Unidade 2

1. Modelagem de sistemas e Paradigma de Orientação a Objetos

A disciplina terá um enfoque mais semântico do que sintático, não pretendendo priorizar os seus aspectos notacionais em detrimento dos conceitos subjacentes.

Conceito: Um **Sistema de Informação** consiste em uma combinação de pessoas, dados, processos, interfaces, redes de comunicação e outras tecnologias que interagem com o objetivo de dar suporte e melhorar o processo de negócio de uma organização, com ênfase na informação que ela utiliza e que ela gera.

Expressões associadas:

> "caráter estratégico da informação no mundo atual",
>
> "globalização",
>
> "vantagem competitiva",
>
> "adição de valor" (custo - $ investido nos diversos tipos de recurso X benefício – quantitativo X qualitativo → $)

Visões sobre a Tecnologia:

> necessidade para a sobrevivência organizacional (o que acaba fazendo-a parecer um fim em si mesma) X mais um instrumento ao serviço da organização;
>
> uso "burro" X uso inteligente;
>
> automatização X inovação

2.1 Modelagem de sistemas de software

Uma das características intrínsecas aos sistemas de software é a complexidade de seu desenvolvimento, que aumenta à medida que o sistema cresce e, também, com o aumento do seu escopo de uso.

Para a construção de sistemas de software, é necessário um planejamento análogo àquele normalmente realizado na construção civil.

Conceito: Um **modelo** é pode ser visto como uma representação idealizada de um sistema a ser construído.

Exemplos: maquetes de edifícios e de aviões, plantas de circuitos eletrônicos,...

Motivações para a construção de modelos:

- **gerenciamento da complexidade:** Dada a dificuldade humana em lidar com problemas complexos, a construção de modelos de um mesmo produto correspondente a diversas visões (partes elétrica e aerodinâmica de um avião) torna a complexidade passível de tratamento. Os modelos seguem o **princípio da abstração**, que faz com que a preocupação se restrinja aos aspectos realmente relevantes do problema, às características essenciais da solução;
- **comunicação entre as pessoas envolvidas:** O desenvolvimento de sistemas envolve uma série de atividades. Essas atividades se traduzem em informação

sobre o sistema em desenvolvimento, grande parte da qual compõe os modelos do sistema. Os modelos agem como veículo de comunicação entre os indivíduos envolvidos no desenvolvimento;

- **controle de erros e redução dos custos de desenvolvimento:** A ocorrência de erros e o custo da sua correção são significativamente diminuídos na abordagem da construção de modelos, pois eles são detectados e corrigidos antes da construção propriamente dita do produto. Além disso, as diversas visões do software expostas pelos diferentes modelos torna explícitas as divergências em tempo de projeto;
- **previsão do comportamento futuro do sistema:** A análise dos modelos permite a previsão do comportamento do sistema, servindo como "laboratório", em que diferentes soluções podem ser experimentadas.

As formas que os modelos adotam podem ser variadas.

Conceito: Modelos gráficos que seguem algum padrão notacional são chamados **diagramas**. Diagramas apresentações de conjuntos de elementos gráficos que possuem significados pré-definidos.

Motivação e mito: "Uma figura vale por mil palavras." Texto (preciso) também é necessário!

Conceito: Os diagramas, juntamente com a informação textual, formam a **documentação do sistema**.

Resumo: A modelagem de sistemas consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representem as partes essenciais de um sistema, considerando-se várias perspectivas diferentes e complementares.

2. O paradigma da Orientação a Objetos

Conceito: Um **paradigma** é uma realização científica que gera modelos que, por período mais ou menos longo e de modo mais ou menos explícito, orientam o desenvolvimento posterior das pesquisas na busca da solução para os problemas por ela suscitados.

Conceito (informal): De maneira informal, um paradigma pode ser visto como uma forma de abordar um problema, uma **abordagem**.

