

Gabarito da AD 1 de Programação I

Rio de Janeiro, 10 de setembro de 2010

1. Funcionamento geral

O sistema proposto na AD 1 deve executar a conversão de 3 valores monetários inseridos pelo usuário, podendo ser de 3 moedas diferentes (R\$, US\$ e E\$). Isso será feito através de 2 formulários: um de inserção de valores e um de conversão.



Formulário de inserção de dados pessoais e valores. O formulário é dividido em duas seções: "Dados Pessoais" e "Valores".

Dados Pessoais

Nome:

CPF:

Valores

Valor 1:

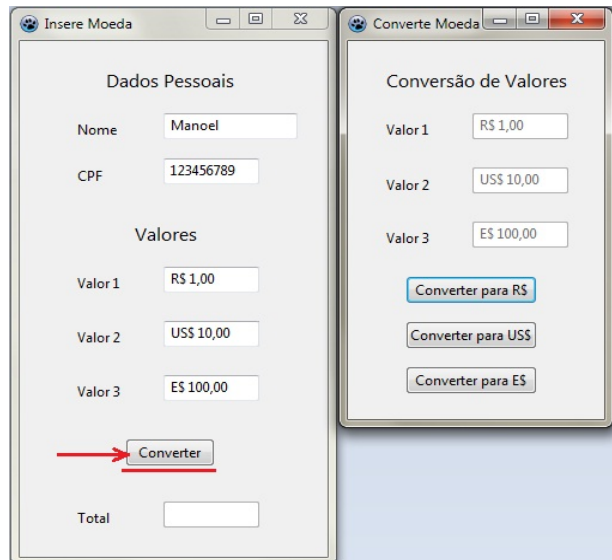
Valor 2:

Valor 3:

Botão: Converter

Total:

Figura 1: Formulário de inserção



Formulário de conversão de valores. O formulário é dividido em duas seções: "Dados Pessoais" e "Conversão de Valores".

Dados Pessoais

Nome: Manoel

CPF: 123456789

Valores

Valor 1: R\$ 1,00

Valor 2: US\$ 10,00

Valor 3: E\$ 100,00

Botão: Converter

Conversão de Valores

Valor 1: R\$ 1,00

Valor 2: US\$ 10,00

Valor 3: E\$ 100,00

Botões: Converter para R\$, Converter para US\$, Converter para E\$

Figura 2: Formulário de conversão

O formulário de inserção (Figura 1) deve conter 5 campos a serem preenchidos pelo usuário (*Nome*, *CPF*, *Valor 1*, *Valor 2* e *Valor 3*), um campo que conterá o somatório dos três valores inseridos, devidamente convertidos para uma moeda específica (*Total*) e um botão que chamará o segundo formulário, responsável pela conversão dos valores (botão *Converter*).

Quando o botão *Converter* for pressionado, o formulário de conversão (Figura 2) deverá ser aberto. Esse formulário deve conter os mesmos 3 campos de valor do primeiro formulário e com os mesmos valores digitados. Além desses campos, ele deve ter 3 botões: *Converter para R\$*, *Converter para US\$* e *Converter para E\$*.

Quando um desses botões for clicado, os campos de valor de ambos os formulários deverão ser convertidos para a moeda de conversão respectiva ao botão e o campo *Total* do formulário de conversão, até então vazio, deverá ser preenchido com o montante dos 3 valores presentes nos 2 formulários.

Observação 1: A linguagem ObjectPascal oferece um conjunto de *procedures* de verificação de conversão de tipos, muito úteis para a validação de campos de formulário:

```
Boolean tryStrToInt(String texto, Integer numero);
Boolean tryStrToCurr(String texto, Currency monetario);
Boolean tryStrToFloat(String texto, Float numeroDecimal);
(...)
```

Elas podem ser facilmente usadas para realizar a validação dos campos monetários. Outra forma de se fazer isso é manualmente, através de comparações de *Strings* (pode ser caracter por caracter ou através do objeto **TStringList**, que facilita a partição de um texto em fragmentos, através da definição de um separador).

Observação 2: Para realizar a atualização mútua entre os dois formulários, deve-se realizar uma pequena mudança na forma de comunicação entre *Units* aprendida em aula. Ao invés de declarar as *Units* na cláusula **uses** da seção de **interface**, deve-se fazê-lo na seção de implementação **implementation** em pelo menos uma delas. No exemplo abaixo, está sendo realizada a intercomunicação entre as *Units* 1 e 2 através da seção **interface**. Caso fosse de interesse passar a declaração da *Unit2* pra a seção **implementation** da *Unit1*, a mudança seria do código abaixo

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms, Dialogs, StdCtrls, Unit2;
(...)
```

para este código:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms, Dialogs, StdCtrls;
(...)

implementation
uses
  Unit2;
(...)

