

Escola Técnica Estadual “São Paulo” - ETESP

Linguagem C# - Visual Studio 2019

Classe “Abstrata”

É uma classe “especial” que NÃO PODERÁ SER INSTANCIADA (poderá ser herdada), destinada a ser uma classe base para outras classes.

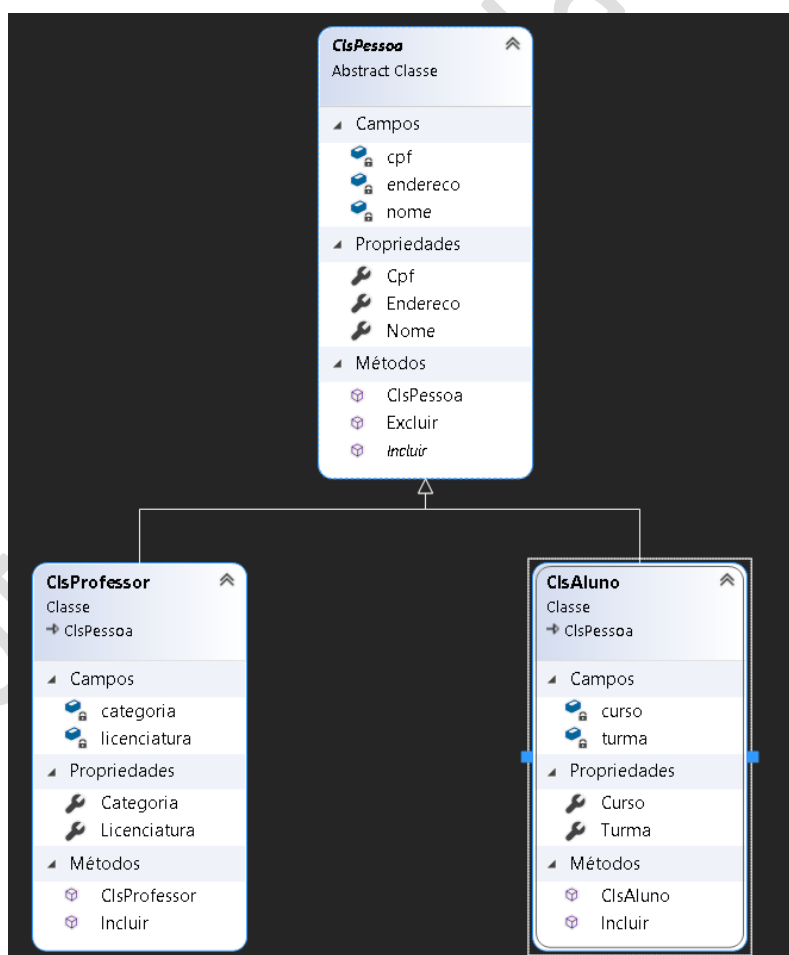
A classe abstrata poderá conter atributos, propriedades (gets/sets), métodos abstratos, métodos concretos (“normais”) e o método construtor.

O método abstrato somente poderá ser declarado em uma classe abstrata. Quando declarado, o método “abstrato” não poderá ser implementado (conter “instruções”). DEVERÁ conter apenas a sua “assinatura”, exemplo: `public abstract void Calcular();`

Sua implementação será obrigatória e deverá ser realizada na classe derivada, utilizando o modificador “override”. Exemplo: `public override void Calcular();`

Vamos a um exemplo prático...

Observe o diagrama de classe abaixo:



O projeto terá “dois formulários” (FrmMenu e FrmCadastro) e “três classes” (ClsPessoa, ClsProfessor e ClsAluno).

A classe “ClsPessoa” será “abstrata”, terá o método Incluir (abstrato) e o método Excluir (concreto).

Prof. Roberto de Castro

As classes `ClsProfessor` e `ClsAluno` serão classes concretas, que herdarão da classe base `ClsPessoa`, os atributos, propriedades e métodos e que, OBRIGATORIAMENTE, deverão implementar o método abstrato "Incluir".

