

using System;

using System.Windows.Forms;

using System.IO;

namespace CollegeSearchWinForm

{

public partial class CollegeCityForm : Form

{

// declaring this string here so it is scoped for whole class. Load and Save can access

string filepath = @"C:\Users\benge\source\repos\CollegeSearch\CollegeSearchWinForm\bin\Debug\CollegeCity.txt";

public CollegeCityForm()

{

InitializeComponent();

}

/// <summary>

/// Runs on load of application. Open, reads, and closes the .txt file

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

// use streamreader to read

StreamReader inputFile;

try

{

// open file

inputFile = File.OpenText(filepath);

// read every item out of the file

while (!inputFile.EndOfStream)

{

// adds each item to the LstBox

LstBoxColleges.Items.Add(inputFile.ReadLine());

}

//close file

inputFile.Close();

}

catch (Exception ex)

{

// would show on read error

MessageBox.Show(ex.Message);

}

}

/// <summary>

/// Saves the user input to the .txt file when the Save button is clicked

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

// use streamwriter to write

StreamWriter outputFile;

try

{

// obtain user input and format for save

string inputToSave = $"{TxtBoxCollege.Text}, {TxtBoxCity.Text}";

// open

outputFile = File.AppendText(filepath);

// use

outputFile.WriteLine(inputToSave);

// close

outputFile.Close();

// clear results and reset cursor focus to first box

TxtBoxCollege.Clear();

TxtBoxCity.Clear();

TxtBoxCollege.Focus();

// update LstBox

LstBoxColleges.Items.Add(inputToSave);

// notify user of successful svae

MessageBox.Show($"{inputToSave} was successfully saved");

}

catch (Exception ex)

{

// would show on write error

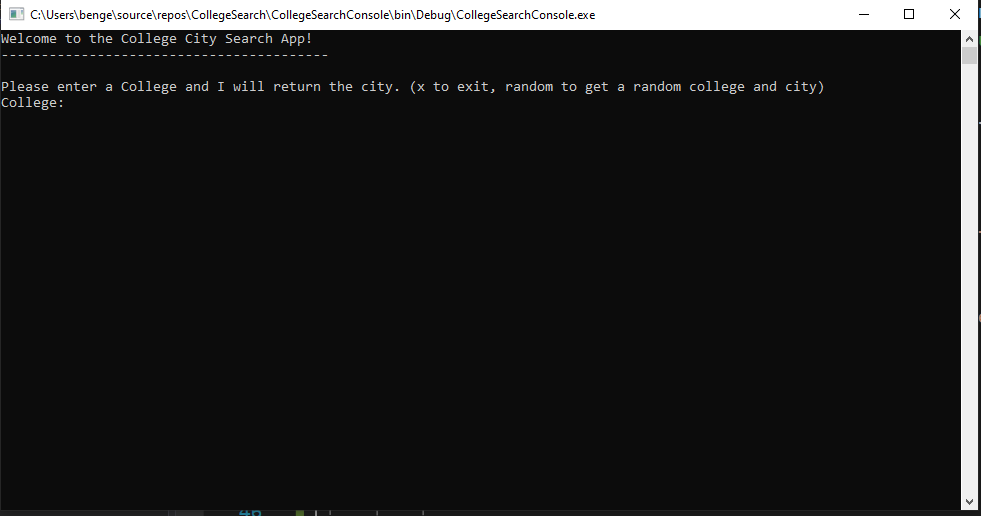
MessageBox.Show(ex.Message);

}

}

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CollegeSearchConsole

{

class Program

{

static void Main(string[] args)

{

// call the constructor method to open the file and create a dictionary

CollegeLookup collegeCity = new CollegeLookup();

Console.WriteLine("Welcome to the College City Search App!");

Console.WriteLine("-----------------------------------------");

Console.WriteLine();

Console.WriteLine("Please enter a College and I will return the city. (x to exit, random to get a random college and city)");

Console.Write("College: ");

// waits for user input

string userInput = Console.ReadLine();

// Need an infinite loop to keep the console open. Otherwise it closes automatically once the code runs

while (userInput != "x")

{

// passes the userInput to SearchDict and returns the city

string city = collegeCity.SearchDict(userInput.ToUpper());

// uppercase version of userInput

string upperUserInput = userInput.ToUpper();

// gives the user the random college and city

if(upperUserInput == "RANDOM")

{

Console.WriteLine($"Random College: {city}");

}

else

{

// if a valid value is returned then return the string below

if (city != null)

{

Console.WriteLine($"The location of {userInput} is{city}");

}

else

{

Console.WriteLine($"No results were found for {userInput}");

}

}

Console.WriteLine();

Console.WriteLine("Please enter a College and I will return the city. (x to exit, random to get a random college and city)");

Console.Write("College: ");

// waits for user input

userInput = Console.ReadLine();

}

}

}

}

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace CollegeSearchConsole

{

class CollegeLookup

{

// holds the filepath for reference

string filepath = @"C:\Users\benge\source\repos\CollegeSearch\CollegeSearchWinForm\bin\Debug\CollegeCity.txt";

// calls the constructor method for a dictionary with a key value pair

Dictionary<string, string> collegeCityDict = new Dictionary<string, string>();

/// <summary>

/// this is a constructor method because it has the same name as the class

/// </summary>

public CollegeLookup()

{

// shows we are reading the file. Incoming, not outgoing

StreamReader inputFile;

// try catch is used in case of read errors due to permissions

try

{

// open

inputFile = File.OpenText(filepath);

// read

while (!inputFile.EndOfStream)

{

// reads the .txt file and saves the key and values into the array

string[] collCity = inputFile.ReadLine().Split(',');

// takes the first item in the array and converts to uppercase

string college = collCity[0].ToUpper();

// saves the value in the city string

string city = collCity[1];

// adds the key value pair into the dictionary

collegeCityDict.Add(college, city);

}

// close

inputFile.Close();

}

catch (Exception ex)

{

// displays any error message

Console.WriteLine(ex.Message);

}

}

/// <summary>

/// Used to search the dictionary for the key (userInput) and returns the value

/// </summary>

/// <param name="userInput"></param>

/// <returns></returns>

public string SearchDict(string userInput)

{

if (userInput == "RANDOM")

{

// creates instance of random class

Random rand = new Random();

// returns the count of the Dictionary

int count = collegeCityDict.Count;

// returns a random number within the count of dictionary

int totalColleges = rand.Next(count);

// returns the key at the random index

string randCollege = collegeCityDict.ElementAt(totalColleges).Key;

// Looks up the key and returns the value

if (collegeCityDict.TryGetValue(randCollege, out string city)) { }

return $"{randCollege}, {city}";

}

else

{

string city = null;

// TryGetValue() takes the key and returns the corrosponding value

if (collegeCityDict.TryGetValue(userInput, out city))

{

return city;

}

// return null

return city;

}

}

}

}

