A Programação Orientada a Objetos

D*esde que o homem passou a usar máquinas de computação, surgiu a necessidade de repassar as instruções e os dados para a obtenção da solução esperada. Simultaneamente, de acordo com a evolução tecnológica, aumentou também o grau de dificuldade dos problemas propostos. Veremos neste trabalho um pouco do histórico da evolução das metodologias de desenvolvimento de sistemas e alguns dos conceitos básicos de uma técnica que vem sendo amplamente utilizada: a orientação a objetos.*

Técnicas Estruturadas

Durante muitos anos os computadores foram utilizados somente por grandes empresas. Até que no princípio da década de 70 houve uma queda no preço dos equipamentos de informática e algumas empresas de médio e pequeno porte puderam se aventurar em transferir para os sistemas informatizados algumas funções de caráter operacional.

Todo o conhecimento que se tinha até então de técnicas de desenvolvimento de software não era suficiente para contornar os problemas de desenvolvimento de sistemas, principalmente se produzidos em grande escala, como passou a se exigir com a demanda de um público consumidor de programas.

E desta necessidade surgiu uma técnica que até hoje é bastante utilizada e difundida chamada de *programação estruturada*, seguida pelo conceito de *desenvolvimento estruturado de sistemas*. Como uma alternativa para sanar as dificuldades de um desenvolvimento em grande escala, a metodologia estruturada pregava alguns princípios:

Abstração: a solução de um problema pode ser encontrada mais facilmente se o mesmo for analisado de forma a separar os demais aspectos que possam atrapalhar numa etapa (relevar os detalhes não necessariamente importantes);

Formalidade: deve ser seguido um caminho rigoroso e metódico para solucionar um problema;

Dividir para conquistar: dividir o problema em partes menores, independentes e com possibilidade de serem mais simples de entender e solucionar;

Hierarquização: os componentes da solução devem ficar dispostos em uma estrutura hierárquica. O sistema deve ser entendido e construído nível a nível, onde cada novo nível acrescenta mais detalhes.

Com estes princípios que facilitavam a vida dos desenvolvedores, estas técnicas tiveram grande sucesso e ainda hoje são amplamente utilizadas.

Um novo ambiente

A revolução ocorrida na década de 70 voltou a ocorrer no final da década de 90, onde os preços dos equipamentos voltaram a cair. Um bom número de empresas, de diversos portes, está com parte de seus sistemas com um considerável nível de informatização, foi amplamente divulgado o uso da Internet como meio de comunicação e busca maciça de informação e finalmente o computador passou a ser um eletrodoméstico dos indivíduos de classe média e uma ferramenta de trabalho diário para uma grande quantidade de pessoas.

Surge então a necessidade de se produzir softwares mais atraentes, dinâmicos e com alto poder de troca de informações. Tais aplicações se caracterizam por:

- Grande interação com o usuário;
- Uso de interfaces gráficas (GUI-Graphics User Interface) como o Windows;
- Necessidade permanente de alteração e expansão, dada a velocidade de mudanças na tecnologia do hardware;
- Interação com outros sistemas, possibilitando a troca de dados entre estes;
- Portabilidade para diversas plataformas e sistemas operacionais;

As técnicas oferecidas pela metodologia estruturada não eram suficientes para atender com satisfação desejada a elaboração deste tipo de aplicação. Era necessário partir para outro tipo de metodologia, que permitisse o desenvolvimento de sistemas com estas novas características. A técnica que começou a ser adotada por parte dos profissionais da área de desenvolvimento de sistemas foi a da *metodologia orientada a objetos*.

Orientação a Objetos

Apesar de não ser um conceito totalmente novo no meio acadêmico, somente nos últimos anos a Orientação a Objetos vem ganhando força também no mercado de software. Basta citar que as grandes empresas da área de desenvolvimento de sistemas oferecem ferramentas se não totalmente fundamentadas, pelo menos baseadas nos conceitos de objetos.

A tecnologia orientada a objetos é fundamentada no que coletivamente chamamos de modelo de objetos, que engloba os princípios da abstração, hierarquização, encapsulamento, classificação, modularização, relacionamento, simultaneidade e persistência. Estes conceitos não são novos, mas o que é importante no modelo de objetos é que estes elementos estão agora agrupados de forma sinérgica.

