

O LÚDICO NO PROCESSO DE INTRODUÇÃO AOS CONCEITOS DE ORIENTAÇÃO A OBJETOS EM DISCIPLINAS BÁSICAS DE INFORMÁTICA NAS ENGENHARIAS

Alexandreta R. C. Bhering – alexandretacb@yahoo.com

Centro Universitário do Leste de Minas Gerais – UnilesteMG, Diretoria de Ciências Exatas

Av. Tancredo Neves, 3500 – Bairro Universitário

35170-056 – Coronel Fabriciano – Minas Gerais

Ivone P. Terra - ivone@unilestemg.br

Resumo: Em geral os cursos de engenharia oferecem uma ou duas disciplinas de informática com carga horária reduzida, com ementas voltadas para algoritmos básicos. Tradicionalmente, para facilitar o aprendizado dos conceitos, o professor opta por utilizar programação estruturada em linguagem de alto nível voltada ao ambiente DOS. Nos cursos de engenharia do UnilesteMG esta tradição também é seguida. Porém não desejando que o aluno saia do curso julgando que a programação para ambientes Windows® e similares é totalmente diferente e mais complexa, oito horas-aula são dedicadas a mostrar, de uma maneira breve e simples, os conceitos de orientação a objetos e programação para “Windows” utilizando o Delphi™, para que o aluno saiba que a lógica de programação é sempre a mesma independente do ambiente. Como apresentar este assunto em tão curto espaço de tempo? Motivando o aluno. Como motivá-lo? Fazendo o ensino/aprendizado divertido. Como fazê-lo divertido? Utilizando recursos simples de realidade virtual; criando um ambiente lúdico; o velho “aprender fazendo” em pequenos passos progressivos; trazendo para a sala/laboratório de aula, o jogo eletrônico que tanto o cativa. Este trabalho pretende mostrar os exemplos lúdicos que são apresentados aos nossos alunos; de como eles são apresentados de forma incompleta, semi-pronta; e de relatar como os alunos se entusiasmam e se sentem desafiados a completar os exemplos/jogos.

Palavras-chave: Orientação a objetos, Delphi, Programação.

1. INTRODUÇÃO

Os cursos de engenharia em geral oferecem uma ou duas disciplinas de fundamentos da computação com carga horária reduzida. As ementas na maioria dos casos estão voltadas à construção e análise de algoritmos. Tradicionalmente o professor ministra a disciplina apresentando os conceitos em algoritmos estruturados *top-down*, utilizando uma linguagem de alto nível, por exemplo Pascal, C ou C++, no ambiente DOS.

Nos cursos de Engenharia Sanitária e Ambiental e Engenharia de Materiais do Centro Universitário do Leste de Minas Gerais – UnilesteMG – esta tradição também é seguida nas disciplinas Fundamentos da Computação I e II que possuem carga horária de 36 horas-aula cada, distribuídas numa proporção aproximada de 60% de aulas teóricas e 40% de aulas práticas em microcomputadores. A linguagem adotada é o Pascal.

Não desejando porém que o aluno saia destas disciplinas julgando que os programas para os ambientes gráficos e interativos, como por exemplo o Microsoft® Windows¹, são totalmente diferentes, muito mais difíceis e complexos daqueles aprendidos para o DOS, as

¹ Microsoft e Windows são marcas registradas da Microsoft Corporation.

oito horas-aula finais de Fundamentos da Computação II são dedicadas a uma breve introdução do paradigma de orientação a objetos e dos fundamentos de uma linguagem visual voltada ao desenvolvimento de programas “para *Windows*”; no presente caso é utilizado o *Borland® Delphi™ 4.0*. Com isto o aluno pode verificar que a lógica da construção de algoritmos aprendida anteriormente ou seja, os comandos/estruturas básicos de entrada, saída, atribuição, decisão e repetição, bem como as estruturas de dados e a modularização também se aplicam neste caso.

Como porém introduzir os conceitos de classe e de objetos com suas propriedades, funções e eventos em tão pouco tempo? Motivando os alunos.

