstração.
- A UML se propõe a ser a linguagem definitiva para modelagem de sistemas orientados a objetos por ser unificada e facilitar que grupos de desenvolvimento de software interpretem de uma maneira correta e sem ambiguidades modelos gerados por outros analistas ou grupos de desenvolvimento.



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

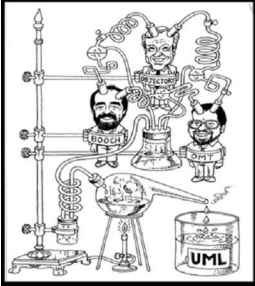
Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

14

3. Histórico da UML

- UML teve início com o “Método Unificado”, com a participação de Grady Booch e Jim Rumbaugh. (Veja apresentação na OOPSLA’95).
- No mesmo ano se uniu a eles Ivar Jacobson. Os “Tres Amigos” tornaram-se sócios na empresa Rational Software, a qual fabricava a ferramenta CASE Rational Rose.
- A IBM comprou a Rational Software em 2000.



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

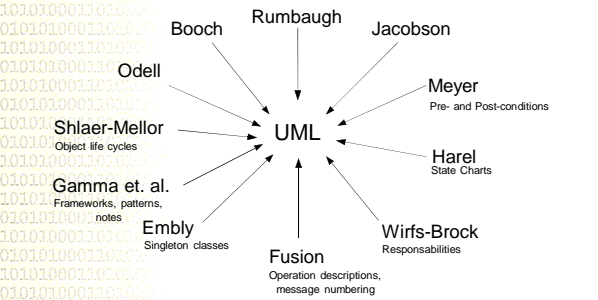
Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

15

3. Histórico da UML

Contribuições



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML


Ana Paula G. Serra

16

3. Histórico da UML

UML : *Unified Modeling Language*

- ☐ Linguagem para especificação, construção, visualização e documentação de artefatos de um sistema.
- ☐ Padrão da OMG (*Object Management Group*) – www.omg.org.
- ☐ Adotado por empresas e instituições de todo o mundo.
- ☐ Existem mais de 50 ferramentas comerciais e acadêmicas para modelagem com UML.



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

17

3. Histórico da UML

UML – Características principais



- ☐ Semântica e notação para tratar um grande número de tópicos atuais de modelagem.
- ☐ Semântica para tratar tópicos de tecnologia de componentes, computação distribuída, frameworks e Internet.
- ☐ Mecanismos de extensão que permitem que futuras aproximações e notações possam usar UML.
- ☐ Semântica e sintaxe para facilitar a troca de modelos entre ferramentas distintas.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu


Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

18

3. Histórico da UML

UML – Objetivos na sua concepção



- ☐ Definição de uma linguagem de modelagem padrão.
- ☐ Independente da linguagem de programação, ferramentas CASE e processo de desenvolvimento.
- ☐ Para diferentes projetos, diferentes metodologias, mas a mesma linguagem de modelagem.
- ☐ Simplicidade.
- ☐ Coerência na unificação de elementos de outras notações (Booch, OMT, OOSE, ...).

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

19

3. Histórico da UML

UML – Objetivos na sua concepção

- Modelar sistemas utilizando a orientação a objetos.
- Estabelecer uma relação clara entre os modelos conceituais e a implementação.
- Definir uma linguagem de modelagem:
 - Notação: símbolos
 - Regras:
 - Sintáticas: definem o formatos e como os símbolos são combinados.
 - Semânticas: definem o significado dos símbolos e suas interpretações.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

20

4. Diagramas da UML

Modelos & Diagramas.

- Um modelo captura uma visão de um sistema referente ao mundo real. Ele é uma abstração do mundo real considerando um certo propósito. Desta forma, o modelo descreve completamente aqueles aspectos do sistema que são relevantes ao propósito do modelo, e com um apropriado nível de detalhe.
- Diagrama é uma representação gráfica de uma coleção de elementos modelados.

OMG UML 1.4 Specification

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu


Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

21

4. Diagramas da UML

- A técnica de modelagem utilizada deve oferecer um conjunto de modelos, os quais permitam expressar o produto a ser desenvolvido com relação as várias perspectivas de interesse.
- O código fonte do sistema é também um modelo. Ele é um modelo mais detalhado do sistema (e também é um modelo executável). Entretanto, outros modelos também são necessários.



