№3 ( уровень 1)

public struct Answer

{

private string \_name;

private int \_count;

public Answer(string Name, int Count)

{

\_name = Name;

\_count = Count;

}

public string Name => \_name;

public int Count => \_count;

public void PrintAnswer()

{

Console.Write($"Name: {\_name}, Count: {\_count} ");

}

}

class Program

{

static void Main()

{

Answer[] Ans = new Answer[7]

{

new Answer("Bob", 10),

new Answer("Mark", 20),

new Answer("Dylan", 14),

new Answer("Kate", 17),

new Answer("Mary", 11),

new Answer("John", 5),

new Answer("Pete", 8)

};

int SumAns = 0;

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

{

SumAns += Ans[i].Count;

}

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

{

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

{

if (Ans[i].Count < Ans[j].Count)

{

(Ans[i], Ans[j]) = (Ans[j], Ans[i]);

}

}

}

for(int i = 0; i < 5; i++)

{

Ans[i].PrintAnswer();

Console.Write($"Per cent: {Ans[i].Count / (SumAns \* 1.0) \* 100}%");

Console.WriteLine();

}

}

}

№5 ( уровень 2)

public struct Sportsman

{

private string \_name;

private int[] \_score;

private int \_distance;

private int \_totalScore;

public Sportsman(string Name, int[] Score, int Distance)

{

\_name = Name;

\_score = Score;

\_distance = Distance;

\_totalScore = CalculateTotalScore();

}

public string Name => \_name;

public int[] Score => \_score;

public int Distance => \_distance;

public int TotalScore => \_totalScore;

private int CalculateTotalScore()

{

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

{

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

{

if (Score[j] > Score[j + 1])

{

int temp = Score[j];

Score[j] = Score[j + 1];

Score[j + 1] = temp;

}

}

}

int sum = 0;

for (int i = 1; i < \_score.Length - 1; i++)

{

sum += \_score[i];

}

int distancePoints = 60 + ((\_distance - 120) \* 2);

return sum + distancePoints;

}

public void PrintResults()

{

Console.WriteLine($"Name: {\_name}, Total Score: {\_totalScore}");

}

}

class Program

{

static void Main()

{

Sportsman[] sp = new Sportsman[7]

{

new Sportsman("Bob", new int[] { 14, 19, 18, 17, 16 }, 110),

new Sportsman("Mark", new int[] { 12, 20, 15, 16, 17 }, 124),

new Sportsman("Dylan", new int[] { 13, 16, 13, 15, 17 }, 120),

new Sportsman("Kate", new int[] { 17, 19, 18, 17, 18 }, 128),

new Sportsman("Mary", new int[] { 17, 16, 14, 15, 16 }, 121),

new Sportsman("John", new int[] { 17, 16, 17, 15, 17 }, 122),

new Sportsman("Pete", new int[] { 18, 17, 20, 18, 19 }, 130)

};

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

{

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

{

if (sp[i].TotalScore < sp[j].TotalScore)

{

Sportsman temp = sp[i];

sp[i] = sp[j];

sp[j] = temp;

}

}

}

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

{

Console.WriteLine($"Place {i + 1}:");

sp[i].PrintResults();

}

}

}

№5 ( уровень 3)

struct Team

{

private string \_name;

private int \_goals\_scored;

private int \_goals\_conceded;

private int \_points;

public Team(string name, int goalsScored, int goalsConceded)

{

\_name = name;

\_goals\_scored = goalsScored;

\_goals\_conceded = goalsConceded;

if (goalsScored > goalsConceded)

{

\_points = 3;

}

else if (goalsScored == goalsConceded)

{

\_points = 1;

}

else

{

\_points = 0;

}

}

public string Name => \_name;

public int GoalsScored => \_goals\_scored;

public int GoalsConceded => \_goals\_conceded;

public int Points => \_points;

public void PrintTeam()

{

Console.WriteLine($"Team: {\_name}, Points: {\_points}");

}

}

class Program

{

static void Main()

{

Team[] teams = new Team[7]

{

new Team("Team1", 4, 2),

new Team("Team2", 3, 3),

new Team("Team3", 1, 4),

new Team("Team4", 1, 1),

new Team("Team5", 5, 2),

new Team("Team6", 2, 5),

new Team("Team7", 3, 1)

};

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

{

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

{

if (CompareTeams(teams[j], teams[j + 1]) > 0)

{

Team temp = teams[j];

teams[j] = teams[j + 1];

teams[j + 1] = temp;

}

}

}

for (int i = 0; i < 7; i++)

{

Console.Write($"Place {i + 1}: ");

teams[i].PrintTeam();

}

}

static int CompareTeams(Team team1, Team team2)

{

if (team1.Points != team2.Points)

return team2.Points.CompareTo(team1.Points);

else

return (team2.GoalsScored - team2.GoalsConceded).CompareTo(team1.GoalsScored - team1.GoalsConceded);

}

}