

# Souhrn datových typů (Visual Basic)

20. 07. 2015 6 min ke čtení Přispěvatelé 

## V tomto článku

[Využití paměti](#)

[Viz také:](#)

Následující tabulka uvádí datové typy jazyka Visual Basic, jejich podpůrné běžných typů modulu runtime jazyka, jejich přidělení nominální úložiště a jejich rozsahy hodnot.

Typ jazyka Visual Basic	Obvyklá struktura typu modulu runtime jazyka	Přidělení úložiště nominální	Rozsah hodnot
<a href="#">Datový typ Boolean</a>	<a href="#">Boolean</a>	Závisí na implementaci platformy	<input type="checkbox"/> True <input type="checkbox"/> Nebo <input type="checkbox"/> False
<a href="#">Bajť</a>	<a href="#">Byte</a>	1 bajt	0 až 255 (unsigned)
<a href="#">Char – (jeden znak)</a>	<a href="#">Char</a>	2 bajty	0 až 65535 (unsigned)
<a href="#">Datum</a>	<a href="#">DateTime</a>	8 bajtů	0:00:00 (půlnoc) na 1. ledna 0001 až 11:59:59 odp v 31. prosince 9999
<a href="#">Decimal</a>	<a href="#">Decimal</a>	16 bajtů	0 až +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9... E + 28) <sup>†</sup> s žádné desetinné čárky, 0 až +/-7,9228162514264337593543950335 28 míst napravo od desetinné čárky;  nejmenší nenulové číslo je +/-0,00000000000000000000000000000001 (+/-1E-28) <sup>†</sup>
<a href="#">Dvojitě (dvojitě přesnosti s plovoucí desetinnou čárkou)</a>	<a href="#">Double</a>	8 bajtů	-1.79769313486231570E + 308 do -4.94065645841246544E-324 <sup>†</sup> pro záporné hodnoty;  4.94065645841246544E-324 prostřednictvím 1.79769313486231570E + 308 <sup>†</sup> pro kladné hodnoty
<a href="#">Integer</a>	<a href="#">Int32</a>	4 bajty	-2,147,483,648 prostřednictvím 2 147 483 647 (se znaménkem)

Typ jazyka Visual Basic	Obvyklá struktura typu modulu runtime jazyka	Přidělení úložiště nominální	Rozsah hodnot
Dlouhé (long integer)	Int64	8 bajtů	-9,223,372,036,854,775,808 prostřednictvím 9,223,372,036,854,775,807 (9.2... E + 18 <sup>†</sup> ) (se znaménkem)
objekt	Object (třídy)	4 bajtů na 32bitové platformě  8 bajtů na 64bitové platformě	Libovolný typ může být uložen v proměnné typu Object
SByte –	SByte	1 bajt	-128 až 127 (se znaménkem)
Krátký (krátké celé číslo)	Int16	2 bajty	-32 768 až 32 767 (se znaménkem)
Jeden (jednoduchou přesnost s plovoucí desetinnou čárkou)	Single	4 bajty	-3.4028235E + 38-1, 401298E-45 <sup>†</sup> pro záporné hodnoty;  1, 401298E-45 prostřednictvím 3.4028235E + 38 <sup>†</sup> pro kladné hodnoty
Řetězec (proměnné délky)	String (třídy)	Závisí na implementaci platformy	0 na přibližně 2 miliardy znaků Unicode
UInteger	UInt32	4 bajty	0 až 4 294 967 295 (unsigned)
ULong	UInt64	8 bajtů	0 až 18,446,744,073,709,551,615 (1.8... E + 19 <sup>†</sup> ) (unsigned)
Uživatelem definované (struktura)	(dědí z ValueType)	Závisí na implementaci platformy	Každý člen struktury má rozsah určující jeho datový typ a nezávislé z rozsahů jiných členů
UShort	UInt16	2 bajty	0 až 65535 (unsigned)

<sup>†</sup> v vědecký zápis, "E" odkazuje na násobky 10. Takže 3.56E + 2 označuje 3.56 x 10<sup>2</sup> nebo 356 a 3.56E-2 označuje 3.56 / 10<sup>2</sup> nebo 0.0356.

## ⓘ Poznámka

U řetězců obsahující text, použijte [StrConv](#) funkce pro převod z jednoho formátu do jiného.

Kromě zadání datový typ v příkazu deklarace, můžete vynutit datový typ některé programovací prvky pomocí znak typu. Zobrazit [znaky](#).

## Využití paměti

Při deklaraci základní datový typ není bezpečné předpokládat, že jeho využití paměti je stejné jako jeho přidělení nominální úložiště. Je to z důvodu následující aspekty:

- **Přiřazení úložiště.** Modul common language runtime můžete přiřadit úložiště založené na aktuální vlastnosti platformy, na kterém vaše aplikace provádí. Pokud paměti je téměř plná, může ji co nejbližší k sobě pack vaše deklarované elementy co nejvíce. V ostatních případech to může být zarovnat jejich adresy paměti na hranice fyzických hardwarových vazeb pro optimalizaci výkonu.
- **Šířka platformy.** Přiřazení úložiště na 64bitové platformě se liší od přiřazení na 32bitové platformě.

## Složené datové typy

Jaké jsou požadavky stejné platí pro každý člen složené datového typu, jako je například struktura nebo pole. Nelze spoléhat na jednoduše sečtením nominálních přidělení úložiště členů typu. Kromě toho existují další důležité informace, jako je následující:

- **Režie.** Některé složené typy mají požadavky na paměť navíc. Například pole používá extra paměť pro pole samotného a také pro jednotlivé rozměry. Na 32bitové platformě právě tato dodatečná režie 12 bajtů plus 8 bajtů pro jednotlivé rozměry. Na 64bitové platformě je dvojitá tento požadavek.
- **Rozložení úložiště.** Nelze bezpečně předpokládat, že pořadí úložiště v paměti je stejné jako vaše pořadí prohlášení. I nelze vytvořit předpoklady o zarovnání bajt, jako jsou hranice bajtu 2 nebo 4 bajty. Pokud definujete třídu nebo strukturu a je nutné řídit rozložení úložiště z jejích členů, můžete použít [StructLayoutAttribute](#) atribut třídy nebo struktury.

## Režijní náklady na objekt

`Object` Vztahující se k libovolným datům základní nebo složený typ používá 4 bajty kromě data obsažená v datovém typu.

## Viz také:

- [StrConv](#)

- [StructLayoutAttribute](#)
- [Funkce pro převod typů](#)
- [Souhrn převodu](#)
- [Znaky typu](#)
- [Účinné používání datových typů](#)