Programming in C# (without Visual Studio)

Variables in C# and VBA:

* Both VBA and C# allow to define variables and arrays.
* Both allow object-oriented-programming, but the main difference is, that C# has the hierarchy (Method – Class – Namespace) whereas VBA has the hierarchy (Method – Module). VBA adds more complexity, because there also exist classes, but they are something additional. In VBA, every Method is first and foremost in a Module.
  + That is why in the C# examples I use **OtherClass.globalVariableInOtherClass** whereas in VBA I use **OtherModule.globalVariableInOtherModule**
* C# is a strongly typed language, which means, that when defining, we always have to add the type of the variable. (**string myString = “ABC”;**). VBA on the other hand has a more liberal approach. We can add the type of the variable, but we don’t have to. So both **dim myString as string** and **dim myString** are valid. VBA is even so liberal, that we don’t have to define variables. So **myString = “ABC”** is also valid.
* Public variables in C# and VBA differ in the sense, that in VBA the variables are really global. They even exists after the VBA program has finished. (Very bad ☺). In C#, a global variable exists within a class. After the program finishes the variable is deleted.
* When working with array, we always start with element 0. So the first element of an array has the index 0.
  + VBA: myStringArray(0) and C#: myStringArray[0]
* Pay attention to the slight difference when defining an array with a fixed number of items.
  + When we want an array with three elements, we write **Dim myStringArray2(2) As String** in VBA, whereas in C# we write **string[] myStringArray2 = new string[3];** So C# takes the number of elements, whereas VBA takes the number of the last index.

Good programming style:

According to Johann.

* Let us prefer hard-typed-programming.
  + In VBA this means:
    - always define variables and always add the type.
    - Use Option Explicit (this forces you to define variables
  + In C# this means:
    - C# always forces you to hard-type, but it is possible to use the type **var** which is de-facto soft-typing through the back-door. **Let’s not use var**
  + Lack of hard-typing means, that the program has to find the type of a variable on run-time, which makes the program slower.
  + In my experience there are only few cases where a variant array is needed.
* Try to use as few global variables as possible. Global variables make your code more enigmatic. To increase the readability of your code, call functions with parameters. In case, that your parameters are huge, make use of ByRef in VBA or ref/out in C#.