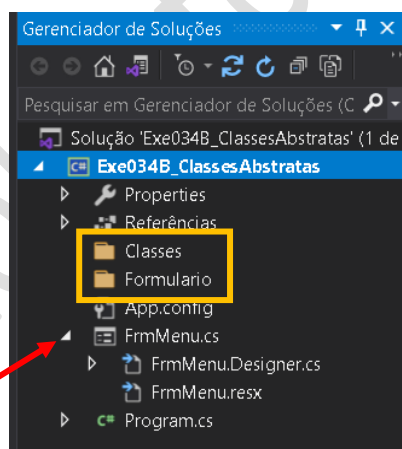
No Visual Studio, crie um novo projeto (`Exe034_ClassesAbstratas`) e altere as seguintes propriedades do formulário:

CONTROLE	PROPRIEDADE	CONTEÚDO
Form	Name	FrmMenu
	MaximizeBox	False
	StartPosition	CenterScreen
	Text	Classes Abstradas
	WindowState	Maximized
	IsMdiContainer	True

Antes de prosseguirmos com a configuração deste formulário, vamos "organizar" nosso projeto com a criação de duas pastas: uma que irá conter os formulários e outra que irá conter as classes.

Na janela do "Gerenciador de Soluções", pressione o botão direito do mouse sobre o nome do projeto e selecione a opção: Adicionar → Nova Pasta. Dê o nome de "Formulario" (sem acento). Repita a operação mais uma vez: Pressione novamente o botão direito do mouse sobre o nome projeto e selecione a opção: Adicionar → Nova Pasta. Dê o nome de "Classes".

Após a criação das pastas, o "Solution Explorer" deverá estar com a aparência abaixo:

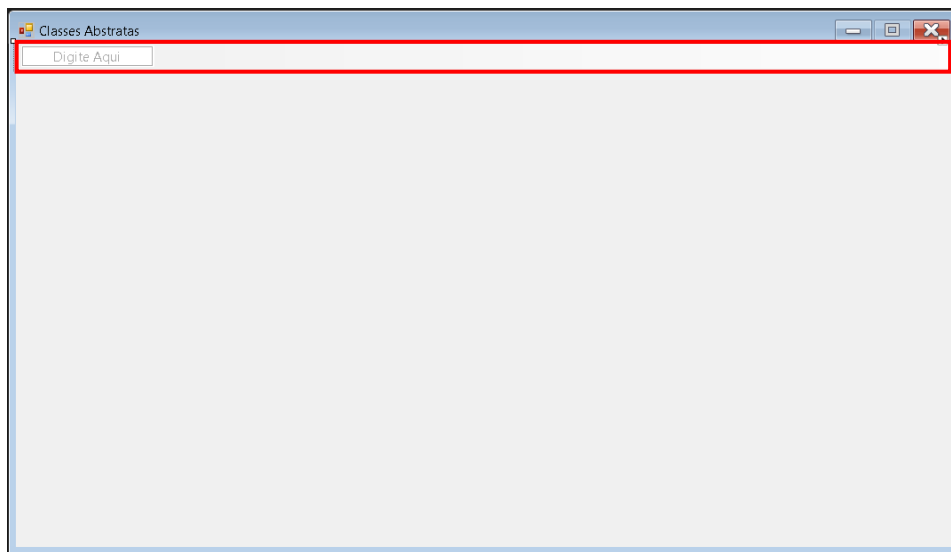


Clique no "FrmMenu" e "arraste" para a pasta "Formulario" (será exibida uma mensagem solicitando a confirmação). O formulário "FrmMenu" será transferido para a pasta "Formulario".

Faça um "pequeno" teste, execute o projeto para confirmar se o funcionamento está ok! O formulário DEVERÁ ser exibido maximizado.

