

## Preguntas Java Cuestionario 4

1. 1 FINN 1 JAKE 2 JAKE 3 FINN 3 JAKE

2. B: `protected int lightsaber(long x){ return 0; }`  
C: `private int lightsaber (long x){ return 0; }`  
E: `protected long lightsaber (int x, int y){ return 0; }`  
F: `public int lightsaber (int x){ return 0; }`  
G: `protected long lightsaber(long x){ return 0; } }`

El método no puede ser `private` porque no se puede reducir la visibilidad, por lo que la primera opción queda descartada. La segunda, tercera, quinta y séptima opción son un `overload`. La cuarta opción es un `override` que no está bien hecho. La sexta opción es un `override`

3. 15 16 6

Pasa por todos los constructores, llamándolos con `this()`

4. `arachnid spider s s`

Al crear un objeto `Spider`, primero pasa por la clase `Arachnid` y su constructor vacío (imprime `"arachnid"`). Después pasa por el constructor vacío de `Spider` (imprime `"spider"`). Después imprime `s` con `this.type`. Después vuelve a imprimir `s` con `super.type` ya que aunque se declaran diferentes valores en las clases, es la misma variable de referencia, que se modificó el objeto al que apuntaba cuando pasó por por la función `run` de `Spider`

5. 2 2

Por la forma en que está declarado el post-decremento, sí afecta a la variable. El pre-incremento se hace antes de imprimir y también afecta a la variable, por lo que al final del programa `b = 2`

6. 997 997

7. 1 4 2

Para escoger qué función se va a usar, se va de lo más particular a lo más general, dejando los `varargs` como último recurso

8. `ClassCastException`

El objeto al que apunta la variable de referencia es `Base1`, clase hermana de `Base2`, por lo que lanza una excepción

9. Error de compilación

Trata de asignar una suma de enteros a un `String`

10.      `Number1: 7 Number2: 7`

El primer entero que se declara está en binario, convertido a decimal es 7. El segundo entero que se declara está en hexadecimal, sin embargo, el programa imprime dos veces el primer número

11.      20

Cuando llama a la función `calc(2)` asigna a la variable `num` (del objeto) `2 * 10`, es decir, 20

12. La sentencia del import esta incompleta

La librería en la que se encuentra Random es java.util, no java.lang, por lo que arroja un error de compilación

13. 10 12

La primera vez que imprime, primero imprime y luego hace el incremento. La segunda vez que imprime primero hace el incremento y luego imprime

14. sg= 5

El for inicia en 11 y decrementa de uno en uno mientras sea mayor a 6, lo que son 5 veces. Por cada una de esas veces, sg aumenta en 1, comenzando en 0

15. A: Un ArrayList es mutable

B: Un arreglo tiene un tamaño fijo

D: Un arreglo permite múltiples dimensiones

E: ArrayList esta ordenado (tiene índice)

F: array esta ordenado (tiene índice)

16. Error de compilación

En el while se necesita algo que retorne un booleano

17. 1 3 5

Sin usar el Stream

18.     Object x = new Object(); *//clase*  
       (interface) y;  
       x = y;

19.     int

No se pueden hacer operaciones con byte porque se puede desbordar la información

20.     JDBC

La aplicación estándar para acceder a las bases de datos

21.     Los paquetes permiten limitar el acceso a las clases, métodos o variables de las clases fuera del paquete

22.     boolean a = (3>6);

23.     RuntimeException

Ya que el if está vacío, se arroja la RuntimeException, que se imprime en el catch

24.     Error de compilación

No se puede reducir la visibilidad de la función, tiene que ser public

25.     Spider Arth

Aunque Spider solo acepta int en su método, hereda de Arthropod y esta sí acepta double en su método, por lo que entra en esta función la segunda vez

26.        Wed Wed Wed

Por cada ciclo en el for se crea una variable nueva llamada d2, a la que se le asignan todos los valores del enum. Entonces, se imprime el mismo elemento todas las veces que haya un ciclo en el for

27.        true 20 false

$x = (y \neq 10) \wedge (z = \text{false}) = (\text{true}) \wedge (\text{false}) = \text{true}$

En xor (^) si los valores son iguales es false, si son diferentes es true

28.        2 4 6 8 5

Cuando se ejecuta una clase, aunque no haya nada en la función main se ejecutan los métodos anónimos estáticos. Así se imprime "2" y "4" y se crea un nuevo objeto, lo que hace que se ejecuten los métodos anónimos que no son estáticos y se imprime "6" y "8". Finalmente pasa por el constructor, que imprime "5"

29.        They dont match

El operador != pregunta si apuntan a diferentes objetos, lo que así es ya que un String se encuentra en el pool de Strings y el otro es un objeto independiente

30.        1 2 4 NullPointerException

Cuando entra al try imprime el 2 pero al intentar manejar una referencia que no apunta a nada lanza una NullPointerException. El catch atrapa la excepción el imprime el 4. Después arroja nuevamente la excepción, para la que no hay ningún catch y se desborda el programa

