

TRABAJO PRÁCTICO N°1

INTEGRANTES:

- **Máximo Aguilera**
- **Máximo Ceverino**
- **Valentin Fernández**
- **José Fausti**
- **Franco Orihuela**
- **Joaquín Riveros**

Ejercicio 1:

1. válido
2. invalido no puede empezar por un número
3. válido
4. invalido palabra reservada
5. válido
6. válido
7. invalido no puede llevar espacio
8. válido
9. invalido no puede llevar guion medio
10. válido
11. invalido usa un carácter especial
12. válido
13. invalido empieza con un número
14. válido
15. invalido usa un carácter reservado
16. válido
17. válido
18. válido
19. válido
20. invalido usa punto
21. válido
22. invalido usa un carácter especial
23. válido
24. válido
25. válido
26. válido
27. válido
28. válido
29. invalido empieza con numero
30. invalido usa un carácter especial
31. invalido empieza con numero
32. invalido palabra reservada

Ejercicio 2:

1. válido double decimal
2. válido double decimal

3. válido double decimal
4. válido double decimal
5. válido int decimal
6. invalido no puede usar _
7. invalido valor no valido para exponente
8. invalido tiene espacio
9. válido int octal
10. invalido tiene letras pero no tiene x
11. válido int hexadecimal
12. válido int hexadecimal
13. válido long decimal
14. invalido el guion bajo no puede ir en esa posición
15. válido int binario
16. válido int octal
17. invalido el literal sólo puede tener un punto
18. válido float hexadecimal
19. válido int binario
20. válido long decimal
21. válido int binario
22. válido long hexadecimal
23. invalido el guion bajo no puede ir en esa posición
24. invalido si fuese binario tendría que empezar por 0b
25. válido int hexadecimal
26. válido double decimal
27. invalido falta el valor del exponente entre E y F
28. válido float decimal
29. invalido el guion bajo no puede aparecer antes o después de e
30. invalido tendría que empezar por 0x
31. válido int hexadecimal
32. válido long hexadecimal
33. invalido tendría que empezar por 0x
34. válido int octal
35. válido double decimal
36. válido double decimal
37. invalido el guion bajo no puede aparecer antes o después de e
38. válido double decimal
39. invalido falta el valor del exponente
40. invalido un número binario sólo debe contener ceros y unos
41. válido int decimal
42. válido int hexadecimal
43. válido int hexadecimal
44. válido double decimal
45. invalido tendría que empezar por 0b
46. invalido solo el literal sólo puede tener un punto
47. válido float decimal
48. invalido g no es un caracter válido para un número hexadecimal

Ejercicio 3 :

1. válido
2. válido
3. válido
4. invalido debe contener un solo carácter a no ser que sea una secuencia de escape
5. válido
6. invalido debe contener un solo carácter a no ser que sea una secuencia de escape
7. invalido debe ir entre comillas simples
8. válido
9. invalido debe contener un solo carácter a no ser que sea una secuencia de escape
10. válido
11. invalido debe contener un solo carácter a no ser que sea una secuencia de escape y debe ir entre comillas simples
12. válido
13. válido
14. válido
15. invalido debe ir entre comillas simples
16. invalido debe ir entre comillas simples
17. válido
18. válido
19. válido
20. válido

Ejercicio 4:

- A. invalido debe ir entre comillas dobles
- B. válido
- C. invalido le falta las comillas del final
- D. válido
- E. válido
- F. válido
- G. válido
- H. válido
- I. invalido las comillas del final tienen que ser dobles
- J. válido
- K. invalido no contiene las comillas interiores como secuencias de escape tendría que ser así : "programación \"Java\"".
- L. invalido tiene que ir entre comillas dobles
- M. invalido tiene que ir entre comillas dobles
- N. válido
- O. válido

Ejercicio 5:**Act 1:**

1. `int p = 75, q = 2;`
2. `float x = 10.25F, y = 2.5F, z = 55.0F;`
3. `c) char a = 'a', b = 'b', c = 'c';`

