

C# Language & .NET Platform

4th Lecture:

C# Type System

<http://d3s.mff.cuni.cz/~jezek>

Department of
Distributed and
Dependable
Systems



Pavel Ježek

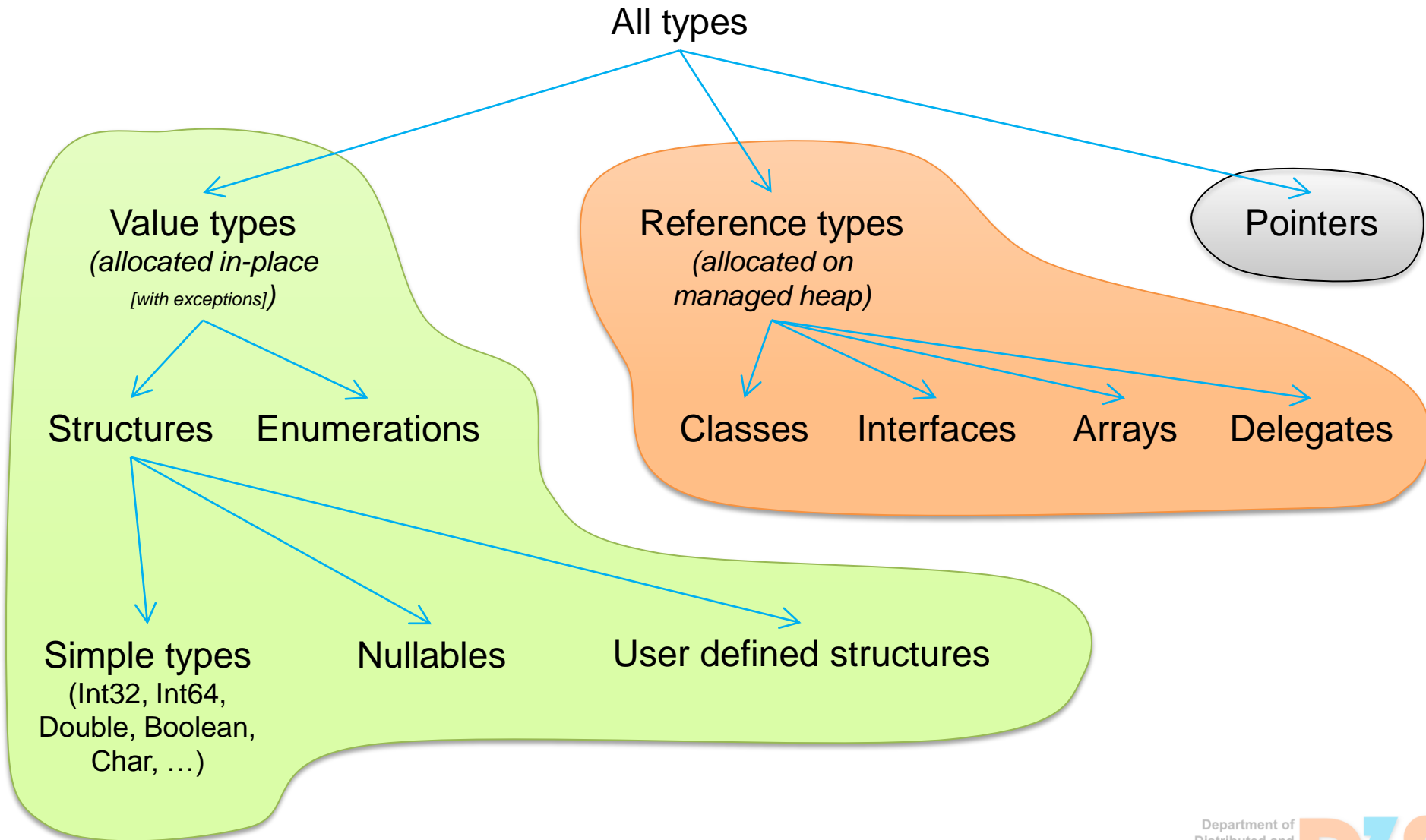
pavel.jezek@d3s.mff.cuni.cz



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

CLI Type System



Simple Types (are value types)