Como a abstração e a hierarquização já foram detalhados na parte da metodologia estruturada, vamos analisar os outros princípios:

Encapsulamento: mecanismo pelo qual podemos ocultar detalhes de uma estrutura complexa, que poderiam interferir durante o processo de análise. Como um exemplo, todos nós sabemos que um carro é composto de motor, lataria, bancos, etc. O motor por sua vez é composto por uma grande quantidade de peças e elementos de ligações. Mas na hora de preencher um cadastro financeiro para solicitação de um empréstimo pessoal, nos campos para colocar os detalhes do carro, não será necessário dizer quais são as peças que formam o motor do carro, apesar de sabermos que sem estas, o valor do carro cai bastante. Estes detalhes serão encapsulados pelas características gerais do carro.

Classificação: é o ato de associar um objeto analisado a uma determinada categoria. Ao classificarmos um objeto, estamos afirmando que este pertence a uma determinada classe. Esta associação é feita comparando as características e funções do objeto em questão com as características e funções dos objetos que pertencem àquela categoria. Como exemplo, quando vemos um gato podemos afirmar que ele pertence à classe dos mamíferos, porque conhecendo as características e funções dos mamíferos e dos gatos, fica fácil chegar a esta conclusão.

Modularização: em um sistema previamente dividido, podemos juntar partes com algumas semelhanças. Note que a idéia de modularizar facilita bastante a aplicação dos outros princípios.

Relacionamento: para o funcionamento do todo, é necessário que as partes funcionem separadamente, mas em conjunto. Esta cooperação é possível porque as partes se relacionam entre si.

Paralelismo: mesmo em um sistema simples, podem haver diversos ações a serem executadas em tempos quase que simultâneos. É necessário um correto gerenciamento dos recursos computacionais para haver uma correta distribuição do tempo entre as tarefas a serem executadas.

Persistência: um objeto em um programa utiliza algum espaço para armazenamento e manipulação e existe por um período de tempo em particular. Geralmente o tempo de vida de um objeto supera o tempo de vida de um sistema que o manipula. Este princípio prega que deve haver uma atenção especial nesta manipulação.

Como pudemos perceber, alguns destes princípios devem trabalhar em conjunto, pois um pode ajudar diretamente os outros e facilitar o entendimento geral da solução do problema.

Classes e objetos

Para entendermos o conceito de *classe*, vamos começar vendo o conceito de *objeto*.

Usamos o termo objeto para representar um determinado elemento do mundo real. Mas somente analisaremos os objetos que tem relevância para a solução de um determinado problema. Portanto, o objeto é uma entidade do mundo real que merece representação para o ambiente estudado.

Como exemplos de objetos, podemos citar os objetos físicos (um livro, uma mercadoria), funções de pessoas para os sistemas (cliente, vendedor), eventos (uma compra, um telefonema), interações entre outros objetos (um item de uma nota fiscal é uma interação entre uma compra e um produto do estoque) e lugares (loja matriz, revenda norte).

Um objeto pode ser simples ou composto de demais objetos. Em geral a maioria dos objetos são compostos, pois sempre podemos dividi-los em partes menores até chegarmos a elementos realmente atômicos (indivisíveis).

Um sistema é um grande objeto composto de outros objetos, formando um mecanismo.

Um objeto é composto de atributos e métodos e tem a capacidade de trocar mensagens com outros objetos.

Os atributos, ou propriedades, representam as características dos objetos e podem ser fixos ou variáveis. Como exemplo, o objeto computador tem como características a marca, o modelo, a quantidade de memória RAM, o tamanho do disco rígido, se tem ou não CD-ROM, etc.

O estado de um objeto é o conjunto de valores de seus atributos em um determinado instante.

Os serviços ou métodos são as funções que operam sobre o mesmo.

O comportamento de um objeto é como ele age e reage em termos de suas mudanças de estado e troca de mensagens com outros objetos.

A identidade é a característica que um objeto deve ter de ser distinguido dos outros objetos.

Agora que já conhecemos melhor o que seria um objeto, vamos analisar o conceito de classe.

Uma classe representa um conjunto de objetos que possuem características e comportamentos comuns e de agora em diante, diremos que um objeto é uma instância de uma determinada classe, ou seja, criaremos nossos objetos baseados nas características definidas nas classes.

