Respuestas del Examen de Fundamentos de Java

1. c) String Explicación: String es un tipo de dato de referencia, no un tipo primitivo. Los tipos primitivos en Java son byte, short, int, long, float, double, boolean y char.

2. b) final Explicación: En Java, se utiliza la palabra clave 'final' para declarar constantes. 'const' se usa en otros lenguajes, 'static' se usa para variables de clase, y 'var' es para inferencia de tipos.

3. b) 11 Explicación: La multiplicación tiene precedencia sobre la suma, por lo que primero se realiza 3 * 2 = 6, y luego 5 + 6 = 11.

4. b) == Explicación: En Java, '==' se usa para comparar igualdad. '=' es para asignación, '!=' para desigualdad, y '===' no existe en Java.

5. a) switch Explicación: 'switch' es una estructura de control de flujo. 'print' es para salida, 'class' define una clase, e 'import' se usa para importar paquetes.

6. c) Devuelve el residuo de una división Explicación: El operador % es el operador módulo, que devuelve el residuo de una división.

7. c) int[] numbers = new int[5]; Explicación: Esta es la sintaxis correcta para declarar un array de enteros con 5 elementos en Java.

8. c) Scanner.nextLine() Explicación: Scanner.nextLine() es el método más comúnmente usado para leer entrada del usuario. Las otras opciones no son métodos estándar en Java para leer entrada.

9. b) 16 Explicación: Primero, x se incrementa en 3 (x = 8), luego se multiplica por 2 (x = 16).

10. b) Incrementa x después de usar su valor Explicación: El operador '++' postfijo incrementa la variable después de que su valor actual ha sido usado en la expresión.

11. d) repeat-until Explicación: 'repeat-until' no es una estructura de repetición en Java. Las estructuras válidas son for, while y do-while.

12. b) break Explicación: 'break' se usa para salir inmediatamente de un bucle. 'continue' salta a la siguiente iteración, 'return' sale del método, y 'exit' no es una palabra clave en Java.

13. a) true Explicación: (true && false) es false, pero false || true es true.

14. b) Niega una expresión booleana Explicación: El operador '!' es el operador de negación lógica en Java.

15. c) public static void main(String[] args) Explicación: Esta es la firma correcta del método main en Java. Debe ser public, static, y void, con un array de String como parámetro.

16. c) 3 Explicación: Los índices de arrays en Java comienzan en 0, por lo que arr[2] se refiere al tercer elemento del array, que es 3.

17. b) Inicializar objetos Explicación: El propósito principal de un constructor es inicializar nuevos objetos de una clase.

18. c) new Explicación: La palabra clave 'new' se usa para crear nuevas instancias de objetos en Java.

19. b) false Explicación: El valor por defecto de una variable boolean en Java es false.

20. b) Devuelve el número de caracteres Explicación: El método length() en un String devuelve el número total de caracteres en la cadena.

21. b) 3 Explicación: En una división entre enteros, Java trunca el resultado a un entero. 10 / 3 da como resultado 3.

22. b) class Explicación: En Java, se usa la palabra clave 'class' para definir una clase.

23. c) Pueden comenzar con un número Explicación: En Java, los nombres de variables no pueden comenzar con un número, aunque pueden contener números.

24. b) Concatena las cadenas Explicación: Cuando se usa con Strings, el operador '+' concatena las cadenas.

25. a) true Explicación: Java realiza una conversión implícita del int 5 a double 5.0 antes de la comparación, por lo que son iguales.

26. b) Integer.parseInt() Explicación: Integer.parseInt() es el método estático utilizado para convertir un String a un int en Java.

27. b) '==' compara referencias, '.equals()' compara contenido Explicación: Para Strings, '==' compara las referencias de los objetos, mientras que '.equals()' compara el contenido de las cadenas.

28. b) AND lógico Explicación: El operador '&' realiza un AND lógico cuando se usa con valores booleanos.

29. b) 5 Explicación: Con el operador postfijo '++', y toma el valor de x antes de que x se incremente.
30. b) void Explicación: En Java, 'void' se usa para indicar que un método no devuelve ningún valor.