5/4/21 Data Types

Tuesday, May 4, 2021 11:48 AM

- Value Types
- Reference Types
- Everything inherits from System.Object

- Value types are stored in Stack, Reference in the Heap

Common Type System (CTS)
The CTS is a standard defn of the types in .NET compliant languages.
This allows for the Language Interoperability
Basically, in one solution, your projects can be written in multiple .NET compliant language.

Value Types
Analogous to java's primitives
Types that derive from System.Value
Stored in the stack and not the heap
This means that when accessing the value of a variable set to a value type, you get the value directly and not a reference to where the value is stored in heap. (Because variables are stored in stack, and the actual objects they reference are stored in heap)
There is no separate heap allocation or garbage collection overhead for value-type variables
Two categories
Structs - used to create custom value types
(INT32, Bool, Numeric primitives)
Enums - defines a set of named integral constants
(ie 1 - French, 2 - Italian, 3 - American, etc..)

Reference Types
A type that is defined as a class, delegate, array, or interface
At run time, when you declare a variable of a ref type, the var contains the value null until you explicitly create an object by using the new operator or assign it an object that has been created elsewhere
Note that ref types are stored in the heap. The stack holds the ref to a place in heap that contains the actual value of the object

What's the difference between the CLS and CTS?
CLS is common language specification that is the subset of CTS (Common Type System)