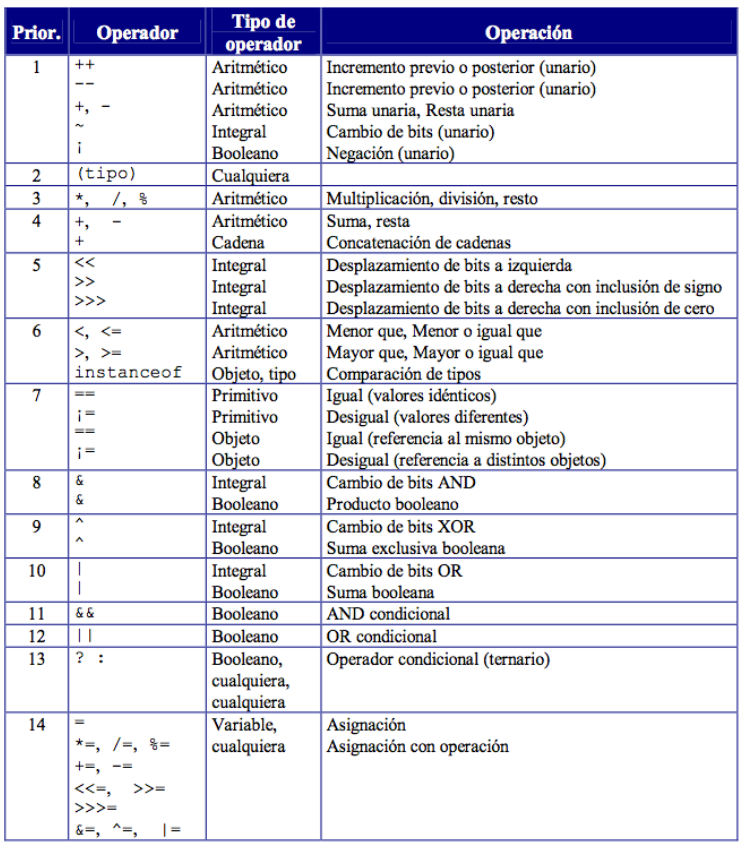
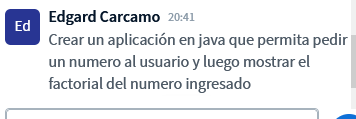
# Ejecución 8 del 6 de mayo

Crear una tabla con los tipos de operadores, función y datos en los que pueden ser aplicados.

|  |  |
| --- | --- |
| Operador de asignación **=** | Es un operador binario que asigna el valor del término de la derecha al operando de la izquierda. El operando de la izquierda suele ser el identificador de una variable. El término de la derecha es, en general, una expresión de un tipo de dato compatible; en particular puede ser una constante u otra variable. Como caso particular, y a diferencia de los demás operadores, este operador no se evalúa devolviendo un determinado valor. |
| Operadores aritméticos **( +, -, \*, /, %)** | El lenguaje de programación Java tiene varios operadores aritméticos para los datos numéricos enteros y reales |
| Operadores aritméticos incrementales **( ++ ) incremento**  **( -- ) decremento** | Los operadores aritméticos incrementales son operadores unarios (un único operando). El operando puede ser numérico o de tipo char y el resultado es del mismo tipo que el operando. Estos operadores pueden emplearse de dos formas dependiendo de su posición con respecto al operando. |
| Operadores aritméticos combinados **+= -= \*= /= %=** | Combinan un operador aritmético con el operador asignación. Como en el caso de los operadores aritméticos pueden tener operandos numéricos enteros o reales y el tipo específico de resultado numérico dependerá del tipo de éstos |
| Operadores de relación **== ¡= < > <= >=** | Realizan comparaciones entre datos compatibles de tipos primitivos (numéricos, carácter y booleanos) teniendo siempre un resultado booleano. Los operandos booleanos sólo pueden emplear los operadores de igualdad y desigualdad. |
| Operadores lógicos o booleanos **¡ | ^ & || &&** | Realizan operaciones sobre datos booleanos y tienen como resultado un valor booleano. |
| Operador condicional **?=** | Este operador ternario tomado de C/C++ permite devolver valores en función de una expresión lógica.  Si el resultado de evaluar la expresión lógica es verdadero, devuelve el valor de la primera expresión, y en caso contrario, devuelve el valor de la segunda expresión. |
| Operadores de bit **^ | ~ & << >> >>>** | Tienen operandos de tipo entero (o char) y un resultado de tipo entero. Realizan operaciones con dígitos (ceros y unos) de la representación binaria de los operandos. Exceptuando al operador negación, los demás operadores son binarios. |
| Operador concatenación de cadenas **+** | El operador concatenación +, es un operador binario que devuelve una cadena resultado de concatenar las dos cadenas que actúan como operandos. Si sólo uno de los operandos es de tipo cadena, el otro operando se convierte implícitamente en tipo cadena. |

Prioridad de los operadores y datos en los que pueden ser aplicado





package clase;

import java.util.Scanner;

public class CicloWhile {

public static void main(String[] args) {

Scanner lector = new Scanner(System.***in***);

System.***out***.println("Ingrese un número: ");

int n = lector.nextInt();

int factorial = 1;

int contador = 0;

while (contador < n) {

contador++;

if (contador == n) {

System.***out***.print(contador+" = "+n+"!");

} else {

System.***out***.print(contador+" x ");

}

factorial = factorial \* contador;

}

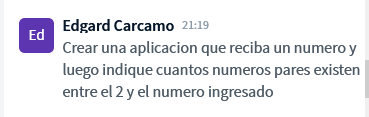
System.***out***.println("");

System.***out***.println("El resultado del factorial es: "+factorial);

lector.close();

}

}



package clase;

import java.util.Scanner;

public class CicloFor {

public static void main(String[] args) {

Scanner lector = new Scanner(System.***in***);

System.***out***.println("Ingrese un número: ");

int n = lector.nextInt();

int contador = 0;

while (n <= 2);

for (int i = 2; i <= n ; i+=2) {

System.***out***.print(i+" ");

contador++;

}

lector.close();

System.***out***.println(" ");

System.***out***.println("La cantidad de numeros pares contenidos en "+n+" son: "+contador);

}

}