

ANÁLISE E PROJETO ORIENTADO A OBJETOS

Professores:

Ana Paula Gonçalves Serra

André Luiz Ribeiro

Keity Yamamoto

Modelagem Orientada a Objetos

1. Conceito de Orientação 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 (UML, OMT, Booch, etc) são regras, conceitos e representações gráficas do sistema. E no desenvolvimento de sistemas através de linguagens de programação orientada a objetos (Java, C++, Delphi, etc).

A Orientação a Objetos é uma metodologia de desenvolvimento de software.

2. Objetos e Classes

No dicionário pode-se encontrar as seguintes definições de objetos:

- Tudo que é manipulável e/ou manufaturável;
- Tudo que é perceptível por qualquer dos sentidos;
- Coisa, peça, artigo de compra e venda;
- Matéria, assunto, Agente; motivo, causa.

Pelo conceito de Orientação a Objeto, pode-se definir objeto como uma abstração, com limites e significados bem definidos, em relação ao problema considerado. Cada objeto tem sua própria identidade, que lhe é inerente.

Exemplos:



Carro Zimba



Computador do
LAB SIM I



Sofá da sala
de estar do João

Objetos com a mesma característica (atributos) e mesmo comportamento (operações ou métodos) são agrupados em uma classe. Cada classe descreve um conjunto possivelmente infinito de objetos individuais. Cada objeto é chamado de instância de sua classe.

Objetos carros (instância da classe carro)

Classe CARRO



Atributos

Marca

Modelo

Ano

Cor

Chassi





Operações

Liga

Desliga

Anda

Para

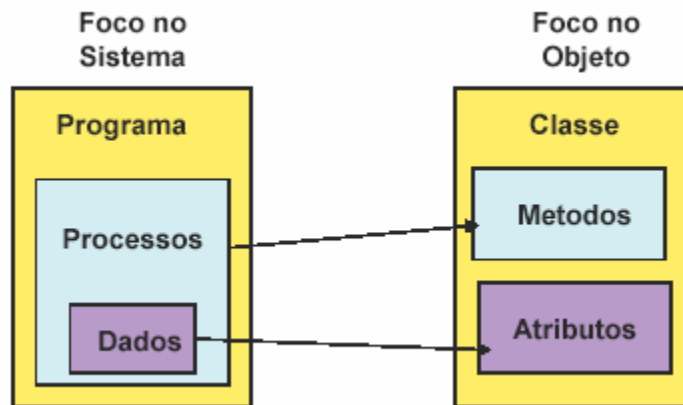
3. Metodologia Estruturada X Metodologia Orientada a Objetos

Na metodologia estruturada o foco é na especificação e decomposição da funcionalidade do sistema enquanto que na metodologia orientada a objetos, primeiro identificam-se os objetos contidos no domínio do sistema e depois os procedimentos relativos a ele.

Em resumo:

- Metodologia estruturada = o foco é na especificação e decomposição da funcionalidade do sistema

- Metodologia orientada a objetos = primeiro identificam-se os objetos contidos no domínio do sistema e depois os procedimentos relativos a ele.



Exemplos:

- 1) De Grady Booch; Object-Oriented Analysis e Design with Applications 1994.



2) Jogo de Futebol

- Análise de um campo de futebol utilizando metodologia estruturada
 - Passe da bola;
 - Fazer gol;
 - Cobrar lateral;

- Cobrar tiro de meta;
 - Driblar o jogador;
 - Cobrar falta;
 - Marcar penault;
 - Passar a bola.
- Análise de um campo de futebol utilizando metodologia orientada a objetos
 - Jogador;
 - Juiz;
 - Bola;
 - Campo.

4. Alguns Conceitos Utilizados na Orientação a Objetos

Durante o curso conceitos como abstração, atributo, entre outros serão utilizados para a elaboração dos modelos Orientados a Objetos. Para uma visão inicial e geral são apresentados esses conceitos a seguir.

- **Abstração:** o termo abstração significa ressaltar os aspectos essenciais de uma entidade e ignorar os aspectos não relevantes para o enfoque considerado;
- **Atributo:** característica da classe que pode ter valor diferenciado para cada objeto instanciado;
- **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;
- **Polimorfismo:** é a implementação de uma mesma operação em diversas classes. Através do polimorfismo, a mesma operação se comporta de forma distinta em classes distintas. Exemplo: operação incluir, pode possuir métodos distintos, como: incluir cliente, incluir fornecedor, ...
- **Herança:** é o compartilhamento de atributos e operações entre as classes que possuem relacionamento de hierarquia. Exemplo: um cliente pode ser pessoa física ou pessoa jurídica;
- **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;
- **Encapsulamento:** é um pacote de atributos e operações o qual representa um estado em um tipo de objeto, de tal forma que o estado é acessível ou modificado somente pela interface provida pelo encapsulamento.
- **Cardinalidade:** quantidade de instâncias determinada para o relacionamento entre duas classes. Por exemplo: 1 para n, 1 para 1, m para n.

