

Otázka: Základní programové konstrukce

Předmět: Programování, Informatika

Přidal(a): Ondřej Rethy

Proměnné

- Jedná se o místo v paměti, které je určeno pro uložení dat
- Každá proměnná má svůj datový typ (v některých jazycích povinné deklarovat při založení, např. C#, Java)
- Typová kontrola podle typového systému
 - Statická (Java, C#)
 - Proměnné je přiřazen datový typ a podle něj i rozsah a typ informace kterou může uchovávat
 - Například integer nemůže uchovávat string
 - Dynamická (Python)
 - Samotné hodnoty mají datový typ, nikoliv proměnné
 - Tyto typy jsou kontrovány až za běhu programu
- Deklarace datového typu
 - Explicitní
 - Zapisuje se před vlastní použitím proměnné
 - `int priklad;`
`priklad = 5;`
 - Implicitní
 - Proměnou použijeme v podprogramu
 - Jedná se o deklaraci „za běhu“

- Implicitní deklarace má rychlejší zápis programového kódu

```
• int funkce()
  {
    return 5;
  }
int priklad = funkce();
```

Operátory

Aritmetické

- **+**, **-** ... sčítání a odčítání
- *****, **/** ... násobení a dělení
- **%** ... modulo (zbytek po dělení)
- **++x** ... preinkrementace
- **x++** ... postinkrementace
- **-x** ... predekrementace
- **x-** ... postdekrementace

Logické

- **&&**, **||** ... logické AND, logické OR
- **!** ... logické NOT

Bitové

- **&**, **|** ... bitové AND, bitové OR
- **~** ... bitové NOT
- **<<**, **>>** ... bitový posun vlevo a vpravo

Relační

- <, > ... porovnání menší a větší
- <=, >= ... porovnání menší nebo rovno a větší nebo rovno
- == ... porovnání rovnosti
- != ... porovnání nerovnosti

Přístupové

- [] ... indexer
- přístup ke členům
- () ... invokace

Ostatní

Název	Symbol
operátor přiřazení	=
lambda operátor	=>
ternární operátor	?:
operátor nullového sjednocení	??
kvalifikátor aliasu oboru názvů	::
await operátor	await
default operátor	default
delegate operátor	delegate
nameof operátor	nameof
new operátor	new
sizeof operátor	sizeof

stackalloc operátor	stackalloc
true/false operátor	true/false

Priorita operátorů

Operátory	Kategorie nebo název
$x.y$, $x?.y$, $x?[y]$, $f(x)$, $a[i]$, $x++$, $x-$, new, typeof, checked, unchecked, default, nameof, delegate, sizeof, stackalloc, $x \rightarrow y$	primární
$+x$, $-x$, $!x$, $\sim x$, $++x$, $--x$, $(T)x$, await, $\&x$, $*x$, true a false	unární
$x * y$, x/y , $x\%y$	multiplikativní
$x + y$, $x - y$	přičítáním
$x \ll y$, $x \gg y$	posunu
$x < y$, $x > y$, $x \leq y$, $x \geq y$, is, as	relační testování a testování typu
$x == y$, $x != y$	rovnost
$x \& y$	logický operátor AND nebo bitový logický operátor AND

Operátory	Kategorie nebo název
$x \wedge y$	logická hodnota XOR nebo bitový logický operátor XOR

<code>x y</code>	logická OR logická logická hodnota nebo
<code>x && y</code>	podmiňovací operátor AND
<code>x y</code>	podmiňovací operátor OR
<code>x?? y</code>	operátor slučování null
<code>r ? t : f</code>	podmíněný operátor
<code>x = y, x + = y, x ^ = y, x << = y, x >> = y, =></code>	přiřazení a deklarace lambda

Datové typy

- Hodnotové
 - Instance hodnotových typů (proměnné) jsou ukládány na zásobník programu a program s nimi pracuje přímo
 - Obsahují pouze hodnotu daného typu
- Referenční
 - Referenční datové typy neuchovávají přímo hodnotu samotnou, nýbrž odkaz na místo v paměti

a) Hodnotové

Logické

- **bool** ... hodnoty true, false

Čísla

- **byte** ... 8bitové beznaménkové celé číslo 0 - 255
- **ushort** ... 16bitové beznaménkové celé číslo 0 - 65535
- **uint** ... 32bitové beznaménkové celé číslo 0 - 4294967295
- **ulong** ... 64bitové beznaménkové celé číslo 0 - 18446744073709551615
- **sbyte** ... 8bitové znaménkové celé číslo -128 - 127
- **short** ... 16bitové znaménkové celé číslo -32768 - 32767
- **int** ... 32bitové znaménkové celé číslo -2147483648 - 2147483647
- **long** ... 64bitové znaménkové celé číslo -9223372036854775808 - 9223372036854775807
- **float** ... 32bitové znaménkové reálné číslo (1bit znaménko, 8bit exponent a 23bit mantisa)
- **double** ... 64bitové znaménkové reálné číslo (1bit znaménko, 11bit exponent a 52bit mantisa)

b) Referenční

- **string** ... slouží k uložení textů
 - Jedná se o alias ke třídě System.String()
 - Lze jí přímo naplnit (bez použití new)
 - Při porovnání se porovnává hodnota objektů, nikoliv hodnota odkazů, jak tomu je u ostatních tříd
- **object** ... ostatní třídy rozšiřují tuto třídu
 - Alias třídy System.Object()
 - Object je tedy společným základem pro ostatní typy
- **třída (class)**
- **rozhraní (interface)**
- **pole**
- **delegát (delegate)**

Základní programové konstrukce

- Větvení (if, switch/case)
 - Úplné/Neúplné
 - Vícenásobné
- Cyklus (for, while)
 - Podmínka na začátku
 - Podmínka na konci
 - Známý počet průchodů

Podprogram

- logický celek programu
- část programu definovaná před startem programu
- po startu hlavního programu je podprogram z hlavního programu spouštěn

Využití podprogramů

- zkrácení, zpřehlednění a zjednodušení hlavního programu
- rozdělení algoritmu na více jednodušších částí
- jednoduché kontrola a odstranění chyb
- správné rozdělení programu na podprogramy je součástí programátorského umění
- zkušení programátoři si vyvíjí celé knihovny podprogramů – programové jednotky

Volání podprogramu

- podprogram je volán svým jménem

- volání probíhá z hlavního programu nebo z nadřazeného podprogramu