4. **d) double raiz1 = 3.25, raiz2 = 30.5;**
5. **e) long contador = 1;**
6. **f) short indicador = 23;**
7. **int indice = 754;**
8. **double precio = 150.50, precioFinal = 100.50;**
9. **char car1 = 'm', car2 = 's';**
10. **byte valor = 3;**
11. **boolean primero = false, ultimo = true;**
12. **String nombre = "Maximo Aguilera";**

Act 2:

- A. int edad;
- B. String códigoPostal;
- C. double altura;
- D. char genero;
- E. String nombre;
- F. int númeroDeHijos;
- G. double iva;
- H. int tallaCamisa;
- I. double peso;
- J. double precio;
- K. String alumnoRepetidor;
- L. String mensaje;
- M. char letra;
- N. int minutos;
- O. int días;
- P. String matrículaCoche;
- Q. int contador
- R. boolean mayorDeEdad;
- S. String tallaCamiseta;

Ejercicio 9:

Calcular el tipo del resultado en operaciones aritméticas.

En aquellas expresiones aritméticas en las que intervienen operandos de distinto tipo (int, double, char, etc.) java convierte el tipo de los operandos que intervienen al tipo del operando de mayor precisión y este será el tipo del resultado obtenido en la operación.

Esta conversión de tipos se realiza de forma temporal, solamente para el cálculo de la operación. Los tipos originales de los operandos que intervienen siguen siendo los mismos después de la operación.

Debemos tener en cuenta que cuando se realiza la conversión temporal de tipos, los operandos de tipo short, byte y char se convierten a int.

Teniendo en cuenta todo esto, realiza el siguiente ejercicio.

Ejercicio: A partir de las siguientes declaraciones de variables:

byte b;

short s;

long ln;

int i, j;

float f;

double d;

char c;

a) $i + c = (\text{int})$

b) $f - c = (\text{float})$

c) $d + f = (\text{double})$

d) $d + i = (\text{double})$

e) $i / f = (\text{float})$

f) $s + j = (\text{int})$

g) $d + j = (\text{double})$

h) $s * c = (\text{int})$

i) $d + c = (\text{double})$

j) $b + c = (\text{int})$

k) $b / c + s = (\text{int})$

l) $c + c = (\text{int})$

m) $i + \text{ln} + d = (\text{double})$

n) $\text{ln} + c = (\text{long})$

o) $5 / j = (\text{int})$

p) $5.2 / j = (\text{double})$

q) $i * f * 2.5 = (\text{double})$

r) $\text{ln} * f * 2.5F = (\text{float})$

s) $j - 4L = (\text{long})$

t) $j - 4L * 2.5F = (\text{float})$

u) $b + 2.5 * i + 35F = (\text{double})$

v) $'a' + b = (\text{int})$

w) $'a' + c = (\text{int})$

x) $c + 2 = (\text{int})$

y) $c - \ln / 2 = (\text{int})$

z) $2 / i + 2.0 / j = (\text{double})$

Ejercicio 10:

Ejercicios sobre operadores lógicos Java

Ejercicio 1: Si X, Y y Z son variables de tipo boolean con valores X = true, Y = false, Z = true, determina el valor de las siguientes expresiones lógicas:

a) $(X \ \&\& \ Y) \ || \ (X \ \&\& \ Z) = \text{true}$ b) $(X \ || \ !Y) \ \&\& \ (!X \ || \ Z) = \text{true}$

c) $X \ || \ Y \ \&\& \ Z = \text{true}$ d) $!(X \ || \ Y) \ \&\& \ Z = \text{false}$

e) $X \ || \ Y \ || \ X \ \&\& \ !Z \ \&\& \ !Y = \text{true}$ f) $!X \ || \ !Y \ || \ Z \ \&\& \ X \ \&\& \ !Y = \text{true}$

Ejercicio 2: Si W, X, Y y Z son variables de tipo boolean con valores W = false, X = true, Y = true, Z = false, determina el valor de las siguientes expresiones lógicas:

a) $W \ || \ Y \ \&\& \ X \ \&\& \ W \ || \ Z$ b) $X \ \&\& \ !Y \ \&\& \ !X \ || \ !W \ \&\& \ Y$

c) $!(W \ || \ !Y) \ \&\& \ X \ || \ Z$ d) $X \ \&\& \ Y \ \&\& \ W \ || \ Z \ || \ X$

e) $Y \ || \ !(Y \ || \ Z \ \&\& \ W)$ f) $!X \ \&\& \ Y \ \&\& \ (!Z \ || \ !X)$

