Atividade Terreno

Terreno.cs

```
namespace Atividade_Terreno
{
    public class Terreno
    {
        private double frente;
        private double lateral;
        private int lote;
        private string quadra;
        private int tipo;
        private string bairro;


        public double Frente
        {
            get { return frente; }
            set { frente = value; }
        }

        public double Lateral
        {
            get { return lateral; }
            set { lateral = value; }
        }

        public int Lote
        {
            get { return lote; }
            set { lote = value; }
        }

        public string Quadra
        {
            get { return quadra; }
            set { quadra = value; }
        }

        public int Tipo
        {
            get { return tipo; }
            set { tipo = value; }
        }

        public string Bairro
        {
            get { return bairro; }
            set { bairro = value; }
        }

        public double CalcularArea()
        {
            return frente * lateral;
        }

        public double CalcularPerimetro()
        {
            return 2 * (frente + lateral);
        }
```

```
        public double IPTU()
        {
            if (tipo == 1) { return (frente * lateral) * 0.50; }
            else { return (frente * lateral) * 1.50; }
        }

        public Terreno()
        {
            frente = 10;
            lateral = 25;
            lote = 0;
            quadra = " ";
            tipo = 1;
            bairro = "Não Informado";
        }
    }
}
```

program.cs

```
namespace Atividade_Terreno
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Terreno T = new Terreno();

            Console.WriteLine("Frente: {0}", T.Frente.ToString());
            Console.WriteLine("Lateral: {0}", T.Lateral.ToString());
            Console.WriteLine("Lote: {0}", T.Lote.ToString());
            Console.WriteLine("Quadra: {0}", T.Quadra.ToString());
            Console.WriteLine("Tipo: {0}", T.Tipo.ToString());
            Console.WriteLine("Bairro: {0}", T.Bairro.ToString());
            Console.WriteLine("Area: {0}", T.CalcularArea().ToString("N2"));
            Console.WriteLine("Perimetro: {0}", T.CalcularPerimetro().ToString("N2"));
            Console.WriteLine("IPTU: {0}", T.IPTU().ToString("N2"));
        }
    }
}
```

Academia
frmAcademia.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Academia
```

```
{
    public partial class frmAcademia : Form
    {
        int Cortesia = 0;
        double valor = 0;

        public frmAcademia()
        {
            InitializeComponent();
        }

        private void frmAcademia_Load(object sender, EventArgs e)
        {
            txtNome.CharacterCasing = CharacterCasing.Upper;
            txtEmail.CharacterCasing = CharacterCasing.Lower;
            lstDisponiveis.Sorted = true;
            lstSelecionados.Sorted = true;
            Inicializar();

        }
        private void Inicializar()
        {
            txtNome.Clear();
            txtEmail.Clear();
            rbManha.Checked = false;
            rbTarde.Checked = false;
            rbNoite.Checked = false;
            chkCromoterapia.Checked = false;
            chkFengShui.Checked = false;
            chkMeditacao.Checked = false;
            chkTaiChiChuan.Checked = false;
            chkYoga.Checked = false;
            grpModalidades.Enabled = false;
            grpCortesia.Enabled = false;

            lstDisponiveis.Items.Clear();
            lstSelecionados.Items.Clear();
            lstDisponiveis.Items.Add("Musculação");
            lstDisponiveis.Items.Add("Natação");
            lstDisponiveis.Items.Add("Zumba");
            lstDisponiveis.Items.Add("Pilates");
            lstDisponiveis.Items.Add("RPM");
            lstDisponiveis.Items.Add("Ginástica Ritmica");
            lstDisponiveis.Items.Add("Hidroginástica");
            lstDisponiveis.Items.Add("Patinação");
            lstDisponiveis.Items.Add("Squash");

            valor = 0;
            Cortesia = 0;
            TestarSelecao();
        }
        private void TestaCampos()
        {
            if (txtNome.Text != string.Empty && txtEmail.Text != string.Empty &&
(rbManha.Checked || rbTarde.Checked || rbNoite.Checked))
            {
                grpModalidades.Enabled = true;
                grpCortesia.Enabled = true;
            }
            else
            {
                grpModalidades.Enabled = false;
                grpCortesia.Enabled = false;
            }
        }
```

```
        private void txtNome_TextChanged(object sender, EventArgs e)
        {
            TestaCampos();

        }

        private void txtEmail_TextChanged(object sender, EventArgs e)
        {
            TestaCampos();
        }

        private void rbManha_CheckedChanged(object sender, EventArgs e)
        {
            TestaCampos();
            if (valor > 0) TestarSelecao();
        }

        private void rbTarde_CheckedChanged(object sender, EventArgs e)
        {
            TestaCampos();
            if (valor > 0) TestarSelecao();