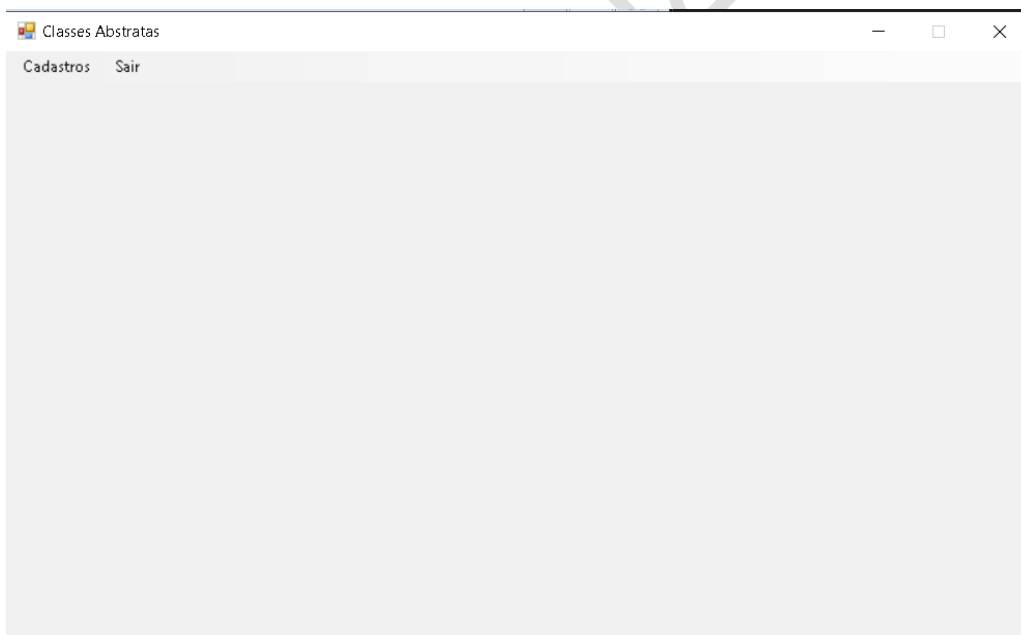
Se tudo ok, prosseguiremos com a configuração do formulário "Menu", que está dentro da pasta "Formulario".

Vamos inserir neste formulário o controle "MenuStrip". Este controle será inserido automaticamente no "topo" do formulário.



Vamos incluir neste menu, duas opções: Cadastros e Sair.

O menu deverá ficar como o modelo abaixo:

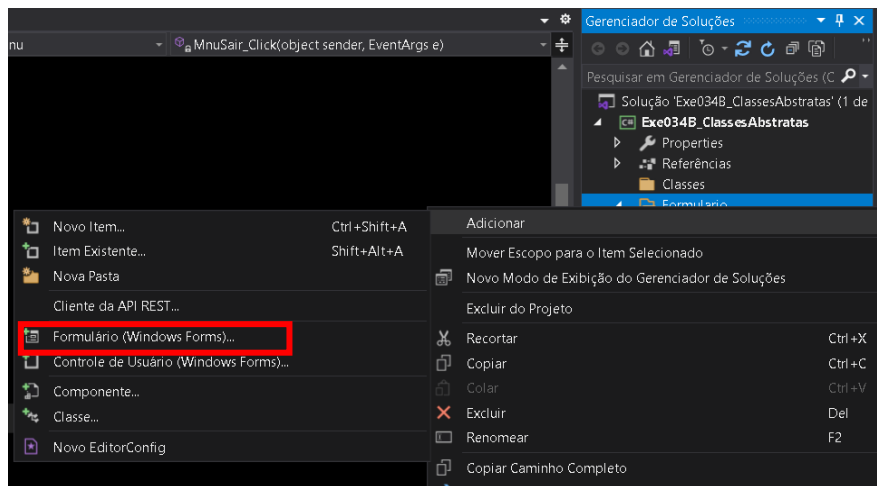


Altere a propriedade "name" destes "menus" para: MnuCadastros e MnuSair.

