Klasa Student

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Oktobar\_25

{

internal class Student

{

private string ime;

private string prezime;

private string grad;

private double ocena1;

private double ocena2;

private double ocena3;

private double ocena4;

public string Ime {

get { return ime; }

set { ime = value; }

}

public string Prezime

{

get { return prezime; }

set { prezime = value; }

}

public string Grad {

get { return grad; }

set { grad = value; }

}

public double Ocena1 {

get { return ocena1; }

set { ocena1 = value; }

}

public double Ocena2 {

get { return ocena2; }

set { ocena2 = value; }

}

public double Ocena3 {

get { return ocena3; }

set { ocena3 = value; }

}

public double Ocena4 {

get { return ocena4; }

set { ocena4 = value; }

}

public Student(string i, string p, string g, double o1, double o2, double o3, double o4)

{

ime = i;

prezime = p;

grad = g;

ocena1 = o1;

ocena2 = o2;

ocena3 = o3;

ocena4 = o4;

}

public override string ToString()

{

return Ime+" "+Prezime+" "+Grad+" "+", ocene: ["+Ocena1+", "+Ocena2+", "+", "+Ocena3+", "+Ocena4+"]";

}

public double Poeni()

{

return Math.Round((ocena1+ocena2+ocena3 + ocena4)\*2,2);

}

}

}

namespace Oktobar\_25

{

internal class Program

{

static void Main(string[] args)

{

string path = @"c:\users\sanjam\desktop\students\_data.txt";

List<Student> students = new List<Student>();

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

HashSet<string> names = new HashSet<string>();

string[] lines=File.ReadAllLines(path);

foreach (string line in lines)

{

var parts = line.Split(',');

if (parts.Length == 7)

{

var student=new Student (

parts[0],

parts[1],

parts[2],

double.Parse(parts[3]),

double.Parse(parts[4]),

double.Parse(parts[5]),

double.Parse(parts[6])

);

students.Add (student);

//brojanje koliko puta se koji grad javlja

if(counts.ContainsKey(student.Grad))

{

counts[student.Grad]++; //ako grad vec postoji u dictionary

}

else

{

counts[student.Grad] = 1; //dodavanje grada u dictionary

}

//lista jedinstvenih imena

names.Add(student.Ime);

}

}

//lista sa imenima i ukupno poena

List<double> studentiPoeni = new List<double>();

foreach(var student in students)

{

studentiPoeni.Add(student.Poeni() );

}

studentiPoeni.Sort();

studentiPoeni.Reverse();

Console.WriteLine("Lista");

foreach(var p in studentiPoeni)

{

Console.WriteLine(p);

}

Console.WriteLine("Broj pojavljivanja gradova: ");

foreach(var g in counts)

{

Console.WriteLine(g.Key + ": " + g.Value);

}

Console.WriteLine("Jedinstvena imena: ");

foreach (var g in names)

{

Console.WriteLine(g);

}

}

}

}