

Standardní datové typy jazyka C# - otázka č. 19

Obecně:

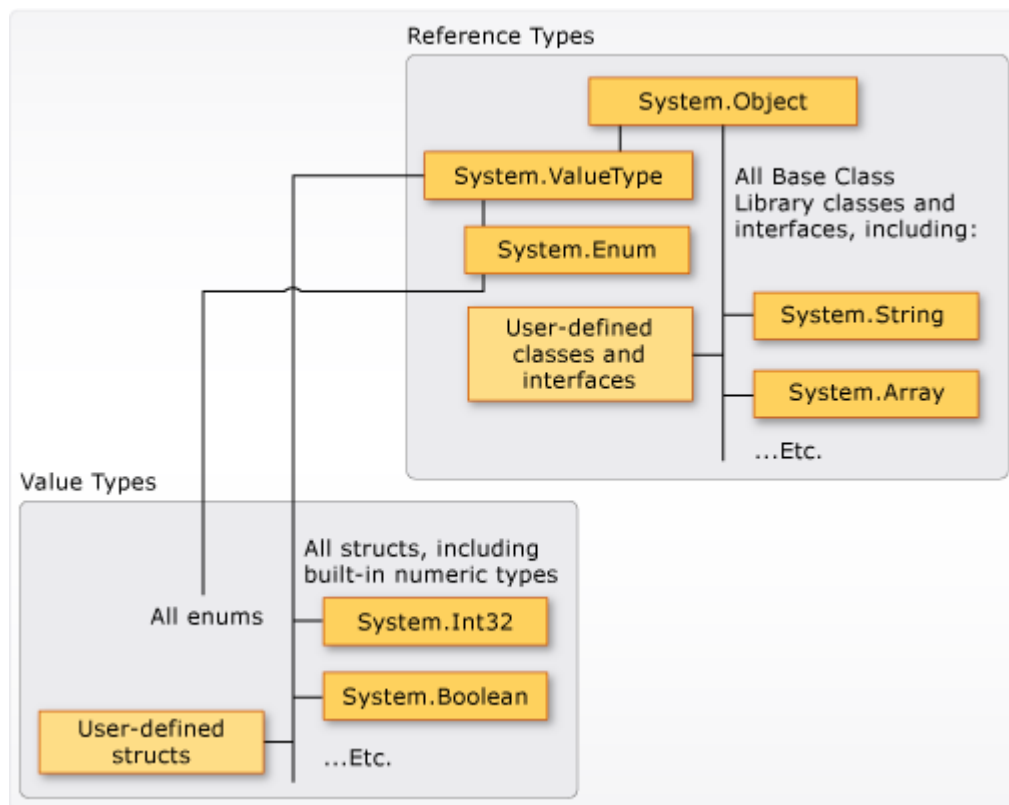
- strongly vs weakly/loosely typed jazyk -> každá proměnná a konstanta musí mít **typ**
 - pomáhá odchytit více chyb ještě během vývoje
 - chování programu je předvídatelnější
 - například c, c++, java
 - neexistuje jedna ucelená definice
 - pozor na rozdíl mezi **dynamic** a **static**
 - dynamic => během runtime
 - static => během kompilace

Konkrétně pro C#:

- typ může obsahovat zejména následující informace o dané proměnné
 - prostor (úložiště, kterou bude proměnná potřebovat)
 - typy operací, které jsou povoleny provádět s danou proměnnou
 - hodnoty, kterých může proměnná nabývat
 - která rozhraní implementuje
 - základní typ od něhož dědí
- při deklaraci definujeme její typ, který poté už nelze změnit
- kompilátor kontroluje, jestli jsou všechny operace, které provádíme bezpečné
- přetypování
 - implicitní - automaticky pokud nedojde ke ztrátě dat, např.: menší => větší celá čísla
 - explicitní - tzv. "cast", dojde ke ztrátě dat a my informujeme kompilátor, že tak činíme vědomě, např.: (int)x
 - s pomocnými třídami - převody mezi nekompatibilními typy string => int, např.: int.Parse()
- vlastní typy
 - enum, class, record, struct, record

Common type system (CTS, běžný systém typů)

- princip dědičnosti - všechny typy dědí od základního typu *System.Object*
- reference vs value type (hodnotový a odkazový typ)
 - reference - odkazuje na nějakou hodnotu, která je uložena v heap memory, např.: class, record
 - value - ukládá hodnotu na stack, např.: standardní typy



Předdefinované typy:

- hodnotové typy
 - bool
 - byte
 - sbyte - signed byte - 1 bit je rezervován pro znaménko
 - short - 16 bitů
 - ushort - unsigned short - nemá zápornou hodnotu, protože zde nemáme onen 1 bit pro znaménko
 - int - 32 bitů
 - uint
 - long - 64 bitů
 - ulong
 - float - 32 bitů
 - double - 64 bitů
 - decimal - 128 bitů
 - char - 16 bitů
- odkazové typy
 - object
 - string
 - delegate
 - dynamic

- typy používané pro OOP
 - class
 - mutable
 - heap
 - struct
 - mutable
 - stack
 - record (class/struct)
 - immutable
 - stack/heap
 - porovnávání na základě hodnoty