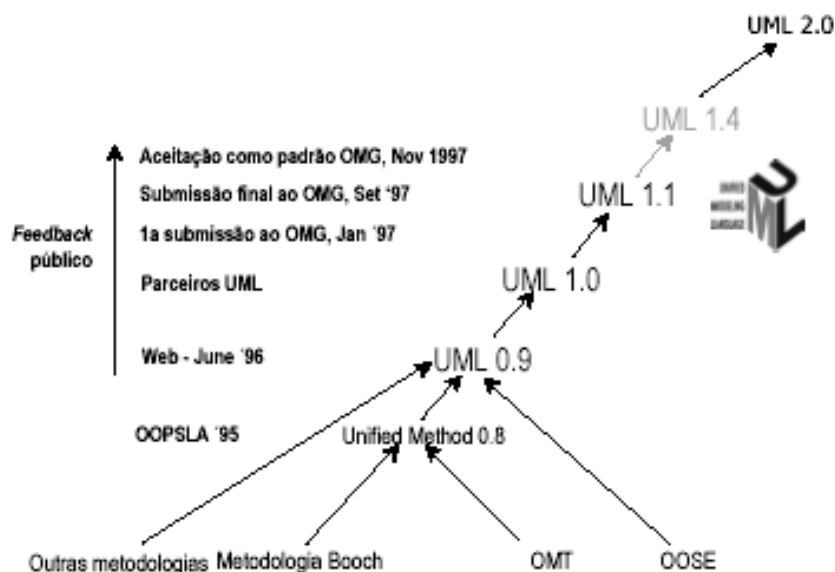
5. Notação UML

A UML (*Unified Modeling Language*) é uma linguagem padrão para a elaboração da estrutura de projetos de software, que surgiu em meados da década de 90 visando unificar as diversas notações de modelagem orientadas a objetos, como por exemplo: OMT (*Object Modeling Technique* – Rumbaugh), OOSE (*Object-Oriented Software Engineering* – Jacobson), Booch, entre outras. Cada uma dessas notações citadas possui suas qualidades e deficiências, o OMT é forte em análise e mais fraco em projeto, Booch é forte em projeto e mais fraco em análise, Jacobson é forte em comportamento de análise e mais fraco em outras áreas. Com isso, a UML é uma notação unificada e padrão de todas as notações mais utilizadas na modelagem de sistemas.

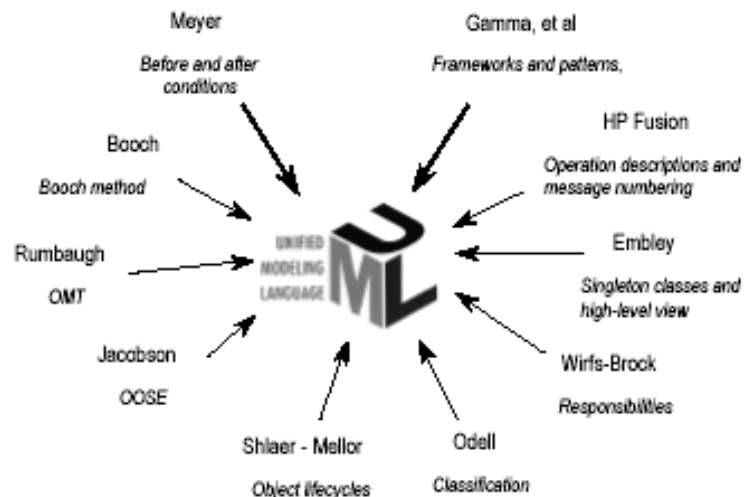
Pode-se definir UML como uma linguagem de modelagem usada para especificar, visualizar, construir e documentar sistemas orientados a objetos.

Durante o curso o UML será utilizado como notação principal para a metodologia orientada a objetos em conjunto com a ferramenta Rational Rose.