.NET Type	C# Keyword	CLS Compliant	In-place Size in Bytes (bits)	Range
System.Byte	byte	Yes	1 B (8 b)	0 ... 255
System.SByte	sbyte	-	1 B (8 b)	-128 ... 127
System.UInt16	ushort	-	2 B (16 b)	0 ... 65,535
System.Int16	short	Yes	2 B (16 b)	-32,768 ... 32,767
System.UInt32	uint	-	4 B (32 b)	0 ... 4,294,967,295
System.Int32	int	Yes	4 B (32 b)	-2,147,483,648 ... 2,147,483,647
System.UInt64	ulong	-	8 B (64 b)	0 ... 18,446,744,073,709,551,615
System.Int64	long	Yes	8 B (64 b)	-9,223,372,036,854,775,808 ... 9,223,372,036,854,775,807
System.Single	float	Yes	4 B (32 b)	IEEE 754: 1-bit sign + 23(+1)-bit mantissa + 8-bit signed exponent: $\pm 3.402823 \cdot 10^{38}$
System.Double	double	Yes	8 B (64 b)	IEEE 754: 1-bit sign + 52(+1)-bit mantissa + 11-bit signed exponent: $\pm 1.79769313486232 \cdot 10^{308}$
System.Boolean	bool	Yes	-	true, false
System.Decimal	decimal	Yes	16 B (128 b)	1-bit sign + 96-bit integer / 10^{0-28} , i.e.: $\pm 2^{96} / 10^{0-28}$
System.Char	char	Yes	2 B (16 b)	UTF-16 characters (Unicode)

Compare: in-place size of reference types is platform dependent, e.g. a string variable on a 32-bit platform = 4 B, but on a 64-bit platform = 8 B

Simple Types (are value types)

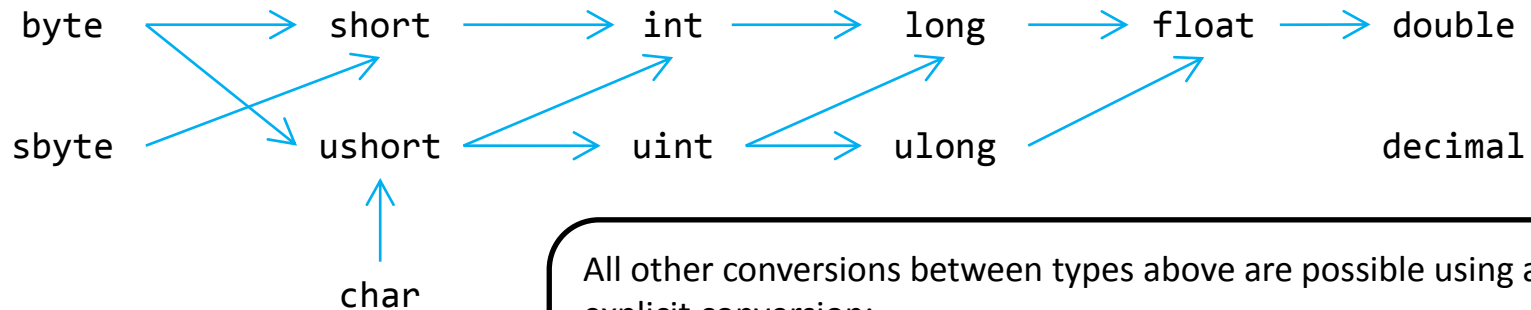
.NET Type	C# Keyword	CLS Compliant	In-place Size in Bytes (bits)	Range
System.Byte	byte	Yes	1 B (8 b)	0 ... 255
System.SByte	sbyte	-	1 B (8 b)	-128 ... 127
System.UInt16	ushort	-	2 B (16 b)	0 ... 65,535
System.Int16	short	Yes	2 B (16 b)	-32,768 ... 32,767
System.UInt32	uint	-	4 B (32 b)	0 ... 4,294,967,295
System.Int32	int	Yes	4 B (32 b)	-2,147,483,648 ... 2,147,483,647
System.UInt64	ulong	-	8 B (64 b)	0 ... 18,446,744,073,709,551,615
System.Int64	long	Yes	8 B (64 b)	-9,223,372,036,854,775,808 ... 9,223,372,036,854,775,807
System.Single	float	Yes	4 B (32 b)	IEEE 754: 1-bit sign + 23(+1)-bit mantissa + 8-bit signed exponent: $\pm 3.402823 \times 10^{38}$
System.Double	double	Yes	8 B (64 b)	IEEE 754: 1-bit sign + 52(+1)-bit mantissa + 11-bit signed exponent: $\pm 1.79769313486232 \times 10^{308}$
System.Boolean	bool	Yes	-	true, false
System.Decimal	decimal	Yes	16 B (128 b)	1-bit sign + 96-bit integer / 10^{0-28} , i.e.: $\pm 2^{96} / 10^{0-28}$
System.Char	char	Yes	2 B (16 b)	UTF-16 characters (Unicode)

