

**Prova de Avaliação de Conhecimentos**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Curso** | | | | | | | | | **Unidade curricular** | | | |
| Sistemas Multimédia | | | | | | | | | Programação I | | | |
| **Época normal** | | | | | | | | **Outras épocas** | | **Data** | **Duração** | |
| Teste intercalar Teste final Teste avaliação global | | | | | | | | Recurso Especial | | 01.01.2020 | 2h | |
| **Docente(s)** | | | | | | | | | | | | |
| António Jorge Machado da Costa | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  | | | | |  |
|  |  |  |  |  |  |  |  | | | | |  |
| **Observações** | | | | | | | | | | | | |
| * O enunciado da prova de avaliação de conhecimentos é composto por 3 páginas incluindo esta página de rosto. * Quaisquer pressupostos adicionais devem ser claramente apresentados. * Desenvolva todas as questões da prova, usando as mesmas normas de programação definidas no Exemplo \*.SLN. * As respostas devem ser apresentadas em Linguagem Microsoft Visual c#.NET. | | | | | | | | | | | | |
| Bom trabalho! | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **Grupo I** | **Programação** | **(20 valores)** |

**[7 valores]**

1. Preencha o código nas 3 Camadas.
2. Crie a camada de dados da class ”Viagem”
3. Crie a camada de negócio da class ”Viagem”
4. No Form ”Form Viagem”, coloque as opções novo, gravar, eliminar e sair a funcionar.

**[2 valores]**

1. Form ”FormViagem”, usando o objeto viagemErrorProvider para validar o registo ao gravar.
   1. A data de início tem de ser superior ou igual à data de fim.
   2. A distância em dias entre a data de início e a data de fim tem de ser inferior a 32 dias.
   3. A data de início e a data de fim tem de ser do mesmo ano.
   4. Os quilómetros têm de ser superiores a 10.
   5. O ID e a descrição não podem ser vazios.

**[4 valores]**

1. Form ”FormViagem”.
   1. Altere o texto dos labels para resources.
   2. Utilize o mesmo evento nos menus e botões equivalentes.
   3. Ao fazer “enter” no ”IdVagemTextBox”, deve validar/obter o conteúdo da textbox e mostrar o registo da viagem no form.
   4. Preencha a ListBox ”viagemListBox” do form ”FormViagem” com a lista descrição da viagem e a duração em dias das viagens que foram efetuadas da data definida no objecto ”dataFiltroDateTimePicker”.
   5. Preencha o label ”resultadoLabel”, com o resultado de viagens feitas no ano corrente.

**[3 valores]**

1. Na class library ”CamadaNegocios.Funcoes”.
   1. Crie uma função static “ObterNome” que devolva uma string com o resultado da formatação de um texto dado como argumento. Ex: Argumento texto = “António Jorge Costa”, a função deve retornar “Costa, A. J.”.
   2. Utilize a função “FormatarNome” da classe ”CamadaNegocios.Funcoes” no Form ”FormConsultaAdvogado”, de forma a mostrar no label “autorLabel” o resultado da função.

**[4 valores]**

1. No user control ”UserControlData”.
   1. Crie a propriedade “Data” e mostre o dia da semana no label “diaSemanaLabel”.
   2. Crie um evento “TextoDuploClick”, que atua sempre que se faz duplo click sobre o label “diaSemanaLabel”.
   3. Exemplifique no Form ”FormViagem” a propriedade e o evento do UserControlData.

CREATE TABLE [dbo].[Viagem](

[IDViagem] [bigint] NOT NULL,

[DescricaoViagem] [nvarchar](100) NULL,

[DataInicio] [datetime] NULL,

[DataFim] [datetime] NULL,

[Kilometros] [real] NULL,

CREATE Procedure [dbo].[Gravar\_Viagem]

@IDViagem bigint,

@DescricaoViagem nvarchar(100),

@DataInicio datetime,

@DataFim datetime,

@Kilometros real

CREATE Procedure [dbo].[Obter\_Viagem]

@IDViagem bigint

CREATE Procedure [dbo].[Listar\_Viagem]

CREATE Procedure [dbo].[Eliminar\_Viagem]

@IDViagem bigint

**2 Form ”FormViagem”, usando o objeto viagemErrorProvider para validar o registo ao gravar.**

* 1. **A data de início tem de ser superior ou igual à data de fim.**
  2. **A distância em dias entre a data de início e a data de fim tem de ser inferior a 32 dias.**
  3. **A data de início e a data de fim tem de ser do mesmo ano.**

