C# and Microsoft .NET

Types

Trainer: Georgi Panayotov
E-mail: smg@smg-bg.net
Phone / Viber: +359887347912

PRAGMATIC
www.pragmatic.bg

Last time…

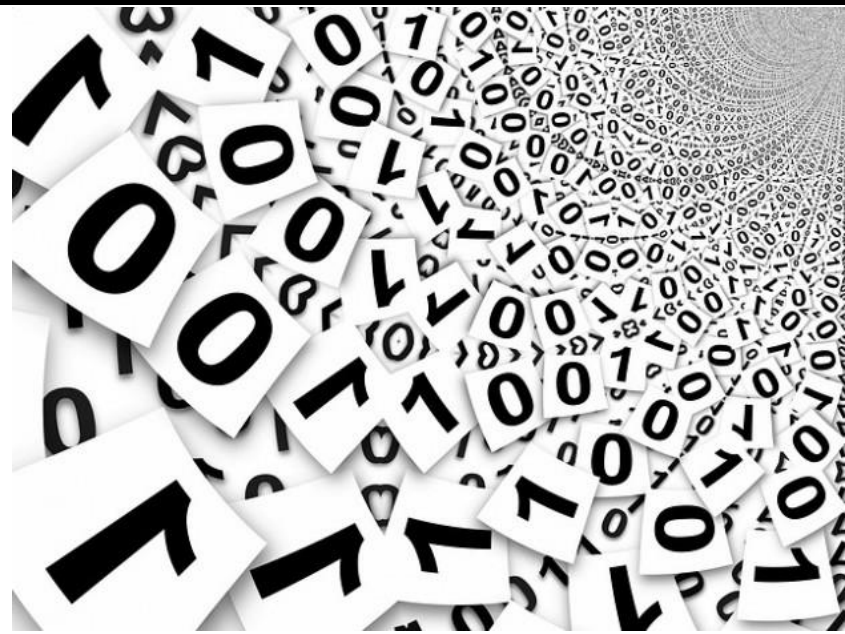
- http://learn.pragmatic.bg
- Overview of .NET framework
- Visual Studio 2017 Community
- Hello World
  - Application structure
  - Console output… continue

Console

- Working with the Console
  - The Console.ReadLine() method
  - The Console.Write(**value**) method
  - The Console.WriteLine(**value**) method

Bits and Bytes

- What is bit
- What is byte
- What about kilobyte, megabyte, gigabyte, terabyte, petabyte
- What are the real abbreviations
  - Is it kB, KB, kb or kbit?



Variables

- What is a Data Type
- What is a Variable
- Variable declaration syntax
  - **data_type variable_name;**
  - **data_type variable_name = value;**
  - **data_type variable_name_1, variable_name_2, ...;**
  - **data_type variable_name_1 = value_1, variable_name_2 = value_2, ...;**
- Constants
- Variable types
- Naming a variable
  - Start with a character a-z, A-Z, _, unicode is allowed but not recommended
  - Can contain _, numbers
  - Cannot be a c# keyword - https://msdn.microsoft.com/en-us/library/x53a06bb.aspx
  - Recommendations

Built-in Data Types

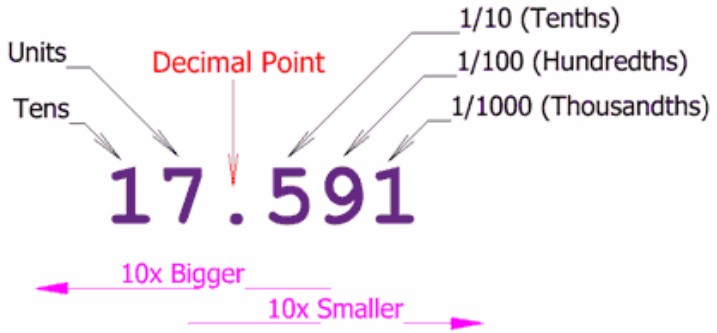
C# Type	Description	.NET Data Type
bool	1 bit - a boolean value	System.Boolean
byte	8 bit – integer value	System.Byte
sbyte	8 bit – signed integer value	System.SByte
char	16 bit character	System.Char
decimal	128 bit floating point value	System.Decimal
double	64 bit A floating point value	System.Double
float	32 bit floating point value (-3.40282e+038f to 3.40282e+038f)	System.Single
int	32 bit integer value	System.Int32
uint	32 bit unsigned integer value	System.UInt32
long	64 bit integer value	System.Int64
ulong	64 bit unsigned integer value	System.UInt64
short	16 bit integer value	System.Int16
ushort	16 bit unsigned integer value	System.UInt16
string	A string of characters	System.String