- Cada modelo é completo com relação a determinado ponto de vista. Contudo, existem relações de integridade entre os modelos, as quais devem ser respeitadas.


Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

22

Quais Modelos UML vocês conhecem ou já ouviram falar?




Pós-Graduação em Eng. de Software
Orientado a Objetos com UML

Ana Paula G. Serra

23

4. Diagramas da UML



- ☐ Diagrama de Casos de Uso
- ☐ Diagrama de Classes **
- ☐ Diagrama de Objetos
- ☐ Diagrama de Sequência **
- ☐ Diagrama de Colaboração ou Comunicação (na UML 2.0) *
- ☐ Diagrama de Atividades **
- ☐ Diagrama de Estados **
- ☐ Diagrama de Componentes **
- ☐ Diagrama de Implantação **
- ☐ Diagrama de Pacotes (na UML 2.0) **
- ☐ Diagrama de Interação (na UML 2.0)
- ☐ Diagrama de Tempo (na UML 2.0)
- ☐ Diagrama de Estruturas Compostas (na UML 2.0)

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Diagrama de Caso de Uso

Diagrama de Classes

Diagrama de Sequência

Diagrama de Colaboração

25

4. Diagramas da UML

Diagrama de Estados

Diagrama de Componentes

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

26

4. Diagramas da UML

Mostram os diferentes aspectos do sistema:

- ☐ Funcional
Casos de Uso
- ☐ Estático (Estrutural)
Diagrama de Classes
- ☐ Dinâmico (Comportamental)
Diagrama de Sequência
Diagrama de Atividades
Diagrama de Estados

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

27

4. Diagramas da UML


Modelo de Casos de Uso

- ☐ A modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema (diagrama + especificação).
- ☐ Descrevem o que o novo sistema deve fazer.
- ☐ Agrupa os requisitos e descreve as funcionalidades do sistema.
- ☐ Preocupa-se com as necessidades de negócio.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu


4. Diagramas da UML

Diagrama de Casos de Uso

28

Composto de:

- **Atores**
 - ✓ Um ator representa um “papel” de negócio.
 - ✓ Um ator pode ser um usuário, um hardware ou um software que interage com o sistema.
 - ✓ Estão fora do sistema, e usualmente fora do controle do sistema.
 - ✓ Impõem requisitos sobre o qual o sistema sendo construído deve fazer.




Nome do Ator

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra



Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu


4. Diagramas da UML

Diagrama de Casos de Uso

29

Composto de:

- Casos de Uso



Nome do Caso de Uso

- ✓ Um caso de uso define uma sequência de ações realizadas por um sistema e que geram um resultado de valor observável para um ator.
- ✓ São iniciados por um ator.
- ✓ Podem envolver mais de um ator.
- ✓ Descrevem como um sistema e seus atores colaboram para atingir pelo menos um dos objetivos.

Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Modelo de Casos de Uso

Diagram illustrating the Use Case Model (Modelo de Casos de Uso) in UML. The diagram shows three use cases (Caso de Uso 1, Caso de Uso 2, and Caso de Uso 3) contained within a large green rectangle. Two actors, Ator 1 and Ator 2, are shown as stick figures. A large red rectangle highlights the use cases. A red document icon represents the specification of a use case. Dashed lines connect the actors to the use cases and the specification icon to the use cases.

Ator 1

Caso de Uso 1

Caso de Uso 2

Caso de Uso 3

Ator 2

Especificação do Caso de Uso 1

Especificação do Caso de Uso 2

Especificação do Caso de Uso 3

Especificação do Caso de Uso 2

90

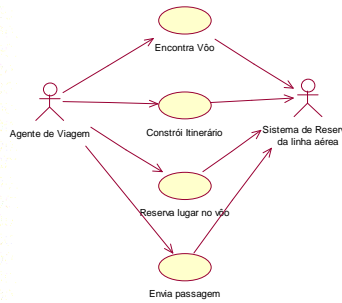
Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Exemplo de Diagrama de Caso de Uso



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Diagrama de Classes

Diagrama de classes demonstra a estrutura estática das classes de um sistema onde estas representam as “coisas” que são gerenciadas pela aplicação modelada.

