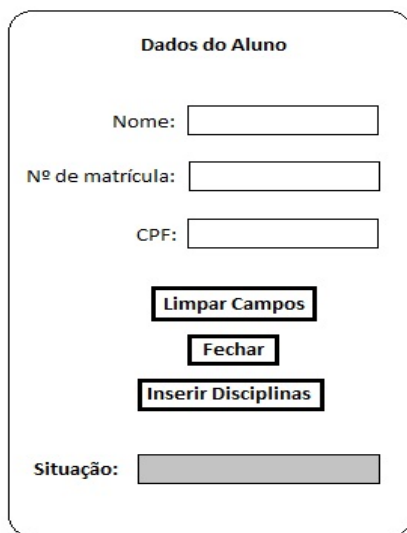


Gabarito da AP 1 de Programação I

Rio de Janeiro, 24 de setembro de 2010

1. Questão 1

Desenho dos formulários:



Formulário 1: Dados do Aluno

Nome:

Nº de matrícula:

CPF:

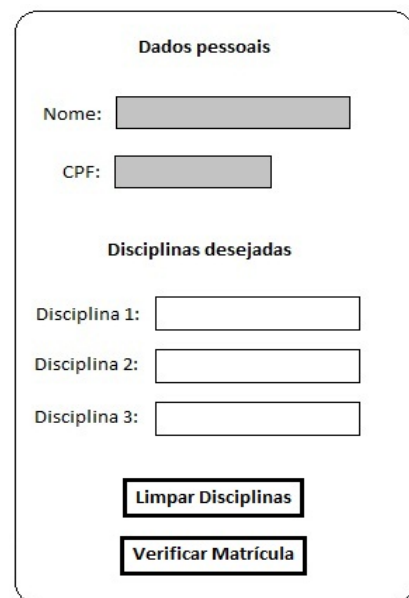
Limpar Campos

Fechar

Inserir Disciplinas

Situação:

Figura 1: Formulário 1



Formulário 2: Dados pessoais

Nome:

CPF:

Disciplinas desejadas

Disciplina 1:

Disciplina 2:

Disciplina 3:

Limpar Disciplinas

Verificar Matrícula

Figura 2: Formulário 2

A Figura 1 e a Figura 2 correspondem, respectivamente, aos desenhos do Formulário Principal e ao Formulário Secundário. Os campos de texto com fundo cinza são campos não-editáveis e os retângulos com linhas mais grossas representam botões.

Segue abaixo as declarações das variáveis responsáveis por armazenar os valores preenchidos pelo usuário do sistema, assim como as componentes de cada formulário:

- Formulário 1:

type

```

TFormInsererMatricula = class(TForm)
    LimparButton: TButton;
    FecharButton: TButton;
    InserirButton: TButton;
    EditNome: TEdit;
    EditNumMat: TEdit;
    EditCPF: TEdit;
    EditSituacao: TEdit;
    LabelDadosPessoais: TLabel;
    LabelNome: TLabel;
    LabelNumMat: TLabel;
    LabelCPF: TLabel;
    LabelSituacao: TLabel;

    procedure LimparButtonClick(Sender: TObject);
    procedure FecharButtonClick(Sender: TObject);
    procedure InserirButtonClick(Sender: TObject);

    private

    public
end;

var
FormInsererMatricula : TFormInsererMatricula;

// Ano de matricula
anoMat : Integer;
end;

```

- Formulário 2:

```

type

TFormInserirDisciplina = class(TForm)
    LimparButton: TButton;
    VerificarButton: TButton;

```

```

    EditNome: TEdit;
    EditCPF: TEdit;
    EditDisciplina1: TEdit;
    EditDisciplina2: TEdit;
    EditDisciplina3: TEdit;
    LabelDadosPessoais: TLabel;
    LabelDisciplinas: TLabel;
    LabelNome: TLabel;
    LabelCPF: TLabel;
    LabelDisciplina1: TLabel;
    LabelDisciplina2: TLabel;
    LabelDisciplina3: TLabel;

    procedure LimparButtonClick(Sender: TObject);
    procedure VerificarButtonClick(Sender: TObject);

    private

    public
end;

var
    FormInsereDisciplina : TFormInsereDisciplina;

    siglaDisciplina : String
    numeroDisciplina: String

    // Periodos de cada disciplina inserida
    periodo1 : Integer;
    periodo2 : Integer;
    periodo3 : Integer;

    temErro : Boolean;

end;

```

2. Questão 2

- Formulário 1:

Labels (sem valor funcional, apenas texto):

- LabelDadosPessoais: texto ‘Dados do Aluno’;
- LabelNome: texto ‘Nome:’;
- LabelNumMat: texto ‘Nº de matrícula:’;
- LabelCPF: texto ‘CPF:’;
- LabelSituacao: texto ‘Situação:’

EditBoxes:

- EditNome: campo onde será preenchido o nome do aluno;

validação: nenhuma.

funcionalidade: preencherá o campo ‘Nome’ do Formulário Secundário.

