[0320] C# Basics: Variables and Data Types

Variables

Variables store data.

Variables must be declared (defined in your code) before they can be used. A typical variable declaration looks like this:

<type> <variable-name> <optional-initialization-stuff>;

where:

> ***<type>*** can be any valid datatype like int, string, Object, etc.
> ***<variable-name>*** can be any valid identifier name. Again, you want to make variable names as "readably useful" as possible!
> ***<optional-initialization-stuff>*** is typically an equals sign followed by a literal constant.

Examples:

```
int i = 0;
string name = "Fred";
boolean finished = false;
int[,] grid = new int[10,10];
```

C# is a **strongly typed** language - i.e., all variables must have be declared using a single data type and only data of that type can be stored in that variable.

Variables have **scope** - a region of your program where the variable can be used - usually limited to the function/method where the variable was declared.

All variables live on "**The Stack**" - a region of memory that grows and shrinks as your program runs. Some variables point to data that lives in "**The Heap**" (a different area of memory for storing large, long-lived items) but the variable itself (i.e, the pointer) lives on The Stack. (We will talk more about the Stack vs. the Heap later.)

The Data Type Hierarchy

C# supports many different datatypes. Many of the datatypes that come with C# are actually part of the .NET Framework (remember the CTS?). C# datatypes are related to

each other via the hierarchy shown below:

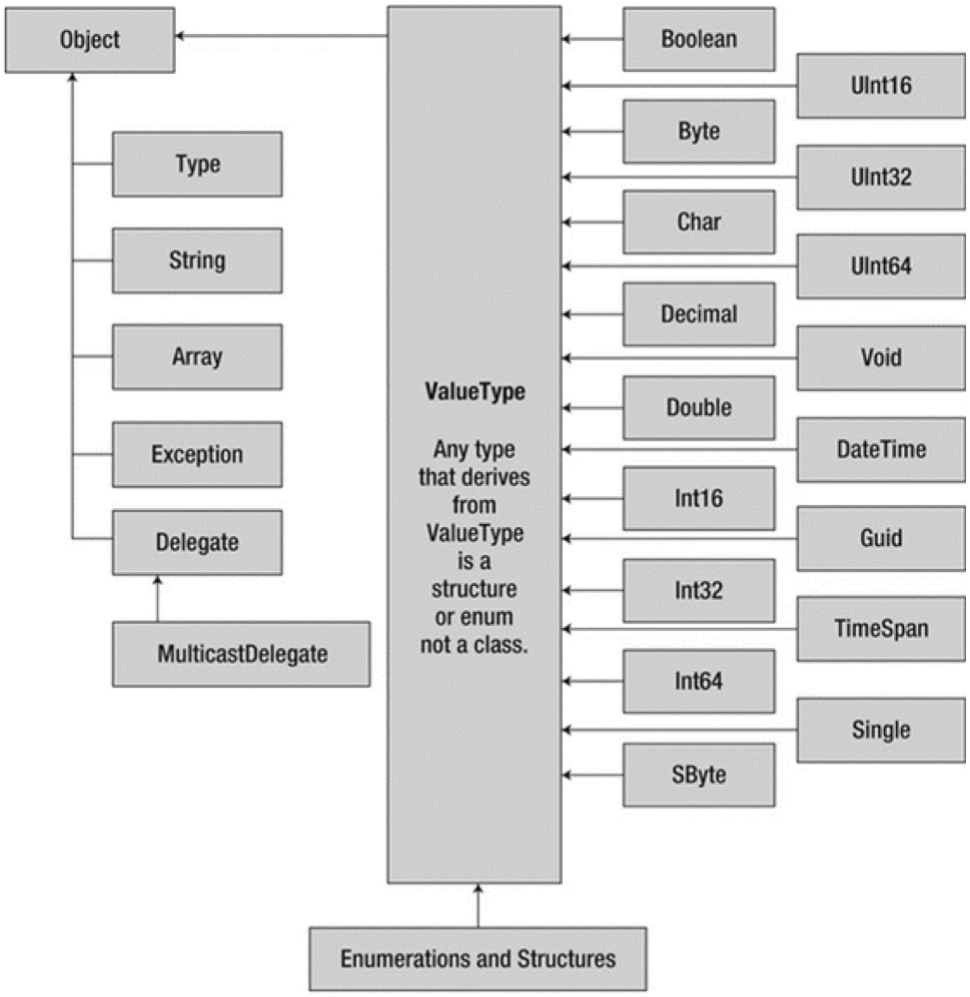

Figure 3-2. The class hierarchy of system types

Notes on Figure 3-2

- Everything is a subclass of Object. E-v-e-r-y-t-h-i-n-g.
- There are three categories of data types:
  - Object-like stuff (left side) - lives on the heap
  - Pure ValueType stuff (right side) - lives on the stack
  - Enum/Struct stuff (bottom) - also lives on the stack

Intrinsic Data Types

- Intrinsic data type names (int, long, double, etc.) are shorthand for .NET system data types
- You should declare and initialize local variables on the same line whenever possible

ValueTypes

- ValueType types are allocated on The Stack
- ValueType types include:
  - The Numerical Data Types
  - System.Boolean
  - System.Char
  - System.DateTime / System.TimeSpan

Notes

- Use "**<type>.Parse(string)**" to convert strings into a numeric type (once you know how to use exceptions).
  - FOR NOW, use "**<type>.TryParse(string, out variable)**" to convert strings since we don't know about exceptions yet.