Conceito: **O paradigma de Orientação a Objetos** é uma forma de abordar a análise e o projeto de sistemas computacionais.

Analogia com a Biologia: Alan Kay, um dos autores do paradigma OO formulou a analogia com a Biologia.

- Ele imaginou um sistema de software como um ser vivo;
- Nesse sistema, cada "célula" interagiria com outras células por meio de mensagens para realizar um objetivo de interesse comum;
- Cada célula se comportaria de maneira autônoma.

De uma forma geral, Kay estabeleceu os seguintes princípios da OO:

- Qualquer coisa pode ser um **objeto**;

- Objetos realizam tarefas por meio da requisição de **serviços** a outros objetos;
- Cada objeto pertence a uma determinada **classe**. Uma classe agrupa objetos semelhantes;
- A classe é um repositório para o comportamento associado ao objeto;
- Classes são organizadas em hierarquias.

Analogia com atividades do mundo real: pedido e entrega de uma pizza

- João quer uma pizza;
- José solicita a pizza ao atendente (José);
- José pede para Maria fazer a pizza;
- Maria faz a pizza e a entrega a José, que a entrega a Antônio;
- João recebe a pizza das mãos de Antônio, o entregador.

É possível observar, na analogia:

- O objetivo do João foi atingido graças à colaboração de diversos agentes, os funcionários da pizzaria;
- Na terminologia da OO, esses agente são chamados de objetos;
- Há diversos objetos na história, João, Maria, José e Antônio (princípio 1);
- Cada um deles colabora com uma parte e o objetivo é alcançado (princípio 2);
- O comportamento esperado do Antônio é o mesmo esperado de qualquer entregador. Dizemos que Antônio é um objeto da classe Entregador (princípio 3);
- Um comportamento comum a todo entregador é o de entregar mercadoria no endereço especificado (princípio 4);
- Finalmente, José, o atendente, como todos os outros indivíduos envolvidos na atividade, é um ser humano, um mamífero, um animal,... (princípio 5).

Antes do aparecimento do paradigma de OO, era utilizado o paradigma da **Análise Estruturada**.

Na Análise Estruturada, os elementos são **dados** e **processos**.

Resumo: O paradigma da OO enxerga um sistema de software como uma coleção de agentes interconectados chamados **objetos**. Cada objeto é responsável por realizar tarefas específicas. É por meio da interação entre objetos que as tarefas são realizadas.

Hipótese: Acredita-se que a OO, como abordagem para a modelagem de sistemas, diminui a diferença semântica entre a realidade modelada e os modelos construídos.

3. Objetos e classes

Conceito: O mundo real é formado de coisas, concretas e abstratas. No paradigma OO, ao se fazer a transposição da realidade para os modelos, as coisas são denominadas **objetos**.

Exemplos: cliente, loja, produto, venda, compra, fornecedor.

Conceitos: O ser humano costuma organizar estes objetos para apreender a complexidade do mundo. É bem mais fácil entender o que significa "cavalo" do que entender todos os cavalos que existem. Na OO, uma idéia ou conceito de uma coisa é chamada de **classe**. Uma classe é a descrição de **atributos** e **serviços** comuns a um grupo de objetos. Diz-se que um objeto é uma **instância** de uma classe.

Exemplos: Se considerarmos "pessoa" como uma classe, uma objeto ou instância dessa classe não será "uma pessoa qualquer", mas, sim, "uma pessoa em particular", com nome e sobrenome (por exemplo, qualquer aluno da turma que a gente selecionar).

Conceitos: A rigor, uma classe é uma **abstração** das características de um conjunto de coisas do mundo real. A abstração é um conceito associado aos conceitos de **modelagem** e de **representação**, conceitos que freqüentemente implicam restrições e/ou simplificações da realidade considerada.

Exemplo: Certamente podemos nos deparar com um cavalo de 3 patas. No entanto, provavelmente, o nosso modelo mental de um cavalo nos permitirá identificar o animal corretamente.