Este trabalho visa mostrar como estes conceitos são apresentados aos alunos da Engenharia Sanitária e Ambiental e Engenharia de Materiais do UnilesteMG de forma a motivá-los; utilizando exemplos simples, lúdicos e atrativos, tornando o aprendizado mais fácil.

2. COMO ACONTECE E O QUE É DADO NAS OITO HORAS-AULA DEDICADAS AO ASSUNTO EM QUESTÃO

2.1 Informações sobre as condições em que são dadas estas aulas

Todas as oito aulas acontecem em laboratório onde existem microcomputadores ligados em rede. Cada computador é utilizado por um ou no máximo dois alunos sendo que cada turma possui no máximo 24 alunos. Cada aula corresponde a 50 minutos e estão agrupadas duas a duas, ou seja, 100 minutos de aula corrida por semana. A cada semana é apresentada uma breve teoria sobre o assunto principal a ser tratado; é dado um aplicativo exemplo semi-pronto para que o aluno apreenda melhor os conceitos teóricos; é solicitado que o aluno complete o exemplo dado e por fim é proposto um exercício a ser entregue na próxima semana. As seções a seguir mostrarão o que é dado em cada uma das quatro semanas.

2.2 Primeira semana

Resumo da primeira semana:

- Preâmbulo
- Breve introdução a objetos
- Assunto principal: Objetos de interface e propriedades (Exemplo1: Tela com os objetos mais comuns)
- Exemplo 2: Aplicativo da lâmpada (3 fases)
- Atividade a ser entregue na próxima semana: Semáforo

Preâmbulo

No início da aula é lembrado ao aluno que a programação vista anteriormente, voltada ao ambiente DOS, se baseia no paradigma estruturado *top-down* onde a seqüência de execução dos comandos acontece de forma linear de cima para baixo cabendo ao usuário seguir a seqüência ditada pelo programador podendo apenas de vez em quando escolher uma dentre algumas poucas opções (“*lembram-se: “SE” o usuário digitar a letra A então será calculado*”).

A seguir é dito que para ambientes do tipo *Windows*, onde o usuário tem plena liberdade de escolher o que quer fazer em determinado momento (“*no Editor de Texto clicar no botão Negrito ou no das Ferramentas de Desenho por exemplo*”) o paradigma utilizado é o da orientação a objetos. O aluno é informado que a Orientação a Objetos é um assunto extenso e que não faz parte da ementa da disciplina corrente. Que os alunos interessados no tema devem

² *Borland e Delphi* são marcas registradas da *Borland Software Corporation*.

matricular-se na disciplina Programação Orientada a Objetos no curso de Sistemas de Informação existente na Instituição. Mas que na presente disciplina ele verá rudimentos de programação baseada em objetos e em eventos. Que será utilizado o *Delphi*, pois o mesmo usa a linguagem *Object Pascal* que engloba a linguagem Pascal anteriormente aprendida.

Breve Introdução a Objetos

O assunto é apresentado de forma sucinta: O aplicativo *Delphi* faz uso de Objetos. Objetos similares pertencem a uma mesma classe. O *Delphi* pode gerar objetos predefinidos, os objetos de interface, e o programador pode criar seus próprios objetos. É esclarecido que será visto apenas como trabalhar com os objetos de interface gerados pelo *Delphi*.

Objetos possuem propriedades (“*dimensões, posição na tela, cor, etc*”) e funções/procedimentos. Aos objetos podem ser associados eventos que são procedimentos executados a partir da ação do usuário (“*ao clicar em um botão por exemplo abre-se uma nova tela*”).

É esclarecido que cada aplicativo *Delphi* gera vários arquivos, muitos deles feitos automaticamente e que os arquivos que o aluno trabalhará principalmente são os Formulários ou Telas para o Usuário e as Unidades de código; que cada Formulário, que também é um objeto, possui uma Unidade associada; que cada aplicativo deve ser salvo em uma pasta específica.

Assunto principal: Objetos de interface e propriedades (Exemplo1: Tela com os objetos mais comuns)

Para que o aluno entenda melhor o que é um objeto de interface, como gera-lo, quais são os objetos mais comuns e suas principais propriedades o aluno é solicitado a construir uma tela similar ao da “Figura 1”.

