Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Programování - Programování C#

Ivan Pomykacz

esf evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Licence

Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

http://dumy.odbornaskola.cz/pomykacz/.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

Programování

Programování C#

Obsah

- Priorita operátorů

Priorita operátorů

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_23_PRG_445		
Název tématické oblasti (sady)	Programování		
Název materiálu	Priorita operátorů		
Anotace	Text seznamuje se základními (probíranými) operátory v programovacím jazyce. Vysvětluje pojmy výraz, operand, operátor. Objasňuje typy operátorů a příklady využití. Velký důraz je kladen na prioritu operátorů.		
Autor	Ivan Pomykacz	**Jazyk**	čeština
Očekávaný výstup	Pojmenuje operátor a operand ve výrazu. Sestavuje výrazy s operátory a operandy. Rozumí prioritě – způsobu vyhodnocování operátorů.		
Klíčová slova	operátor, operand, priorita, unární operátor, binární operátor, ternární operátor		
Druh výukového zdroje	Výklad	**Věková skupina žáků**	15+
Typ interakce	aktivita	**Ročník**	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	21.10.2013	**Celková velikost**	

Obsah

- Shrnutí
- Výraz
  - Význam operátorů
  - Priorita
  - Typy operátorů
- Vyhodnocování výrazů
  - Priority operátorů

Shrnutí

Výraz

Výraz je tvořen sekvencí operátorů a operandy.

Ukažme si to na příkladu z matematiky, tam se to operátory a operandy jen hemží. Např. výraz `x = 1 + 2`.

Kolik je zde operátorů a operandů?

- operátory: `+`, `=`
- operandy: `1`, `2` a `x`

Ano, patrně jste si odvodili, že operátor je ten kdo s něčím operuje (pracuje). Operand je ten flákač, který se sebou nechá dělat, co se vám zlíbí.

Význam operátorů

Jakmile si uvědomíme významy jednotlivých operátorů, budeme výrazům lépe rozumět.

Operátor `+` provádí operaci součet, tj. dokáže sečíst dva operandy, které jsou po jeho levé a pravé straně. Ano, na tom totiž záleží. V našem programovacím jazyce není možné napsat např. `x = + 1 2` .

Operátor `=` provádí přiřazení hodnoty zprava doleva. Tzn. výsledek operace la pravé straně uloží do operandu na levé straně.

Priorita

Zcela automaticky chápeme výraz `x = 1 + 2`. Když se vás někdo zeptá kolik je *x*, okamžitě odpovíte, že 3.

Pokud se máme bavit o nějaké prioritě operátorů, tak určitě víte, že pokud si chceme ujasnit, co se má provést (vypočítat) dřív (nejprve), tak to dáme do závorek.

Kdyby jste dostali za úkol "ozávorkovat" náš výraz `x = 1 + 2`, tak vás nejspíš napadne jediná možnost: `x = (1 + 2)`. A to zároveň vystihne i prioritu, v jaké se provedou jednotlivé operace. Nejprve se provede operátor `+`, protože je v závorce.

Teprve potom se provede `=`. Ono to funguje ale i bez závorek. To proto, že `+` má vyšší prioritu, než `=`.

Představte si situaci, kdyby tomu tak nebylo. Kdyby `=` mělo vyšší prioritu, než `+`. Pak bychom museli náš výraz ozávorkovat takto: `(x = 1) + 2`. To už vypadá trochu méně smysluplněji, že?

Proto mají operátory svou prioritu, abychom věděli, kde s vyhodnocováním výrazu začít. Tedy my ne, ale programovací jazyk. Nám se vyplatí znát pravidla, abychom mohli výrazy vhodně poskládat.

Platí, že čím nižší priorita, tím později operátor provede svou akci.

Typy operátorů

V programovacím jazyce se můžeme setkat se třemi typy operátorů:

- **unární** - mají pouze jeden operand. Např. `x++`.
- **binární** - mají klasicky dva operandy, jako `x + y`.

- **ternární** - je pouze jeden `?:`, a ano má tři operandy: `c ? a : b`

Vyhodnocování výrazů

Pořadí, v jakém se vyhodnocují jednotlivé operátory je dáno prioritou. To ale nestačí.

Mějme výraz `x + y * z`. Ze základní školy víme, že násobení má přednost před sčítáním. To platí samozřejmě i zde. Kdybychom měli aplikovat závorky na náš výraz, tak by to dopadlo takto: `x + (y * z)`.

Ale co když bude náš výraz takovýto: `x + y + z`?

V jakém pořadí se vyhodnotí teď? A není to jedno? Přeci `x + (y + z)`, je to samé jako `(x + y) + z`. Ano i ne.

Počítač je striktně deterministický stroj. Pro náhodu zde není místo. (Generování náhodný čísel je proto trochu oříšek, ale to je zase jiná pohádka) Vyhodnocování výrazů se musí řídit přesnými pravidly.

Proto se u operátorů ještě definuje tzv. asociativnost zleva nebo zprava.

Kromě operátoru `=`, jsou všechny binární operátory asociativní zleva. To znamená, že se vyhodnocují zleva doprava. Tudíž výraz uvedený výše se vyhodnotí jako: `(x + y) + z`.

Priority operátorů

Tabulka obsahuje výčet některých operátorů. Priorita je od shora (nejvyšší) dolů (nejnižší) a zleva (vyšší) doprava (nižší).

priorita	kategorie	operátory
nejvyšší	Primární	`x++`, `x--`
	Unární	`++x`, `--x`
	Multiplikativní	`*`, `/`
	Aditivní	`+`, `-`
	Relační	`<`, `>`, `<=`, `>=`
	Porovnávací	`==`, `!=`
	Podmiňovací	`&&`
	Podmiňovací	`||`
nejnižší	Přiřazovací	`=`, `*=`, `/=`, `+=`, `-=`