O princípio da abstração permite que os problemas complexas sejam passíveis de tratamento (solução). Ao se modelar um sistema, selecionam-se os aspectos relevantes ao problema e à sua solução.

4. Operações e mensagens

Conceitos: Uma **operação** é alguma ação que um objeto sabe realizar quando solicitado. A execução de uma operação é determinada pelo envio de um estímulo ao objeto. Diz-se que o objeto recebe uma **mensagem** solicitando a execução de uma operação.

No contexto de sistemas OO, dizer que "os objetos estão trocando mensagens" significa dizer que eles estão se comunicando com o objetivo de realizar alguma tarefa (operação) do sistema onde estão inseridos.

5. Aplicações da abstração na Orientação a Objetos

Conceito: A **abstração** é um processo mental pelo qual os seres humanos nos atemos aos aspectos mais importantes das coisas e ignoramos os menos relevantes. Este processo mental permite tratar a complexidade do mundo.

Conceito: É crucial notar que a relevância de um objeto no mundo real depende das circunstâncias, da situação, do **contexto**. Aspectos relevantes num contexto podem não ser importantes em outro. Aspectos relegados num problema podem ser extremamente relevantes em outra situação.

5.1 Encapsulamento

Conceitos: O mecanismo de **encapsulamento** é uma forma de restringir o acesso ao comportamento interno de um objeto. Um objeto que necessite da colaboração de outro objeto para realizar alguma operação simplesmente envia uma mensagem a este último. Segundo o princípio de encapsulamento, o **método** que um objeto utiliza para realizar a operação não é conhecido pelos objetos requisitantes. Em outras palavras, o objeto remetente da mensagem não precisa conhecer a forma como o objeto destino irá executar a operação; tudo o que importa ao primeiro é ter a operação realizada.

Conceitos: Os objetos necessitam conhecer quais as operações que os outros objetos sabem realizar e que informação eles detêm. Para isso, a classe de um objeto descreve o seu comportamento. Na OO, a **interface** de um objeto corresponde à descrição do que o objeto conhece e do que ele sabe fazer. Mas esta descrição não informa como o objeto conhece a informação ou como ele realiza a operação. Se enxergarmos um objeto como um provedor de **serviço**s, a interface determina quais os serviços que ele pode fornecer. A implementação dos serviços pode ser feita de variadas formas, mas isso não interessa a quem o requisita.

O encapsulamento permite que a implementação de uma operação possa ser modificada ou mesmo substituída sem que os objetos requisitantes da mesma precisem ser alterados. Isso faz com a a propagação de mudanças seja menor. Com o software cada vez mais complexo, a independência dos módulos componentes de um sistema em relação aos outros é extremamente relevante. Tudo isto leva, também, a uma maior produtividade no desenvolvimento, na medida em que para utilizarmos um objeto não precisamos entender como ele funciona.

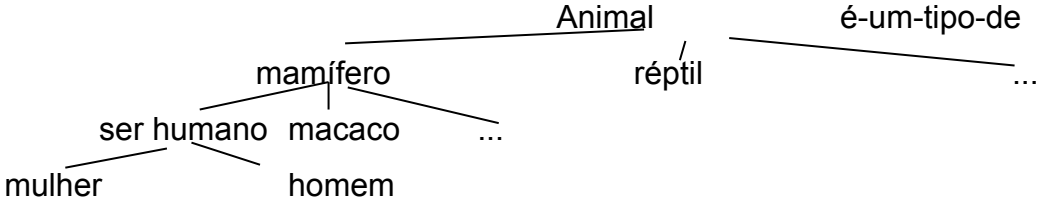
Ao esconder os detalhes do funcionamento interno dos objetos, está sendo aplicado o conceito de abstração.

5.2 Generalização (Herança)

Conceito: As características (atributos) e o comportamento (operações) comuns a um conjunto de objetos podem ser abstraídos numa classe. A **generalização** (ou **herança**) pode ser vista como a extensão da abstração para cima daquela encontrada entre objetos e classes.