Soluciones:

Ejercicio 1:

A_ $(X \ \&\& \ Y) \ || \ (X \ \&\& \ Z) \Rightarrow \text{TRUE}$

$(T \ \&\& \ F) \ || \ (T \ \&\& \ T)$

$F \ || \ T$

T

B_ $(X \ || \ !Y) \ \&\& \ (!X \ || \ Z) \Rightarrow \text{TRUE}$

$(T \ || \ !F) \ \&\& \ (!T \ || \ T)$

$(T \ || \ T) \ \&\& \ (F \ || \ T)$

$T \ \&\& \ T$

T

C_ X || Y && Z => TRUE

T || F && T

T || F

T

D_ !(X || Y) && Z => FALSE

!(T || F) && T

!T && T

F && T

F

E_ X || Y || X && !Z && !Y => TRUE

T || F || T && !T && !F

T || F || T && F && T

T || F || F && T

T || F || F

T || F

T

F_ !X || !Y || Z && X && !Y => TRUE

!T || !F || T && T && !F

F || T || T && T && T

F || T || T && T

F || T || T

T || T

T

Ejercicio 2:

A_ $W \vee Y \wedge X \wedge W \vee Z \Rightarrow \text{FALSE}$

$F \vee T \wedge T \wedge F \vee F$

$F \vee T \wedge F \vee F$

$F \vee F \vee F$

$F \vee F$

F

B_ $X \wedge \neg Y \wedge \neg X \vee \neg W \wedge Y \Rightarrow \text{TRUE}$

$T \wedge \neg T \wedge \neg T \vee \neg F \wedge T$

$T \wedge F \wedge F \vee T \wedge T$

$F \wedge F \vee T \wedge T$

$F \vee T$

T

C_ $\neg(W \wedge \neg Y) \wedge X \vee Z \Rightarrow \text{TRUE}$

$\neg(F \wedge \neg T) \wedge T \vee F$

$\neg(F \wedge F) \wedge T \vee F$

$\neg F \wedge T \vee F$

$T \wedge T \vee F$

$T \vee F$

T

D_ $X \wedge Y \wedge W \vee Z \vee X \Rightarrow \text{TRUE}$

$T \wedge T \wedge F \vee F \vee T$

$T \wedge F \vee F \vee T$

$F \vee F \vee T$

$F \vee T$

T

E_ X || !(Y || Z && W) => TRUE

T || !(T || F && F)

T || !(T || F)

T || !(T)

T || F

T

F_ !X && Y && (!Z || !X) => FALSE

!T && T && (!F || !T)

F && T && (T || F)

F && T

F

Ejercicio 11:

- a) False
- b) False
- c) True
- d) False
- e) True
- f) True
- g) True
- h) False
- i) False
- j) False
- k) True
- l) True
- m) True
- n) false
- o) true
- p) ERROR
- p>false
- q) true
- r) false
- s) true
- t) false
- u) false
- v) true
- w) True
- x) false
- y) true

z) false

Ejercicio 12:

1)

```
int x = 10;
System.out.println(x);
10
x++;
System.out.println(x);
11
System.out.println(++x);
12
System.out.println(x++);
12
System.out.println(x);
13
System.out.println(x++);
13
System.out.println(++x);
15
System.out.println(++x);
16
++x;
x++;
System.out.println(++x);
19
System.out.println(x++);
19
System.out.println(++x);
21
```