A imagem mostra uma janela de aplicativo com o título "Form1". O formulário contém os seguintes elementos de interface:

- Um grupo de texto "Dados do Professor" no topo.
- Dois campos de texto para "Nome" e "Endereço".
- Um campo de lista suspensa para "Cidade" com o valor "Ipatinga" selecionado.
- Um grupo de botões de opção para "Cargo" com as opções: Auxiliar, Assistente (selecionado), Adjunto e Titular.
- Um campo de lista suspensa para "Lotado no Curso de ..." com as opções: Aplicadas, Exatas, Humanas e Saúde.
- Dois campos de verificação: "Tem Mestrado?" (marcado) e "Tem Doutorado?" (desmarcado).
- Um campo de texto grande para "Observações".
- Dois botões de ação: "CADASTRAR" e "LIMPAR".

Figura 1 – Formulário ou Tela com os principais objetos de interface

O aluno é informado que a tela correspondente à “Figura 1” tem como único objetivo o aprendizado de: 1) criação dos principais objetos utilizados em aplicativos “para *Windows*” e 2) alteração dos valores iniciais das principais propriedades destes objetos. É informado que o desenvolvimento deste software completo exigiria conhecimentos de Banco de Dados que não faz parte da ementa da disciplina corrente mas que o aluno interessado no tema pode

matricular-se na disciplina homônima no curso de Sistema de Informação existente na instituição.

Em turmas que constroem esta tela com rapidez é dado o evento associado ao clicar do botão “Limpar”.

Exemplo 2: Aplicativo da lâmpada (3 fases)

O aluno fica um pouco frustrado pelo aplicativo anterior não funcionar. Portanto é dado um outro exemplo, o exemplo da lâmpada. Este exemplo é bastante simples e de fácil compreensão e por embutir realidade virtual causa grande impressão no aluno.

A idéia no uso destes exemplos é dá-lo semipronto para que o aluno decifre o código já construído e o tenha como dica para desenvolver o restante do código necessário.

O exemplo da lâmpada é desenvolvido em três fases.

1ª) É pedido ao aluno que construa a tela da “Figura 2a”, onde o círculo representa uma lâmpada que inicialmente está apagada (cor cinza) e que ao clicar no botão “Ligar” ela se acenderá (cor amarela). Em seguida é pedido ao aluno que modifique a propriedade *name* dos objetos que irão participar do código. É explicado que todos os nomes devem seguir a seguinte regra:

NomeDaClasseNomeSignificativo (ex.: ShapeLampada, BotaoLigar)
para facilitar a leitura da unidade de código. É explicado também que quando se deseja modificar/usar a propriedade de determinado objeto no *Delphi* é seguida a seguinte regra:

NameDoObjeto.Propriedade ou NameDoObjeto.Propriedade.SubPropriedade

É dito também para modificar valor de propriedade em tempo de execução utiliza-se o já conhecido **comando básico de atribuição**.

A seguir é mostrado ao aluno o código a ser vinculado ao clique do botão “Ligar” e como vincula-lo ao botão (a parte semipronta). Depois o aluno é solicitado a pensar e criar ele mesmo o evento a ser vinculado ao clique do botão “Desligar” que por ser bastante parecido com o anterior é feito rapidamente.



(a)

(b)

(c)

Figura 2 – Aplicativo da Lâmpada em 3 etapas

2ª) Tendo feito e rodado o aplicativo da “Figura 2a” o aluno é alertado “*mas na vida real só temos uma chave para ligar e desligar a lâmpada*”. O aluno é solicitado então a modificar o aplicativo anterior para que só seja necessário um único botão “*lembram-se do Se...Então...Senão...? Se a cor da lâmpada for cinza então torna-la amarela senão...*”. A maioria dos alunos consegue completar esta tarefa que é mostrada na “Figura 2b”. Aos que após algum tempo não conseguem completá-la é mostrado no quadro o código do evento.

3ª) Aos alunos que sozinhos conseguiram completar a 2ª fase é dito “*aqui no laboratório uma única chave comanda várias lâmpadas, vamos também colocar duas lâmpadas no aplicativo?*”. A maioria dos alunos que completou a 2ª fase consegue também fazer esta 3ª fase mostrada na “Figura 2c”.