7 decimal digit precision

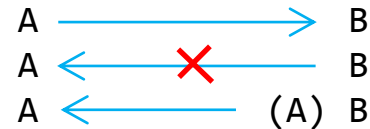
15 decimal digit precision

28 decimal digit precision

Simple Types: Implicit Conversions



All other conversions between types above are possible using an explicit conversion:



e.g.:

```
long a = 1;  
int b = (int) a;
```

C# vs. C/C++: No conversions possible to or from `bool` type in C#!

`bool`

Simple Types: Literals

Integer literals

- Default type of integer literals is `int` – larger type is selected if literal's value does not fit into an `int`, e.g. `1000000000000` → `long`
- A literal can be implicitly converted to a smaller type if literal's value fits, e.g.:
`byte a = 1;`
- Can be decimal, e.g. `254`
- Can be hexadecimal, e.g. `0xFE`
- A literal case-insensitive suffix forces a subset of possible types:

`u U` → `uint, ulong`

`l L` → `long, ulong`

`ul uL Ul UL` → `ulong`

`lu lU Lu LU`

Real literals

- Default type of real literals is `double`.
- Format: `[123456789].123456789[E[+/-]123]`, e.g. `10.0`, `10E15`, `10E-4`, `.14159`
- A literal case-insensitive suffix forces literal type:

`f F` → `float`

`d D` → `double`

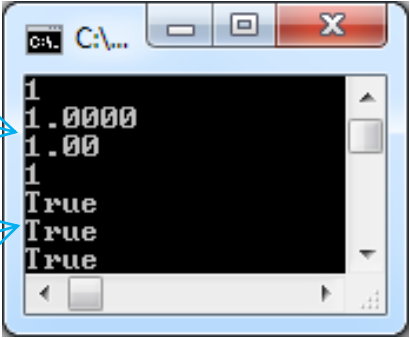
`m M` → `decimal`

Char Literals & Escape Sequences

Char literal: a single character or escape sequence in quotes:
'A' or '\n' or '\x5C'

Escape Sequence	Character	Unicode Code (in hex)
\'	'	27
\"	"	22
\\	\	5C
\0	Null	0
\b	Backspace	8
\n	New line	A
\r	Carriage return	D
\t	Tab (horizontal)	9
\xWXYZ	<i>Any character</i>	WXYZ
\a	Alert	7
\f	Form feed	C
\v	Vertical tab	B

Note on Decimals

.NET Type	C# Keyword	CLS Compliant	In-place Size in Bytes (bits)	Range
System.Byte	byte	Yes	1 B (8 b)	0 ... 255
<p>Values in Decimals are not normalized:</p> <pre> decimal a = 1; decimal b = 1.0000M; decimal c = 1.00M; decimal d = (decimal) 1.000; Console.WriteLine(a); Console.WriteLine(b); Console.WriteLine(c); Console.WriteLine(d); Console.WriteLine(a == b); Console.WriteLine(a == c); Console.WriteLine(a == d); </pre> 				
System.Double	double	Yes	8 B (64 b)	±1.7976931348623157 × 10 ³⁰⁸
System.Decimal	decimal	Yes	16 B (128 b)	1-bit sign + 96-bit integer / 10 ⁰⁻²⁸ , i.e.: ±2 ⁹⁶ / 10 ⁰⁻²⁸
System.Char	char	Yes	2 B (16 b)	UTF-16 characters (Unicode)

```

double a = 1;
double b = 1.0000;
double c = 1.00;
                    
```

```

Console.WriteLine(a);
Console.WriteLine(b);
Console.WriteLine(c);
                    
```