2)

```
char A = 'c';
char B;
System.out.println(A++);
c
```

```
System.out.println(A++);
```

d

```
System.out.println(++A);
```

f

```
B = --A;
```

```
System.out.println(++A);
```

g

```
A++;
```

```
--B;
```

```
System.out.println(B++);
```

d

```
System.out.println(++B);
```

f

```
System.out.println(++A);
```

h

```
System.out.println(B--);
```

f

```
System.out.println(A);
```

h

```
System.out.println(B);
```

e

3)

a) i = ++j;

i=2

j=2

k=1

b) i = k++;

i=1

k=2

j=1

c) i = k + ++j;

i=3

k=1

j=2

```

d) i = i + j++;
i=2
e) i = j + ++k;
i=3
k=2
j=1
f) i = ++j + k++;
i=3
k=2
j=2
g) j = k-- + --i;
j=1
k=0
i=0
h) i = k + 1 + ++j;
i=4
k=1
j=2
i) i = ++i + --j + k--;
i=3
j=0
k=0
j) k = j-- + ++k;
k=3
j=0
i=1

```

Ejercicio 17:

1. $2 * n$
2. $3 * n$
3. $(A - B) * 6$
4. $(A * B) - (C + D)$
5. $(A \% 10) / 2$
6. $(N / 100) + ((N / 10) \% 10) + (N \% 10)$
7. $(N \% 10) + (M \% 10)$

8. $N \% 2 == 0$ and $N \% 3 == 0$
9. $(N \% 10) \% 2 == 0$
10. $(N / 100) \% 2 != 0$
11. $(N / 1000) \% 2 == 0$
12. $(A >= 'A') \text{ and } (A <= 'Z')$
13. $(A >= 'A') \&\& (A <= 'Z') \parallel (A >= 'a') \&\& (A <= 'z')$
14. $(A < 'A') \parallel (A > 'Z')$
15. $((A < 'A') \parallel (A > 'Z')) \&\& ((A < 'a') \parallel (A > 'z'))$
16. $((N \% 10 == 0)) \parallel ((N \% 10) == 7)$
17. $(\text{precio} >= 10) \&\& (\text{precio} < 50)$
18. $N = N + 77$ ó $N += 77$
19. $N = N - 3$ ó $N -= 3$
20. $N = N * 2$ ó $N *= 2$
21. $N = N + (N \% 10)$ ó $N += (N \% 10)$
22. $(N / 1000 == N \% 10) \&\& (N / 100 \% 10 == N / 10 \% 10)$
23. $M == 4 \parallel M == 6 \parallel M == 9 \parallel M == 11$
24. $N = N / 10$
25. $N = N \% 10000$
26. $(C >= '0') \&\& (C <= '9')$
27. $X = N / 1000 + N / 100 \% 10 + N / 10 \% 10 + N \% 10$
 $Y = M / 1000 + M / 100 \% 10 + M / 10 \% 10 + M \% 10$
 $Z = X + Y$
28. $MN) \parallel (AA - AN == 18) \&\& (MA == MN) \&\& (DA >= DN)(AA - AN > 18) \parallel (AA - AN == 18) \&\& (MA >$

Ejercicio 18:

1. $N / 1000 \% 2 == 0$
2. $N / 10000 == N / 1000 \% 10$
3. $N / 10000 == N \% 10 \&\& N / 1000 \% 10 == N / 10 \% 10$
4. $C == 'A' \parallel C == 'E' \parallel C == 'I' \parallel C == 'O' \parallel C == 'U'$
5. $C != 'A' \&\& C != 'E' \&\& C != 'I' \&\& C != 'O' \&\& C != 'U'$
6. $(A == 'a' \parallel A == 'e' \parallel A == 'i' \parallel A == 'o' \parallel A == 'u') \&\& B == 'a' \parallel B == 'e' \parallel B == 'i' \parallel B == 'o' \parallel B == 'u')$
7. $A \% 4 == 0 \&\& A \% 100 != 0 \parallel A \% 400 == 0$
8. $N / 10 == N \% 10$
9. $(N / 10 + N \% 10) \% 2 == 0$
10. $N / 10 \% 10 > N / 100 \&\& N / 10 \% 10 > N \% 10$
11. $N / 1000 == 4 \parallel N / 100 \% 10 == 4 \parallel N / 10 \% 10 == 4 \parallel N \% 10 == 4$
12. $X = N / 100$
13. $N = N - N \% 10$
14. $M = M - N \% 10$
15. $N = N + (N > 5 ? 5 : 100);$
16. $N = N < 0 ? 5 : 100;$