❑ Classes podem se relacionar com outras através de diversas maneiras:

- ✓ associação (conectadas entre si)
- ✓ dependência (uma classe depende ou usa outra classe)
- ✓ especialização/especialização (uma classe é uma especialização de outra classe)
- ✓ pacotes (classes agrupadas por características similares).

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

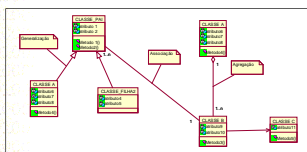
Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Diagrama de Classes

☐ Todos estes relacionamentos são mostrados no diagrama de classes juntamente com as suas estruturas internas, que são os atributos e operações.



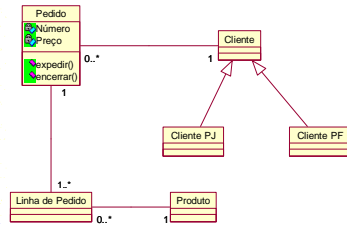
Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Sarco

4. Diagramas da UML

Exemplo de Diagrama de Classes



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

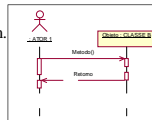
Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Diagrama de Sequência

- ❑ Mostra a colaboração dinâmica entre os vários objetos de um sistema.
- ❑ Ênfase na comunicação dos objetos através da passagem de mensagem entre os mesmos.
- ❑ Mostra a interação entre os objetos, alguma coisa que acontecerá em um ponto específico da execução do sistema.
- ❑ Ênfase na ordem temporal que os eventos acontecem.
- ❑ Devem ser construídos por cenários de casos de uso.
- ❑ Faz parte do conjunto de diagramas de interação.



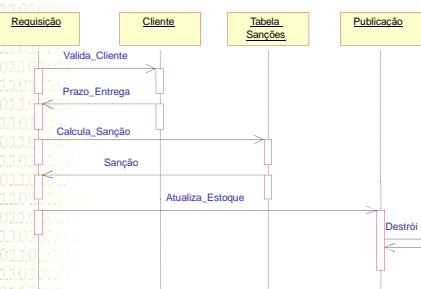
Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Exemplo de Diagrama de Sequência



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

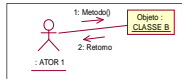
Ana Paula G. Sarco

- 4. Diagramas da UML
- Diagrama de Colaboração
- O diagrama de colaboração mostra de maneira semelhante ao diagrama de sequência, a colaboração dinâmica entre os objetos. Normalmente pode-se escolher entre utilizar o diagrama de colaboração ou o diagrama de sequência.
- No diagrama de colaboração, além de mostrar a troca de mensagens entre os objetos, percebe-se também os objetos com os seus relacionamentos. A interação de mensagens é mostrada em ambos os diagramas.

4. Diagramas da UML

Diagrama de Colaboração

- Se a ênfase do diagrama for o decorrer do tempo, é melhor escolher o diagrama de sequência, mas se a ênfase for o contexto do sistema, é melhor dar prioridade ao diagrama de colaboração.
- Também chamado de **diagrama de comunicação – UML 2.0**
- Geralmente as ferramentas CASE de modelagem geram automaticamente o diagrama de colaboração baseado no diagrama de sequência e vice-versa.



O diagrama de colaboração UML 2.0 ilustra a interação entre um ator e um objeto. O ator, representado por um símbolo de pessoa e rotulado 'ATOR 1', envia uma mensagem '1: Mensagem' para um objeto retangular rotulado 'Objeto : CLASSE B'. Em resposta, o objeto envia uma mensagem de retorno '2: Retorno' de volta ao ator. O diagrama é encerrado por uma caixa retangular com uma borda tracejada.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Exemplo de Diagrama de Colaboração

```
sequenceDiagram
    participant Requisição
    participant Cliente
    participant Tabela_Sanções as Tabela Sanções
    participant Publicação

    Requisição->>Cliente: 1: Valida_Cliente
    Cliente-->>Requisição: 2: Prazo_Entrega
    Requisição->>Tabela_Sanções: 3: Calcula_Sanção
    Tabela_Sanções->>Requisição: 4: Sanção
    Requisição->>Publicação: 5: Atualiza_Estoque
    Publicação->>Publicação: 6: Destrói
```

The diagram illustrates a collaboration between four objects: **Requisição**, **Cliente**, **Tabela Sanções**, and **Publicação**. The interactions are as follows:

- Requisição** sends a message **1: Valida_Cliente** to **Cliente**.
- Cliente** returns a message **2: Prazo_Entrega** to **Requisição**.
- Requisição** sends a message **3: Calcula_Sanção** to **Tabela Sanções**.
- Tabela Sanções** returns a message **4: Sanção** to **Requisição**.
- Requisição** sends a message **5: Atualiza_Estoque** to **Publicação**.
- Publicação** performs a self-message **6: Destrói**.

