

## Programação Orientada a Objeto

A orientação a objetos, também conhecida como Programação Orientada a Objetos (POO), ou ainda em inglês Object-Oriented Programming (OOP) é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.

- **Classe** representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos. Exemplo de classe: Os seres humanos.
- **Subclasse** é uma nova classe originada de sua classe pai.
- **Objeto** é uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Humanos: João, José, Maria.
- **Atributos** são características de um objeto. Basicamente a estrutura de dados que vai representar a classe. Exemplos: Funcionário: nome, endereço, telefone, CPF, ...; Carro: nome, marca, ano, cor, ...; Livro: autor, editora, ano. Por sua vez, os atributos possuem valores. Por exemplo, o atributo cor pode conter o valor azul. O conjunto de valores dos atributos de um determinado objeto é chamado de estado.
- **Métodos** definem as habilidades dos objetos. Bidu é uma instância da classe Cachorro, portanto tem habilidade para latir, implementada através do método deUmLatido. Um método em uma classe é apenas uma definição. A ação só ocorre quando o método é invocado através do objeto, no caso Bidu. Dentro do programa, a utilização de um método deve afetar apenas um objeto em particular; Todos os cachorros podem latir, mas você quer que apenas Bidu dê o latido. Normalmente, uma classe possui diversos métodos, que no caso da classe Cachorro poderiam ser sente, coma e morda.

## Delphi

# A Linguagem de Programação Delphi

O ambiente de desenvolvimento Delphi se baseia em uma extensão orientada a objetos da linguagem de programação Pascal, também conhecida como Object Pascal. Recentemente, a Borland declarou sua intenção de se referir à linguagem como 'a linguagem Delphi', provavelmente devido ao fato de que a empresa quer poder dizer que o Kylix usa a linguagem Delphi, e também porque a Borland fornecerá a linguagem Delphi na plataforma .NET da Microsoft. Em virtude do hábito, usaremos os dois nomes.

As linguagens de programação mais modernas fornecem suporte para a *programação orientada a objetos* (POO). As linguagens de POO se baseiam em três conceitos fundamentais: o encapsulamento (em geral implementado com as classes), a herança e o polimorfismo ou ligação tardia (*late binding*). Embora seja possível escrever código Delphi sem entender os principais recursos de sua linguagem, você não poderá dominar esse ambiente sem ter uma compreensão completa da linguagem de programação.

## Recursos Básicos da Linguagem

A linguagem Delphi é uma extensão orientada a objetos da linguagem Pascal clássica, a qual a Borland promoveu por muitos anos em seus compiladores Turbo Pascal. A sintaxe da linguagem Pascal é bastante prolixa e mais legível do que, por exemplo, a linguagem C. Sua extensão orientada a objetos segue a mesma estratégia, apresentando o mesmo poder da geração mais recente de linguagens de POO, do Java até o C#.

Mesmo as linguagens básicas estão sujeitas a alterações contínuas, mas poucas afetarão as suas necessidades diárias de programação. No Delphi 6, por exemplo, a Borland incluiu o suporte para diversos recursos mais ou menos relacionados com o desenvolvimento do Kylix, a versão Linux do Delphi:

Delphi é um compilador, uma IDE e uma linguagem de programação, produzido antigamente pela Borland Software Corporation e atualmente produzido pela Embarcadero. O Delphi, originalmente direcionado para a plataforma Windows, chegou a ser usado para desenvolvimento de aplicações nativas para Linux e Mac OS, através do Kylix.

O Delphi é muito utilizado no desenvolvimento de aplicações desktop, aplicações multicamadas e cliente/servidor, compatível com os bancos de dados mais conhecidos do mercado. O Delphi pode ser utilizado para diversos tipos de desenvolvimento de projeto, abrangendo desde Serviços a Aplicações Web e CTI.

O nome **Delphi** é inspirado na cidade de Delfos, o único local na Grécia antiga em que era possível consultar o Oráculo de Delfos. O nome deve-se ao fato de que os desenvolvedores do compilador buscavam uma ferramenta capaz de acessar o banco de dados Oracle - daí o trocadilho: "a única maneira de acessar o oráculo é usando **Delphi**".

## HISTÓRIA

Quando lançado em 1995 para a plataforma Windows 16 bits, foi o primeiro a ser descrito como ambiente *RAD* (em português, **Desenvolvimento Rápido de Aplicações**). A segunda versão, lançada um ano depois, já produzia aplicativos para a plataforma Windows 32 bits (também existe uma versão em C++ - o C++ Builder). Em 2001, uma versão para plataforma Linux, conhecida como Kylix (uma espécie de junção do Delphi e do C++ Builder para o Linux), foi disponibilizada.