Na generalização, classes semelhantes são reagrupadas em uma hierarquia. Cada nível dessa hierarquia pode ser visto como um nível de abstração.

Cada classe num dado nível da hierarquia **herda** as características e o comportamento das classes superiores às quais está associada. Além disto, cada classe pode ter características e comportamento particulares. Assim, uma classe pode ser criada por meio do **reuso** de classes pré-existentes.



5.3 Polimorfismo

Conceito: O **polimorfismo** se refere à capacidade de abstrair diferentes implementações numa mesma interface. No contexto da OO, o polimorfismo diz respeito à capacidade de duas ou mais classes de objetos responderem a uma mesma mensagem, cada uma do seu modo.

Exemplo: O exemplo clássico de polimorfismo no desenvolvimento de software é o das

formas geométricas. Imagine uma coleção de formas geométricas que inclua círculo, quadrado, retângulo, losango e outras formas específicas. Pelo princípio do polimorfismo, quando uma região de código precisa desenhar os elementos daquela coleção, não é necessário conhecer os tipos específicos das figuras existentes, basta que cada elemento da coleção receba uma mensagem solicitando que desenhe a si próprio.

É importante notar que isto simplifica o código do objeto cliente (o que solicitou o desenho das figuras), na medida em que ela não precisa conhecer o tipo de cada figura. Ao mesmo tempo, essa região de código não precisa ser alterada quando, por exemplo, é incluída uma nova forma geométrica na coleção.

Esta é uma nova forma de aplicação do princípio da abstração, pois um objeto pode enviar a mesma mensagem para objetos semelhantes, que implementam a sua interface de formas diferentes.

5.4 Composição

Conceito: No mundo real, é comum pensarmos em objetos como coisas compostas por outros objetos. Este é o princípio da **composição**. A composição permite que criemos objetos a partir da reunião de outros objetos.

Exemplos: Livros são compostos de páginas. Páginas são compostas de parágrafos. Parágrafos são compostos de frases, e assim por diante. Um televisor antigo é formado por um painel de controle, uma tela, um tubo de imagem,...

6 Linguagem de Modelagem Unificada (Unified Modelling Language – UML)

A Linguagem de Modelagem Unificada (daqui para a frente chamada de UML) foi definida na busca do aproveitamento do melhor das características das notações pré-existentes na Engenharia de Software.

Atualmente, a notação está na sua versão 2.0 de 2003.

Conceito: A UML é uma linguagem visual para modelar sistemas OO. Ela define elementos gráficos que representam os conceitos do paradigma de OO e podem ser utilizados na modelagem de sistemas, determinando diversas perspectivas de um sistema.

Cada elemento gráfico da UML possui uma sintaxe e uma semântica associadas. A sintaxe corresponde à forma de desenhar o elemento. A semântica determina o significado do elemento e o objetivo da sua utilização. A sintaxe e a semântica dos elementos UML são extensíveis. Essa extensibilidade permite que a UML seja adaptada às características específicas de cada projeto.

A UML é independente tanto de linguagens de programação quanto de processos de desenvolvimento. Este fator torna a notação bem popular entre desenvolvedores, na medida em que ela se aplica ao desenvolvimento de sistemas bem diversos.

A definição completa da UML está contida na Especificação da Linguagem de Modelagem Unificada na OMG e pode ser obtida gratuitamente do site www.uml.org. No entanto, ela não é de fácil leitura pois é direcionada principalmente a pesquisadores e desenvolvedores de ferramentas de suporte ao desenvolvimento de sistemas (ferramentas CASE).

Conceito: Um processo de desenvolvimento que utilize a UML como linguagem de suporte à modelagem envolve a criação de diversos documentos. Cada um destes documentos pode ser textual ou gráfico. Na modelagem UML estes documentos são chamados de **artefatos**.