- EditNumMat: campo onde será preenchido o número de matrícula;

validação: verificar se o campo preenchido segue o formato “xxxxxx”, onde xxx são dígitos.

funcionalidade: os dois últimos dígitos devem ser armazenados na variável ‘anoMat’ para se realizar a validação da inscrição em disciplinas.

- EditCPF: campo onde será preenchido o CPF do aluno;

validação: nenhuma.

funcionalidade: preencherá o campo ‘CPF’ do Formulário Secundário.

- EditSituacao (somente leitura): campo onde será preenchido o resultado da inscrição em disciplinas do aluno;

validação: nenhuma.

funcionalidade: será usado como visualizador do resultado da inscrição.

Botões:

- LimparButton: limpará os campos 'Nome', 'Nº de matrícula' e 'CPF';
- FecharButton: fechará a aplicação;
- InserirButton: abrirá o Formulário Secundário, passando os valores em 'Nome' e 'CPF' para ele.

- Formulário 2:

Labels (sem valor funcional, apenas texto):

- LabelDadosPessoais: texto 'Dados Pessoais';
- LabelDisciplinas: texto 'Disciplinas desejadas';
- LabelNome: texto 'Nome:';
- LabelCPF: texto 'CPF:';
- LabelDisciplina1: texto 'Disciplina 1:';
- LabelDisciplina2: texto 'Disciplina 2:';
- LabelDisciplina3: texto 'Disciplina 3:';

EditBoxes:

- EditNome (somente leitura): campo que conterà o nome do aluno, informado no Formulário Principal;

validação: nenhuma.

funcionalidade: nenhuma.

- EditCPF (somente leitura): campo que conterà o CPF do aluno, informado no Formulário Principal;

validação: nenhuma.

funcionalidade: nenhuma.

- EditDisciplina1: campo onde será preenchido o código da primeira disciplina escolhida pelo aluno;

validação: verificar se o campo preenchido segue o formato “SIxxx”, onde xxx são dígitos.

funcionalidade: o primeiro dígito deve ser armazenado na variável ‘periodo1’ para se realizar a validação da inscrição em disciplinas.

- EditDisciplina2: campo onde será preenchido o código da segunda disciplina escolhida pelo aluno;

validação: verificar se o campo preenchido segue o formato “SIxxx”, onde xxx são dígitos.

funcionalidade: o primeiro dígito deve ser armazenado na variável ‘periodo2’ para se realizar a validação da inscrição em disciplinas.

- EditDisciplina3: campo onde será preenchido o código da terceira disciplina escolhida pelo aluno;

validação: verificar se o campo preenchido segue o formato “SIxxx”, onde xxx são dígitos.

funcionalidade: o primeiro dígito deve ser armazenado na variável ‘periodo3’ para se realizar a validação da inscrição em disciplinas.

Botões:

- LimparButton: limpará os campos ‘Disciplina 1’, ‘Disciplina 2’ e ‘Disciplina 3’;

- VerificarButton: fechará o Formulário Secundário e preencherá o campo ‘EditSituacao’ do Formulário Principal com a mensagem “Matrícula Autorizada” ou com a mensagem “Matrícula Não Autorizada”, dependendo do sucesso ou da falha em inscrever disciplinas.

3. Questão 3

- Formulário 1:

- procedure **LimparButtonClick**(Sender: TObject):

A limpeza dos campos ‘Nome’, ‘Nº de matrícula’ e ‘CPF’ é feita através da atribuição de ‘’ ao texto de cada um desses campos. Por exemplo, para limpar o campo ‘Nome’:

```
EditNome.Text := '';  
(...)
```

- procedure **FecharButtonClick**(Sender: TObject):

Para fechar a aplicação, basta usar a procedure *Terminate* da instância de Application em execução. Isso é feito através da seguinte linha de código:

```
Application.Terminate;  
(...)
```

- procedure **InserirButtonClick**(Sender: TObject):

Esse botão deverá abrir o segundo formulário, passando os valores dos campos 'Nome' e 'CPF' para ele. Antes de tudo, o código da matrícula deve ser validado para que se garanta o formato especificado na questão 2. Isso pode ser feito através da função *tryStrToInt(texto, numero)* (para verificar se apenas números foram digitados), combinada com uma verificação de tamanho de texto (como por exemplo: `Length(EditNumMat.Text) = 5`). Caso a entrada não esteja de acordo com esse formato, o sistema deve avisar o usuário do erro ao preencher a matrícula, através do procedimento *ShowMessage(mensagem)*.

Observação: é necessário importar a Unit contendo o segundo formulário, através da cláusula **Uses** da seção de **interface**.