É necessário frisar que alguns alunos desenvolvem esta e outras tarefas mais rapidamente que outros; é importante portanto que haja novas tarefas a serem completadas pelos que terminaram cedo, para que todos tenham trabalho a desenvolver. É necessário porém ficar claro para os alunos que nem todos precisam completar todas as tarefas pois o aluno de desenvolvimento mais lento não pode ao final da aula se sentir um derrotado.

Atividade a ser entregue na próxima semana: Semáforo

A fim de reforçar os conceitos apresentados em aula é solicitado ao aluno ao final de cada aula que faça um aplicativo a ser entregue na próxima semana. Tal aplicativo é sempre parecido com o feito em sala de aula porém com temática diferente. É essencial que não haja muitas novidades na atividade de casa para que o aluno possa completá-la sem muito esforço e sentir-se um “senhor” programador, ficando portanto motivado para a próxima aula.

Na primeira semana é pedido o aplicativo do semáforo. O formulário do mesmo é apenas esboçado no quadro. Aqui, na “Figura 3”, é apresentado como ele fica já entregue pelo aluno. É dito ao aluno que quando o usuário clicar no botão “Troca”, para trocar a “cor do semáforo”, é necessário que a seqüência seja a mesma de um semáforo normal, ou seja, vermelho – verde – amarelo – vermelho – verde e etc (“gente, vai ser necessário utilizar um ninho de Ses”).

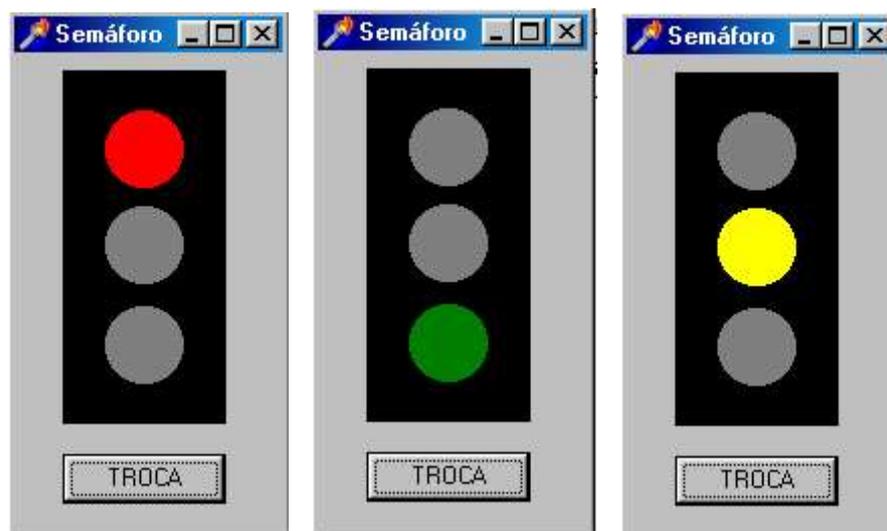


Figura 3 – A atividade do semáforo em tempo de execução

2.3 Segunda semana

Resumo da segunda semana:

- Assunto Principal: Eventos
- Exemplo: Carrinho
- Atividade a ser entregue na próxima semana: Avião

Assunto Principal: Eventos

Na primeira semana o tema “eventos” é tratado rapidamente, apenas o suficiente para a construção do exemplo da lâmpada. Nesta segunda semana é falado com um pouco mais de detalhe o que é evento, os eventos disponíveis para cada objeto, como vincular um evento a um objeto e os cuidados para com os cabeçalhos de eventos criados pelo *Delphi* que não podem ser modificados, nem copiados nem apagados, enfim não mexer com eles.

Exemplo: Carrinho

Este exemplo é dado semipronto para o aluno, ou seja, a tela já está completa com todos os objetos, porém os eventos previstos para os objetos ou não existem ainda ou estão incompletos. O aplicativo exemplo do carrinho é colocado na rede e cada aluno faz uma cópia para si. O aluno então é solicitado a rodar este aplicativo e testa-lo para verificar o que funciona e o que não funciona. Este exemplo é mostrado na “Figura 4”.