**//Form Viagens validação de datas**

**private bool DatasValidas(out string mensagemErro)**

**{**

**mensagemErro = string.Empty;**

**bool ok = true;**

**DateTime dataInicio = this.dataInicioDateTimePicker.Value;**

**DateTime dataFim = this.dataFimDateTimePicker.Value;**

**if (dataInicio > dataFim)**

**{**

**ok = false;**

**mensagemErro = "Data de inicio superior à data de fim";**

**}**

**else if (dataInicio.AddDays(32) < dataFim)**

**{**

**ok = false;**

**mensagemErro = "Distancia entre datas superior a 32 dias";**

**}**

**else if (dataInicio.Year != dataFim.Year)**

**{**

**ok = false;**

**mensagemErro = "Datas em anos diferentes";**

**}**

**return ok;**

**}**

* 1. **Os quilómetros têm de ser superiores a 10.**

**//Validar Kilometros**

**private bool KilometrosValidos(out string mensagemErro)**

**{**

**mensagemErro = string.Empty;**

**bool ok = true;**

**float kilometros = 0;**

**if (!float.TryParse(this.kilometrosTextBox.Text, out kilometros))**

**{**

**ok = false;**

**mensagemErro = "Kilometros invalidos";**

**}**

**else if (kilometros <= 10)**

**{**

**ok = false;**

**mensagemErro = "Kilometros tem de ser superior a 10";**

**}**

**return ok;**

**}**

* 1. **O ID e a descrição não podem ser vazios.**

**//Validar Descricao**

**private bool DescricaoValido(out string mensagemErro)**

**{**

**mensagemErro = string.Empty;**

**bool ok = true;**

**string idViagem = this.descricaoViagemTextBox.Text;**

**if (string.IsNullOrEmpty(idViagem))**

**{**

**ok = false;**

**mensagemErro = "Descrição invalido";**

**}**

**return ok;**

**}**

**3.1 - Altere o texto dos labels para resources.**

private void Form1\_Load(object sender, EventArgs e)

{

this.idViagemLabel.Text = Properties.Resources.VIAGEM\_ID;

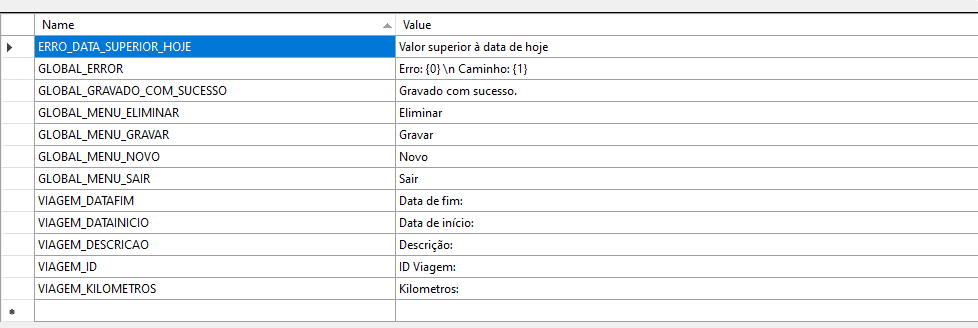
this.descricaoViagemLabel.Text = Properties.Resources.VIAGEM\_DESCRICAO;

this.dataInicioLabel.Text = Properties.Resources.VIAGEM\_DATAINICIO;

this.dataFimLabel.Text = Properties.Resources.VIAGEM\_DATAFIM;

this.kilometrosLabel.Text = Properties.Resources.VIAGEM\_KILOMETROS;

}

**Alterar resources**

**3.2 Utilize o mesmo evento nos menus e botões equivalentes.**

private void eliminar\_Click(object sender, EventArgs e)

{

this.Eliminar();

}

private void eliminar\_Click(object sender, EventArgs e)

{

this.Eliminar();

}

private void gravar\_Click(object sender, EventArgs e)

{

this.Gravar();

}

private void novo\_Click(object sender, EventArgs e)

{

this.Novo();

}

Nos eventos associar o mesmo evento nos botões e no menu strip

**3.3 Ao fazer “enter” no ”IdVagemTextBox”, deve validar/obter o conteúdo da textbox e mostrar o registo da viagem no form. Evento key UP**

