Tp 1 (Introducción) Laboratorio Java.

Primera tarea Java.

Franco Sardi

Ejercicios: 1 -4 ; 8 ; 17;18

**1) Ejercicio sobre indentificadores Java**

1. registro1: válido.
2. 1registro: No válido. Un identificador no puede comenzar por un dígito.
3. archivo\_3: válido.
4. while: No válido. Es una palabra reservada del lenguaje.
5. $impuesto: válido.
6. año: válido.
7. primer apellido: no válido, un identificador no puede contener espacios en blanco.
8. En todo caso serían dos identificadores.
9. primer\_apellido: válido.
10. primer-apellido: No válido. Un identificador no puede contener el carácter – y en general no puede contener
11. caracteres que representen operadores: +, -, \*, /, %, <, >, etc.
12. primerApellido: válido.
13. Tom’s: No válido. Un identificador no puede contener el carácter comilla simple
14. C3PO: válido.
15. 123#: No válido. El carácter # no es válido en un identificador.
16. PesoMáximo: válido. Las vocales acentuadas pueden usarse para crear identificadores.
17. %descuento: No válido. Contiene el carácter no válido %
18. Weight: válido
19. $$precioMínimo: válido
20. \_$Único: válido
21. tamaño\_màximo: válido
22. peso.maximo: No válido. Un identificador no puede contener el carácter ‘.’
23. Precio\_\_\_: válido
24. matrícula?: No válido. Un identificador no puede contener el carácter ‘?’
25. cuántoVale: válido
26. high: válido
27. barça: válido
28. piragüista: válido
29. B\_011: válido
30. X012AB: válido
31. 70libro: No válido. Un identificador no puede comenzar por un dígito.
32. nombre&apellido: No válido. Un identificador no puede contener el carácter ‘&’
33. 0X1A: No válido. Un identificador no puede comenzar por un dígito.
34. else: No válido. Es una palabra reservada del lenguaje.

**2) Ejercicio sobre literales Java**

LITERAL VÁLIDO TIPO SISTEMA NUMERACIÓN

1) 0.5 SI double decimal

2) .5 SI. double decimal

Si la parte entera es 0 se puede omitir

3) 9.3e12 SI. double decimal

Representa el valor 9.3\*1012

4) 9.3e-12 SI. double decimal

Representa el valor 9.3\*10-12

5) 12345678 SI int decimal

6) 12345678\_L NO. Posición no válida para el carácter \_

7) 0.8E+0.8 NO. Valor no válido para el exponente en un literal Java.

No puede contener decimales. Intenta representar el valor 0.8\*100.8

8) 0.8E 8 NO. Hay un espacio en blanco entre E y 8

9) 05\_15 SI int octal

10) 018CDF NO. Si el número comienza por 0 está indicando que es un entero

escrito en octal pero contiene los caracteres CDF no válidos para el

sistema octal. Si fuese un entero hexadecimal debería comenzar por 0X.

11) 0XBC5DA SI int hexadecimal

12) 0x87e3a SI int hexadecimal

13) 234567L SI long decimal

14) 0\_B11 NO. El carácter \_ no puede separar los caracteres 0B que indican que

el número está escrito en binario.

15) 010101 SI int octal

16) 0\_557 SI int octal

17) .00.8E2 NO. El literal solo puede contener un punto (.) que separe la parte

entera de la parte decimal.

18) .3e3f SI float decimal

19) 0b111 SI int binario

20) 12\_234L SI long decimal

21) 0Xabcd SI int hexadecimal

22) 0xabcEL SI long hexadecimal

23) \_234 NO. El carácter \_ no puede aparecer al principio del número.

24) 1010B NO. Si fuese binario debería empezar por 0B: 0B1010

25) 0x1010B SI int hexadecimal

26) 1\_234.2E-2 SI double decimal

Representa el valor 1234.2\*10-2

27) 1234.2EF NO. Falta el valor del exponente entre los caracteres E y F

28) 1234.2E3F SI float decimal

Representa el valor 1234.2\*103

29) 1\_1.2e\_2 NO. En un literal Java de tipo float o double el carácter \_ no puede

aparecer antes o después del carácter e.