Integer Types

- sbyte (SByte) - 8bit
  - -128 to 127 (SByte.MinValue to SByte.MaxValue)
- byte (Byte) – 8bit unsigned
  - 0 to 255
- short (Int16) – 16bit
  - -32768 to 32767
- ushort (UInt16) – 16bit
  - 0 to 65535
- int (Int32) – 32bit
  - -2 147 483 648 to 2 147 483 647
  - int number = 16;
  - int hexNumber = 0x10;
- uint (UInt32) – 32bit
  - 0 to 4 294 967 295
- long (Int64) – 64bit
  - -9 223 372 036 854 775 808 to 9 223 372 036 854 775 807
- ulong (UInt64) – 64bit
  - 0 to 18 446 744 073 709 551 615

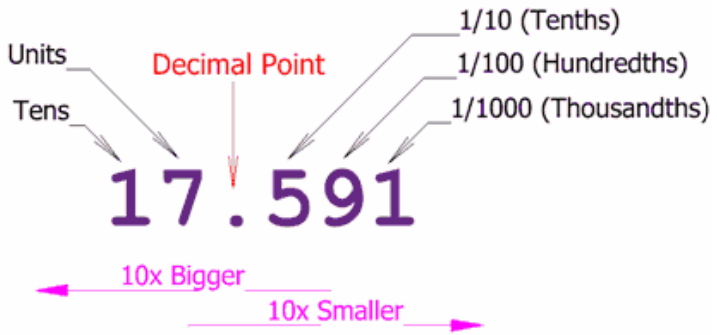
Real Floating-Point Types

- float (Single) - 32bit
  - single precision real number
  - accuracy up to 7 decimal places
  - -3.4×10^{38} to $+3.4 \times 10^{38}$
  - Negative infinity -∞ (Single.NegativeInfinity).
    - Obtained when dividing -1.0f by 0.0f.
  - Positive infinity +∞ (Single.PositiveInfinity).
    - Obtained when dividing 1.0f by 0.0f.
  - Uncertainty (Single.NaN) – when invalid operation is performed on real numbers.
    - Obtained when dividing 0.0f by 0.0f
    - when calculating square root of a negative number



Real Floating-Point Types

- double (Double) - 64bit
  - Double precision real number
  - accuracy up to 15/16 decimal places
  - $\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$
  - Negative infinity -∞ (Double.NegativeInfinity).
    - Obtained when dividing -1.0d by 0.0d.
  - Positive infinity +∞ (Double.PositiveInfinity).
    - Obtained when dividing 1.0d by 0.0d.
  - Uncertainty (Double.NaN) – when invalid operation is performed on real numbers.
    - Obtained when dividing 0.0d by 0.0d
    - when calculating square root of a negative number.



Real Types with Decimal Precision

- decimal (Decimal) – 128bit
  - Presented in the decimal numeral system rather than the binary one
  - precision from 28 to 29 decimal places
  - No loss of accuracy
  - -7.9×10^{28} to $+7.9\times10^{28}$
  - Used in financial calculations
  - Calculations with decimal are slower than float/double

Boolean Type

- bool (Bool) - 1bit
  - Two values – true and false
  - Used to keep result of logical expressions

Character Type

- char (Char) – 16 bit
  - Unicode 16-bit character
  - Represented in memory from 0 to 65535
  - Example:
    - char a = 'X'; // Character literal
    - char b = '\x0058'; // Hexadecimal
    - char c = (char)88; // Cast from integral type
    - char d = '\u0058'; // Unicode - see http://www.unicode.org/charts/
  - Escaping

Strings

- string (String)
  - Unlimited* sequences of characters
- string a = “This is a string”;
- string b = "\"This is quoted text\"";
- string c = @"\This is quoted text\";
- string d = @""""This is quoted text"""";
- string e = @”This is

a new line”;

Escape Symbols

Escaping symbol	Description
\'	symbol in a string literal
\"	" symbol in a string literal
\\	\ symbol in a string literal
\b	deletes the preceding character in a string literal
\n	new line
\r	carriage return
\t	horizontal tabulation

Enumerations

- Why enumerations?
- Definition of an enumeration type (**syntax)**

```
enum type_name
{
   key_1 = value_1,
   key_2 = value_2,
   ...
}
```

Object Type

- object (Object) – reference type
  - Can contain a reference to any value
  - Object a = “text”;
  - Object b = 5;
  - Object c = 4.16;
  - Object d = null;

Value and Ref Data Types

- Value Types
  - Working with Value Types
  - Value Types and the stack memory
- Reference Types
  - Working with Reference Types
  - Reference Types and the dynamic memory
  - The **null** value