        }

        private void rbNoite_CheckedChanged(object sender, EventArgs e)
        {
            TestaCampos();
            if (valor > 0) TestarSelecao();
        }

        private void btnSelecionarUm_Click(object sender, EventArgs e)
        {
            if (lstDisponiveis.SelectedIndex > -1)
            {
                lstSelecionados.Items.Add(lstDisponiveis.SelectedItem);
                lstDisponiveis.Items.Remove(lstDisponiveis.SelectedItem);
                TestarSelecao();
            }
            else
            {
                MessageBox.Show("É necessário Selecionar um item", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnSelecionarTodos_Click(object sender, EventArgs e)
        {
            for (int i = 0; i < lstDisponiveis.Items.Count; i++)
            {
                lstSelecionados.Items.Add(lstDisponiveis.Items[i]);
            }
            lstDisponiveis.Items.Clear();
            TestarSelecao();
        }

        private void btnRecuperarUm_Click(object sender, EventArgs e)
        {
            if (lstSelecionados.SelectedIndex > -1)
            {
                lstDisponiveis.Items.Add(lstSelecionados.SelectedItem);
                lstSelecionados.Items.Remove(lstSelecionados.SelectedItem);
                TestarSelecao();
            }
            else
```

```
            {
                MessageBox.Show("É necessário Selecionar um item", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }


        }

        private void btnRecuperarTodos_Click(object sender, EventArgs e)
        {
            for (int i = 0; i < lstSelecionados.Items.Count; i++)
            {
                lstDisponiveis.Items.Add(lstSelecionados.Items[i]);
            }
            lstSelecionados.Items.Clear();
            TestarSelecao();
        }

        private void TestarSelecao()
        {
            int Selecionados = lstSelecionados.Items.Count;
            int Disponiveis = lstDisponiveis.Items.Count;
            btnSelecionarUm.Enabled = Disponiveis > 0;
            btnSelecionarTodos.Enabled = Disponiveis > 0;
            btnRecuperarUm.Enabled  = Selecionados > 0;
            btnRecuperarTodos.Enabled = Selecionados > 0;


            txtQuantidade.Text = Selecionados.ToString();

             if (Selecionados != 0)
             {
                 if (Selecionados < 3)
                     valor = 100;
                 else if (Selecionados < 5)
                     valor = 150;
                 else if (Selecionados < 7)
                     valor = 200;
                 else
                     valor = 250;
             }


             if (rbTarde.Checked)
                 valor = valor * 0.85;

             txtValor.Text = valor.ToString("N2");
        }


        private void chkYoga_CheckedChanged(object sender, EventArgs e)
        {
            if (chkYoga.Checked)
                Cortesia += 1;
            else
                Cortesia -= 1;
            testarCortesia();

        }

        private void chkMeditacao_CheckedChanged(object sender, EventArgs e)
        {

            if (chkMeditacao.Checked)
                Cortesia += 1;
```

```
            else
                Cortesia -= 1;

            testarCortesia();

        }

        private void chkTaiChiChuan_CheckedChanged(object sender, EventArgs e)
        {
            if (chkTaiChiChuan.Checked)
                Cortesia += 1;
            else
                Cortesia -= 1;
            testarCortesia();

        }

        private void chkFengShui_CheckedChanged(object sender, EventArgs e)
        {
            if (chkFengShui.Checked)
                Cortesia += 1;
            else
                Cortesia -= 1;
            testarCortesia();

        }

        private void chkCromoterapia_CheckedChanged(object sender, EventArgs e)
        {
            if (chkCromoterapia.Checked)
                Cortesia += 1;
            else
                Cortesia -= 1;
            testarCortesia();

        }

        private void btnLimpar_Click(object sender, EventArgs e)
        {
            Inicializar();
            txtNome.Focus();
        }



        private void btnSair_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("Confirma Saída?", "Sair", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
                Application.Exit();
        }

        void testarCortesia()
        {
            if (Cortesia >= 3)
            {
                chkCromoterapia.Enabled = chkCromoterapia.Checked;
                chkMeditacao.Enabled = chkMeditacao.Checked;
                chkTaiChiChuan.Enabled = chkTaiChiChuan.Checked;
                chkFengShui.Enabled = chkFengShui.Checked ;
                chkYoga.Enabled = chkYoga.Checked;

            }
```

```
            else
            {
                chkCromoterapia.Enabled = true;
                chkMeditacao.Enabled = true;
                chkTaiChiChuan.Enabled = true;
                chkFengShui.Enabled = true;
                chkYoga.Enabled = true;
            }
            txtCortesia.Text = Cortesia.ToString();
        }