En este caso se quiere representar el valor 11.2\*102

30) 0bABCDL NO. Comienza por 0b. Si fuese hexadecimal debería comenzar por 0X.

31) 0X1A SI int hexadecimal

32) 0X12AL SI long hexadecimal

33) abcd NO. Si fuese hexadecimal debería comenzar por 0X.

34) 0125 SI int octal

35) .01011 SI double decimal

36) 3e12 SI double decimal

Representa el valor 3\*1012

37) 3\_e12 NO. En un literal Java de tipo float o double el carácter \_ no puede

aparecer antes o después del carácter e.

En este caso se quiere representar el valor 3\*1012

38) -3E-1\_2 SI double decimal

Representa el valor -3\*10-12

39) 0.8E NO. Falta el valor del exponente

40) 0B1212 NO. Un número binario (0B) solo puede contener ceros y unos

41) 1\_2\_3 SI int decimal

42) 0xedad SI int hexadecimal

43) 0XBE2 SI int hexadecimal

44) 101e2 SI double decimal

Representa el valor 101\*102

45) B1101 NO. Si fuese binario debería empezar por 0B

46) 1.34.5 NO. El literal solo puede contener un punto (.) que separe la parte

entera de la parte decimal.

47) 12.3E4F SI float decimal

Representa el valor 12.3\*104

48) 0X12AG NO. G no es un carácter válido para un número hexadecimal (0X)

**3) Ejercicios Sobre literales de tipo char**

LITERAL VÁLIDO

1) 'a' SI

2) '$' SI

3) '\n' SI Es una secuencia de escape

4) '/n' NO No válido. Un literal de tipo carácter debe estar formado por un

solo carácter a no ser que sea una secuencia de escape.

5) '\\' SI Es una secuencia de escape

6) '\ñ' NO Un literal de tipo carácter debe estar formado por un solo

carácter a no ser que sea una secuencia de escape.

7) "T" NO Los literales de tipo carácter deben ir entre comillas simples.

8) 'ñ' SI

9) 'xyz' NO Un literal de tipo carácter debe estar formado por un solo

carácter a no ser que sea una secuencia de escape.

10) '\u0066' SI Representa un valor Unicode.

11) "XYZ" NO Los literales de tipo carácter están formados por un solo

carácter y deben ir entre comillas simples.

12) '4' SI

13) '\t' SI Es una secuencia de escape

14) '\b' SI Es una secuencia de escape

15) k NO Los literales de tipo char deben inr entre comillas simples.

16) + NO Los literales de tipo char deben inr entre comillas simples.

17) '+' SI

18) '?' SI

19) 'â' SI

20) ':' SI

**4)Ejercicios sobre literales de tipo String**

LITERAL VÁLIDO

a) '8:15 P.M.' NO Los literales de tipo String deben ir entre

comillas dobles

b) "Rojo, Blanco y Azul" SI

c) "Nombre: NO Faltan las comillas finales

d) "Capítulo \’3\’" SI Un literal de tipo String puede contener secuencias

de escape. En este caso contiene la secuencia \'

e) "1.3e-1-2" SI

f) "" SI Representa una cadena vacía.

g) " " SI Es un String formado por un solo espacio en blanco

h) "A" SI

i) "FP' NO Las comillas finales en un String deben ser dobles.

