09

Faça como eu fiz

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Vá na classe **Cliente** e implemente todos os testes das propriedades. Os testes completos ficarão da seguinte forma:

```
[Required(ErrorMessage = "Código do Cliente é obrigatório.")]
[RegularExpression("([0-9]+)",ErrorMessage = "Código do Cliente somente aceita valores numéricos
[StringLength(6, MinimumLength = 6, ErrorMessage = "Código do Cliente deve ter 6 dígitos.")]
public string Id { get; set; }

[Required(ErrorMessage = "Nome do Cliente é obrigatório.")]
[StringLength(50,ErrorMessage = "Nome do Cliente deve ter no máximo 50 caracteres.")]
public string Nome { get; set; }

[StringLength(50, ErrorMessage = "Nome do Pai deve ter no máximo 50 caracteres.")]
public string NomePai { get; set; }

[Required(ErrorMessage = "Nome da Mãe é obrigatório.")]
[StringLength(50, ErrorMessage = "Nome da Mãe deve ter no máximo 50 caracteres.")]
public string NomeMae { get; set; }

public bool NaoTemPai { get; set; }

[Required(ErrorMessage = "CPF obrigatório.")]
[RegularExpression("([0-9]+)", ErrorMessage = "CPF somente aceita valores numéricos.")]
[StringLength(11, MinimumLength = 11, ErrorMessage = "CPF deve ter 11 dígitos.")]
public string Cpf { get; set; }

[Required(ErrorMessage = "Genero obrigatório.")]
public int Genero { get; set; }

[Required(ErrorMessage = "CEP obrigatório.")]
[RegularExpression("([0-9]+)", ErrorMessage = "CPF somente aceita valores numéricos.")]
[StringLength(8, MinimumLength = 8, ErrorMessage = "CPF deve ter 8 dígitos.")]
public string Cep { get; set; }

[Required(ErrorMessage = "Logradouro é obrigatório.")]
[StringLength(100, ErrorMessage = "Logradouro deve ter no máximo 100 caracteres.")]
public string Logradouro { get; set; }

[Required(ErrorMessage = "Complemento é obrigatório.")]
[StringLength(100, ErrorMessage = "Complemento deve ter no máximo 100 caracteres.")]
public string Complemento { get; set; }

[Required(ErrorMessage = "Bairro é obrigatório.")]
[StringLength(50, ErrorMessage = "Bairro deve ter no máximo 50 caracteres.")]
public string Bairro { get; set; }

[Required(ErrorMessage = "Cidade é obrigatória.")]
[StringLength(50, ErrorMessage = "Cidade deve ter no máximo 50 caracteres.")]
public string Cidade { get; set; }
```

```
[Required(ErrorMessage = "Estado é obrigatório.")]
[StringLength(50, ErrorMessage = "Estado deve ter no máximo 50 caracteres.")]
public string Estado { get; set; }

[Required(ErrorMessage = "Número do telefone é obrigatório.")]
[RegularExpression("([0-9]+)", ErrorMessage = "Número do telefone somente aceita valores numéri
public string Telefone { get; set; }

public string Profissao { get; set; }

[Required(ErrorMessage = "Renda familiar é obrigatória.")]
[Range(0, double.MaxValue, ErrorMessage = "Renda familiar deve ser um valor positivo.")]
public Double RendaFamiliar { get; set; }
```

2) Continuando na classe **Cliente**, inclua um novo método para validar os testes que dependam de várias propriedades distintas:

```
public void ValidaComplemento()
{
    if (this.NomePai == this.NomeMae)
    {
        throw new Exception("Nome do Pai e da Mãe não podem ser iguais.");
    }
    if (this.NaoTemPai == false)
    {
        if (this.NomePai == "")
        {
            throw new Exception("Nome do Pai não pode estar vazio quando a propriedade Pai Desc
        }
    }
    bool validaCPF = Cls_Uteis.Valida(this.Cpf);
    if (validaCPF == false)
    {
        throw new Exception("CPF inválido.");
    }
}
```

3) Coloque a declaração da classe **Cliente** como sendo pública:

```
public class Cliente
```

4) No **Gerenciador de Soluções**, no projeto **CursoWindowsForms**, clique com o botão direito do mouse sobre **Referências** e adicione a DLL **Microsoft.VisualBasic**.

5) No código fonte do formulário **Frm_CadastroCliente_UC**, adicione a referência mostrada abaixo:

```
using Microsoft.VisualBasic;
```

6) Ainda no código fonte do formulário **Frm_CadastroCliente_UC**, crie a rotina que atribui os dados do formulário a uma instância da classe `Cliente` . Insira o seguinte código:

```
Cliente.Unit LeituraFormulario()
{
    Cliente.Unit C = new Cliente.Unit();
    C.Id = Txt_Codigo.Text;
    C.Nome = Txt_NomeCliente.Text;
    C.NomeMae = Txt_NomeMae.Text;
    C.NomePai = Txt_NomePai.Text;
    if (Chk_TemPai.Checked)
    {
        C.NaoTemPai = true;
    }
    else
    {
        C.NaoTemPai = false;
    }
    if (Rdb_Masculino.Checked)
    {
        C.Genero = 0;
    }
    if (Rdb_Feminino.Checked)
    {
        C.Genero = 1;
    }
    if (Rdb_Indefinido.Checked)
    {
        C.Genero = 2;
    }
    C.Cpf = Txt_CPF.Text;

    C.Cep = Txt_CEP.Text;
    C.Logradouro = Txt_Logradouro.Text;
    C.Complemento = Txt_Complemento.Text;
    C.Cidade = Txt_Cidade.Text;
    C.Bairro = Txt_Bairro.Text;

    if (Cmb_Estados.SelectedIndex < 0)
    {
        C.Estado = "";
    }
    else
    {
        C.Estado = Cmb_Estados.Items[Cmb_Estados.SelectedIndex].ToString();
    }

    C.Telefone = Txt_Telefone.Text;
    C.Profissao = Txt_Profissao.Text;

    if (Information.IsNumeric(Txt_RendaFamiliar.Text))
    {
        Double vRenda = Convert.ToDouble(Txt_RendaFamiliar.Text);
        if (vRenda < 0 )
        {
            C.RendaFamiliar = 0;
        }
        else
        {
            C.RendaFamiliar = vRenda;
```

```
            }
        }
        return C;
    }
```

7) Altere o código do evento **Click**, do botão **Novo**, da barra de ferramentas, para o seguinte:

```
    private void novoToolStripButton_Click(object sender, EventArgs e)
    {
        try
        {
            Cliente.Unit C = new Cliente.Unit();
            C = LeituraFormulario();
            C.ValidaClasse();
            C.ValidaComplemento();
            MessageBox.Show("Classe foi inicializada sem erros", "ByteBank", MessageBoxButtons.OK, M
        }
        catch (ValidationException Ex)
        {
            MessageBox.Show(Ex.Message, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        catch (Exception Ex)
        {
            MessageBox.Show(Ex.Message, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
```