

# *Operatörler* *(İşleç-Operators)*

- *Kelime anlamı;*
- *Herhangi bir işlemde, işlenenler üzerinde yapılması öngörülen işi tanımlayan özel damga, damga dizgisi vb. göstergeler.*
- *Tanımları önceden yapılmış **matematiksel** ya da **mantıksal** işlemleri yapmak için kullanılan özel karakterlere operatör denir.*

# *Operatör Türleri*

- *Aritmetik Operatörler(Arithmetic Operators)*
- *Karşılaştırma Operatörleri(Comparison operators)*
- *Mantıksal Operatörler(Logical Operators)*
- *Atama ve İşlemlili Atama Operatörleri(Assignment Operators)*

# Aritmetik Operatörler(Arithmetic Operators)

- *Toplama, çıkarma, çarpma, bölme, mod alma vb. matematiksel işlemlerin yapıldığı operatör türüdür.*

Aritmetiksel Operatörler			
Operatör	Açıklama	Kullanım 1	Kullanım 2
+	Ekleme(Toplama)	a=12+5;	a=sayi+3;
-	Çıkarma	a=12-5;	a=sayi-3;
*	Çarpma	a=12*5;	a=sayi*3;
/	Bölme	a=12/5;	a=sayi/3;
%	Kalan ( <b>mod</b> )	a=12%5;	a=sayi%3;
++	Birer Birer Arttırma	a++ ,++a;	z=a++;
--	Birer Birer Eksiltme	a-- ,--a;	z=a--;

# *İşlem Öncelikleri*

- *Sembollerin öncelikleri matematik de ki işlem öncelik sırasına göre işlemektedir.*

İşlem Önceliği	
1	()
2	*, /, %
3	+, -
Aynı seviyedekiler için soldan sağ doğru	

## Örnek

$$\text{sonuç} = \frac{a^2 + 5b - c}{b + c} + 2abc$$

İşlemi nasıl yapılır.  $a=2, b=4, c=3$  vererek sonucu bulunuz.

# Çözüm

- $\text{Sonuç} = (a^2 + 5 * b - c) / (b + c) + 2 * a * b * c$

```
double us = Math.Pow(a,2);
```

```
sonuc = (((Math.Pow(a,2)) + 5 * b - c) / (b + c)) + (2 * a * b * c);
```

# Karşılaştırma Operatörleri(Comparison operators)

- Karar yapılarında(if-else, switch-case) kullanılır. True ya da False dediğimiz Doğru yanlış durumlarında kullanılır. Genellikle Bool değişkeni ile kullanılır.*

Operatör	Açıklama
>	Büyüktür
<	Küçüktür
>=	Büyük Eşit
<=	Küçük Eşit
==	Eşit mi
!=	Eşit Değil mi
is	Tür dönüşümü(kontrol)
as	Tür dönüşümü(atama-dönüşüm)



# *is ve as Karşılaştırma Operatörleri*

- *Burada boxing dediğimiz kutulama yani object veri türüne atama işleminden sonra tekrar geri getirme durumunda(unboxing) atayacak olduğumuz değişkenin veri tipinin değer aralığına uygun bir değer içermiyorsa hata vermesi istenebilir.*
- *Is ve as operatörleri bu noktada bizlere tipler üzerinde kontrol yapma olanağı sağlamaktadır.*
- *Bu durumda değer kaybı yaşamamak için is operatörünü, atama işleminde sıkıntı yok ise as operatörü ile atama işlemini yapıyoruz*

# Mantıksal Operatörler(Logical Operators)

- *Mantıksal operatörler karşılaştırma operatörlerinin devamı niteliğinde olduğu söylenebilir.*
- *Bu operatörler true veya false sabit ya da değişkenleri mantıksal ve, veya, değil işlemine sokarlar.*

Operatör	Operatör Adı	Kullanım
&&	Ve Operatörü	a && b
	Ya da Operatörü	a    b
!	Değil Operatörü	!a

- ***||(veya) operatörü:*** *iki değerden birisi doğru ise doğru, ikisi de yanlış ise yanlış değerini döndürür ayrıca önceliği en az olan mantıksal operatördür.*
- ***&&(ve) operatörü :*** *iki değer in ikisi de doğru ise doğru, en az birisi yanlış ise yanlış değerini döndürür.*
- ***!(değil) operatörü :*** *aritmetik operatörlerden (++)arttırma ve (-- )eksiltme operatörleri ile eşit önceliklidir.*
- *&&(ve), ||(veya) operatörleri aritmetik, karşılaştırma ve bit sel operatörlere göre düşük önceliklidir fakat kendi aralarındaki öncelik sıralaması &&(ve), ||(veya) şeklindedir.*

# *Atama ve İşlemlili Atama Operatörleri (Assignment Operators)*

- *Sol tarafta kesinlikle yalnızca bir tane değişken olmalı. ( Hatalı kullanım:  $a+b=20$ )*
- *= operatöründe işlemler sağdan sola gerçekleşir ve = operatörü operandlarıyla birlikte atadığı değeri tutar*

byte b=7, a=1, c;

c=a=b;

Console.Write(a+" "+b+" "+c); sonuç olarak ; 7 7 7 yazar)

Atama Operatörü	Kullanımı	Örnek
= (Eşittir)	sonuc=5	Değişkene 5 değeri atanıyor
+= (Topla Eşitle)	sonuc+=5	sonuc=sonuc+5
-= (Çıkar Eşitle)	sonuc-=5	sonuc=sonuc-5
*= (Çarp Eşitle)	sonuc*=5	sonuc=sonuc*5
/= (Böl Eşitle)	sonuc/=5	sonuc=sonuc/5
%= (Mod Al Eşitle)	sonuc%=5	sonuc=sonuc%5

- `a=a+b;`    **yerine**    `a+=b;`
- `a=a-b;`    **yerine**    `a-=b;`
- `a=a*b;`    **yerine**    `a*=b;`
- `a=a/b;`    **yerine**    `a/=b;`
- `a=a%b;`    **yerine**    `a%=b;`
- `a=a<<b;`    **yerine**    `a<<=b;`
- `a=a>>b;`    **yerine**    `a>>=b;`
- `a=a&b;`    **yerine**    `a&=b;`
- `a=a^b;`    **yerine**    `a^=b;`
- `a=a|b;`    **yerine**    `a|=b;`