表达式计算

翁恺

表达式

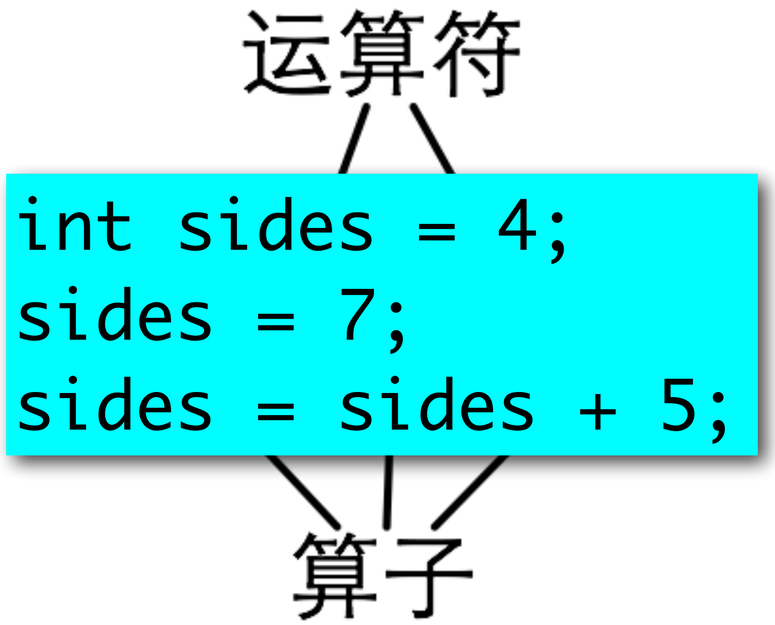
- 一个表达式是一系列运算符和算子的组合，用来计算一个值

```
amount = x * (1 + 0.033) * (1
+ 0.033) * (1 + 0.033);
total = 57;
count = count + 1;
value = (min / 2) * lastValue;
```

运算符

- 运算符（operator）是指进行运算的动作，比如加法运算符“+”，减法运算符“-”。
- 算子（operand）是指参与运算的值，这个值可能是常数，也可能是变量，还可能是一个方法的返回值

运算符



计算

- result = 12 + 6 / 2;
- result = (12 + 6) / 2;
- result = 4 * ((12 – 4) / 2);

四则运算

四则运算	C符号	意义
+	+	加
-	-	减
×	*	乘
÷	/	除
	%	取余
()	()	括号

- %表示取两个数相除以后的余数

计算时间差

- 输入两个时间，每个时间分别输入小时和分钟的值，然后输出两个时间之间的差，也以几小时几分表示

```
int hour1, minute1;
int hour2, minute2;

scanf("%d %d", &hour1, &minute1);
scanf("%d %d", &hour2, &minute2);
```

如果直接分别减，会出现分钟借位的情况：1点40分和2点10分的差?

计算时间差

```c
int hour1, minute1;
int hour2, minute2;

scanf("%d %d", &hour1, &minute1);
scanf("%d %d", &hour2, &minute2);

int t1 = hour1 * 60 + minute1;
int t2 = hour2 * 60 + minute2;

int t = t2-t1;

printf("时间差是%d小时%d分。", t/60, t%60);
```

- hour1*60+minute1—>转换为分钟为单位
- t/60—>小时部分；t%60—>分钟

求平均值

- 写一个程序，输入两个整数，输出它们的平均值

```
int a,b;

scanf("%d %d", &a, &b);

double c = (a+b)/2.0;

printf("%d和%d的平均值=%f\n", a, b, c);
```

运算符优先级

优先级	运算符	运算	结合关系	举例
1	+	单目不变	自右向左	a*+b
1	–	单目取负	自右向左	a*-b
2	*	乘	自左向右	a*b
2	/	除	自左向右	a/b
2	%	取余	自左向右	a%b
3	+	加	自左向右	a+b
3	-	减	自左向右	a-b
4	=	赋值	自右向左	a=b

单目运算符

- 只有一个算子的运算符：+、-

```
int a = 10;
int b = -20;
printf("%d", a * - b);
```

赋值运算符

- 赋值也是运算，也有结果
- a=6的结果是a被赋予的值，也就是6
- a=b=6 —> a=(b=6)

“嵌入式赋值”

```
int a = 6;
int b;
int c = 1+(b=a);
```

- 不利于阅读
- 容易产生错误

结合关系

- 一般自左向右
- 单目+和赋值=自右向左

这样的表达式太复杂，不容易阅读和理解，容易造成读程序时的误解。所以，要避免写出这样的复杂表达式来的。这个表达式应该被拆成若干个表达式，然后以明显的正确的顺序来进行计算。

```
result = a + b + 3 + c;
result = 2;
result = (result = result * 2) * 6 * (result = 3 + result);
```

计算复利

- 在银行存定期的时候，可以选择到期后自动转存，并将到期的利息计入本金合并转存。如果1年期的定期利率是3.3%，那么连续自动转存3年后，最初存入的x元定期会得到多少本息余额?
- 本息合计 = $x(1+3.3\%)^3$

计算复利

```
int x;
scanf("%d", &x);
double amount = x * (1 + 0.033) * (1 + 0.033) *
(1 + 0.033);
printf("%f", amount);
```

要计算任意年以后的本息金额，
就需要做(1+0.033)^n的计算

交换两个变量

- 如果已经有：

```
int a = 6;
int b = 5;
```

如何交换a、b两个变量的值?

程序是按步执行的

- 程序表达的是顺序执行的动作，而不是关系

```
a=b;
b=a;
```

是依次执行的，结果使得a和b都得到b原来的值

交换

```
int t = a;
a = b;
b = t;
```

复合赋值

- 5个算术运算符，+ - * / %，可以和赋值运算符“=”结合起来，形成复合赋值运算符：“+=”、“-=”、“*=”、“/=”和“%=”
  - total += 5;
  - total = total + 5;
- 注意两个运算符中间不要有空格

复合赋值

- total += (sum+100)/2;
  - total = total + (sum+100)/2;
- total * = sum+12;
  - total = total*(sum+12);
- total /= 12+6;
  - total = total / (12+6);

递增递减运算符

- “++”和“--”是两个很特殊的运算符，它们是单目运算符，这个算子还必须是变量。这两个运算符分别叫做递增和递减运算符，他们的作用就是给这个变量+1或者-1。
  - count++;
  - count += 1;
  - count = count + 1;

前缀后缀

- ++和--可以放在变量的前面，叫做前缀形式，也可以放在变量的后面，叫做后缀形式。
- a++的值是a加1以前的值，而++a的值是加了1以后的值，无论哪个，a自己的值都加了1了。

前缀后缀

表达式	运算	表达式的值
count++	给count加1	count原来的值
++count	给count加1	count+1以后的值
count--	给count减1	count原来的值
--count	给count减1	count-1以后的值

++--

- 这两个运算符有其历史来源
- 可以单独使用，但是不要组合进表达式
  - ++i++ -->?
  - i++++ —>?
- a = b+=c++-d+--e/-f