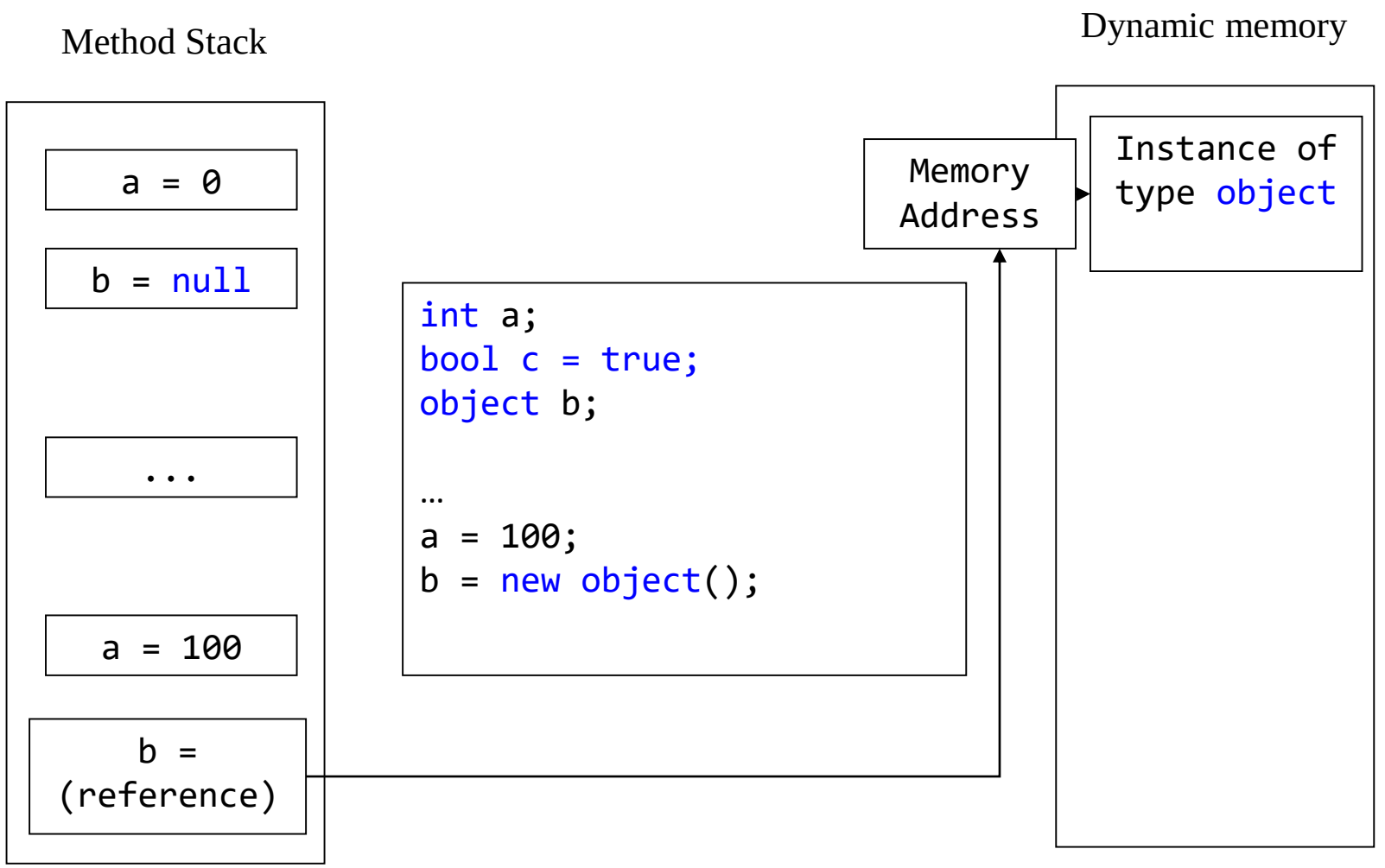
Stack Memory

- Contains value types
- Fast
- Every thread has its own stack
- Space is managed efficiently by CPU, memory will not become fragmented

Dynamic (heap) memory

- Contains reference types values
- Applications usually have 1 heap
- Limited to.. it depends
- Can contain large data blocks
- Slower than stack memory
- Managed by Garbage collector
- Fragmentation

Value and Ref Data Types



Type Casting

- Type casting syntax

  target_type = (target_type)original_type
- Explicit type casting
- Implicit type casting
- Incompatible types and type conversion
  - The **Convert** class

Operators in C#

- Arithmetic
  - +, -, *, /, %, ++, --
- Logical
  - &&, ||, !
- String
  - +
- Comparison
  - ==, !=, >, <, >=, <=
- Assignment
  - =, +=, -=, *=, /=, %=

Math

- Math.Pow(number, power)
- Math.Sqrt(number)
- Math.Abs(number)
- Math.Sin(number)
- Math.Cos(number)

$$2\frac{10^{|x|}}{x+5^2}$$

$$\frac{x^{10}}{2x}-\frac{|x|}{2,}$$

$$5x\sqrt{100}+\frac{x}{2}x$$

$$x+\frac{10^{2x}}{cos(x)}+|100-x^2|$$

$$\frac{sin(x)}{|x|}+10^x$$

$$\sqrt[3]{2x+x}$$


Questions?