A ênfase da metodologia orientada a objetos é dada na criação das classes, e não dos objetos, como se poderia pensar pelo nome.

Todo objeto pertence a uma determinada classe durante sua existência, não podendo modificar sua classificação.

Classe Computador

Atributos

- Marca
- Modelo
- Quantidade de memória RAM

- Tamanho do disco rígido
- Possui CD-ROM?

Métodos

- Receber dados
- Processar informações
- Enviar resultados para impressora

Relacionamento entre classes

As classes devem poder se relacionar para que o sistema possa funcionar. Como formas de relacionamento podemos citar:

Associação: um exemplo típico de associação é a feita entre as classes Aluno e Responsável de um sistema de controle acadêmico. Um objeto da classe Aluno está associado a apenas um único responsável, mas uma mesma pessoa pode ser responsável por mais de um aluno. Podem haver relacionamentos de cardinalidades um-para-um, um-para-muitos e muitos-para-muitos.

Especialização: dada uma determinada classe, criamos uma outra com novos atributos ou serviços que a tornam mais restrita.

Herança: é o mecanismo pelo qual uma classe obtém as características e métodos de outra para expandi-la ou especializá-la de alguma forma.

Agregação: é o ato de agregar, juntar duas ou mais classes para formar uma nova classe.

Além destes conceitos, se analisarmos as obras dos diversos autores que se propõem a estudar as nuances da orientação a objetos, veremos uma quantidade muito grande de variações, conceitos, especificações e documentações que dizem respeito a estes conceitos básicos vistos acima.

Análise, Projeto e Programação Orientados a Objetos

É perfeitamente possível modelar uma solução pensando totalmente orientado a objetos desde a fase de análise, passando pelo projeto do software e chegando à implementação em uma linguagem de programação orientada a objetos.

Uma grande vantagem de se pensar totalmente orientado a objeto é o fato de que um mesmo objeto, concebido em fase de análise, passa com as mesmas características desde o usuário até o programador que será responsável pela codificação final.

Como exemplo de softwares auxiliares, podem ser vistas diversas ferramentas CASE, linguagens de programação como Java, Delphi e outras.

Um dos problemas de se tentar estudar orientação a objetos é o fato de que alguns autores montam suas próprias concepções sobre esta teoria. Surgindo assim várias escolas.

Uma tentativa de resolver este tipo de problema é o que surgiu recentemente chamado de UML (Unified Modeling Language) que tenta concentrar os principais conceitos da modelagem orientada a objetos em um único método. Está sendo bem recebido pelo universo de desenvolvedores e tem grandes chances de se tornar uma tendência.

Considerações finais

A utilização da orientação a objetos é inevitável para quem pensa em começar a desenvolver sistemas adequados ao nosso tempo.

É de grande interesse que os alunos de graduação, possam aprender além dos conceitos da programação estruturada, também os conceitos da programação orientada a objetos.

Hoje existem ferramentas gráficas que facilitam bastante a vida do desenvolvedor, mas não fazem o principal: montar a solução com a escolha correta das classes.

Portanto o grande trabalho continua sendo o do mentor que está por trás de todas as soluções dos problemas deste mundo: o cérebro humano.

Bibliografia Consultada

BOOCH, Grady. Objected-Oriented Analysis and Design with applications. Addison-Wesley, 1998.

CESTA, André Augusto. Tutorial: A Linguagem de Programação Java. Unicamp, 1996.

COAD, Peter e MAYFIELD, Mark. Projeto de Sistemas em Java: Construindo aplicativos e melhores applets. São Paulo, Makron Books, 1998.

COUGO, Paulo. Modelagem Conceitual e Projeto de Banco de Dados. Rio de Janeiro: Campus, 1997.

LEWIS, John e LOFTUS, William. Java: Software Solutions. Addison-Wesley, 1998.

OLIVEIRA, Adelize Generini de. Análise, Projeto e Programação Orientados a Objetos.Bookstore, 1996.

OLIVEIRA, Adelize Generini de. Java: a linguagem de programação da Internet.Bookstore, 1996.

ROWE, Glenn. An Introduction to Data Structures and Algorithms with Java. Prentice hall Europe, 1998.