```

Por fim, segue abaixo alguns exemplos de conversão (Figuras 3, 4 e 5):

The image shows two side-by-side windows. The left window, titled 'Inserir Moeda', contains a form with the following fields: 'Nome' (Manoel), 'CPF' (123456789), 'Valores' (Valor1: R\$ 1,00, Valor2: R\$ 22,00, Valor3: R\$ 280,00), a 'Converter' button, and a 'Total' field (R\$ 303,00). The right window, titled 'Converte Moeda', shows the same 'Valores' fields but with the 'Converter' button disabled. Instead, there are three buttons: 'Converter para R\$', 'Converter para US\$', and 'Converter para E\$'. A red arrow points to the 'Converter para R\$' button.

Figura 3: Conversão para R\$

The image shows two side-by-side windows. The left window, titled 'Inserir Moeda', contains a form with the following fields: 'Nome' (Manoel), 'CPF' (123456789), 'Valores' (Valor1: US\$ 0,45, Valor2: US\$ 10,00, Valor3: US\$ 127,27), a 'Converter' button, and a 'Total' field (US\$ 137,73). The right window, titled 'Converte Moeda', shows the same 'Valores' fields but with the 'Converter' button disabled. Instead, there are three buttons: 'Converter para R\$', 'Converter para US\$', and 'Converter para E\$'. A red arrow points to the 'Converter para US\$' button.

Figura 4: Conversão para US\$

2. Características obrigatórias

- O sistema deve validar cada um dos campos monetários preenchidos pelo usuário ao apertar no botão de conversão. O sistema deve permitir **somente** valores no formato <R\$, US\$, E\$> xxx,xx, sendo que x é um algarismo. Em caso de qualquer outro tipo de preenchimento, ao clicar em *Converter*

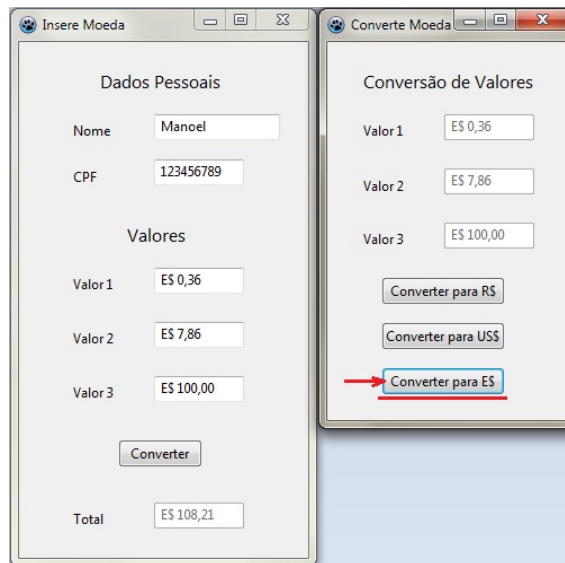


Figura 5: Conversão para E\$

o usuário deverá ser notificado do erro e deverá ser instruído sobre a forma correta de preenchimento, como pode ser visto na Figura 6:

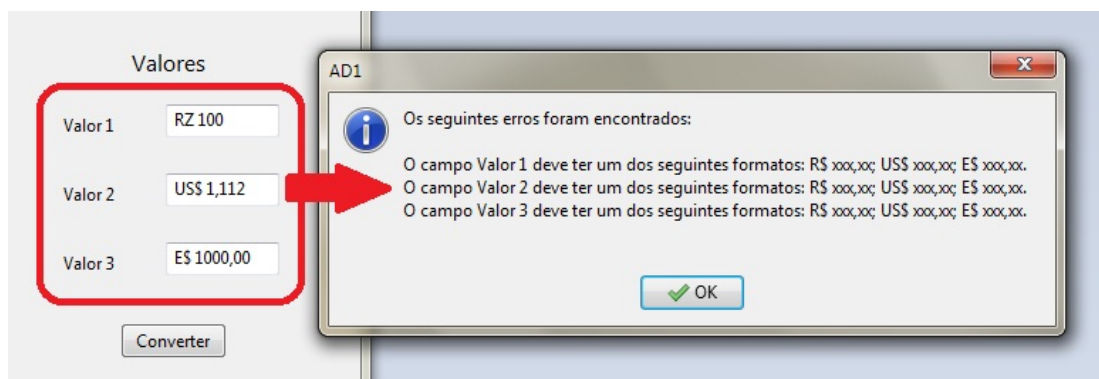


Figura 6: Validação dos campos monetários

- O separador de casas decimais dos valores monetários **deve ser a vírgula (,)**. O modo de se garantir isso é redefinindo o separador decimal *default* do Lazarus, como pode ser visto no código abaixo:

```
// Configurando o estilo monetário a se usar (x,xx)
DecimalSeparator := ',';
```

- O sistema deve permitir múltiplas conversões em uma mesma execução, portanto quaisquer conversões errôneas em cliques seguidos nos botões de qualquer

um dos formulários serão penalizadas.

