Operators

C#’da operatörler diğer programlama dillerinde olduğu gibi kullanılmaktadır.

* Operatörlerle birlikte kullanılan ifadelere **operand** adı verilmektedir.

(Aritmetik Operatörler)

|  |  |
| --- | --- |
| Operator | Description |
| + | String ifadeleri birleştirir, sayısal ifadeleri toplar. |
| - | Çıkarma yapmak için kullanılır. |
| \* | Çarpma yapmak için kullanılır. |
| / | Bölme yapmak için kullanılır. |
| % | Bölümden kalanı bulmak için kullanılır. |

(Atama Operatörleri)

|  |  |
| --- | --- |
| Operator | Description |
| = | Soldaki değere sağdakini atar. |
| += | Topla ve ata. |
| -= | Çıkar ve ata. |
| \*= | Çarp ve ata. |
| /= | Böl ve ata. |
| %= | Mod al ve ata. |

Example

//(i += 5) ile (i = i + 5) aynı anlama gelmektedir.

int i = 5;

i += 5;

Console.WriteLine(i); //10

(Arttırma ve Azaltma Operatörleri)

|  |  |
| --- | --- |
| Operator | Description |
| x++ | x’i bir arttır. |
| ++x | x’i bir arttır. |
| x-- | x’i bir azalt. |
| --x | x’i bir azalt. |

* x++ ile ++x arasındaki fark aşağıdaki örnekte gösterilmiştir.

Example

int x = 1;

int y = 2;

Console.WriteLine(x++); //1

Console.WriteLine(x); //2

Console.WriteLine(++y); //3

Console.WriteLine(y); //3

(Karşılaştırma Operatörleri)

|  |  |
| --- | --- |
| Operator | Description |
| == | Eşitse. |
| != | Eşit değilse. |
| < | Küçükse. |
| > | Büyükse. |
| <= | Küçük ya da eşitse. |
| >= | Büyük ya da eşitse. |

* == operatörü referans türleri karşılaştırırken operandların referanslarının aynı olup olmadığını kontrol etmektedir.

(Mantıksal Operatörler)

|  |  |
| --- | --- |
| Operator | Description |
| && | Mantıksal AND operatörü. Tüm koşulların sağlanması durumunda true aksi halde false döner. |
| || | Mantıksal OR operatörü. Koşullardan en az biri sağlanıyorsa true hiçbir koşulun sağlanmaması durumunda false döner. |
| ! | Mantıksal NOT operatörü. Kendisinden sonra gelen değerin değilini(tersini, zıttını) döndürür. |

Example

Console.Write(!false); //true

(Bit Düzeyinde İşlem Yapan Operatörler)

|  |  |
| --- | --- |
| Operator | Description |
| & | AND operatörü. |
| | | OR operatörü. |
| ~ | NOT operatörü. |
| ^ | XOR operatörü. |

Example

/\*

25 : 00011001

14 : 00001110

& : 00001000 ->8

\*/

Console.Write(25 & 14); //8

Example

/\*

25 : 00011001

14 : 00001110

| : 00011111 ->31

\*/

Console.Write(25 | 14); //31

Console.Write(255 | 0); //255

Example

/\*

26 : 00011010

~ : 11100101 ->-27

\*/

Console.Write(~26); //-27

Example

/\*

25 : 00011001

14 : 00001110

^ : 00010111 ->23

\*/

Console.Write(25 ^ 14); //23

Console.Write(255 ^ 0); //255

(Kaydırma Operatörleri)

|  |  |
| --- | --- |
| Operator | Description |
| << | Sola kaydırma. |
| >> | Sağa kaydırma. |

Example

/\*

16 : 00010000

<< : 00100000 ->32

\*/

Console.WriteLine(16 << 1); //32 (Sayı 2 ile çarpılır)

Console.WriteLine(16 << 2); //64 (Sayı 2 defa 2 ile çarpılır)

Console.WriteLine(17 << 1); //34

Console.WriteLine(17 << 2); //68

Example

/\*

16 : 00010000

>> : 00001000 ->8

\*/

Console.WriteLine(16 >> 1); //8 (Sayı 2'ye bölünür)

Console.WriteLine(16 >> 2); //4 (Sayı 2 defa 2'ye bölünür)

//17/2 = 8.5 olmasına rağmen virgülden sonrası kaybolur.

Console.WriteLine(17 >> 1); //8

Console.WriteLine(17 >> 2); //4

(Ternary Operatörü)

Example

//Sözdizim : (condition) ? Process1 : Process2

Console.Write((1 == 1) ? "Equal" : "Not Equal"); //Equal

(as Operatörü)

as operatörü tip dönüşümü yapmak için kullanılır.

Example

object data = "Apple";

string fruit = data as string;

Yukarıdaki örnekte data değişkeni string türde veri içermeseydi hata meydana gelmezdi ancak geriye null değer dönerdi.

* as operatörü sadece referans tipler ile beraber kullanılabilmektedir.

(is Operatörü)

is operatörü tip kontrolü yaparken kullanılır. Geriye bool türünde değer döndürür.

Example

int number = 1;

bool result = number is int;

Console.Write(result); //True

Example

class Class1

{

}

class Class2 : Class1

{

}

class Program

{

static void Main(string[] args)

{

Class2 obj = new Class2();

bool result = obj is Class1;

Console.WriteLine(result); //True

Console.ReadLine();

}

}