Se a validação não acusar erros, para terminar, basta exibir o segundo formulário e passar os valores:

```
FormInsereDisciplina.EditNome.Text :=  
    FormInsereMatricula.EditNome.Text;  
FormInsereDisciplina.EditCPF.Text :=  
    FormInsereMatricula.EditCPF.Text;  
  
FormInsereDisciplina.Show;
```

- Formulário 2:

- procedure **LimparButtonClick**(Sender: TObject):

Assim como no primeiro formulário, a limpeza dos campos ‘Disciplina 1’, ‘Disciplina 2’ e ‘Disciplina 3’ é feita através da atribuição de ‘’ ao texto de cada um desses campos.

- procedure **VerificarButtonClick**(Sender: TObject):

Esse procedimento realiza quatro tarefas: a validação das disciplinas inseridas, a validação do período das matérias com o período vigente do aluno, o preenchimento do campo ‘Situação’ do Formulário Principal e o fechamento do Formulário Secundário.

- (a) Validação das disciplinas - é feito através da verificação da ocorrência do padrão ‘SIxxx’, sendo que x são dígitos. Isso pode ser feito através da separação do texto inserido em 2 partes, usando a função *AnsiMidString*:

```
// Exemplo para a disciplina 1

siglaDisciplina :=
  AnsiMidStr(FormInsereDisciplina.EditDisciplina1.Text, 1, 2);
numeroDisciplina :=
  AnsiMidStr(FormInsereDisciplina.EditDisciplina1.Text, 3, 3);

if ((CompareStr(siglaDisciplina, 'SI') = 0) &&
    (Length(numeroDisciplina) = 3) &&
    TryStrToInt(numeroDisciplina)) then
begin
  // Disciplina inserida OK!

  (...)
end;
```

Caso haja algum erro, é necessário abortar a funcionalidade e avisar o usuário do ocorrido usando *ShowMessage(mensagem)*.

- (b) Validação do período vigente - o período da disciplina (representado pelo primeiro dígito do código preenchido) deve ser compatível com o período do aluno. Como o período do aluno está contido no número de matrícula, a Unit contendo o Formulário Secundário deve possuir uma declaração

da Unit do Formulário Principal usando a cláusula **uses**. Porém, essa declaração deve ser feito dentro da seção **implementation**, para não ocorrer dependência cíclica.

A compatibilidade de período das disciplinas inseridas podem ser verificadas da seguinte forma (continuando o código do item anterior):

```
// Exemplo para a disciplina 1

siglaDisciplina :=
  AnsiMidStr(FormInserDisciplina.EditDisciplina1.Text, 1, 2);
numeroDisciplina :=
  AnsiMidStr(FormInserDisciplina.EditDisciplina1.Text, 3, 3);

if ((CompareStr(siglaDisciplina, 'SI') = 0) &&
    (Length(numeroDisciplina) = 3) &&
    TryStrToInt(numeroDisciplina)) then
begin
  // Disciplina inserida OK!

  // Obtendo o ano de matricula
anoMat := StrToInt(FormInserMatricula.EditNumMat.Text) mod 100;

// Obtendo o periodo de cada materia
periodo1 := StrToInt(numeroDisciplina[1]);

  if ( (11 - anoMat) * 2 >= periodo1 ) then
  begin
    // Disciplina valida para esse aluno!
  end;
end;
```

- (c) Preenchimento do campo ‘Situação’ - é feito de acordo com o sucesso ou falha da validação do período (item anterior). Se as três disciplinas são válidas para o aluno, então o campo ‘Situação’ do Formulário Principal deve ser preenchido com o texto “Matrícula Autorizada”. Caso contrário, ele deverá conter a mensagem “Matrícula Não Autorizada”. Isso pode ser feito da seguinte forma (expandindo o código anterior):

```
// Exemplo para a disciplina 1
```

```

// No começo da procedure, escreva:
temErro := F;

siglaDisciplina :=
  AnsiMidStr(FormInserDisciplina.EditDisciplina1.Text, 1, 2);
numeroDisciplina :=
  AnsiMidStr(FormInserDisciplina.EditDisciplina1.Text, 3, 3);

if ((CompareStr(siglaDisciplina, 'SI') = 0) &&
  (Length(numeroDisciplina) = 3) &&
  TryStrToInt(numeroDisciplina)) then
begin
  // Disciplina inserida OK!

  // Obtendo o ano de matricula
anoMat := StrToInt(FormInserMatricula.EditNumMat.Text) mod 100;

// Obtendo o periodo de cada materia
periodo1 := StrToInt(numeroDisciplina[1]);

  if ( (11 - anoMat) * 2 >= periodo1 )
  then else
    temErro := T;
  end;
end
else
  temErro := T;
end;

(Fazer o mesmo para as outras duas disciplinas)
(...)

if(temErro = T) then
  FormInserMatricula.EditSituacao.Text :=
    'Matrícula Não Autorizada';
else
  FormInserMatricula.EditSituacao.Text :=
    'Matrícula Autorizada';

```

```
end;
```

(d) Fechamento do Formulário Secundário - é feito através da chamada *Close*:

```
(...)
```

```
Close;
```

```
end;
```