3. Funcionalidades adicionais

A seguir, alguns exemplos de funcionalidades desejáveis ao sistema (que enriquecerão a AD de quem as implementou) serão listadas. É claro que quaisquer outras funcionalidades de interesse à ideia do sistema serão bem-vindas.

- Todos os campos do primeiro formulário podem ser validados para que seu preenchimento seja obrigatório. Caso algum campo não seja preenchido pelo usuário e o botão *Converter* seja acionado, uma situação semelhante à da Figura 7 pode ser implementada:

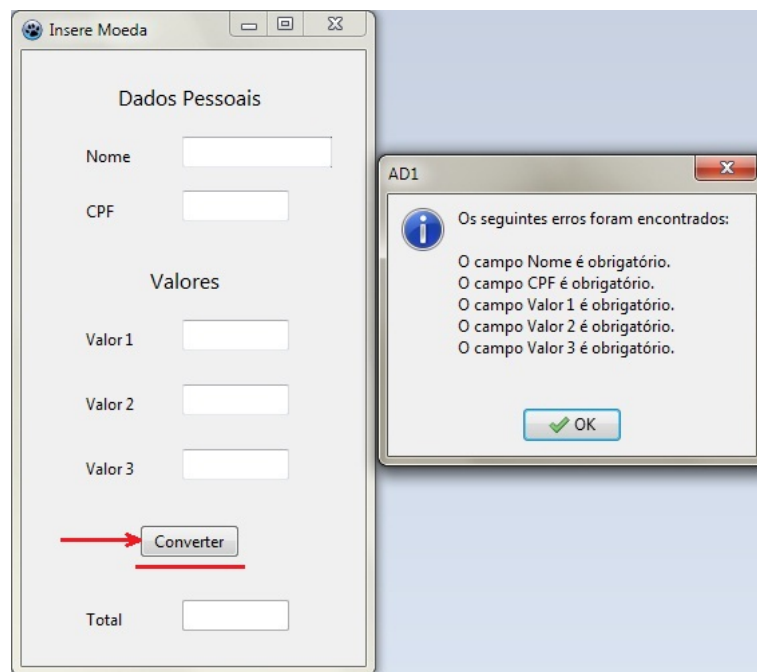


Figura 7: Validação de campos obrigatórios

- Pode-se adicionar validações aos campos *Nome* e *CPF*, tais como forçar o preenchimento de um CPF de 9 dígitos. A figura 8 exemplifica esse caso.

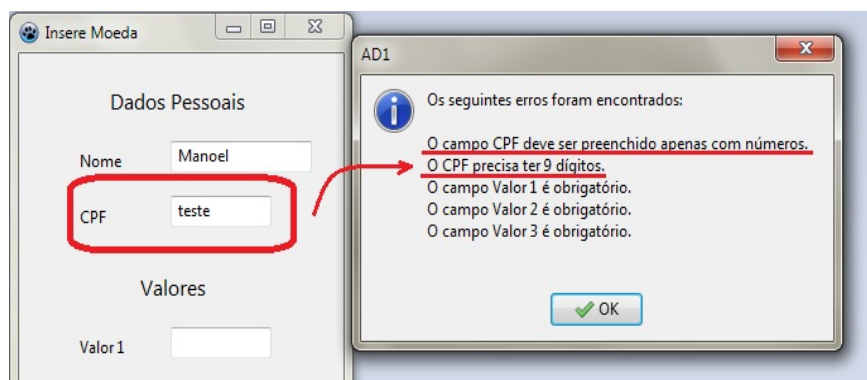


Figura 8: Validação do CPF

- É interessante permitir a escrita apenas nos valores monetários do primeiro formulário (proibindo edição nos valores do formulário de conversão e no valor total), de forma que os outros campos sirvam apenas de “visualizadores de informação”. Isso pode ser feito através da propriedade *Enabled* do campo em que se deseja impedir a edição (no Lazarus, ao se clicar um um *EditText*, por exemplo, essa propriedade torna-se visível no *ObjectInspector*).
- Por último, também vale a pena tomar o cuidado de não exibir o formulário de conversão em cima do formulário de inserção de valores. Para evitar isso, basta editar as propriedades *Top* e *Left* do formulário de conversão de forma apropriada (novamente, encontradas no *ObjectInspector* do Lazarus).