

INTRODUCCIÓN

Uno de los grandes amigos que tendrá el diseñador de videojuegos a la hora de programar las mecánicas y dinámicas de los videojuegos serán las matemáticas. Por ese motivo los algoritmos generalmente involucrarán operaciones matemáticas. En esta sección se analizarán las expresiones aritméticas y se ejercitará el desarrollo de algoritmos simples que involucren operaciones matemáticas.

QUE ES UNA EXPRESIÓN ARITMÉTICA

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres especiales.

Las expresiones algebraicas son aquellas que representan ideas que son utilizadas en la notación tradicional de la matemática, por ejemplo

$$a + (b + 3) + \sqrt{c}$$

Es una expresión algebraica, que es probable que le sea familiar: los paréntesis indican una prioridad en el cálculo, la $\sqrt{}$ representa a la raíz cuadrada y el símbolo + expresa suma.

Por otro lado a, b y c son variables que almacenan un valor. Los valores de la variable son involucrados en los cálculos realizados; mientras que el número 3 se denomina **argumento literal o simplemente valor constante**.

La suma es uno de los diferentes operadores aritméticos disponibles. La conexión entre variables (o variables y constantes) por medio de operadores los convierte en operandos. Es decir, por ejemplo, en $(b + 3)$, tanto b como 3 son operandos; mientras que + es el operador.

En los algoritmos es posible representar expresiones algebraicas mediante las denominadas expresiones aritméticas. **Una expresión aritmética** es por lo tanto una traducción dentro de los algoritmos de una expresión algebraica.

El resultado de una expresión aritmética es un tipo de datos numérico, por lo cual puede ser un entero o un real. En el caso de que una variable, constante o resultado de una operación sea real, el separador decimal se expresa con un punto (.). Por ejemplo, suponga que $b = 4.5$. Como puede observar su valor es un real, pero el separador decimal es el punto en lugar de la coma.

OPERADORES ARITMÉTICOS

Los operadores aritméticos son análogos a los usados en ecuaciones matemáticas. Resulta lógico que se apliquen sobre variables y constantes numéricas. Los operadores aritméticos son:

+	Suma
-	Resta
*	Multiplicación
/	División
** o ^	Potenciación
div	División entera
mod, %	Módulo (resto)

Ejemplo 1: Represente las siguientes operaciones matemáticas como una expresión dentro de un algoritmo e indique el resultado

Operación matemática	Expresión aritmética	Resultado
5×7	$5 * 7$	35
$\frac{6}{4}$	$6 / 4$	1.5
3^7	$3 ^ 7$	2187

OPERADORES DE INCREMENTO Y DECREMENTO

Muchos lenguajes de programación incorporan sus propios operadores aritméticos con el objetivo de simplificar ciertas operaciones. Algunos de estos han trascendido al uso de lenguaje debido a su recurrente uso, tales como los operadores de incremento y decremento.

También son conocidos como operadores unitarios (o unarios) y poseen la siguiente forma:

- **(++)**: operador de incremento: aumenta el valor de su operando en una unidad
- **(--)**: operador de decremento: disminuye el valor de su operando en una unidad

El valor resultante dependerá de que el operador se emplee como prefijo (antes de la variable) o como sufijo (después de la variable).

Si actúa como prefijo, el operador cambia el valor de la variable y devuelve este nuevo valor; en caso contrario, si actúa como sufijo, el resultado de la expresión es el valor de la variable, y después se modifica esta variable.

Ejemplo 2: Suponga que define una variable *i* con un valor por ejemplo 5. Entonces si a esa variable que inicialmente tiene el valor 5 se la somete al operador unario se obtendrá:

Algoritmo	Descripción	Resultado
<pre>resultado ← 4 i ← 4 resultado ← ++i Mostrar resultado, i</pre>	Incrementa <i>i</i> en 1. Luego se asigna el valor de <i>i</i> a resultado.	<pre>resultado = 5 i = 5</pre>
<pre>resultado ← 4 i ← 4 resultado ← i++ Mostrar resultado, i</pre>	Asigna el valor actual de <i>i</i> a resultado. Luego se incrementa el valor de <i>i</i> en 1.	<pre>resultado = 4 i = 5</pre>
<pre>resultado ← 4 i ← 4 resultado ← --i Mostrar resultado, i</pre>	Decrementa <i>i</i> en 1. Luego se asigna el valor de <i>i</i> a resultado.	<pre>resultado = 3 i = 3</pre>
<pre>resultado ← 4 i ← 4 resultado ← i-- Mostrar resultado, i</pre>	Asigna el valor actual de <i>i</i> a resultado. Luego se decrementa el valor de <i>i</i> en 1.	<pre>resultado = 4 i = 3</pre>

OPERADORES DE INCREMENTO Y DECREMENTO

Como se expresó anteriormente, el tipo de datos de la variable o de la constante, así como el operador involucrado brindan una idea del tipo de datos que corresponde al resultado de la operación. Esto se resume en la tabla 1.



