Уровень 1

using System;

struct Student

{

private string name1;

private int grade1;

private int missedClasses1;

public string Name => name1;

public int Grade => grade1;

public int MissedClasses => missedClasses1;

public Student(string name, int grade, int missedClasses)

{

name1 = name;

grade1 = grade;

missedClasses1 = missedClasses;

}

public void Print()

{

Console.WriteLine("имя: {0} \t балл: {1} \t пропуски: {2}", name1, grade1, missedClasses1);

}

}

class Program

{

static void Main(string[] args)

{

Student[] stu = new Student[7];

stu[0] = new Student("Kate", 2, 3);

stu[1] = new Student("Anna", 4, 5);

stu[2] = new Student("Kirill", 2, 7);

stu[3] = new Student("Lev", 3, 1);

stu[4] = new Student("Maria", 2, 6);

stu[5] = new Student("Mira", 4, 2);

stu[6] = new Student("Aleksey", 3, 4);

for (int i = 0; i < stu.Length - 1; i++)

{

int maxMissed = stu[i].MissedClasses;

int maxIndex = i;

for (int j = i + 1; j < stu.Length; j++)

{

if (stu[j].MissedClasses > maxMissed && stu[j].Grade == 2)

{

maxMissed = stu[j].MissedClasses;

maxIndex = j;

}

}

Student temp = stu[maxIndex];

stu[maxIndex] = stu[i];

stu[i] = temp;

}

Console.WriteLine("Неуспевающие студенты:");

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

{

if (stu[i].Grade == 2)

{

stu[i].Print();

}

}

Console.ReadKey();

}

}

Уровень 2

using System;

using System.Security.Cryptography.X509Certificates;

struct ChessTournoment

{

private string \_name;

private int \_wincount;

private double \_drawcount;

private int \_loosecount;

private double \_finalscore;

public ChessTournoment(string name, int wins, double draws, int looses)

{

\_name = name;

\_wincount = wins;

\_drawcount = draws;

\_loosecount = looses;

\_finalscore = wins \* 1 + draws / 2;

}

public double Finalscore => \_finalscore;

public void Print()

{

// name не больше 5 символов иначе вывод ломаесться

Console.WriteLine("{0,-5} | {1,-3} | {2,-3} | {3,-7} | {4,-3}", \_name, \_wincount, \_drawcount, \_loosecount, \_finalscore);

}

}

internal class Program

{

static void Sort(ChessTournoment[] participants)

{

for (int i = 1; i < participants.Length; i++)

{

for (int j = 1; j < participants.Length; j++)

{

if (participants[j].Finalscore > participants[j - 1].Finalscore)

{

ChessTournoment temp = participants[j];

participants[j] = participants[j - 1];

participants[j - 1] = temp;

}

}

}

}

static void Main(string[] args)

{

ChessTournoment[] participants = new ChessTournoment[7]

{

new ChessTournoment("Vlad", 1,2,7),

new ChessTournoment("Pavel", 4,3,2),

new ChessTournoment("Lera", 0,2,9),

new ChessTournoment("Diana", 2,8,1),

new ChessTournoment("Vera", 3,3,7),

new ChessTournoment("Denis", 3,6,3),

new ChessTournoment("Anton", 4,6,3)

};

Console.WriteLine("Список участников");

Console.WriteLine("Имена Победы Ничьи Поражения Результаты");

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

{

participants[i].Print();

}

Sort(participants);

Console.WriteLine("Sorted");

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

{

participants[i].Print();

}

Console.ReadKey();

}

}

Уровень 3

using System;

struct Group

{

public string Name;

public int[] Scores;

public Group(string name, int[] scores)

{

Name = name;

Scores = scores;

}

public double GetAverageScore()

{

int sum = 0;

foreach (int score in Scores)

{

sum += score;

}

return (double)sum / Scores.Length;

}

}

class Program

{

static void Main()

{

Group[] groups = new Group[]

{

new Group("Группа 1", new int[] {5, 4, 4, 4, 5}),

new Group("Группа 2", new int[] {5, 4, 3, 3, 4}),

new Group("Группа 3", new int[] {5, 5, 4, 4, 3})

};

// Сортируем группы по убыванию среднего балла

Array.Sort(groups, (x, y) => y.GetAverageScore().CompareTo(x.GetAverageScore()));

// Выводим результаты в виде таблицы

Console.WriteLine("Группа\t\tСредний балл");

foreach (var group in groups)

{

Console.WriteLine($"{group.Name}\t{group.GetAverageScore()}");

}

}

}