

С++ тилинин амалдары

Арифметикалык амалдар

С++ тилинде каалагандай эле туюнтма амалдардын белгиси менен байланышкан операндалардан турат.

Операнда-бул амал белгисинин оң жана сол жагындагы орундар.

Амал белгиси - бул компиляторго же арифметикалык же логикалык иш аракетерди жүргүзүүнү кабарлайт. С++ тилинде операндалардын санына жараша унардык жана бинардык амалдар бар. Унардык арифметикалык амалга минус (-) терс сандарды билгизе турган белги, ал эми унардык логикалык амалга тануу (not) амалы кирет.

Бинардык арифметикалык амалдарга кошу (+), кемитүү (-), көбөйтүү (*) жана бөлүү амалдары кирет. Бөлүүнүн эки түрү каралган бөлүү (/) жана бүтүн сандарды бөлүүдө, калдыгын табуучу бөлүү амалы (%). Бөлүү амалын жүргүзүүдө, төмөндөгүдөй эреже иштейт. Эгерде бөлүнүүчү да бөлүүчү да бүтүн типте болушса натыйжа бүтүн типте болот б.а бөлчөк бөлүгү эске алынбайт. Арифметикалык амалдарды аткарууга мисалдар:

Туюнтмалар	Жыйынтыктар
-7	-7
1+3*4-2	11
1+3/4-2	-1
1+3%4-2	2
1+3,0%4-2	-0,25

Бөлүү амалында операндалар бүтүн жана анык сандар болгон учурун карайлы.

Мисал-1;

```
init n,m;  
float r; n=9;  
m=4;  
r=n/m;
```

Бул программада $r=2.00$, себеби n жана m операндалары бүтүн типтүү, адегенде бүтүн сандарды бөлүү аткарылат жыйынтыгы 2 болот 0.25 болгон бөлүгү эске алынбайт, 2 бүтүн саны r калкыма чекиттүү өзгөрмөнүн тибине өзгөртүлүп 2.00 деп чыгарылат.

Мисал-2; float

```
x,y,z; x=9;  
y=4;  
z=x/y;
```

Бул мисалда z тин мааниси 2.25 болот, себеби бөлүү амалындагы эки операндасы тең калкыма чекиттүү өзгөрмөлөр.

Мисал-3; Бул мисалда типтерди келтирүү амалын бөлүүдө колдонулушун карайлы. Мейли `int n`, `float x` жана `n=7` болсун

1) $x=n/2$; жыйынтык: $x=3.000000$

2) $x=(float)n/2$; жыйынтык: $x=3.500000$ себеби n бүтүн тиби, калкыма чекиттүү

типке өзгөртүлүп андан кийин бөлүнүп жатат.

3) $x=(\text{float})(n/2)$; жыйынтык: $x=3.000000$ 1-учурга окшош адегенде бүтүн сандардын бөлүүсү аткарылып андан кийин калкыма чекиттүү типке келтирилет.

Чоңойтуучу жана кичирейтүүчү амалдар.

C++ тилинде чоңойтуучу амал катары “++” белгиси ал эми кичирейтүүчү амал катары “--” белгиси алынат. Чоңойтуучу жана кичирейтүүчү белги өзгөрмөнүн ысымынын алдына же өзгөрмөнүн ысымынан кийин коюлат ++x жана x++ деген жазуу $x=x+1$; операторуна эквиваленттүү б.а x тин маанисин 1 ге чоңойт дегенди билгизет, ал эми --x же x-- жазуусу x тин маанисин 1 ге кичирейт дегенди билдирет. Чоңойтуу жана кичирейтүү белгилеринин өзгөрмөнүн ысымынан мурун же кийин жазылышы өзүнчө мааниге ээ. Эгерде чоңойтуучу (кичирейтүүчү) белги өзгөрмөнүн ысымынан мурун келсе мисалы: ++a же --b, анда өзгөрмөнүн мааниси 1ге чоңойуп же 1ге кемип(азайып) туюнтманы эсептөөгө катышат, ал эми чоңойтуучу же кичирейтүүчү белги өзгөрмөнүн ысымынан кийин келсе мисалы a++ же b-- анда өзгөрмөнүн маанилери бирге чоңойтулбай же бирге кемитилбей (азайтылбай) (өзгөрмөнүн эски мааниси боюнча) туюнтманы эсептөөгө катышып туюнтма эсептелип бүткөндөн кийин гана өзгөрмөнүн мааниси 1 ге чоңойтулат же 1ге кичирейтилет.

