namespace Vezba\_30oktobar

{

internal class Program

{

static void Main(string[] args)

{

List<Vozilo> vozila = new List<Vozilo>();

// 1. Učitavanje linija iz data3.txt u listu

string path3 = @"c:\tmp\data3.txt";

List<string> linije = File.ReadAllLines(path3).ToList();

// 2. Obrada linija i kreiranje objekata Vozilo

foreach (var line in linije)

{

string[] data = line.Split('-');

//trimujemo da bi dobili podatke kako treba, da ne bude

//npr. "BMW ", nego "BMW"

for (int i = 0; i < data.Length; i++)

{

data[i] = data[i].Trim();

} if (data.Length == 4)

{

string tip = data[0];

double nabavnaCena = double.Parse(data[1]);

double prodajnaCena = double.Parse(data[2]);

int godinaProizvodnje = int.Parse(data[3]);

Vozilo vozilo = new Vozilo(tip, nabavnaCena, prodajnaCena, godinaProizvodnje);

vozila.Add(vozilo);

}

}

// 3. Ispis vozila sa prodajnom cenom 10% većom od nabavne cene

Console.WriteLine("\nVozila sa prodajnom cenom za 10% većom od nabavne cene:");

foreach (var vozilo in vozila)

{

if (vozilo.ProdajnaCena >= vozilo.NabavnaCena \* 1.10)

{

Console.WriteLine(vozilo);

}

}

// 4. Ispis vozila tipa BMW i godine proizvodnje od 2010 do 2015

Console.WriteLine("\nVozila tipa BMW proizvedena između 2010. i 2015:");

foreach (var vozilo in vozila)

{

if (vozilo.Tip == "BMW" && vozilo.GodinaProizvodnje >= 2010 && vozilo.GodinaProizvodnje <= 2015)

{

Console.WriteLine(vozilo);

}

}

// 5. Broj vozila po tipu

Console.WriteLine("\nBroj vozila po tipu:");

Dictionary<string, int> brojPoTipu = new Dictionary<string, int>();

foreach (var vozilo in vozila)

{

if (brojPoTipu.ContainsKey(vozilo.Tip))

{

brojPoTipu[vozilo.Tip]++;

}

else

{

brojPoTipu[vozilo.Tip] = 1;

}

}

foreach (var tip in brojPoTipu)

{

Console.WriteLine($"{tip.Key}: {tip.Value}");

}

// 6. Prosek nabavne cene za proizvoljni tip (Audi)

Console.WriteLine("\nProsečna nabavna cena za proizvoljni tip (Audi):");

string trazeniTip = "Audi";

double sumaCena = 0;

int brojVozilaTipa = 0;

foreach (var vozilo in vozila)

{

if (vozilo.Tip == trazeniTip)

{

sumaCena += vozilo.NabavnaCena;

brojVozilaTipa++;

}

}

double prosecnaCena = brojVozilaTipa > 0 ? sumaCena / brojVozilaTipa : 0;

Console.WriteLine($"Prosečna nabavna cena za {trazeniTip}: {prosecnaCena}");

/\*

// 1. Učitaj sadržaj fajla data1.txt u string

string path = @"c:\tmp\data1.txt";

string text1 = File.ReadAllText(path);

// Brojanje malih slova 'i'

int count\_i = CountOccurrences(text1, 'i');

Console.WriteLine($"Broj pojavljivanja slova 'i': {count\_i}");

// Brojanje malih i velikih slova 'c' i 'C'

int count\_c = CountOccurrences(text1, 'c') + CountOccurrences(text1, 'C');

Console.WriteLine($"Broj pojavljivanja slova 'c' (malo i veliko): {count\_c}");

// 2. Ispis teksta velikim slovima

Console.WriteLine("\nTekst velikim slovima:");

Console.WriteLine(text1.ToUpper());

// 3. Promeni tekst da sadrži samo velika slova

string upperText = text1.ToUpper();

Console.WriteLine("\nTekst sa samo velikim slovima:");

Console.WriteLine(upperText);

// 4. Učitaj sadržaj drugog fajla i dodaj ga prethodnom

string path2 = @"c:\tmp\data2.txt";

string text2 = File.ReadAllText(path2);

string combinedText = upperText + text2;

Console.WriteLine("\nSpojeni tekst:");

Console.WriteLine(combinedText);

Console.WriteLine($"Dužina spoženog teksta: {combinedText.Length}");

// 5. Pronađi lokacije reči "the"

Console.WriteLine("\nLokacije reči 'the':");

int index = combinedText.IndexOf("the");

while (index != -1)

{

Console.WriteLine(index);

index = combinedText.IndexOf("the", index + 1);

}

// 6. Prvih 20 karaktera u novom nizu

string first20Chars = combinedText.Substring(0, 20);

Console.WriteLine("\nPrvih 20 karaktera:");

Console.WriteLine(first20Chars);

// 7. Rad sa nizom stringova

string[] str1 = { "jun", "jul", "maj", "avg", "sep" };

List<string> monthsList = new List<string>(str1);

// Obrnuti redosled

monthsList.Reverse();

Console.WriteLine("\nLista u obrnutom redosledu:");

foreach (string month in monthsList)

{

Console.WriteLine(month);

}

// Sortiraj listu

monthsList.Sort();

Console.WriteLine("\nSortirana lista:");

foreach (string month in monthsList)

{

Console.WriteLine(month);

}

// 8. Unija dve liste

List<int> list1 = new List<int> { 10, 11, 12, 13, 14 };

List<int> list2 = new List<int> { 15, 16, 17, 18, 19 };

// Unija bez duplikata

List<int> unionList = new List<int>(list1);

foreach (int item in list2)

{

if (!unionList.Contains(item))

{

unionList.Add(item);

}

}

Console.WriteLine("\nUnija liste 1 i liste 2:");

foreach (int number in unionList)

{

Console.WriteLine(number);

}

}

// Funkcija za brojanje pojavljivanja određenog karaktera

static int CountOccurrences(string text, char character)

{

int count = 0;

foreach (char c in text)

{

if (c == character)

{

count++;

}

}

return count;

}

\*/

}

}

}

Klasa Vozilo

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vezba\_30oktobar

{

class Vozilo

{

// Privatni atributi

private string tip;

private double nabavnaCena;

private double prodajnaCena;

private int godinaProizvodnje;

// Konstruktor sa parametrima

public Vozilo(string tip, double nabavnaCena, double prodajnaCena, int godinaProizvodnje)

{

this.tip = tip;

this.nabavnaCena = nabavnaCena;

this.prodajnaCena = prodajnaCena;

this.godinaProizvodnje = godinaProizvodnje;

}

// Get i set metode za svaki atribut

public string Tip

{

get { return tip; }

set { tip = value; }

}

public double NabavnaCena

{

get { return nabavnaCena; }

set { nabavnaCena = value; }

}

public double ProdajnaCena

{

get { return prodajnaCena; }

set { prodajnaCena = value; }

}

public int GodinaProizvodnje

{

get { return godinaProizvodnje; }

set { godinaProizvodnje = value; }

}

// ToString metoda koja vraća atribute u string formatu

public override string ToString()

{

return $"Tip: {tip}, Nabavna Cena: {nabavnaCena}, Prodajna Cena: {prodajnaCena}, Godina Proizvodnje: {godinaProizvodnje}";

}

}

}