Inclua a programação no menu "Sair". Dê dois cliques sobre o texto "Sair" e insira a programação:

```
private void MnuSair_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Vamos incluir o segundo formulário... Botão direito sobre a pasta "Formulario" e selecione a opção: Adicionar
→ Formulario



Para este segundo formulário dê o nome de FrmCadastro.

Altere as propriedades abaixo:

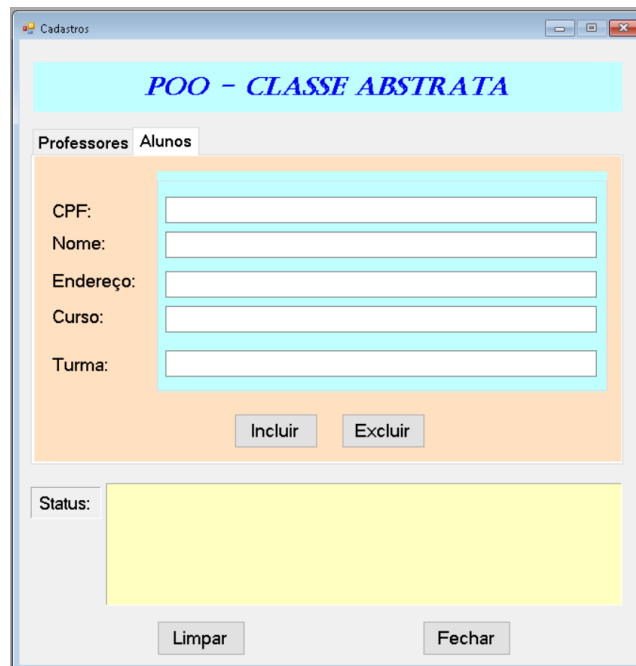
CONTROLE	PROPRIEDADE	CONTEÚDO
Form	Name	FrmCadastro
	MaximizeBox	False
	StartPosition	CenterScreen
	Text	Cadastros

Inclua o controle “TabControl” (funcionará como “abas”).

Vamos configurar as “abas”. Selecione o controle “TabControl” e clique na propriedade “TabPage” e altere, respectivamente, o “text” para “Professores” e “Alunos”.

Na aba “Professores”, inclua os controles abaixo:

Na aba “Alunos”:

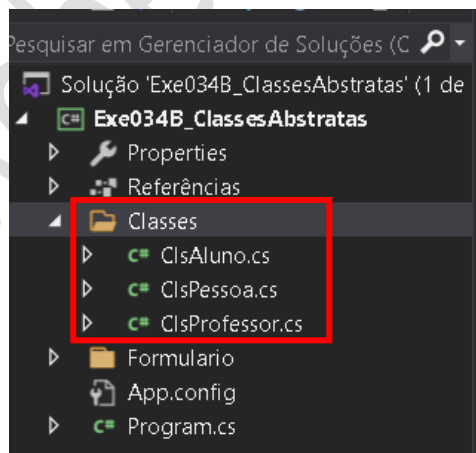


Finalizada a etapa de criação e configuração dos formulários, vamos agora para a programação das classes!!

Vamos incluir três classes. Clique com o botão direito do mouse sobre a pasta “Classes”, selecione a opção Adicionar → Classe.

A primeira classe se chamará “ClsPessoa”, a segunda “ClsProfessor” e a terceira “ClsAluno”.

Após a criação das três classes, a pasta “Classes” deverá estar conforme modelo abaixo:



Vamos incluir a programação da classe “ClsPessoa” (classe abstrata que será a classe base para as demais).

```
namespace Exe034B_ClassesAbstratas.Classes
{
    public abstract class ClsPessoa
    {
        //Declaração dos atributos - campos
        private string cpf;
        private string nome;
        private string endereco;

        //Declaração das propriedades (gets / sets)
        public string Cpf { get => cpf; set => cpf = value; }
        public string Nome { get => nome; set => nome = value; }
        public string Endereco { get => endereco; set => endereco = value; }

        //Método construtor
        public ClsPessoa()
        {
            cpf = "";
            nome = "";
            endereco = "";
        }

        //Método abstrato que DEVERÁ ser implementado nas subclasses: Aluno e Professor
        public abstract string Incluir();


        //Método CONCRETO que será herdado pelas subclasses: Aluno e Professor
        public string Excluir()
        {
            return "Dados excluídos com sucesso!!!";
        }
    }
}
```

Classe "ClsProfessor" (que herda a classe abstrata "ClsPessoa"). Observe o destaque em "vermelho":

```
namespace Exe034B_ClassesAbstratas.Classes
{
    public class ClsProfessor : ClsPessoa
    {
        //Declaração dos atributos / campos
        private string licenciatura;
        private string categoria;

        //Declaração das propriedades
        public string Licenciatura { get => licenciatura; set => licenciatura = value; }
        public string Categoria { get => categoria; set => categoria = value; }

        //Declaração do construtor
    }
}
```

A diagram consisting of a horizontal line with an arrow pointing left to the word 'ClsPessoa' in the line 'public class ClsProfessor : ClsPessoa'. A vertical line extends upwards from the horizontal line, then turns left to point at the text 'Observe o destaque em "vermelho":'.

```
public ClsProfessor()
{
    licenciatura = "";
    categoria = "";
}

//implementação do método abstrato que está classe base - ClsPessoa
public override string Incluir()
{
    return "Dados do professor incluídos com sucesso:\n"
        + Cpf + "\n"
        + Nome + "\n"
        + Endereco + "\n"
        + Licenciatura + "\n"
        + Categoria;
}
}
```

Classe "ClsAluno" (que herda a classe abstrata "ClsPessoa"). Observe o destaque em "vermelho":

```
namespace Exe034B_ClasesAbstratas.Classes
{
    //Observe os dois pontos ":" após o nome da classe ClsAluno
    //--> indica que a classe ClsAluno está herdando da classe ClsPessoa
    public class ClsAluno : ClsPessoa
    {
        //Declaração dos atributos / campos
        private string curso;
        private string turma;