5.1. Evolução da UML



5.2. Contribuições:



5.3. O que é a UML ?

- Linguagem visual para a especificação de sistemas orientados a objetos;
- Visa:
 - Visualizar
 - Especificar
 - Construir
 - Documentar
- É independente de:
 - Fases de desenvolvimento de software
 - Processo de software utilizado
 - Linguagens de programação

5.4. Objetivos da UML

- Modelar sistemas utilizando conceitos de Orientação a Objetos;
- Estabelecer uma clara ligação entre os modelos conceituais e de implementação;
- Explicitar os pontos notáveis em sistemas complexos ou de missão crítica;
- Definir uma linguagem de modelagem passível de uso por pessoas e máquinas.

5.5. Aspectos da UML

- Funcional
 - Casos de Uso

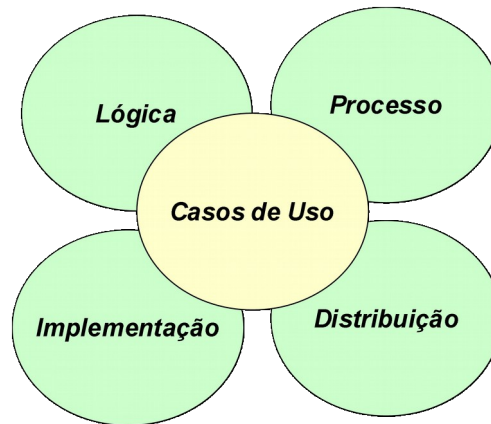
- Estático (Estrutural)
 - Diagrama de Classes
- Dinâmico (Comportamental)
 - Diagrama de Seqüência, Atividades e Estados

6. Visão Geral dos Modelos da UML

- **Diagrama de Casos de Uso:** descreve as funcionalidades do sistema e os usuários e entidades externas, organizando o comportamento do sistema. Além do diagrama há toda a descrição de atores e casos de uso.
- **Diagrama de Classes:** descreve a estrutura de solução do sistema, através de um conjunto de classes (compostas de atributos e operações), e relacionamentos. Geralmente é dividido em diagrama de classes de análise (domínio) e diagrama de classes de projeto (implementação).
- **Diagrama de Objetos:** descreve um conjunto de objetos e seus relacionamentos. Esse diagrama ilustra a estrutura de dados e instâncias do diagrama de classes.
- **Diagrama de Seqüência:** faz parte do conjunto de diagramas de interação, descreve o comportamento do sistema, enfatizando a comunicação dos objetos através da passagem de mensagem entre os mesmos;
- **Diagrama de Colaboração ou Comunicação (na UML 2.0):** faz parte do conjunto de diagramas de interação, descreve o comportamento do sistema, enfatiza a organização estrutural dos objetos que enviam e recebem mensagens;
- **Diagrama de Atividades:** descreve o comportamento do sistema, através do fluxo de controle de funções.
- **Diagrama de Estados:** descreve o comportamento do sistema, enfatizando os estados que o objeto pode possuir.
- **Diagrama de Componentes:** descreve os componentes que irão ser criados no sistema e a comunicação entre eles;
- **Diagrama de Implantação/Distribuição:** descreve a arquitetura física e os componentes utilizados no sistema.
- **Diagrama de Pacotes (na UML 2.0):** fornece um mecanismo de organização para os elementos da UML. O pacote é utilizado para agrupar elementos da modelagem.
- **Diagrama de Interação – Visão Geral (na UML 2.0):** faz parte do conjunto de diagrama de interação, apresenta a visão geral, de congregação, dos outros diagramas de interação (seqüência, comunicação e tempo).
- **Diagrama de Tempo (na UML 2.0):** faz parte do conjunto de diagrama de interação, descreve o comportamento de um ou mais objetos em dado período de tempo, mostrando suas alterações de estados.

- **Diagrama de Composição de Estrutura (na UML 2.0):** descreve a composição de diversos elementos de modelagem, como: interfaces, objetos ou classes, mas que não perdem suas características em combinação com outras.

7. 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, seqüência, estados e atividades.

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 seqüência, estados e atividades.

Visão de Implementação

- Descreve os módulos de implementação, suas estruturas e dependências;
- Utilizada pelos desenvolvedores;

- Representada pelo diagrama de componentes.

Visão de Implantação/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.

8. Algumas Ferramentas:

- Rational Modeler / Rational Rose
- OMondo
- ArgoUML
- Astah
- Enterprise Architect
- StarUML

Copyright © 2012 - 2017 Profa. Ana Paula Gonçalves Serra, Prof. André Luiz Ribeiro e Prof. Keity Yamamoto. Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, da Profa. Ana Paula Gonçalves Serra, Prof. André Luiz Ribeiro e Prof. Keity Yamamoto.