**//Evento do form viagem**

private void viagemTextBox\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

**this.VerificarMostrarViagem();**

}

}

**//Funcao do Form Viagem**

**private void VerificarMostrarViagem()**

{

string mensagemErro = string.Empty;

string caminhoErro = string.Empty;

**if (this.ValoresValidos(true))**

{

Viagem viagem = **this.GetViagem(true);**

if (viagem != null)

{

**viagem.ObterViagem**(out mensagemErro, out caminhoErro);

this.MostrarViagem(viagem);

}

}

}

**//Funcao do formViagem**

private Viagem **GetViagem(**bool soChave)

{

Viagem viagem = new Negocio.Viagem();

long idViagem = 0;

if (long.TryParse(idViagemTextBox.Text, out idViagem))

{

viagem.IDViagem = idViagem;

}

if (!soChave)

{

float kilometros = 0;

viagem.DescricaoViagem = this.descricaoViagemTextBox.Text;

viagem.DataInicio = this.dataInicioDateTimePicker.Value;

viagem.DataFim = this.dataFimDateTimePicker.Value;

if (float.TryParse(this.kilometrosTextBox.Text, out kilometros))

{

viagem.Kilometros = kilometros;

}

}

return viagem;

}

**3.4 Preencha a ListBox ”viagemListBox” do form ”FormViagem” com a lista descrição da viagem e a duração em dias das viagens que foram efetuadas da data definida no objecto ”dataFiltroDateTimePicker”.**

**//Funçao do form Viagem**

private void EventoObterListaViagem()

{

**//Chama Camada de Negocio Obter Listas**

ViagemCollection viagens = Viagem.ObterListaViagem();

**this.PreencherListaViagem**(viagens); //chama função no form (abaixo)

}

**//aqui estou na camada de negocio**

public static ViagemCollection ObterListaViagem()

{

//chama função lista Viagem

DataTable dataTable = **Viagem.ListaViagem**();

ViagemCollection viagens = new ViagemCollection(dataTable);

return viagens;

}

#################################################################

**//função na camada de negocio**

public static DataTable **ListaViagem()**

{

**//chama camada de dados Obter Datatable**

return Dados.Viagem.ObterListaViagem();

}

**//Dentro do formViagem (função formviagens)**

private void PreencherListaViagem(ViagemCollection viagens)

{

if (viagens == null)

{

return;

}

DateTime dataFiltro = this.dataFiltroDateTimePicker.Value;

this.viagemListBox.Items.Clear();

foreach (Viagem viagem in viagens)

{

if (viagem.DataInicio <= dataFiltro &&

viagem.DataFim >= dataFiltro)

{

string registo = viagem.DescricaoViagem +

" Duração=" + viagem.DuracaoViagem().ToString();

this.viagemListBox.Items.Add(registo);

}

}

}

##################################################################

**//[3.4] V2**

**private void EventoObterListaViagem\_V2()**

**{**

**DateTime dataFiltro = this.dataFiltroDateTimePicker.Value;**

**IEnumerable<Viagem> viagens = Viagem.ObterListaViagemNaData(dataFiltro);**

**this.PreencherListaViagem\_V2(viagens);**

**}**

**//[3.4] V2**

**private void PreencherListaViagem\_V2(IEnumerable<Viagem> viagens)**

**{**

**if (viagens == null)**

**{**

**return;**

**}**

**this.viagemListBox.Items.Clear();**

**foreach (Viagem viagem in viagens)**

**{**

**string registo = viagem.DescricaoViagem + " Duração=" + viagem.DuracaoViagem().ToString();**

**this.viagemListBox.Items.Add(registo);**

**}**

**}**

**3.5 Preencha o label ”resultadoLabel”, com o resultado de viagens feitas no ano corrente.**

**//formViagem**

**private void totalButton\_Click(object sender, EventArgs e)**

**{**

**this.CalcularNumeroViagensAnoCorrente();**

**}**

**//formViagem**

**private void CalcularNumeroViagensAnoCorrente()**

**{**

**int numeroViagens = 0;**

**ViagemCollection viagens = Viagem.ObterListaViagem();//função camada negocio**

**numeroViagens = viagens.GetNumeroViagensAnoCorrente();**//função camada negocio collection

**this.resultadoLabel.Text = numeroViagens.ToString();**

**}**

**####################################################################**

**public int GetNumeroViagensAnoCorrente()** //ViagemCollection : Collection<Viagem>

**{**

**//V1**

**//int viagens = (from Viagem viagem in this**

**// where viagem.IsAnoCorrente()**

**// select viagem).Count();**

**//V2**

**int viagens = (from element in this**

**where element.IsAnoCorrente()**

**select element).Count();**

**return viagens;**

**}**

**4 Na class library ”CamadaNegocios.Funcoes”.**

**4.1 Crie uma função static “ObterNome” que devolva uma string com o resultado da formatação de um texto dado como argumento. Ex: Argumento texto = “António Jorge Costa”, a função deve retornar “Costa, A. J.”.**

