Kifejezések, értékadás

A nyelv számunkra fontos operátorai a prioritásnak megfelelő sorrendben a következők:

Egyoperandusú operátorok:

new a futás közbeni, azaz dinamikus helyfoglalás operátora. Az operandusa határozza meg, hogy milyen típusú objektumnak foglal helyet. Sikeres helyfoglalás eredménye a lefoglalt memória címe, sikertelené pedig a null mutató. Pl.:

x=new int[3];

//3 egészet tartalmazó x nevű tömb számára foglal helyet a memóriában

A .NET keretrendszer automatikusan elvégzi a lefoglalt memóriaterület felszabadítását, ha a programnak már nincs rá szüksége.

-

aritmetikai negálás

!

logikai negálás

~

bitenkénti negálás

++, --

eggyel növelés, ill. csökkentés

Pl.:

j=++i;

// i=i+1; j=i; tömören írva

j=i++;

// j=i; i=i+1; tömören írva

double x=6.1;

int i =(int) x;

// i=6

Az egyoperandusú operátorok és az értékadás operátora esetén jobbról balra történik a végrehajtás, egyébként azonos precedenciájú operátoroknál balról jobbra.

Háromoperandusú operátor

Valójában egy logikai elágazás utasítás, a tömörebb írásmódot szolgálja. Alakja: kif1 ? kif2 : kif3 Ha kif1 értéke igaz (nem 0), akkor az egész kifejezés értéke kif2, különben kif3. Pl.:

x=a>b?a+2:b+2;

// ennek eredménye megegyezik az alábbi kód eredményével

if (a>b) x=a+2; else x=b+2;

Kétoperandusú operátorok

*

szorzás

/

osztás

%

maradékképzés, a maradékképzés operandusai egészek.

Mindig igaz: (a/b)*b+a%b=a

+

összeadás, szöveg esetén összefűzés

-

kivonás

<<

bitenkénti balra léptetés

<, >

kisebb, nagyobb relációs operátorok

<=, >=

kisebb vagy egyenlő, nagyobb vagy egyenlő operátorok

==, !=

egyenlő, nem egyenlő relációs operátorok

&

bitenkénti és

^

bitenkénti kizáró vagy

|

bitenkénti vagy

&&

logikai és

||

logikai vagy

Kétoperandusú értékadó operátorok

= értékadás operátora. Az értékadás során a bal oldal megkapja a jobb oldali kifejezés értékét(, és egyben ez lesz a kifejezés értéke is). További értékadó operátorok, amelyek a hatékonyabb, tömörebb programkódot eredményeznek: +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, |= A kif1 valamelyik_operátor_az_előző_sorból kif2 kifejezés megfelel a kif1 = kif1 op kif2 kifejezésnek, ahol op az egyenlőségjel előtti operátor.

Ha matematikai számításokra van szükség, akkor a System névtér Math osztálya segít, pl.:

Math.Sqrt(x)

x négyzetgyöke

Math.Abs(x)

x abszolút értéke

Math.Round(x)

kerekítés a matematikai szabályok szerint

Math.Ceiling(x)

felfelé kerekítés

Math.Floor(x)

lefelé kerekítés

Math.Power(x,y)

hatványozás, x az alap, y a kitevő

Math.PI

3.14159265358979323846

Gyakran van szükség véletlen számokra. A System névtér Random osztálya biztosítja a pszeudo-véletlenszámok generálásának lehetőségét. Pl.:

Random x=new Random();

// x véletlenszám objektum (a rendszeridőből generálva)

Random x1=new Random(10);

// x1 véletlenszám objektum (a 10 értékből kiindulva)

int i=x.Next();

// véletlen egész szám 0 és MaxInt között, 0 lehet, MaxInt nem // MaxInt a legnagyobb egész számot jelöli

int i1=x.Next(100);

// véletlen egész szám 0 és 100 között, 0 lehet 100 nem

int i2=x.Next(100,200);

// véletlen egész szám 100 és 200 között, 100 lehet 200 nem

double a=x.NextDouble();

// véletlen valós szám 0 és 1 között, 0 lehet 1 nem

Az escape sorozatok teljes listája a következő:

\a figyelmeztető jelzés (bell, csengő)

\b visszalépés (backspace)

\f lapdobás (formfeed)

\n új sor (new line)

\r kocsi vissza (carriage return)

\t vízszintes tabulátor (horizontal tab, HTAB)

\v függőleges tabulátor (vertical tab, VTAB)

\\ fordított törtvonal (backlash)

\? kérdőjel

\' aposztróf

\" idézőjel

\ooo oktális szám

\xhh hexadecimális szám