j) " programación " SI

k) "programación "Java"" NO Para que fuese válido, el String debería contener las

comillas interiores como secuencias de escape

"programación \"Java\""

l) programación NO Los literales de tipo String deben en ir entre

comillas dobles

m) 'W' NO Los literales de tipo String deben en ir entre

comillas dobles

n) "\n" SI

o) "4 + 5 \* 2" SI

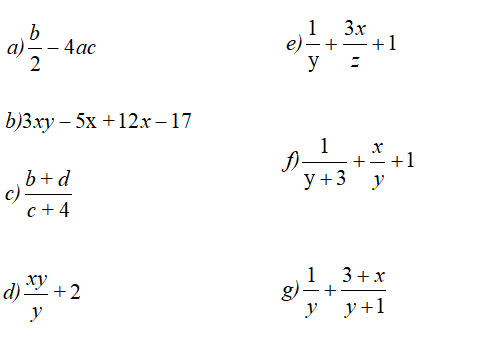
**8) Ejercicio: Convierte en expresiones algebraicas las siguientes expresiones algorítmicas.**

a) b/2-4\*a\*c e) 1/y+3\*x/z+1

b) 3\*x\*y-5\*x+12\*x-17 f) 1/(y+3)+x/y+1

c) (b+d)/(c+4) g) 1/y+(3+x)/(y+1)

d) (x\*y)/y+2



**Ejercicio 17**

1. El doble de un número N

2 \* N

2. El triple de un número N

3 \* N

3. Seis veces la diferencia de dos números enteros A y B

6 \* (A – B)

4. La diferencia entre el producto de A por B y la suma de C más D

A \* B – ( C + D )

5. La mitad de la última cifra de un número entero A

A % 10 / 2

6. La suma de los dígitos de un número entero N de 3 cifras

N/100 + N%10 + N/10%10

7. La suma de la última cifra de los números enteros N y M

N%10 + M%10

8. Comprobar si un número entero N es múltiplo de 2 y de 3

N%2 == 0 && N%3 == 0

9. Comprobar si la última cifra de un número entero N es par

N%10%2 == 0

10. Comprobar si la primera cifra de un número entero N de 3 cifras es impar

N/100%2 != 0 También es válida esta solución -> N/100%2 == 1

11. Comprobar si la primera cifra de un número entero N de 4 cifras es par

N/1000%2 == 0

12. Comprobar si una variable A de tipo carácter contiene una letra mayúscula

A >= ’A’ && A <= ’Z’

13. Comprobar si una variable A de tipo carácter contiene una letra mayúscula o minúscula

A >= ’A’ && A <= ’Z’ || A >= ’a’ && A <= ’z’

14. Comprobar si una variable A de tipo carácter no contiene una letra mayúscula

A < ’A’ || A > ’Z’

15. Comprobar si una variable A de tipo carácter no contiene una letra mayúscula o minúscula

(A < ’A’ || A > ’Z’) && (A < ’a’ || A > ’z’)

También es válida esta solución:

!(A >= ’A’ && A <= ’Z’ || A >= ’a’ && A <= ’z’)

16. Comprobar si el contenido de la variable N termina en 0 ó en 7

N%10 == 0 || N%10 == 7

17. Comprobar si el contenido de la variable precio es igual o mayor que 10€ y menor que 50€

precio >= 10 && precio < 50

18. Modificar el valor de la variable entera N incrementándolo en 77.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 87.

N = N +77 ó N+=77

19. Modificar el valor de la variable entera M disminuyéndolo en 3.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 7.

N = N – 3 ó N-=3

20. Modificar el valor de la variable entera X duplicando su valor.

Por ejemplo, si N contiene el valor 10, después de la operación contendrá el valor 20.

N = N \* 2 ó N\*=2

21. Sumar el dígito más a la derecha de N al propio valor de N.

Por ejemplo, si N contiene el valor 463 después de la operación contendrá el valor 466 ( 463 + 3 )

N = N + N%10 ó N+=N%10

22. Comprobar si un número entero N de cuatro cifras es capicúa.

Un número es capicúa si se puede leer igual de derecha a izquierda que de izquierda a derecha.

N/1000 == N%10 && N/100%10 == N/10%10

23. Una variable entera M contiene un número de mes. Comprobar si corresponde a un mes de 30 días.

M == 4 || M == 6 || M == 9 || M == 11

24. Quitarle a un número entero N su última cifra. Supondremos que N tiene más de una cifra.

Por ejemplo si N contiene el valor 123 después de la operación contendrá el valor 12.