40

4. Diagramas da UML

Diagrama de Atividades

- Capturam ações e seus resultados.
- Mostram o fluxo sequencial das atividades
- É normalmente utilizado para demonstrar as atividades executadas por uma operação específica do sistema.
- Consistem em estados de ação, que contém a especificação de uma atividade a ser desempenhada por uma operação do sistema.
- Decisões e condições, como execução paralela, também podem ser mostradas na diagrama de atividade.




Diagrama de Atividades exemplo: Um fluxo que começa em um nó inicial, passa por uma ação 'Ação 1', depois uma decisão 'Condição 1'. Se a condição for verdadeira, segue para 'Ação 2'; se falsa, para 'Ação 3'. Ambas as ações convergem para 'Ação 4', que termina em um nó final.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

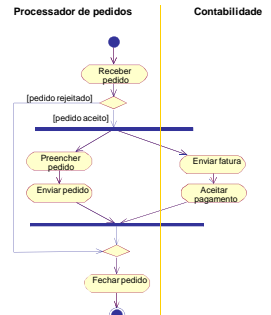
Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

41

4. Diagramas da UML

Exemplo de Diagrama de Atividades



Exemplo de Diagrama de Atividades: O diagrama é dividido em duas swimlanes: 'Processador de pedidos' e 'Contabilidade'. No 'Processador de pedidos', o fluxo começa com 'Receber pedido', seguido de uma decisão '[pedido rejeitado]' ou '[pedido aceito]'. Se aceito, segue para 'Preencher pedido' e 'Enviar pedido'. Na 'Contabilidade', o fluxo recebe 'Enviar fatura' e 'Aceitar pagamento'. Ambos os fluxos convergem para 'Fechar pedido' no 'Processador de pedidos', que termina no nó final.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

42

4. Diagramas da UML

Diagrama de Estados

- É tipicamente um complemento para a descrição das classes.
- Mostra todos os estados possíveis que objetos de uma certa classe podem se encontrar e mostra também quais são os eventos do sistemas que provocam tais mudanças.
- Não são escritos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos e onde o comportamento das classes é afetado e modificado pelos diferentes estados.
- Capturam o ciclo de vida dos objetos, subsistemas e sistemas. Eles mostram os estados que um objeto pode possuir e como os eventos (mensagens recebidas, timer, erros, e condições sendo satisfeitas) afetam estes estados ao passar do tempo.

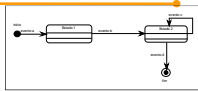


Diagrama de Estados exemplo: Um diagrama de estado com um único estado 'Estado 1' que possui uma transição para si mesmo, representando um estado estacionário.

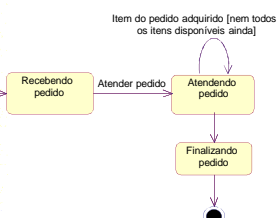
Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

4. Diagramas da UML

Exemplo de Diagrama de Estados



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

5. Ferramentas CASE

Algumas Ferramentas:

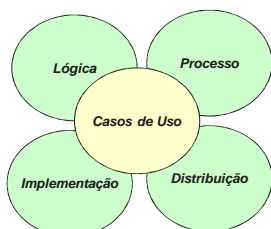
- ❑ Rational Rose (www.ibm.com)
- ❑ OModel (www.omondo.com)
- ❑ ArgoUML (www.argouml.tigris.org)
- ❑ Astah (<http://astah.net/editions/community>)
- ❑ Enterprise Architect (www.sparxsystems.com)
- ❑ Star UML (www.staruml.sourceforge.net)