Em 2002, foi lançada uma nova versão compatível com Linux (através do Kylix e a paleta de componentes **CLX**), sendo que em 2003 o Delphi 8 passou a incluir suporte para desenvolvimento de aplicações .NET. No final de 2004 foi lançada a versão *Delphi 2005* (correspondente a versão 9), com suporte, na mesma IDE, ao desenvolvimento para plataformas Windows 32 bits e Windows .NET, além de integrar no mesmo ambiente de desenvolvimento a possibilidade de se desenvolver utilizando o ambiente do Delphi, C++ Builder ou C#. No final de 2005 foi lançada a versão *Delphi 2006* (correspondente a versão 10) integrada no Borland Developer Studio 4.0 com destaque para a total integração com o Borland Together, o preview do compilador do C++ Builder, atualização dos drivers DBExpress, novos refactorings e recursos da IDE. Nessa versão o gerenciador de memória utilizado pelo Delphi foi substituído, tornando os aplicativos compilados em Delphi mais rápidos. Como a ferramenta é compilada utilizando o próprio Delphi, esta nova versão também está muito rápida.

O arquiteto por trás do Delphi e de seu predecessor Turbo Pascal foi Anders Hejlsberg, até a sua alteração para a Microsoft em 1996, onde se tornou o arquiteto responsável pelo projeto do C# e foi peça chave na criação do ambiente Microsoft .NET, utilizado no Delphi 8.

O Delphi teve o formato da IDE alterado pela primeira vez na versão 8, basicamente similar ao formato do Microsoft Visual Studio para .NET.

As principais diferenças entre o Delphi/Kylix e outras ferramentas de desenvolvimento são: a programação visual, programação orientada a eventos, as paletas VCL e CLX, forte ênfase na conectividade com diversos bancos de dados e um grande número de componentes produzidos por terceiros, muitos deles disponíveis na internet e grande parte deles com o código fonte disponível. Alguns destacam como vantagens do Delphi: a existência de uma grande quantidade de componentes prontos em sua biblioteca, facilidade de uso e aprendizado e desenvolvimento rápido.

No dia 5 de setembro de 2006 a Borland, juntamente com a Developer Tools Group (grupo responsável pelo desenvolvimento das ferramentas IDE), entrega ao público a linha Turbo, que é praticamente o desmembramento da suíte Borland Developer Studio (BDS 2006). Em duas versões, Explorer (download gratuito; direcionado a estudantes, iniciantes em programação e hobbyistas) e a Professional (pago; direcionado às softwares-houses e profissionais autônomos).

Já no dia 14 de novembro de 2006, a Borland não encontrando um potencial comprador que atendesse os quesitos que se comprometam com a evolução das ferramentas IDE, ela decidiu criar uma subsidiária colocando todos responsáveis pela "DevCo" nessa nova empresa chamada CodeGear. Que, especialmente cuidará das IDEs.

A Borland vendeu sua divisão responsável pelos IDEs, a Codegear para a Embarcadero, que passou a produzir as versões mais recentes do Delphi.

## IDE

### Visão Geral do IDE

Ao trabalhar com um ambiente de desenvolvimento visual, seu tempo é gasto em duas partes diferentes do aplicativo: nos projetistas visuais e no editor de código. Os projetistas permitem que você trabalhe com os componentes no nível visual (quando você inclui um botão em um formulário) ou no nível não visual (quando você inclui um componente DataSet em um módulo de dados). Você pode ver um formulário e um módulo de dados em ação na Figura 1.1. Em ambos os casos, os projetistas permitem que você selecione os componentes necessários e defina o valor inicial das propriedades do componente.

O editor é o local no qual você escreve o código. O modo mais óbvio de escrever código em um ambiente visual é responder aos eventos, iniciando com aqueles ligados às operações executadas pelos usuários do programa, tal como clicar em um botão ou selecionar um item em uma caixa de lista. Você pode usar a mesma abordagem para os eventos internos, como, por exemplo, os que envolvem alterações de banco de dados ou notificações do sistema operacional.

À medida que os programadores conhecem melhor o Delphi, eles passam a escrever principalmente código de manipulação de eventos e, em seguida, passam a escrever suas próprias classes e componentes, e quase sempre acabam passando a maior parte de seu tempo no editor. Como este livro aborda mais do que a programação visual e como lidar com o restante do poder do Delphi, você verá mais código e menos formulários no restante do texto.

## SISTEMA DE INFORMAÇÃO

Um Sistema de Informação é um conjunto de componentes inter-relacionados que coleta dados no ambiente em que opera, usando recursos de sensoriamento e telecomunicações (entrada), analisa esses dados (processamento) e finalmente apresenta o produto como informação útil (saída), sendo construído de forma a atender os interesses de uma organização, de seus clientes internos e externos e de todos aqueles atingidos direta ou indiretamente pelo novo produto.

## SOFTWARE

Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um cliente específico ou para um mercado geral, ou ainda é uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.

Software também é o nome dado ao comportamento exibido por essa sequência de instruções quando executada em um computador ou máquina semelhante.

Software também é um produto e é desenvolvido pela Engenharia de software, e inclui não só o programa de computador propriamente dito, mas também manuais e especificações. Para fins contábeis e financeiros, o Software é considerado um Bens de capital.

## BANCO DE DADOS

Banco de dados é uma coleção de dados relacionados, os dados são fatos que podem ser gravados e que possuem um significado implícito. (Elmasri, 2005, p.4).

Já para Deitel (2005, p.895) um banco de dados é uma coleção organizada de dados, ainda para Deitel (2005, p.496) um grupo de arquivos relacionados costuma ser chamado de banco de dados.

## Tela Principal do Delphi