Tabla 1. Operadores aritméticos y el tipo de datos del resultado

Operador	Significado	Tipos de operandos	Tipo de resultado
+	Signo positivo	Entero o real	Entero o real
-	Signo negativo	Entero o real	Entero o real
*	Multiplicación	Entero o real	Entero o real
/	División	Real	Real
div	División entera	Entero	Entero
mod, %	Módulo (resto)	Entero	Entero
++	Incremento	Entero	Entero
--	Decremento	Entero	Entero

REGLAS DE PRIORIDAD

Cuando en una expresión aritmética intervienen dos o más operandos se requiere de la aplicación de reglas matemáticas que permitan determinar el orden de las operaciones. Estas reglas se denominan reglas de prioridad.

Las reglas de prioridad son las siguientes:

1. Las operaciones que están encerradas entre paréntesis se evalúan primero. Si existen diferentes paréntesis anidados (interiores unos a otros), las expresiones más internas se evalúan primero.
2. Las operaciones aritméticas dentro de una expresión suelen seguir el siguiente orden de prioridad:
 - operador ()
 - operadores ++, -- unitarios,
 - operadores *, /, % (producto, división, módulo)
 - operadores +, - (suma y resta).
3. En los lenguajes que soportan la operación de exponenciación, este operador tiene la mayor prioridad.
4. En caso de coincidir varios operadores de igual prioridad en una expresión o subexpresión encerrada entre paréntesis, el orden de prioridad en este caso es de izquierda a derecha, y a esta propiedad se denomina asociatividad.

Los siguientes ejemplos son tomados de Joyanes Aguilar (2008, p. 98) para dar claridad sobre el uso de las reglas de paridad mencionadas.

Ejemplo 3: ¿Cuál es el resultado de la siguiente expresión?

$$-4 * 7 + 2 ^ 3 / 4 - 5$$

Solución

$$-4 * 7 + 2 ^ 3 / 4 - 5$$

resulta

$$\begin{aligned} & -4 * 7 + 8 / 4 - 5 \\ & -28 + 8 / 4 - 5 \\ & -28 + 2 - 5 \\ & -26 - 5 \\ & -31 \end{aligned}$$

Ejemplo 4: Obtener los resultados de las expresiones

$$a) 3 + 6 * 14$$

$$b) 8 + 7 * 3 + 4 * 6$$

Solución

$$a) \begin{array}{r} 3 + 6 * 14 \\ 3 + 84 \\ \hline 87 \end{array}$$

$$b) \begin{array}{r} 8 + 7 * 3 + 4 * 6 \\ 8 + 21 \quad 24 \\ \hline 29 + 24 \\ \hline 53 \end{array}$$

Ejemplo 5: Convertir en expresiones aritméticas las siguientes expresiones algebraicas

$$\begin{array}{l} 5 \cdot (x + y) \\ \frac{x + y}{u + \frac{w}{a}} \end{array}$$

$$\begin{array}{l} a^2 + b^2 \\ \frac{x}{y} \cdot (z + w) \end{array}$$

Los resultados serán:

$$\begin{array}{l} 5 * (x + y) \\ a ^2 + b ^2 \\ (x + y) / (u + w/a) \\ x / y * (z + w) \end{array}$$

Ejemplo 6: evalúe la precedencia de los paréntesis:

$A * (B + 3)$	la constante 3 se suma primero al valor de B, después este resultado se multiplica por el valor de A.
$(A * B) + 3$	A y B se multiplican primero y a continuación se suma 3.
$A + (B + C) + D$	esta expresión equivale a $A + B + C + D$
$(A + B/C) + D$	equivale a $A + B/C + D$
$A * B/C * D$	equivale a $((A * B)/C) * D$ y no a $(A * B)/(C * D)$.

Ejemplo 7: Evaluar la expresión $12 + 3 * 7 + 5 * 4$

En este ejemplo existen dos operadores de igual prioridad, * (multiplicación); por ello los pasos sucesivos son:

$$\begin{array}{l} 12 + \underline{3 * 7} + 5 * 4 \\ \quad 21 \\ 12 + 21 + \underline{5 * 4} \\ \quad \quad 20 \\ 12 + 21 + 20 = 53 \end{array}$$

BIBLIOGRAFIA

Fundamentos de Programación: Algoritmos, estructuras de datos y objetos. Cuarta edición. Luis Goyanes Aguilar. ISBN: 978-84-481-6111-8