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula C. Sampaio

6. Visões Arquiteturais



Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Appendix C: Summary

46

6. Visões Arquiteturais

- Visão de Casos de Uso
 - Descreve as funcionalidades do sistema – **o quê**;
 - Facilita o entendimento dos usuários;
 - Base para as demais visões;
 - Representada pelos diagramas de casos de uso e diagrama de atividades (eventual).
- Visão Lógica
 - Descreve **como** as funcionalidades serão fornecidas;
 - Define o comportamento do sistema;
 - Representada pelos diagramas de classes, sequência, estados e atividades.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

47

6. Visões Arquiteturais

- Visão de Processo
 - Descreve a divisão do sistema em processos;
 - Centrado na visão **não funcional** do sistema, como desempenho, segurança e concorrência;
 - Representada pelos diagramas sequência, estados e atividades.
- Visão de Implementação
 - Descreve os módulos de implementação, suas estruturas e dependências;
 - Utilizada pelos desenvolvedores;
 - Representada pelo diagrama de componentes.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

48

6. Visões Arquiteturais

- Visão de Distribuição
 - Exibe a distribuição física do sistema através de nós e suas conexões;
 - Mapeia quais programas e objetos são executados em cada computador;
 - Representada pelo diagrama de distribuição.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra


49

7. Fundamentos de Orientação a Objetos

The diagram illustrates the four fundamental pillars of Object-Oriented Programming (OOP) that support the central concept of 'Orientação a Objetos' (Object-Oriented Programming). The pillars are represented by colored boxes: a green box for 'Abstração' (Abstraction), an orange box for 'Encapsulamento' (Encapsulation), a yellow box for 'Hierarquia - Herança' (Hierarchy - Inheritance), and a purple box for 'Polimorfismo' (Polymorphism). These four pillars are arranged in a row, each with a 3D effect, and they all point upwards towards a large light blue box at the top labeled 'Orientação a Objetos'. A red arrow points from the top left towards the central concept.

Orientação a Objetos

- Abstração
- Encapsulamento
- Hierarquia - Herança
- Polimorfismo




Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu


7. Fundamentos de Orientação a Objetos

Abstração

90




- **Abstração** consiste no ato de se concentrar nas qualidades essenciais ou gerais. Em termos de desenvolvimento de sistemas, isto significa concentrar-se no que um objeto *é* e faz antes de se decidir como ele será implementado.
- O uso de abstração preserva a liberdade para tomar decisões de desenvolvimento ou de implementação apenas quando há um melhor entendimento do problema a ser resolvido.
- Muitas linguagens de programação modernas suportam o conceito de abstração de dados; porém, o uso de abstração juntamente com polimorfismo e herança, como suportado em orientação a objetos, é um mecanismo muito mais poderoso.
- O uso apropriado da abstração permite que um mesmo modelo conceitual (orientação a objetos) seja utilizado para todas as fases de desenvolvimento de um sistema, desde sua análise até sua documentação.


 Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

Modelagem de Sistemas
Orientação a Objetos com UML

Ana Paula G. Serra




USP


7. Fundamentos de Orientação a Objetos

Abstração - Exemplo


51




Estudante




Professor





Modelagem Com UML

(Segunda e Quarta-Feira as 19:00hs)



Curso (exemplo: Álgebra)

Pós-Graduação em Eng. de Software

Universidade São Judas Tadeu

Modelagem de Sistemas

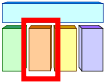
Orientado a Objetos com UML

Ana Paula G. Serra

52

7. Fundamentos de Orientação a Objetos

Encapsulamento



- Encapsulamento é um mecanismo usado para ocultar os dados, a estrutura interna e os detalhes da implementação de algum elemento, como um objeto.
- O uso de encapsulamento evita que um programa torne-se tão interdependente que uma pequena mudança tenha grandes efeitos colaterais.
- Assim como abstração, o conceito de encapsulamento não é exclusivo da abordagem de orientação a objetos. Entretanto, a habilidade de se combinar estrutura de dados e comportamento em uma única entidade torna o encapsulamento mais elegante e mais poderoso do que em linguagens convencionais que separam estruturas de dados e comportamento.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu


Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

53

7. Fundamentos de Orientação a Objetos

Encapsulamento - Exemplo



Professor Pedro

Professor Pedro irá lecionar em 4 turmas no próximo semestre.