31. 1

La operación se realizó correctamente

32. the profits are 676000, 18330000, 27980000

Cuando se requieren los atributos se tienen que tomar en cuenta los atributos de la variable de referencia. Cuando se requieren los métodos se tienen que tomar en cuenta los métodos del objeto

33. DUHAST FEEL YELLOW FIX YOU  
ArrayIndexOutOfBoundsException

El for continua mientras i sea menor o igual a la longitud del arreglo, es decir, hasta el 4. El último índice del arreglo es 3. Entonces imprime el arreglo hasta que entra por última vez en el for y entonces lanza la excepción

34. beans, 2 beans, 3 egg, 2 egg, 3 ham, 2 ham, 3 juice,  
2 juice, 3

35. A: StringBuilder es más rápido que StringBuffer  
B: StringBuffer maneja los hilos de forma segura,  
StringBuilder no

36. `Compila sin errores, no imprime nada`

37.

38. El ciclo for mejorado: `foreach`. Ofrece una sintaxis simple para iterar pero no puede ser usado para eliminar elementos de una colección

39.

40. `Equal`

El método `equalsIgnoreCase()` compara el contenido de las cadenas ignorando si son mayúsculas o minúsculas

41. `apple banana grapes pears`

Lo que hace la función es comparar los primeros 2 elementos, en este caso los intercambia. Después compara "banana" y "pears": los deja como están. Después compara "pears" y "grapes": los intercambia

42. `ArrayIndexOutOfBoundsException`

En el arreglo no existe el índice -1

43. `0`

El valor por default es 0. No pasa por `Salmon()` ya que no es el constructor, los constructores no devuelven nada, ni void

44. Error de compilación

Se usa la variable c2 antes de declararla

45. 611

$++a + b++ * c = (\text{primero se hace el incremento}) 11 + 20 * 30$   
 $= 11 + 600 = 611$

46. Error de compilación

Cuando se hace uso de los carargs, deben ser los últimos que reciba la función

47. plant leaves plant leaves

Primero pasa por el constructor vacío de Plant ("plant"). Pasa por el constructor vacío de Tree. Pasa por el constructor vacío de Forest, ahí se regresa al constructor con un String de Tree ("leaves") y crea un nuevo objeto Tree con un String. Pasa por el constructor vacío de Plant ("plant") y pasa por el constructor con un String de Tree ("leaves")

48. true

La función intern() copia el contenido de un String y lo pone en el pool de Strings

49. Bucle infinito

A: while(true);  
B: while(1==1){}



50.        0

El valor de z nunca llega a alterarse

51.        Exception StackOverflowError

Rekursividad: al crear un objeto, el constructor crea otro y otro y otro...

52.        Error de compilación

Trata de manipular un String vacío

53.        10 +235

La primera vez imprime una suma de enteros (la resuelve antes). La segunda vez imprime el carácter "+" por lo que entiende el resto como concatenación de cadenas, es decir, no resuelve la suma de enteros porque los ve como cadenas

54.        Here1 Here3

55.        Error de compilación

Llama a la clase count como si fuera un método, no hay ningún objeto count creado

56.        magic town

Se ejecuta solo lo que hay dentro de public static void main (String[] args), las demás funciones solo se ejecutan si se llaman dentro de esta función

57. 2 1

La primera vez hace la suma  $0+2$  y luego hace el incremento

La segunda vez  $a=1$  por el incremento que se hizo en la línea anterior

58. Error de compilación

No puede saber que tipo de dato es E

59. Error de compilación

Break es una palabra reservada

60. 3 4

Primero entra al constructor con un entero. Después entra al constructor con un objeto, no puede entrar como si fuera un entero por la L

61. everything

62. Error de compilación

Hace falta un try-catch para las posibles excepciones en la conversión de String a número

63. True

Concatenar una cadena con una cadena vacía no cambia la primera cadena. Ambas variables apuntan al pool de Strings

64. Error de compilación

No se pueden declarar los case de esa manera

65. -100

66. Error de compilación

El case 200 sobrepasa el valor que puede tener un byte

67. ClassCastException

No se puede castear un objeto a su variable hija

68. ANY FRUIT WILL DO APPLE MANGO BANANA

Distingue de mayúsculas y minúsculas y continua hasta que encuentra un break

69. Error de compilación

Animal necesita tener un constructor de parámetros ya que Dog hereda de Animal

70. 333.6789 234.111

Si alguna de las cadenas no fuera un número, lanzaría la excepción y se desbordaría el programa. El throws en el main reconoce la excepción aunque no la maneja

71. p1 p1 p2 p3

La cola es p1, p2, p3. `queue.peek()` devuelve el primer elemento (p1). `queue.poll()` devuelve y elimina el primer elemento (p1) porque es una cola. Entonces, la cola queda así: p2, p3

72.      10 +215

La primera vez imprime una suma de enteros. La segunda vez, gracias a "+", imprime una concatenación de cadenas: imprime +, imprime 2 y al final imprime el resultado de la multiplicación