VB x Delphi - A verdade sobre o Delphi

Se você é programador de Visual Basic, para que se importar com o que podem fazer os programadores Delphi? No mínimo, isto daria uma idéia de que se deve esperar das próximas versões do Visual Basic.

Não estou falando de um compilador de verdade. É lógico que os programadores VB anseiam por um destes há anos. E embora o Delphi seja um deles, a presença ou ausência de um compilador não altera o desempenho da maioria dos aplicativos escritos em qualquer uma dessas linguagens.

INVEJA DOS OBJETOS - O que o Delphi tem que o VB não tem? Objetos. O Delphi emprega um modelo dos objetos consistente que abrange todos os seus aspectos. Enquanto os módulos de Classe e os recursos de construção de objetos do Visual Basic 4.0 são adições recentes derivadas de uma ferramenta originalmente não orientada a objetos, o Delphi é orientado a objetos desde o começo.

O modelo de objetos do Delphi permite obter mais resultados com menos códigos e desenvolver aplicativos melhores com maior rapidez e menor número de erros. Um exemplo: para adicionar uma lista de fontes da impressora a uma caixa de listagem no VB 4.0, você executaria este código:

```
Dim I
For I=0 to Printer.FontCount – 1
List1.AddItem Printer.Fonts (I)
Next I
```

No Delphi 2.0, você pode obter os mesmos resultados com uma instrução:

```
List1.Itens := Printer.fonts;
```

No Delphi, a propriedade Items (itens) de um objeto de caixa de listagem e a propriedade Fonts (fontes) do objeto Printer (impressora) são indicadores de objetos Tstrings, que mantêm uma lista de cadeias de caracteres e são semelhantes a conjuntos de cadeias (assim como as propriedades Items dos componentes Memo, RichtEdit e Outline). Sendo um Tstring tão bom quanto o outro, você pode atribuir a lista de fontes à caixa de listagem, a seu critério.

No VB, por outro lado, a lista de fontes e os itens da lista de alternativas são conjuntos de cadeias, características de um produto que anteriormente não era orientado a objetos. Uma vez que estes conjuntos não são objetos, não há como fazer referência a toda a lista de fontes ou ao conteúdo da caixa de listagem no VB. Seu único recurso é copiar cada item da lista de fontes para a caixa desejada.

A Microsoft introduziu um objeto Collection, que permite fazer referência a um conjunto de itens como uma unidade no VB 4.0. Entretanto, você só pode utilizar objetos Colletion com componentes Windows 95, portanto eles são inúteis quando você está trabalhando com elementos de linguagem (por exemplo, uma lista suspensa ou objeto Printer) anteriores ao Windows 95.

COMPONENTES EXPANDIDOS - Outra grande vantagem do Delphi em relação ao VB é que ele permite personalizar e expandir componentes, além de criar outros totalmente novos. Embora os módulos de Classe do VB 4.0 ainda nem começou a resolver a hereditariedade e a capacidade de reutilização.

Digamos que você precisa de uma lista de fontes em vários aplicativos e deseja que estas listas classifiquem as fontes, exibindo-as em duas colunas e permita seleções múltiplas. No VB, seria necessário definir cada uma destas propriedades todas as vezes que uma caixa de listagem fosse adicionada a um formulário; e, em seguida, adicionar o código necessário para preencher a caixa com a lista de fontes Print. O Delphi, porém, permite criar um componente FontList reutilizável. A declaração de um novo componente TFontList como um tipo TListBox faz com que o TFontList do Delphi possa herdar, automaticamente, todas as propriedades, métodos e eventos do componente padrão. Portanto, é preciso elaborar códigos apenas para as coisas que estão sendo alteradas.

Você pode fazer tudo isto elaborando código VB. E se quisesse que o componente FontList exibisse o nome de cada fonte usando a família de fontes que ele representa? Não é possível fazer isto numa caixa de listagem VB sem utilizar controles personalizados adicionais. Mas no Delphi, é só especificar que seu FontList é uma caixa de listagem proprietária e, depois, adicionar códigos ao manipulador de eventos InDrawItem do componente para exibir cada nome de fonte com sua própria família de fontes.

Embora os usuários VB possam sentir-se ludibriados agora, é óbvio que a Microsoft fará todo o possível para eliminar a vantagem do Delphi. O VB não vai se tornar um ambiente totalmente orientado a objetos, mas vai chegar perto, incorporando mais elementos a cada nova versão. Quanto à capacidade de reutilização, pode ter certeza de que já vimos a última versão do VB, que é incapaz de criar controles ActiveX.

Paul Bonner
Colaborador da Windows Sources