using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyProject

{

class Player

{

public string Name { get; set; }

public string Surname { get; set; }

public DateTime Birthdate { get; set; }

public int Score { get; set; }

public override string ToString()

{

return $"{Name} {Surname} {Birthdate} {Score}";

}

}

public class Program

{

static void Main(string[] args)

{

var players = new List<Player>

{

new Player

{

Name="John",

Surname="Johnlu",

Birthdate=new DateTime(1995,8,15),

Score=100

},

new Player

{

Name="Aysel",

Surname="Eliyeva",

Birthdate=new DateTime(2000,10,11),

Score=65

},

new Player

{

Name="John",

Surname="Mammadov",

Birthdate=new DateTime(1998,7,1),

Score=50

},

new Player

{

Name="Rafiq",

Surname="Mammadli",

Birthdate=new DateTime(1999,1,18),

Score=60

},

new Player

{

Name="Eli",

Surname="Eliyev",

Birthdate=new DateTime(1999,10,16),

Score=70

},

new Player

{

Name="Mehemmed",

Surname="Mehemmedli",

Birthdate=new DateTime(1997,3,4),

Score=90

},

new Player

{

Name="Nezrin",

Surname="Memmedova",

Birthdate=new DateTime(1990,4,28),

Score=80

},

};

//players.ForEach(delegate (Player p)

//{

// Console.Write(p.Name+" ");

// Console.WriteLine(p.Birthdate.DayOfWeek);

//});

//var only2000Greater=players.FindAll(delegate (Player player)

//{

// return player.Birthdate.Year >= 2000;

//});

//only2000Greater.ForEach(delegate (Player player)

//{

// Console.WriteLine(player);

//});

//var list = players.Select(delegate (Player p)

//{

// return p.Name + " " + p.Birthdate.ToLongDateString();

//}).ToList();

//list.ForEach(delegate (string p)

//{

// Console.WriteLine(p);

//});

//Lambda Expression

//players.ForEach(p =>

//{

// Console.WriteLine(p.Name+" "+p.Surname);

//});

//var specialPlayers = players.FindAll(p => p.Name.ToLower().Contains("a"));

//specialPlayers.ForEach(p =>

//{

// Console.WriteLine(p.Name);

//});

//var specialPlayers = players.FindAll(p=>p.Name.StartsWith("A") || p.Name.StartsWith("N"));

//specialPlayers.ForEach(p =>

//{

// Console.WriteLine(p.Name);

//});

//LINQ

//var list=players.Select(p=>p.Birthdate).ToList();

//list.ForEach(b => Console.WriteLine(b));

//var count = players.Count(p => p.Birthdate.Year >= 2000);

//var count = players.Count(p => p.Surname.Length>8);

//Console.WriteLine(count);

//var player = players.First(p => p.Name == "John123");

//Console.WriteLine(player);

//var player = players.FirstOrDefault(p => p.Surname == "Rafiqli");

//if (player == null)

//{

// Console.WriteLine("We did not find");

//}

//else

//{

// Console.WriteLine(player);

//}

//var player = players.Single(p => p.Name == "John");

//Console.WriteLine(player);

//var player = players.SingleOrDefault(p => p.Name == "John");

//if(player == null)

//{

// Console.WriteLine("We did not find");

//}

//else

//{

// Console.WriteLine(player);

//}

//First FirstOrDefault Single SingleOrDefault

//var totalScore = players.Sum(p => p.Score);

//Console.WriteLine(totalScore);

//var avg = players.Average(p => p.Score);

//Console.WriteLine(avg);

//var items=players

// .OrderByDescending(p=>p.Name)

// .ToList();

//items.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//var items = players

// .OrderByDescending(p => p.Score)

// .ToList();

//items.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//Console.WriteLine();

//Console.WriteLine();

//items = items.Take(3).ToList();

//items.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//players.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//Console.WriteLine();

//Console.WriteLine();

////items = items.Skip(5).ToList();

//players = players

// .OrderByDescending(i => i.Score)

// .SkipWhile(s => s.Score > 60)

// .ToList();

//players.ForEach(i =>

//{

// Console.WriteLine(i);

//});

players.ForEach(i =>

{

Console.WriteLine(i);

});

Console.WriteLine();

Console.WriteLine();

//players = players

// .OrderByDescending(i => i.Score)

// .TakeWhile(s => s.Score > 60)

// .ToList();

//players.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//var item = players.FirstOrDefault(p => p.Name == "Aysel");

//var index = players.IndexOf(item);

//Console.WriteLine(index);

//var index = players.FindIndex(p => p.Name == "Aysel");

//if(index == -1)

//{

// Console.WriteLine("We did not find");

//}

//else

//{

// Console.WriteLine($"We found at this index : {index}");

//}

//var item = players.FirstOrDefault(p => p.Name=="Aysel");

//var canDeleted=players.Remove(item);

//players.RemoveAt(3);

//var countOfDelete=players

// .RemoveAll(p => p.Name

// .ToLower()

// .Contains("a"));

//Console.WriteLine(countOfDelete);

//Console.WriteLine("After Delete");

//players.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//var items=players

// .Where(p=>p.Birthdate.Year>=1998)

// .ToList();

//items.ForEach(i =>

//{

// Console.WriteLine(i);

//});

//var items = players

// .Where(p => p.Birthdate.Year >= 1998)

// .OrderByDescending(p => p.Score)

// .Take(2)

// .ToList();

//items.ForEach(i =>

//{

// Console.WriteLine(i);

//});

}

}

}