

ACADEMIA JAVA

RESPUESTAS CUESTIONARIO XIDERAL: SEMANA 1

NOMBRE: JOSE MANUEL JIMENEZ HDZ

1 - R: Bajo mi lógica los operadores que pueden usarse son == para comparar el estado de la bandera y saber si apuntan hacia la misma referencia.

El operador ! también puede ser usado en temas comparativos donde ubicándolos por delante de la referencia bandera sirve para preguntar si el estado de la bandera es diferente al de otra o condiciona para que en caso de ser diferente del estado inicial de la bandera ocurra algo.

Por último, si es posible castear el estado de un boolean utilizando el = cambiando de forma manual o según alguna condición el estado de la bandera.

2 - R: La respuesta correcta es int, log y double, la explicación es sencilla, no se puede meter algo grande en algo pequeño, en este caso no se podría guardar en un short o un byte, ya que el resultado sobrepasa el rango permitido en estos tipos de datos.

3 - R: La respuesta correcta sería el inciso 6, ya que el resultado del Long si es suficiente para entrar en el int, pero por la inferencia de datos es necesario realizar el casteo del Long e indicarle que debe actuar como un int, para estar en un int.

4 - R: El método de la clase candyCounter está bien, pero considero que está mal implementado en el método main, ya que los parámetros del método no son correctos.

5 - R: el primer valor es 4, ya que no se cumple la condición de que $a > 2$, aunque aquí indica que se incrementa uno a b, sigue siendo 4, ya que en este caso lo sumara hasta que b se use de nuevo.

El segundo valor es 5, ya que no se cumple la condición de que a sea diferente de c, y como b ya tenía una suma de +1 pendiente, entonces da 5, aunque ahí indica que ese mismo b = 5 tendrá un +1 en su siguiente llamado.

El tercer valor sería 1, ya que no se cumplen las primeras dos condiciones de que $a > b$ y $B < c$, por lo tanto, salta al else.

6 - R: en este caso, ticketsTaken será igual a 4, ya que su valor inicial es 1, después se multiplica con el número que se le asigna que sería dos y ya traía un incremento pendiente de la operación anterior donde se vio involucrado, con lo cual será $2 * 2 = 4$.

En el caso de ticketsSold, su valor inicial es 3, luego se le suma 1 al mismo momento que se reasigna, sería igual a $3 + 1$ y se le suma el valor de la otra variable que vale 1 hasta aquí, lo cual no daría como resultado 5 y al final nos indica la suma de un long casteado que es igual a 1, con lo cual daría $5 + 1 = 6$.

7 - R: la respuesta sería "just Right", ya que no cumple la condición de que si bien cumple la primera condición del primer if, al no ser else if, no se detiene y se sigue comprando hasta llegar a la última condición que si cumpla.

8 - R: las respuestas serían break y break RABBIT, ya que en este caso el mensaje está fuera del segundo for, pero dentro del primero, así que para llegar al mensaje el segundo for se tendría que romper.

9 - R: Si funcionara daría 11, pero creo que no está bien implementada el ciclo do while.

10 - R: Por la presencia del break antes del mensaje, el código va a correr, pero en cuanto cumpla el if, se va a romper por la cláusula break.

11 - R: El código no va a compilar, ya que se quiere guardar un int en un string en la línea 5

12 - R: El código si compilaría y por los métodos implementados de la clase stringBuilder el resultado sería: abbacca

13 - R: En la línea 18 me parece que el StringBuilder no es capaz de ser comparado con un String bajo el ==, únicamente por el equals, para que esta comparación sea posible, necesitamos sacarle el toString al stringBuilder

14 - R: El resultado sería “roar roar!!!” ya que si bien al String se le agrega el !!!, Este luego es reasignado y queda apuntando a otro objeto, en el caso del stringBuilder, este agrega el !!! y como es mutable, solo le agrega contenido al objeto, pero nunca deja de apuntar al mismo objeto.

15 - R: Se puede usar el método reverse y también la línea `puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length() - 1);`