

演算子とは、演算を表す記号のことです。また演算が作用する対象をオペランドといいます。

目次

- [算術演算子](#)
- [単項演算子](#)
- [比較演算子](#)
- [論理演算子](#)
- [インクリメント演算子、デクリメント演算子](#)
- [ビット演算子](#)
- [代入演算子](#)
- [条件演算子](#)
- [カンマ演算子](#)
- [sizeof演算子](#)
- [ポインタ演算子、アドレス演算子](#)
- [メンバ演算子](#)
- [new演算子、delete演算子](#)

算術演算子

2つのオペランドに対し、加算、減算などを行う演算子です。

演算子	意味	例	答え
+	加算	1 + 2	3
-	減算	2 - 1	1
*	乗算	2 * 3	6
/	除算	2 / 1 3 / 2 3.0 / 2.0	2 1 整数同士の割り算は整数 1.5
%	剰余	2 % 1 3 % 2	0 1

除算では、整数同士の割り算の計算結果は整数となり、0に丸められます(正の場合は切り捨て、負の場合は切り上げ)。小数の計算結果を求める場合は、浮動小数点同士で計算をします。

単項演算子

1つのオペランドの前に使用する演算子を単項演算子といいます。

演算子	意味	例	答え
+	符号はそのまま	+1 +(-1)	1 -1
-	符号を反転	-1 -(-1)	-1 1

比較演算子

2つのオペランドを比較する演算子を比較演算子といいます。

演算子	意味	例
-----	----	---

<	より小さい	X < Y
>	より大きい	X > Y
<=	以下	X <= Y
>=	以上	X >= Y
==	等しい	X == Y
!=	等しくない	X != Y

論理演算子

論理演算を行い真偽を判定する演算子を論理演算子といいます。

演算子	意味	例	答え
&&	論理積(AND)	true && true true && false false && false	true false false
	論理和(OR)	true true true false false false	true true false
!	論理否定(NOT)	!true !false	false true

&&、|| は左オペランドの評価で結果が得られる場合に右オペランドは評価しません(ショートサーキット)。

インクリメント演算子、デクリメント演算子

オペランドの値に1加算する演算子をインクリメント演算子、1減算する演算子をデクリメント演算子といいます。

演算子	意味	例	答え
++	オペランドに1加算される	int a = 0; cout << ++a; 1 cout << ++a; 2 cout << a; 2 int b = 0; cout << b++; 0 cout << b++; 1 cout << b; 2	
--	オペランドから1減算される	int a = 0; cout << --a; -1 cout << --a; -2 cout << a; -2 int b = 0; cout << b--; 0 cout << b--; -1 cout << b; -2	

++ 演算子をオペランドの手前に書くと、オペランドの値を利用する前に1加算されます。オペランドの後方に書くと、オペランドの値を利用した後に1加算されます。同様に、--演算子をオペランドの手前に書くと、オペランドの値を利用する前に1減算されます。オペランドの後方に書くと、オペランドの値を利用した後に1減算されます。

ビット演算子

整数に対してビット単位で演算を行うための演算子をビット演算子といいます。

演算子	意味	例	答え
&	論理積	12 & 10	8
	論理和	12 10	14
^	排他的論理和	12 ^ 10	6
<<	左シフト	15 << 1	30
>>	右シフト	15 >> 1	7

代入演算子

オペランドに値を代入する際用いられる演算子が代入演算子となります。

演算子	意味	例	答え
=	代入	int a = 1;	1
+=	加算代入	int a = 1; a += 2;	3
-=	減算代入	int a = 2; a -= 1;	1
*=	乗算代入	int a = 2; a *= 3;	6
/=	除算代入	int a = 6; a /= 3;	2
%=	剰余代入	int a = 3; a %= 2;	1
&=	AND代入	bool b = true; b &= false;	False
=	OR代入	bool b = true; b = false;	true
^=	排他的OR代入	bool b = true; b ^= false;	True
<<=	左シフト代入	int a = 15; a <<= 1;	30
>>=	右シフト代入	int a = 15; a >>= 1;	7

+= 演算子の a += 2; は a = a + 2; と同じ意味になります。-=, *=, /=, %=, &=, |=, ^=, <<=, >>= も同様です。

条件演算子

条件の真偽に応じ、結果が変わる演算子を条件演算子と言います。

演算子	意味	例	答え
?:	三項条件	true ? 1 : 2 false ? 1 : 2	1 2

1番目のオペランドがtrueの場合、2番目のオペランドが結果となります。1番目のオペランドがfalseの場合、3番目のオペランドが結果となります。

カンマ演算子

複数行にわたるプログラム行を一行で表現する際使用する演算子がカンマ演算子です。

演算子	意味	例
,	カンマ演算子	int a = 1; int b = 2; int a = 1,b =2;

変数の宣言、for文、if文で使用されます。

sizeof演算子

型や変数のメモリサイズを調べる際使用する演算子がsizeof演算子です。

演算子	意味	例	答え
sizeof	sizeof演算子	sizeof(int); sizeof(double); int a; sizeof a; sizeof(a);	4 8 4 4

sizeof(データ型)、sizeof 変数名、sizeof(変数名)の形で使用されます。

ポインタ演算子、アドレス演算子

ポインタの宣言や、アドレスに格納された値を表す際に使用する演算子がポインタ演算子*です。

変数のアドレスを表す際に使用する演算子がアドレス演算子&です。

演算子	意味	例
*	ポインタ演算子	//ポインタの宣言 int *pa; //アドレスの代入 int a = 1; pa = &a;
&	アドレス演算子	//値の代入 *pa = 2;

種類	宣言	値	アドレス
変数	int a;	a	&a
ポインタ	int *pa;	*pa	pa

メンバ演算子

構造体、共用体、クラスのメンバへアクセスする際使用する演算子がメンバ演算子です。

演算子	意味	例
-----	----	---

.	メンバへ直接アクセス	<pre>struct book{ int id; }; book bk; bk.id = 1;</pre>
->	メンバへポインタを介しアクセス	<pre>struct book{ int id; }; book *pbk; pbk->id = 1;</pre>

new演算子、delete演算子

メモリ領域の動的確保に使用される演算子がnew演算子です。
動的確保されたメモリ領域を解放する演算子がdelete演算子です。

演算子	意味	例
new	メモリ領域の動的確保	<pre>int *pn; pn = new int;</pre>
delete	メモリ領域の解放	<pre>delete pn;</pre>