Мисал; int

n,m,r; n=3;

m=3;

r=(n++)+(--m)

жыйынтыгы r=5, n=4,m=2 болот.

Менчиктөө амалы

Менчиктөө амалынан жалпы берилиши.

<өзгөрмөнүн ысымы>=<туюнтма> бул амал аткарылганда

<туюнтманын> тибин өзгөрмөнүн тибине өзгөртүлүп түзүлөт.

Мисал-1; int

x,y,z; x=7;

y=x*2+9;

z=y/x;

Бул мисалдагы менчиктөө операторлорун бир сапка менчиктөө амалы катары жазса болот.

$z=(y=(x=7)*2+9)/4$; Менчиктөө амалын аткаруу ондон солду көздөй жүрөт.

Мисал-2;

$x=y=z=10$ *a биринчи 10*a мааниси эсептелип z ке менчиктелет андан кийин y ке менчиктелет, аягында x өзгөрмөсүнө менчиктелет, демек иш аракеттер ондон солду көздөй жүрөт.

Менчиктөө амалынын да өзгөчө түрү кезигет

<өзгөрмөнүн ысымы> <амал белгиси>= <туюнтма> бул жазуу

<өзгөрмөнүн ысымы> = < өзгөрмөнүн ысымы > <амал белгиси> <туюнтма> деген жазууга тең күчтө.

Мисалы $x = x + 5$ жазуусун $x + 5$ жазсак болот $x + 5$ амалы $x = x + 5$ амалына караганда тез иштейт.

Мисал-3;

int a,b	Жыйынтыктар
$a = b = 5;$	$a = 5, b = 5$
$a += 1;$	$a = 6$
$b -= 3;$	$b = 2$
$a * = b$	$a = 12, b = 2$
$a /= ++b$	$a = 4, b = 3$
$a \% = b--$	$a = 1, b = 2$

Логикалык жана катыш амалдары

C++ тилинде 6 катыш жана 3 логикалык амал бар. “<”-кичине

“<=”-кичине же барабар

“>”-чоң

“>=”-чоң же барабар

“=”-барабар “!=”-

барабар эмес

Логикалык амалдар

&& -“жана” амалы

|| -“же” амалы

! -“эмес” же тануу амалы

Катыш амалдары эки туюнтманын маанилерин салыштырууда колдонулат.

C++ тилиндеги амалдардын таблицасы аткаруу тартиби менен.

Амалдын белгиси	Амалдын аталышы
()	Функцияга кайрылуу
[]	Массивтин элементин бөлүп алуу
.	Жазылыштын элементин бөлүп алуу
!	Логикалык тануу
~	Разряд боюнча тануу
-	Белгини өзгөртүү
++	Бирге чоңойтуу
--	Бирге кичирейтүү
&	Адреси көрсөтүү
*	Адрес боюнча кайрылуу

(tip)	Типти өзгөртүү
Sizeof ()	Байттагы өлчөмдү аныктоо
*	Көбөйтүү
/	Бөлүү
%	Бүтүн бөлүүдө калдыкты аныктоо
+	Кошу
-	Кемитүү
<<	Солду көздөй жылуу
>>	Оңду көздөй жылуу
<	Кичине
<=	Кичине же барабар
>	Чоң
>=	Чоң же барабар
= =	Барабар
!=	Барабар эмес
&	Разряд боюнча ”жана” амалы
^	Эки модулу боюнча кошуу амалы
	Разряд боюнча “же” амалы
&&	Логикалык «жана»
	Логикалык «же»
?:	Шарттуу амалы
=	Менчиктөө амалы