using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Collections.ObjectModel;  
using System.Collections.Specialized;  
using System.Threading.Channels;  
  
namespace OOP12  
{  
 class Test : IEquatable<Test>, IComparable<Test>  
 {  
 private static int *\_id*;  
  
 public readonly int id;  
 public int Mark { get; set; }  
 public string Name { get; set; }  
 static Test() => *\_id* = 0;  
  
 public Test(int mark, string name)  
 {  
 id = ++*\_id*;  
 Name = name;  
 Mark = mark;  
 }  
  
 public override string ToString() => $"Name of test - \"{Name}\". The highest mark you can get is {Mark}";  
  
 public int CompareTo(Test other) => Equals(null, other) ? 0 : Mark.CompareTo(other.Mark);  
 public bool Equals(Test other) => !Equals(null, other) && this.Name.Equals(other.Name);  
  
 public override bool Equals(object obj) => !Equals(null, obj) && Equals(obj as Test);  
  
 public override int GetHashCode() => id;  
 }  
 class Program  
 {  
 delegate void OneParameter<T>(T obj);  
  
 delegate void NoParameters();  
  
 delegate void OneParameterBuiltIn();  
  
 delegate void NoParametersBuiltIn();  
  
 static void Main(string[] args)  
 {  
 OneParameter<int> myAction = value => { Console.WriteLine(value); };  
 myAction?.Invoke((new Random()).Next());  
  
 NoParameters myAction2;  
 myAction2 = delegate { Console.WriteLine("Anonymous method"); };  
 myAction2 += () => { Console.WriteLine("Lambda-expression"); };  
 myAction2?.Invoke();  
  
 Action action1 = () => { Console.WriteLine("Some method"); };  
 action1?.Invoke();  
  
 Action<int> action2 = value => { Console.WriteLine(value); };  
 action2?.Invoke((new Random()).Next());  
  
 object[] arrayObjects = new object[] {"Yehor", 1, 1.0f};  
 Action<object> show = item => Console.WriteLine(item);  
 Array.ForEach(arrayObjects, show);  
  
 List<Test> testsList = new List<Test>(new[]  
 {  
 new Test(1, "First"),   
 new Test(2, "Second"),  
 });  
 Action<List<Test>> sortListAction = value => {  
 foreach (var item in value)  
 {  
 Console.WriteLine(item);  
 }  
 };  
 sortListAction(testsList);  
  
 Test[] tests = new Test[]  
 {  
 new Test(2, "First"),   
 new Test(2, "Second"),  
 };  
  
 Test find = (Array.Find<Test>(tests, test => test.Mark== 2));  
 Console.WriteLine(find);  
 Action <string>del5 =name=>Console.WriteLine("Train "+name+" is in garage");  
   
 del5("smth");  
   
   
 Action<List<Test>> sorting = listTests =>  
 {  
 listTests.Sort((train1, train2) => train1.Mark.CompareTo(train2.Mark));  
 };  
   
 sorting(testsList);  
  
 foreach(var i in testsList)  
 {  
 Console.WriteLine(i);  
 }  
  
 Func<List<Test>, List<Test>> biggestAmount = listTrains => listTrains.FindAll(train => train.Mark >= 2);  
 testsList = biggestAmount(testsList);  
  
 foreach (var i in testsList)  
 {  
 Console.WriteLine(i);  
 }  
 Console.ReadKey();  
 }  
  
 public static void PrintIndexAndValues(IEnumerable myList)  
 {  
 int i = 0;  
 foreach (Object obj in myList)  
 Console.WriteLine("\t[{0}]:\t{1}", i++, obj);  
 Console.WriteLine();  
 }  
  
 }  
}