Figura 4 – O aplicativo do carrinho

Neste aplicativo está funcionando o botão “Pra Frente” (o carrinho anda um pouco a cada clique) e a caixa “Farol aceso” (o farol “acende”, ou melhor, aparece ou desaparece conforme o que se marcar). Os objetos caixas de escolha “Cor do Carro” e “Paisagem” estão incompletos; as cores azul e vermelha funcionam, a verde não; a paisagem “Neve” funciona (conforme mostrado na “Figura 4”), mas “Praia” e “Campo” não. O botão “Marcha Ré” não funciona. A caixa “Disparar” não funciona.

A seguir o aluno é solicitado a observar a propriedade *name* dos objetos que participam do código. Todos os nomes foram escolhidos segundo a regra já sugerida aos alunos: NomeDaClasseNomeSignificativo, como por exemplo: PanelCarro, ShapeCarro, RadioCor, CheckFarol, ShapeFarol). Isto é necessário para que eles compreendam melhor a parte do código que já está pronta.

É pedido ao aluno que complete o programa. Nos momentos necessários é explicado o significado das propriedades que ele ainda não conhece. A ordem que ele deve seguir quando desta complementação é sugerida, pois é conveniente que o aluno faça primeiro os eventos mais simples e lenta e progressivamente passe para os mais complexos. A ordem é a seguinte:

- 1) Observar o código completo do botão “Pra Frente”, compreende-lo e então fazer o evento para o clique do botão “Marcha Ré” que é bastante parecido. Salvar e rodar o aplicativo;
- 2) Observar o código incompleto da caixa “Escolha a Cor do Carro”, compreende-lo e então completá-lo para que a cor verde também possa aparecer no carro. Salvar e rodar o aplicativo;
- 3) Observar o código incompleto da caixa “Escolha a Paisagem”, compreende-lo e então completá-lo para que as paisagens “Campo” e “Praia” também possam aparecer como fundo no aplicativo. Salvar e rodar o aplicativo. OBS.: As duas figuras correspondentes a “Campo” e “Praia” já se encontram arquivadas na mesma pasta “Carro” copiada pelo aluno e isto é mostrado para ele (“*verifiquem o conteúdo da pasta Carro no Windows Explorer; verifiquem que lá se encontram três figuras...*”);
- 4) Observar o código completo da caixa “Farol aceso” e entendê-lo.
- 5) Modificar os procedimentos referentes aos cliques dos botões “Pra Frente” e “Marcha Ré” (“*adicionem um se...então...senão...*”) de tal forma que: se a caixa “Disparar” estiver marcada, ao se clicar em um destes dois botões o carrinho “dispara” para o fim (ou início) da tela, senão anda só um pouquinho como já vinha fazendo anteriormente (“*gente, repare que não quero uma ação imediata na caixa “Disparar” como acontece com o “Farol” e sim uma ação vinculada aos botões “Pra Frente e Marcha Ré...*”)

Atividade a ser entregue na próxima semana: Avião

É pedido ao aluno que na próxima semana traga um aplicativo parecido com o do carrinho porém até mais simples. Em tal aplicativo um avião deve “voar” diagonalmente pela tela ao se pressionar os botões “Decolar” e “Pousar” (“*gente, mexer nas propriedades Left e Top ao mesmo tempo*”). A paisagem de fundo pode ser uma única fixa porém deve ser diferente das do carrinho e estar de acordo com o aplicativo (“*procurem na internet uma paisagem de céu ou de aeroporto*”). O avião de preferência não deve ser “desenhado” como o carrinho o foi com objetos “Shape” sobre um objeto “Panel” pois ficaria muito feio e sim utilizado uma figura de avião em um objeto “Image” como o da paisagem de fundo.

Este aplicativo é muito fácil de ser feito e pode ficar muito bonito. Os alunos quando o mostram rodando na semana seguinte se orgulham de seu feito e disputam entre si qual ficou o mais bonito.