        //Declaração das propriedades
        public string Curso { get => curso; set => curso = value; }
        public string Turma { get => turma; set => turma = value; }

        //Declaração do construtor
        public ClsAluno()
        {
            curso = "";
            turma = "";
        }

        //implementação do método abstrato que está classe base - ClsPessoa
        public override string Incluir()
        {
            return "Dados do aluno incluídos com sucesso:\n"
                + Cpf + "\n"
                + Nome + "\n"
                + Endereco + "\n"
                + Curso + "\n"
        }
    }
}
```

```
+ Turma;  
    }  
}  
}
```

Programação do formulário:

```
using System;  
using System.Windows.Forms;  
using Exe034B_ClassesAbstratas.Classes; //using necessário para acessar as Classes
```

```
namespace Exe034B_ClassesAbstratas.Formulario  
{  
    public partial class FrmCadastro : Form  
    {  
        public FrmCadastro()  
        {  
            InitializeComponent();  
        }  
  
        private void BtnFechar_Click(object sender, EventArgs e)  
        {  
            this.Close();  
        }  
  
        private void BtnIncluir_Click(object sender, EventArgs e)  
        {  
            //Cria a instância da classe Professor  
            //Observe a inclusão do using nas linhas iniciais...  
            //  
            ClsProfessor ObjProfessor = new ClsProfessor();  
  
            //Envia informações para a classe professor  
            ObjProfessor.Cpf = TxtCpfProfessor.Text;  
            ObjProfessor.Nome = TxtNomePro.Text;  
            ObjProfessor.Endereco = TxtEnderecoPro.Text;  
            ObjProfessor.Licenciatura = TxtLicenciatura.Text;  
            ObjProfessor.Categoria = TxtCategoria.Text;  
  
            //executa o método incluir (sobrescrito) da classe Professor  
            LblStatus.Text = ObjProfessor.Incluir();  
        }  
  
        private void BtnExcluir_Click(object sender, EventArgs e)  
        {  
            //Cria a instância da classe Professor  
            ClsProfessor ObjProfessor = new ClsProfessor();  
  
            //Executa o método Excluir da classe base  
            if (TxtCpfProfessor.Text != "")  
            {
```


Prof. Roberto de Castro

```
//Envia informações para a classe professor
ObjProfessor.Cpf = TxtCpfProfessor.Text;
LblStatus.Text = ObjProfessor.Excluir();
}
else
{
    LblStatus.Text = "ATENÇÃO: Digite o número do CPF para exclusão!!";
}
}

private void BtnIncluirAluno_Click(object sender, EventArgs e)
{
    //Cria a instância da classe Aluno
    //Importante: Observe a inclusão do "using" nas linhas iniciais
    //
    ClsAluno ObjAluno = new ClsAluno();

    //Envia informações para a classe Aluno
    ObjAluno.Cpf = TxtCpfAluno.Text;
    ObjAluno.Nome = TxtNomeAluno.Text;
    ObjAluno.Endereco = TxtEnderecoAluno.Text;
    ObjAluno.Curso = TxtCurso.Text;
    ObjAluno.Turma = TxtTurma.Text;

    //executa o método incluir (sobrescrito) da classe Aluno
    LblStatus.Text = ObjAluno.Incluir();
}

private void BtnExcluirAluno_Click(object sender, EventArgs e)
{
    //Cria a instância da classe Aluno
    ClsAluno ObjAluno = new ClsAluno();

    //Executa o método Excluir da classe base
    if (TxtCpfAluno.Text != "")
    {
        //Envia informações para a classe Aluno
        ObjAluno.Cpf = TxtCpfAluno.Text;
        LblStatus.Text = ObjAluno.Excluir();
    }
    else
    {
        LblStatus.Text = "ATENÇÃO: Digite o número do CPF para exclusão!!";
    }
}

private void BtnLimpar_Click(object sender, EventArgs e)
{
    if (tabControl1.SelectedIndex == 0) //a guia Professores está ativa
    {
        foreach (TextBox CaixaTexto in GrpProfessor.Controls)
```

```
{
    CaixaTexto.Text = "";
}
else //a guia Alunos está ativa
{
    foreach (TextBox caixaTexto in GrpAluno.Controls)
    {
        caixaTexto.Text = "";
    }
}

LblStatus.Text = "";
}
}
```