N = N/10

25. Quitarle a un número entero N de 5 cifras su primera cifra.

Por ejemplo si N contiene el valor 12345, después de la operación contendrá el valor 2345.

N = N % 10000

26. Comprobar si una variable C de tipo char contiene un dígito. (Carácter entre 0 y 9)

C >= ‘0’ && C <= ‘9’

27. Dada dos variables enteras N y M de cuatro cifras, sumar las cifras de N y guardar la suma en la variable X, sumar las cifras de M y guardar la suma en la variable Y. Finalmente guarda en la variable Z la suma de X e Y.

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. A partir de las variables enteras DN, MN y AN que contienen el día, mes y año de nacimiento de una persona y las variables DA, MA y AA que contienen el día, mes y año actual, escribe la expresión algorítmica que compruebe si esa persona ya ha cumplido los 18 años.

AA – AN > 18 || AA – AN == 18 && MA > MN || AA – AN == 18 && MA == MN && DA >= DN}

**Ejercicio 18**

1. Dado un número N de cuatro cifras, comprobar si la primera cifra(la más a la izquierda) es impar.

N / 1000 % 2 == 0

2. Dado un número N de cinco cifras, comprobar si la primera cifra(la más a la izquierda) es igual a la segunda

N / 10000 == N / 1000 % 10

3. Determinar si un número entero N de 5 cifras es capicúa

N / 10000 == N % 10 && N / 1000 % 10 == N / 10 % 10

4. Comprobar si una variable C de tipo char contiene una vocal mayúscula

C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’

5. Comprobar si una variable C de tipo char no contiene una vocal mayúscula

Dos soluciones posibles:

!(C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’)

C != ‘A’ && C != ‘E’ && C != ‘I’ && C != ‘O’ && C != ‘U’

6. Dadas dos variables A y B de tipo char, comprobar si las dos contienen una vocal minúscula

(A == ‘a’ || A == ‘e’ || A == ‘i’ || A == ‘o’ || A == ‘u’) &&

B == ‘a’ || B == ‘e’ || B == ‘i’ || B == ‘o’ || B == ‘u’)

7. Dada una variable A que contiene un año, determinar si ese año es bisiesto. Un año es bisiesto si es divisible por 4 y no por 100 ó si es divisible por 400

A % 4 == 0 && A % 100 != 0 || A % 400 == 0

8. Dado un número N de dos cifras, comprobar si las dos cifras son iguales

N / 10 == N % 10

9. Dado un número N de dos cifras, comprobar si la suma de sus cifras es un número par

(N / 10 + N % 10) % 2 == 0

10. Dado un número N de tres cifras, comprobar si la cifra del centro es la mayor

N / 10 % 10 > N / 100 && N / 10 % 10 > N % 10

11. Dado un número N de cuatro cifras, comprobar si alguna de las cifras es un 4

N / 1000 == 4 || N / 100 % 10 == 4 || N / 10 % 10 == 4 || N % 10 == 4

12. Dado un número N de 4 cifras, asigna a una variable X las dos primeras cifras del número

X = N / 100

13. Dado un número entero N, modifícalo restando a N el valor de su última cifra. Por ejemplo, si N = 123, el valor final de N debe ser 120

N = N – N % 10 o también N -= N % 10

14. Dados dos números enteros N y M, modifica M restándole la última cifra de N. Por ejemplo si M = 123 y N = 47, el valor final de M debe ser 116

M = M - N % 10 o también M -= N % 10

15. Si el valor de una variable entera N es positivo, sumarle 5 sino sumarle 10

N = N + (N > 5 ? 5 : 100); o también N += N > 5 ? 5 : 100;

16. Si el valor de una variable entera N es negativo, asignarle el valor 5 sino asignarle el valor 100

N = N < 0 ? 5 : 100;