Como atividade extra, não sujeita a pontuação porém podendo ser utilizada para substituir algum outro trabalho não entregue, é sugerido ao aluno que acrescente mais um formulário ao aplicativo do avião e que invente uma outra brincadeira diferente qualquer para este formulário (“*como se fosse uma segunda fase de um jogo*”) a ser acionada pelo clique a um botão “Nova fase”. Aos alunos interessados é explicado como adicionar novo formulário a um aplicativo já existente.

2.4 Terceira semana

Resumo da terceira semana:

- Assunto Principal: Como utilizar dados digitados pelo usuário para fazer cálculos numéricos
- Exemplo: Aplicativo de Física
- Atividade a ser entregue na próxima semana: Adição de novo formulário ao aplicativo de Física com assunto diverso, ou Aplicativo de Química ou de Cálculo.

Assunto Principal: Como utilizar dados digitados pelo usuário para fazer cálculos numéricos

Os aplicativos feitos nas semanas anteriores eram todos não numéricos, ou seja, não manipulavam valores numéricos, não faziam cálculos. Como em engenharia a maioria dos programas exigirá cálculos e muitos destes cálculos envolverão valores digitados pelo usuário nos objetos da classe *Edit* dos formulários, esta última semana é dedicada ao desenvolvimento de aplicativos numéricos e ao uso das funções *StrToFloat* (ou *StrToInt*) e *FloatToStr* (ou *IntToStr*) que transformam respectivamente valores-texto digitados na caixa *Edit* em valores numéricos para cálculos na unidade de código e valores numéricos calculados pelo programa em valores-texto a serem colocados nos objetos *Edit* ou *Label*.

É explicado ao aluno o significado destas funções e o porquê da necessidade de uso delas.

Exemplo: Aplicativo de Física

O formulário da “Figura 5” é esboçado no quadro e é pedido ao aluno que o construa. Este aplicativo não é lúdico como os demais mas também desperta a curiosidade do aluno por trazer embutido uma certa interdisciplinaridade.



Figura 5 – Aplicativo de Física

É explicado aos alunos que neste aplicativo o usuário digitará os valores de distância e tempo e ao clicar no botão “Calcular” a velocidade uniforme deverá ser calculada e escrita na caixa correspondente. É pedido ao aluno que desabilite a caixa Velocidade e mude sua cor para *clInactiveBorder* e explicado o porquê disso. É pedido ainda que as propriedades *name* dos objetos que participam do código sejam alteradas para: *EditDistancia*, *EditTempo*, *EditVelocidade*, *BotaoCalcular* e *BotaoLimpar* conforme regras já passadas anteriormente a eles. O procedimento associado ao evento *OnClick* do botão “Calcular” é escrito incompleto no quadro como mostrado na “Figura 6” e o aluno é solicitado a tentar compreendê-lo e a completá-lo vinculando-o ao seu botão “Calcular” e, após testar o funcionamento deste botão, fazer o procedimento *OnClick* do botão “Limpar” e testá-lo.

```

procedure TForm1.BotaoCalcularClick(Sender: TObject);
var D,T,V : real;
begin
D:= StrToFloat(EditDistancia.Text);
T:=
V:=
EditVelocidade.Text:=FloatToStr(V);
end;

```

Figura 6 – Código do procedimento *OnClick* do botão “Calcular”

Como este procedimento é o primeiro feito no *Delphi* a fazer uso de variáveis, é ressaltado ao aluno o fato de que a declaração destas variáveis é feita de maneira similar aos dos procedimentos previamente aprendidos na linguagem Pascal para programas para DOS.

Muitos alunos conseguem completar corretamente o código. Após certo tempo o mesmo é completado no quadro para que aqueles que não conseguiram completa-lo sozinhos possam assimilá-lo.

Aos alunos que vão terminando esta tarefa é pedido que desenvolvam um novo aplicativo que faça conversão de unidades, como por exemplo transforme polegadas em centímetros e vice-versa para exercitarem o uso das funções *StrToFloat* e *FloatToStr* e o uso de variáveis.