    }
}
```

IMC

```
namespace IMC
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }


        private void btnLimpar_Click(object sender, EventArgs e)
        {
            //três maneiras diferentes de "limpar" um textbox
            txtAltura.Text = "";
            txtPeso.Text = string.Empty;
            txtIMC.Clear();
            txtClassificacao.Clear();
            //retorno o cursor para leitura de novo peso
            //o "foco" vai para o txtpeso
            txtPeso.Focus();

        }

        private void btnCalcular_Click(object sender, EventArgs e)
        {
            //declaração de variaveis
            double peso, altura, imc;
            peso = double.Parse(txtPeso.Text);
            altura = double.Parse(txtAltura.Text);
            //cálculo do IMC
            imc = peso / (altura * altura);
            //exibir o IMC
            txtIMC.Text = imc.ToString("N1");
            //exibir classificação de acordo com o valor do IMC
            if (imc < 20)
                txtClassificacao.Text = "magro";
            else
                if (imc <= 25)
                txtClassificacao.Text = "peso normal";
            else
                txtClassificacao.Text = "obeso";


        }

        private void btnCalcular_MouseEnter(object sender, EventArgs e)
        {
```

```
        }

        private void Form1_MouseEnter(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
         {
            txtIMC.ReadOnly = true;
            txtClassificacao.ReadOnly = true;
            btnCalcular.Enabled = false;

        }

        private void txtPeso_TextChanged(object sender, EventArgs e)
        {
            btnCalcular.Enabled = TestarCampos();
        }



        private void txtAltura_TextChanged(object sender, EventArgs e)
        {
            btnCalcular.Enabled = TestarCampos();
        }
        private bool TestarCampos()
        {
            if (txtPeso.Text != "" && txtAltura.Text != "")
                return true;
            else
                return false;
        }

        private void btnSair_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("Deseja Sair?", "Sair do Sistema",
                            MessageBoxButtons.YesNo,
                            MessageBoxIcon.Question,
                            MessageBoxDefaultButton.Button2) == DialogResult.Yes)
                Application.Exit();
        }
    }
}
```

Triangulo

```
namespace Triangulo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnClassificar_Click(object sender, EventArgs e)
        {
            double lado1, lado2, lado3;
            lado1 = double.Parse(txtLado1.Text);
            lado2 = double.Parse(txtLado2.Text);
            lado3 = double.Parse(txtLado3.Text);
```

```
            //verifico se é um triângulo
            if (lado1 < lado2 + lado3 && lado2 < lado1 + lado3 && lado3 < lado1 +
lado2)
            {
                //é um triângulo, classifico
                //verifico se tem três lados iguais
                if (lado1 == lado2 && lado2 == lado3)
                {
                    txtTipo.Text = "Equilátero";
                    pbTriangulo.Image = Properties.Resources.equilatero;
                }
                else if (lado1 == lado2 || lado2 == lado3 || lado1 == lado3)
                {
                    txtTipo.Text = "Isósceles";
                    pbTriangulo.Image = Properties.Resources.isosceles;
                }
                else
                {
                    txtTipo.Text = "Escaleno";
                    pbTriangulo.Image = Properties.Resources.escaleno;
                }
            }
            else
            {
                txtTipo.Text = "Não é Triângulo";
                pbTriangulo.Image = Properties.Resources.erro;
            }

        }

        private void Form1_Load(object sender, EventArgs e)
        {
            pbTriangulo.Image = null;
            btnClassificar.Enabled = false;
            txtTipo.ReadOnly = false;
        }

        public void TestarCampos()
        {
            if (txtLado1.Text != "" && txtLado2.Text != "" & txtLado3.Text != "")
                btnClassificar.Enabled = true;
            else
                btnClassificar.Enabled = false;
        }

        private void txtLado1_TextChanged(object sender, EventArgs e)
        {
            TestarCampos();
        }

        private void txtLado2_TextChanged(object sender, EventArgs e)
        {
            TestarCampos();
        }

        private void txtLado3_TextChanged(object sender, EventArgs e)
        {
            TestarCampos();
        }

        private void btnLimpar_Click(object sender, EventArgs e)
        {
            txtLado1.Text = string.Empty;
            txtLado2.Text = string.Empty;
            txtLado3.Text = string.Empty;
```

```
            txtLado1.Focus();
            pbTriangulo.Image = null;
        }

        private void txtLado1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if(char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar))
                {
                e.Handled = false;
            }
            else if (e.KeyChar == ',')
            {
                var tb = (TextBox)sender;
                //impede mais de uma vírgula
                if(tb.Text.Contains(","))
                    e.Handled = true;
                else
                    e.Handled |= false; // aceita a primeira vírgula
            }
            else
            { e.Handled = true; }

        }
    }
}
```