﻿using Arvores;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_16189\_16191\_Projeto4ED

{

public partial class frmManterCidades : Form

{

private ArvoreDeBusca<Cidade> cidades = new ArvoreDeBusca<Cidade>();

private FileStream arqCidade = null;

public frmManterCidades()

{

InitializeComponent();

LerArquivo();

}

private void LerArquivo()

{

OpenFileDialog dlgAbrir = new OpenFileDialog();

if (dlgAbrir.ShowDialog() == DialogResult.OK)

{

arqCidade = new FileStream(dlgAbrir.FileName, FileMode.OpenOrCreate);

NoArvore<Cidade> raiz = new NoArvore<Cidade>();

LeituraDaArvore(0, arqCidade.Length, ref raiz);

cidades.raiz = raiz;

}

}

public void LeituraDaArvore(long inicio, long fim, ref NoArvore<Cidade> atual)

{

if (inicio <= fim)

{

int meio = (int)(inicio + fim) / 2;

RegistroCidade regAluno = new RegistroCidade();

Cidade umaCidade = new Cidade();

regAluno.LerRegistro(arqCidade, meio, ref umaCidade);

atual = new NoArvore<Cidade>(umaCidade);

LeituraDaArvore(inicio, meio - 1, ref atual.esquerdo);

LeituraDaArvore(meio + 1, fim, ref atual.direito);

}

}

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

{

if (CidadeValida(txtCidade.Text))

{

IncluirCidade(txtCidade.Text);

MessageBox.Show("Cidade registrada com sucesso!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

MessageBox.Show("A cidade digitada não é válida!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

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

{

bool cidadesValidas = true;

if (! CidadeValida(txtOrigem.Text))

{

cidadesValidas = false;

MessageBox.Show("A cidade de origem não é válida!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

if (! CidadeValida(txtDestino.Text))

{

cidadesValidas = false;

MessageBox.Show("A cidade de destino não é válida!", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

if (cidadesValidas)

{

SalvarAresta(txtOrigem.Text, txtDestino.Text);

MessageBox.Show("Aresta registrada com sucesso!", "", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void SalvarAresta(string origem, string destino)

{

throw new NotImplementedException();

}

private void IncluirCidade(string cidade)

{

}

private bool CidadeValida(string cidade)

{

if (string.IsNullOrEmpty(cidade))

return false;

foreach (char letra in cidade)

if (char.IsDigit(letra))

return false;

return true;

}

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

{

}

//arvoreBinaria.DesenharArvore(true, arv.Raiz, pnlArvore.Width / 2, 0, Math.PI / 2, Math.PI / 2.5, 150);

}

}