Aos alunos que concluem com mais rapidez esta última tarefa, é solicitado que incluam um novo formulário (e é mostrado como) a este aplicativo de conversão de unidades a fim de fazer novas e diferentes conversões, como de temperatura por exemplo. Assim eles aprendem a trabalhar com diversos formulários e diversas unidades de código correspondentes. Aos alunos com dificuldades nas tarefas anteriores não é exigido que trabalhem com mais de um formulário.

Atividade a ser entregue na próxima semana: aplicativo de Química ou de Cálculo ou adição de novo formulário ao aplicativo de Física com assunto diverso

Ao final da aula é pedido ao aluno que na próxima semana traga um aplicativo que faça cálculos a partir de valores digitados pelo usuário. O aplicativo deve envolver cálculos referentes a outras disciplinas do curso como Química ou Cálculo ou mesmo algum outro tópico de Física.

2.5 Quarta semana

Resumo da quarta e última semana:

- Assunto Principal: Desenvolvimento de um aplicativo numérico mais elaborado
- Exemplo: Calculadora Eletrônica Científica
- Sugestão de exercício para casa: Adição de novas funcionalidades na calculadora.

Assunto Principal: Desenvolvimento de um aplicativo numérico mais elaborado

Os aplicativos numéricos feitos na semana anterior, na aula ou em casa, são em geral bastante simples e têm como objetivo o entendimento da necessidade e do uso das funções de transformação de texto em número e vice-versa.

Nesta última semana o objetivo principal é a construção de aplicativo mais elaborado, que faça cálculos mais complexos; que façam uso de todos os cinco comandos/estruturas básicos de programação; que façam uso tanto de variáveis locais quanto de variáveis globais e que os procedimentos vinculados a eventos também façam uso de funções pré-definidas no *Delphi* (como *sqrt()* e *cos()* por exemplo) e também e principalmente de funções já construídas pelo

aluno quando do desenvolvimento de programas Pascal para DOS (como a função *Fatorial()*).

Um aplicativo que se presta bastante a este papel é o da Calculadora Eletrônica que pode ser desenvolvida a princípio como um tipo mais simples e depois pouco a pouco ver-se transformada em Calculadora Científica com a adição de novas funcionalidades. A próxima seção trata deste exemplo.

Exemplo: Calculadora Eletrônica Científica

No início da aula é pedido que o aluno abra a Calculadora existente como Acessório do *Windows* e faça algumas operações nela. Depois é explicado ao aluno que na presente aula eles deverão desenvolver uma calculadora semelhante no *Delphi*. Ela deverá ser construída a princípio com poucas funções e que novas funções poderão ser adicionadas por aqueles alunos que terminarem as funções básicas.

É esboçado no quadro a interface da primeira versão desejada para a calculadora (mostrada na “Figura 7”). Um roteiro então é passado ao aluno com as tarefas que ele deverá completar uma a uma. No roteiro existem exemplos prontos de alguns procedimentos para facilitar a compreensão do aluno do que deve ser feito. É ressaltado que ele só deverá passar para uma nova tarefa após concluída a tarefa anterior e que já é esperado que nem todos os alunos completarão todas as tarefas; que isto é normal e permitido desde que o aluno esteja sempre trabalhando.



Figura 7 - Versão 1 da Calculadora



Figura 8 – Versão 2 da Calculadora

Os passos do roteiro são mostrados a seguir (os passos de 1 a 4 se referem à versão 1 mostrada na “Figura 7” e os passos de 5 a 7 se referem à versão 2 mostrada na “Figura 8”):

- 1) Construir a interface da calculadora (“Figura 7”) e modificar as propriedades *name* dos objetos para *EditVisor*, *Botao1*, *Botao2*, *BotaoIguar*, etc;
- 2) Construir os códigos dos procedimentos associados aos números (“1”, “2”, “3”, “4”, ..., e “,”) e testa-los. No roteiro é dado o código do botão “1” como exemplo;
- 3) Construir os códigos dos procedimentos associados aos botões de cálculo que utilizam apenas um parâmetro (“raiz”, “x^2”, “1/x” e “+/-”). No roteiro é dado o código do botão “raiz” como exemplo. Lá é lembrado ao aluno que ele já conhece a função *sqrt()* quando do estudo de Pascal.
- 4) Construir os códigos dos procedimentos associados às operações que necessitam de dois operadores, ou seja, os códigos dos botões “+”, “-”, “x”, “/” e “=”). No roteiro é mostrado o código do botão “+” e o código incompleto do botão “=” (contemplando apenas a operação de adição) como exemplos.