**//Class camada negocio funções**

**//Funcao Obter Nome**

public static string ObterNome(string valor)

{

string resultado = string.Empty;

string[] palavras;

valor = valor.Trim();

palavras = valor.Split(' ');

//[Adicionar a última palavra]

resultado = palavras[palavras.Length - 1];

resultado += ", ";

//[percorrer as palavras da 2ª palavra até à antepenultima ]

for (int n = 0; n <= palavras.Length - 2; n++)

{

if (string.IsNullOrEmpty(palavras[n]))

{

resultado += palavras[n][0].ToString();

resultado += ".";

}

}

return resultado;

}

**4.2 Utilize a função “FormatarNome” da classe ”CamadaNegocios.Funcoes” no Form ”FormConsultaAdvogado”, de forma a mostrar no label “autorLabel” o resultado da função**

**//Funcoes class**

**//Funcao Formatar Nome**

public static string FormatarNome(string valor)

{

string resultado = string.Empty;

string[] palavras;

valor = valor.Trim();

palavras = valor.Split(' ');

if (palavras.Length >= 2)

{

resultado = palavras[0];

resultado += ", ";

resultado += palavras[palavras.Length - 1];

}

else

{

resultado = valor;

}

return resultado;

}

**//FormViagem**

**//[4.2]**

private void exemplificarFuncaoButton\_Click(object sender, EventArgs e)

{

string texto = this.autorLabel.Text;

string resultado = Negocio.Funcoes.FormatarNome(texto);

this.autorLabel.Text = resultado;

}

* 1. **No user control ”UserControlData”.**

**5.1 Crie a propriedade “Data” e mostre o dia da semana no label “diaSemanaLabel”.**

**//UserControl**

**//Propriedades**

//[5.1]

private DateTime data;

public DateTime Data

{

get { return data; }

set

{

data = value;

this.diaSemanaLabel.Text = this.GetDiaSemana(data);

}

}

**//UserControl**

**//Metodos**

private string GetDiaSemana(DateTime data)

{

string diaSemana = string.Empty;

switch (data.DayOfWeek)

{

case DayOfWeek.Sunday:

diaSemana = "domingo";

break;

case DayOfWeek.Monday:

diaSemana = "segunda-feira";

break;

case DayOfWeek.Tuesday:

diaSemana = "terça-feira";

break;

case DayOfWeek.Wednesday:

diaSemana = "quarta-feira";

break;

case DayOfWeek.Thursday:

diaSemana = "quinta-feira";

break;

case DayOfWeek.Friday:

diaSemana = "sexta-feira";

break;

case DayOfWeek.Saturday:

diaSemana = "sabado";

break;

}

return diaSemana;

}

**5.2 Crie um evento “TextoDuploClick”, que atua sempre que se faz duplo click sobre o label “diaSemanaLabel”.**

**//UserControl**

**//Eventos**

//[5.2]

public event EventHandler TextoDuploClick;

//[5.2]

public void OnTextoDuploClick()

{

if (this.TextoDuploClick != null)

{

this.TextoDuploClick(this, new EventArgs());

}

}

//[5.2]

private void diaSemanaLabel\_DoubleClick(object sender, EventArgs e)

{

this.OnTextoDuploClick();

}

**5.3 Exemplifique no Form ”FormViagem” a propriedade e o evento do UserControlData.**

**//FormViagem**

//[5.3]

private void exemplificarUserControlButton\_Click(object sender, EventArgs e)

{

this.userControlDiaSemana1.Data = this.dataFiltroDateTimePicker.Value;

**}**

**//FormViagem**

//[5.3]

private void userControlDiaSemana1\_TextoDuploClick(object sender, EventArgs e)

{

MessageBox.Show("Ocorreu o evento Duplo Click");

}