

Interface IComparable

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Interface Comparable

[https://msdn.microsoft.com/en-us/library/system.icomparable\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.icomparable(v=vs.110).aspx)

```
public interface IComparable {  
    int CompareTo(object other);  
}
```

Problema motivador

Faça um programa para ler um arquivo contendo nomes de pessoas (um nome por linha), armazenando-os em uma lista. Depois, ordenar os dados dessa lista e mostra-los ordenadamente na tela. Nota: o caminho do arquivo pode ser informado "*hardcode*".

```
Maria Brown
Alex Green
Bob Grey
Anna White
Alex Black
Eduardo Rose
Willian Red
Marta Blue
Alex Brown
```

```
using System;
using System.IO;
using System.Collections.Generic;

namespace Course {
    class Program {
        static void Main(string[] args) {

            string path = @"c:\temp\in.txt";

            try {
                using (StreamReader sr = File.OpenText(path)) {
                    List<string> list = new List<string>();
                    while (!sr.EndOfStream) {
                        list.Add(sr.ReadLine());
                    }
                    list.Sort();
                    foreach (string str in list) {
                        Console.WriteLine(str);
                    }
                }
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

Outro problema

Faça um programa para ler um arquivo contendo funcionários (nome e salário) no formato .csv, armazenando-os em uma lista. Depois, ordenar a lista por nome e mostrar o resultado na tela. Nota: o caminho do arquivo pode ser informado "hardcode".

```
Maria Brown,4300.00
Alex Green,3100.00
Bob Grey,3100.00
Anna White,3500.00
Alex Black,2450.00
Eduardo Rose,4390.00
Willian Red,2900.00
Marta Blue,6100.00
Alex Brown,5000.00
```

```
using System.Globalization;

namespace Course {
    class Employee {

        public string Name { get; set; }
        public double Salary { get; set; }

        public Employee(string csvEmployee) {
            string[] vect = csvEmployee.Split(',');
            Name = vect[0];
            Salary = double.Parse(vect[1], CultureInfo.InvariantCulture);
        }

        public override string ToString() {
            return Name + ", " + Salary.ToString("F2", CultureInfo.InvariantCulture);
        }
    }
}
```

Interface IComparable

```
namespace System {
    public interface IComparable {
        int CompareTo(object obj);
    }
}
```

```
Console.WriteLine("maria".CompareTo("alex"));
Console.WriteLine("alex".CompareTo("maria"));
Console.WriteLine("maria".CompareTo("maria"));
```

Output:

```
1
-1
0
```

Value	Meaning
Less than zero	The current instance precedes the object specified by the <code>CompareTo</code> method in the sort order.
Zero	This current instance occurs in the same position in the sort order as the object specified by the <code>CompareTo</code> method.
Greater than zero	This current instance follows the object specified by the <code>CompareTo</code> method in the sort order.

```
using System;
using System.Globalization;

namespace Course {
    class Employee : IComparable {

        public string Name { get; set; }
        public double Salary { get; set; }

        public Employee(string csvEmployee) {
            string[] vect = csvEmployee.Split(',');
            Name = vect[0];
            Salary = double.Parse(vect[1], CultureInfo.InvariantCulture);
        }

        public override string ToString() {
            return Name + ", " + Salary.ToString("F2", CultureInfo.InvariantCulture);
        }

        public int CompareTo(object obj) {
            if (!(obj is Employee)) {
                throw new ArgumentException("Comparing error: argument is not an Employee");
            }
            Employee other = obj as Employee;
            return Name.CompareTo(other.Name);
        }
    }
}
```