using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Cs13

{

#region AnonymMethods

//internal class Program

//{

// public delegate void AnonimDelegate();

// public delegate void AnonimDelegate2(int no);

// public delegate int AnonimDelegate3(int num1, int num2);

// static void Foo()

// {

// Console.WriteLine("Hiii Anonims");

// }

// static void Main(string[] args)

// {

// //AnonimDelegate del = new AnonimDelegate(Foo);

// //del.Invoke();

// //AnonimDelegate del = new AnonimDelegate(delegate

// //{

// // Console.WriteLine("Hi Guys, What's up");

// //});

// //del.Invoke();

// //del += Foo;

// //Console.WriteLine();

// //del.Invoke();

// //Console.WriteLine();

// //del += delegate

// //{

// // Console.WriteLine("Goodbye!");

// //};

// //del.Invoke();

// //AnonimDelegate2 del = new AnonimDelegate2(delegate (int no)

// //{

// // Console.WriteLine(no);

// //}

// //);

// //del.Invoke(101);

// //AnonimDelegate3 del = new AnonimDelegate3(delegate (int n1, int n2)

// //{

// // Console.WriteLine(n1 \* n2 + 10 + n1 + n2);

// // return n1 \* n2 + 10 + n1 + n2;

// //});

// //del+=delegate(int n1, int n2)

// //{

// // Console.WriteLine(n1+n2);

// // return n1+n2;

// //};

// //del.Invoke(50,100);

// //Action action = new Action(Foo);

// //action += delegate()

// //{

// // Console.WriteLine("Hi Guys are you okay");

// //};

// //action.Invoke();

// //Func<string, int, bool> func = delegate (string name, int age)

// //{

// // Console.WriteLine($"Name : {name}");

// // Console.WriteLine($"Age : {age}");

// // return true;

// //};

// //func("Ayxan", 16);

// //func.Invoke("Ayxan", 20);

// }

//}

#endregion

#region Task

//class Student

//{

// public Student(string name, string surname, int age, string speciality)

// {

// Id = Guid.NewGuid();

// Name = name;

// Surname = surname;

// Age = age;

// Speciality = speciality;

// }

// public Guid Id { get; set; } = Guid.NewGuid();

// public string Name { get; set; }

// public string Surname { get; set; }

// public int Age { get; set; }

// public string Speciality { get; set; }

// public override string ToString()

// {

// return $"Id : {Id} \nName : {Name} \nSurname : {Surname} \nAge : {Age} \nSpeciality : {Speciality}";

// }

// //public void Show()

// //{

// // Console.WriteLine($"Id : {Id}");

// // Console.WriteLine("Student : ");

// // Console.WriteLine($"Name : {Name}");

// // Console.WriteLine($"Surname : {Surname}");

// // Console.WriteLine($"Age : {Age}");

// // Console.WriteLine($"Speciality : {Speciality}");

// //}

//}

//public delegate void MyDelegate();

//class Program

//{

// static void Main(string[] args)

// {

// Student student = new Student("John", "Johnlu", 22, "Programming");

// MyDelegate del = delegate

// {

// File.WriteAllText("info.log", student.ToString()) ;

// };

// del += delegate ()

// {

// File.WriteAllText("info.txt", student.ToString());

// };

// //del += delegate ()

// //{

// // File.WriteAllText("info.ayxan", student.ToString());

// //};

// del += delegate ()

// {

// Console.WriteLine(student);

// };

// del.Invoke();

// }

//}

#endregion

#region MyRegion

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}";

}

}

class Program

{

static void Main(string[] args)

{

List<Player> players = new List<Player>()

{

new Player()

{

Name = "John",

Surname = "Johnlu",

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

Score = 98

},

new Player()

{

Name = "Tural",

Surname = "Turalli",

BirthDate = new DateTime(1996,8,25),

Score = 76

},

new Player()

{

Name = "Memmed",

Surname = "Memmedli",

BirthDate = new DateTime(1985,4,15),

Score = 25

},

new Player()

{

Name = "Eli",

Surname = "Eliyev",

BirthDate = new DateTime(1998,4,4),

Score = 65

},

new Player()

{

Name = "John",

Surname = "Rafiqli",

BirthDate = new DateTime(1994,5,15),

Score = 34

},

};

//players.ForEach(delegate (Player p)

//{

// Console.WriteLine(p.Name);

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

//});

//var only1997Greater = players.FindAll(delegate (Player player)

//{

// return player.BirthDate.Year >= 1997;

//});

//only1997Greater.ForEach(delegate (Player p)

//{

// Console.WriteLine(p);

//});

//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));

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

//specialPlayers.ForEach(p => Console.WriteLine(p));

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

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

//var count = players.Count(p => p.BirthDate.Year >= 1996);

//var count = players.Count(p => p.Surname.Length > 6);

//Console.WriteLine(count);

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

//Console.WriteLine(player);

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

//if (player == null)

//{

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

//}

//else

//{

// Console.WriteLine(player);

//}

//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(p => Console.WriteLine(p));

//Console.WriteLine();

//Console.WriteLine();

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

//items.ForEach(p => Console.WriteLine(p));

//var items = players

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

// .ToList();

//players.ForEach(p => Console.WriteLine(p));

//Console.WriteLine();

//Console.WriteLine();

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

//players = players.OrderByDescending(s => s.Score).ToList();

//players = players.SkipWhile(s => s.Score > 50).ToList();

//players.ForEach(p => Console.WriteLine(p));

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

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

//Console.WriteLine(deleted);

}

}

#endregion

}