SetMaxLoad(4)

AcceptCourseOffering()

SubmitFinalGrade()

SetMaxLoad()

TakeSabbatical()

Nome: Pedro Silva
Identificação: 567138
Contratação: 25/07/1991
Status: Ativo
Disciplina: Finanças
MaxLoad: 4

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu


Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra

54

7. Fundamentos de Orientação a Objetos

Herança

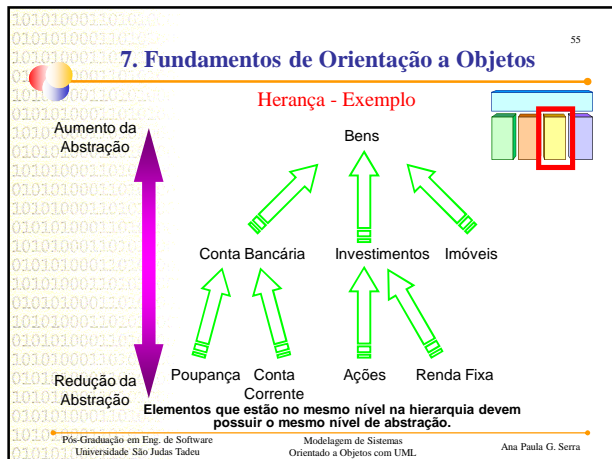


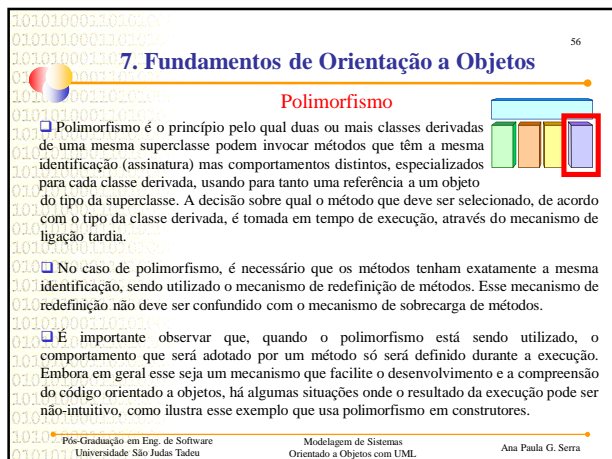
- Generalização e herança são abstrações poderosas para compartilhar similaridades entre classes e ao mesmo tempo preservar suas diferenças.
- A classe sendo refinada é chamada de superclasse ou classe base, enquanto que a versão refinada da classe é chamada uma subclasse ou classe derivada.
- Atributos e operações comuns a um grupo de classes derivadas são colocadas como atributos e operações da classe base, sendo compartilhados por cada classe derivada. Diz-se que cada classe derivada herda as características de sua classe base. Algumas vezes, generalização é chamada de relacionamento is-a (é-um), porque cada instância de uma classe derivada é também uma instância da classe base.
- Generalização e herança são transitivas, isto é, podem ser recursivamente aplicadas a um número arbitrário de níveis. Cada classe derivada não apenas herda todas as características de todos seus ancestrais como também pode acrescentar seus atributos e operações específicos.

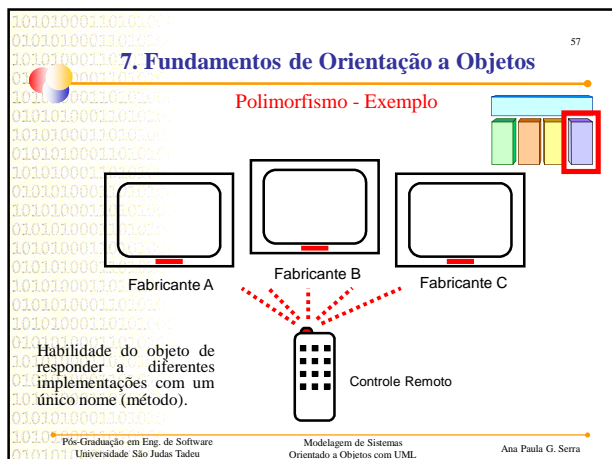
Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra







58

7. Fundamentos de Orientação a Objetos

Polimorfismo - Exemplo

```
Animal[] listaDeAnimais = new Animal[100]; // Criamos um array com 100 posições que armazenam objetos do tipo Animal.
```

```
listaDeAnimais[0] = new Cachorro("Frodo"); // Criamos um novo objeto do tipo Cachorro com o nome Frodo e armazenamos no array.
```


```
listaDeAnimais[1] = new Gato("Alan"); // Criamos um novo objeto do tipo Gato com o nome Alan e armazenamos no array.
```

```
classDiagram
    Animal <|-- Cachorro
    Animal <|-- Gato
    class Animal {
        +fale() void
    }
    class Cachorro {
        +fale() void
    }
    class Gato {
        +fale() void
    }
```

Pós-Graduação em Eng. de Software
Universidade São João Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra




7. Fundamentos de Orientação a Objetos


Polimorfismo - Exemplo

```

1  Classe Animal {
2      Método fale() {imprimaNaTela(" Eu sou mudo! ");}
3
4
5
6
7  Classe Gato {
8      Método fale() {imprimaNaTela(" Miaaaaaauuuuuu! ");}
9
10
11
12
13 Classe Cachorro {
14     Método fale() {imprimaNaTela(" Au au au! ");}
15
16
17
18
19     int cont;
20     para cont de 0 a 100 faça
21         listaDeAnimais[cont].fale();
22     fim para
23 }
                
```




Isto é polimorfismo! Uma mesma mensagem é enviada para diferentes objetos da mesma classe (Animal) e o resultado pode ser diferente, para cada caso.

 Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML


Ana Paula G. Serra



7. Fundamentos de Orientação a Objetos

60

Modularidade




Por exemplo, particionar sistemas complexos em partes menores é mais simples.

Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

Modelagem de Sistemas
 Orientado a Objetos com UML

Ana Paula G. Serra

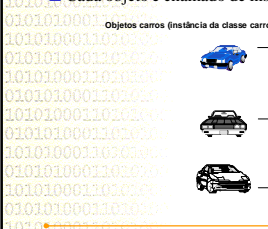


Pós-Graduação em Eng. de Software
 Universidade São Judas Tadeu

7. Fundamentos de Orientação a Objetos

Classe x Objeto

- Um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e comportamento.
- Cada objeto é chamado de instância de sua classe.



```

            graph LR
                subgraph "Objetos caros (instância da classe carro)"
                    direction TB
                    O1[Carro Azul]
                    O2[Carro Preto]
                    O3[Carro Prateado]
                end
                O1 --- C[Classe CARRO]
                O2 --- C
                O3 --- C
                subgraph C [Classe CARRO]
                    direction TB
                    A[Atributos]
                    B[Marca]
                    C[Modelo]
                    D[Ano]
                    E[Cor]
                    F[Chassi]
                    G[Operações]
                    H[Liga]
                    I[Desliga]
                    J[Anda]
                    K[Para]
                end
            
```

Modelagem de Sistemas Orientado a Objetos com UML

Ana Paula G. Serra

62

7. Fundamentos de Orientação a Objetos

Outros Conceitos...

- ❑ **Atributo:** característica ou propriedade nomeada de uma classe.
- ❑ **Operação:** é uma ação ou transformação realizada por um objeto ou sofrida por um objeto.
- ❑ **Propriedade:** é a implementação de um atributo para uma classe.
- ❑ **Método:** é a implementação de uma operação para uma classe com algoritmo específico.
- ❑ **Superclasse/Subclasse:** Ao evidenciar os atributos e as operações de um conjunto de classes, identifica-se uma superclasse (generalização) refinada em subclasses (especialização). Uma subclasse herda atributos e operações das superclasses e possui seus atributos e operações específicas.

Ana Paula G. Serra

63

Referências Bibliográficas

- [] UML – Guia do Usuário. 2ª. ed. Grady Booch; James Rumbaugh; Ivar Jacobson. Editora Campus. 2006. ISBN: 10-85-352-1784-3.
- [] Larman, Craig; Utilizando UML e Padrões, 2a. Edição. Bookman, 2003. ISBN: 85-363-0358-1.
- [] UML 2 Toolkit. Hans-Erik Eriksson; Magnus Penker; Brian Lyons; David Fado. Wiley. 2004. ISBN: 0-471-46361-2.

Pós-Graduação em Eng. de Software
Universidade São Judas Tadeu

Modelagem de Sistemas
Orientado a Objetos com UML

Ana Paula G. Serra
