

PCP — Lecture 15

Fall 2020 November 17, 2020

1 Static

When we write:

```
1 Console.WriteLine(Math.PI);
```

The `Math` actually refers to *a class*, and not to *an object*. How is that?

Actually, everything in the `MATH` class is *static* (`public static class Math`), and the `PI` constant is actually public! (`public const double PI`).

Class attribute: can be static or not, public or private, a constant or variable.

```
1 public const double PI = 3.14159265358979;
```

We also have static methods:

```
1 Math.Min(x,y);
2 Math.Max(x,y);
3 Math.Pow(x,y);
```

A static member (variable, method, etc) belongs to the type of an object rather than to an instance of that type.

1.1 Static Class Members

Class member = methods and fields (attributes)

Motivation: the methods we are using the most (`WriteLine`, `ConsoleRead`) are static, but all the methods we are writing are not (they are “non-static”, or “instance”).

Static Method	Non-static Method
<code>ClassName.MethodName(arguments)</code>	<code>ObjectName.MethodName(arguments)</code>
<code>Math.Pow(2, 5)</code> ($2^5 = 32$)	<code>myRectangle.SetLength(5)</code>

A static class member is associated with the **class** instead of **with the object**.

\	Static Field	Non-static Field
Static method	OK	NO
Non-static method	OK	OK

2 A Static Class for Arrays

```
1  using System;
2      static class Lib
3      {
4          public static int ValueIsIndex(int[] arrayP)
5          {
6              int res = 0;
7              for (int i = 0; i < arrayP.Length; i++)
8                  if (arrayP[i] == i) res++;
9              return res;
10         }
11
12         public static bool AtLeastOneValueIsIndex(int[] arrayP)
13         {
14             return (ValueIsIndex(arrayP) > 0);
15         }
16
17         public static int ValueMatch(int[] arrayP1, int[] arrayP2)
18         {
19             int res = 0;
20             int smallestSize;
21             if (arrayP1.Length < arrayP2.Length) smallestSize = arrayP1.Length;
22             else smallestSize = arrayP2.Length;
23             for (int i = 0; i < smallestSize; i++)
24                 if (arrayP1[i] == arrayP2[i]) res++;
25             return res;
26         }
27     }
28
29 using System;
30 class Program
31 {
32     static void Main(string[] args)
33     {
34         int[] arrayA = {0, 3, 5, 12, 4, 5, 8 };
35         Console.WriteLine(Lib.ValueIsIndex(arrayA));
36         Console.WriteLine(Lib.AtLeastOneValueIsIndex(arrayA));
37
38         int[] arrayB = {3, 5, 4, 12, 5, 8 };
39         Console.WriteLine(Lib.ValueIsIndex(arrayB));
40         Console.WriteLine(Lib.AtLeastOneValueIsIndex(arrayB));
41
42         Console.WriteLine(Lib.ValueMatch(arrayA, arrayB));
43     }
44 }
```