- 5) Modificar a interface da calculadora para que possam ser adicionadas novas funcionalidades conforme mostrado na “Figura 8” e modificar as propriedades *name* dos novos objetos.
- 6) Construir os códigos dos botões fatorial (“n!”) e de potência (“x^y”). É lembrado ao aluno que ele já possui estas duas funções no caderno em Pascal quando do estudo de modularização. Que tais funções podem ser agora definidas globalmente na unidade de código da calculadora e chamadas a trabalhar pelos procedimentos de eventos. Como exemplo é dado no roteiro a função fatorial e o procedimento *OnClick* do botão “n!”;
- 7) Construir o código do botão “pi” e os códigos referentes ao cálculo de seno, cosseno e tangente. É lembrado ao aluno que ele pode utilizar as funções correspondentes do Pascal (sin e cos) e lembrado ainda que as mesmas supõem parâmetros em radianos e daí a necessidade de conversão quando em grau como já aprendido anteriormente. Como exemplo é dado no roteiro o procedimento *OnClick* do botão “sen”.
- 8) Construir os códigos referentes aos botões C, CM, M, M+ e MR. Como exemplos são dados os códigos dos botões M e MR. É interessante notar que muitos alunos que chegam a esta fase não conseguem completar por não estarem familiarizados com o uso da memória de uma calculadora física.
- 9) Criar novas funcionalidades a gosto para os botões em branco.

Como este aplicativo é grande, poucos alunos completam os nove passos durante a aula. É sugerido a eles que procurem o monitor ao longo da semana para que ele os possa auxiliar no desenvolvimento do mesmo pois o seu entendimento será de grande valia como preparação para a prova teórica.

4. CONSIDERAÇÕES FINAIS

A maneira de trabalhar relatada nas seções anteriores tem sido testada e aprimorada por aproximadamente cinco semestres com três a quatro turmas por semestre. Ela tem se mostrado eficiente, ou seja, a grande maioria dos alunos participa com entusiasmo das aulas e o resultado do aprendizado é refletido positivamente na prova que é feita de maneira individual e sem consulta, em sala de aula, sem o uso de microcomputador.

É necessário frisar no entanto que uma determinada turma pode ter um nível de desenvolvimento melhor ou pior que a média. Neste caso é preciso que os programas sejam adaptados à turma, isto é, que o grau de exigência seja majorado ou minorado, sempre com a intenção de manter a motivação do aluno, lembrando que a programação baseada em objetos não faz parte da ementa da disciplina e que é abordada com o único intento de expandir o horizonte do aluno e talvez despertar seu interesse para uma especialização futura.

HOW TO INTRODUCE WITH FUN THE OBJECT-ORIENTED PROGRAMMING CONCEPTS IN THE BASIC COMPUTER SCIENCE DISCIPLINES IN ENGINEERINGS

Abstract: *Most engineering courses offer only one or two computer science disciplines and they usually work with basic algorithms. Traditionally, to facilitate the learning of the basic concepts, the teacher chooses to use structured programming with a high level language for DOS operational system. In the engineering courses of UnilesteMG this tradition also exists, but it is not desirable that the student goes off the course thinking that the programs “for Windows©” and similar are too different and much more hard and complex than that “for DOS”. To show him/her that the programming logic is always the same for the various*

software platform, eight hours are dedicated to show the basic concepts of object-oriented programming “for Windows” with Delphi™ in a brief and simple way. How to show this subject in such a short time? Giving motivation to the student. How to give motivation? Turning the teaching/learning process funny. How to make it funny? Using simple recourse of virtual reality; making games; the old “to do to learn” in small progressing steps; bringing to the laboratory class, the electronic game. This work shows the amused examples that are given to the UnilestMG engineering students and how they are showed, in a incomplete way, half ready. It relates also the enthusiasm of the students in completing the examples/games.

Key-words: *Object- Oriented, Delphi, Programming.*