

Roberto Gonzalez

## Chapter 1

Mis respuestas son las que no estan en parentesis, las que estan en parentesis son las respuestas correctas

1) B,E (D,E)

La razon por la que pense que era B fue por la parte final, pero no vi que le faltaba el void

2) c, e, d (c,e,d)

Elegi estas ya que segui lo que decia el libro, primero package, si no tiene esta bien ya que entra en el package unnamed, luego imports y luego clases.

3) a, e, c (A,E)

Elegi la c porque pense que main era tambien considerada una clase, pero cómo tal no lo es para cuestiones generales

4) b, e, g (b,e,g)

Elegi estas igual por los consejos del libro, puede iniciar con guion bajo pero no puede ser solamente guiones bajos, evitar palabras reservadas (Public no cuenta por la mayuscula) y no puede iniciar con numeros.

5) a, d, f (a, d, f)

Elegi estas respuestas dado a lo que hemos repasado y lo que lei en el libro de garbage collection.

6) b (f)

Elegi esta ya que pense que solo eran las que estaban inmediatamente disponibles en el bloque, no conte las variables anteriores que todavia entran dentro del scope pero esto ya no volvera a pasar.

7) f (c,e)

Aqui pense que no compilaria, debido a las "" que estaban con " normales entremezcladas, pense que causaria un error cómo el caso de no usar simbolos de escape.

8) b, d, g (b,d,e,h)

Me falto tener en cuenta bien la pregunta de que preguntaba si compilaba y me base mas en el runtimeException que tendria cuando fuera ejecutado, tambien pense que no podia apuntar a null por nada, pero resulta que no y puede apuntar una vez ya declarada

9) c, e (e)

Pense que el default de double es 0.0, pero es 0.0d, lo cual tiene mas sentido en retrospectiva

10) a, c, e, f (a,e,f)

Se me paso el elegir una que terminaba en guion bajo

11) e (e)

Igual elegi estas por vil eliminación, lang y system pueden ser ignoradas por cuestion de que son importadas automaticamente. entre aquarium y water se podria quitar el de water sin problemas dejando el de wildcard

12) a (a,c,d)

Elegi A por la razon de que no se pueden declarar dos variables al mismo tiempo separadas por coma, en cuanto a la c y la d si fueron descuidos mios al no fijarme de la declaracion que tenia la linea 4 dentro de un método y tampoco al no darme cuenta que fins no esta dentro del scope del metodo.

13) b (a,b,c)

Se me olvido por completo que se podia obviar jellies con todo y métodos si se importaba todo aquarium, tambien se me paso el caso de jellies.water

14) a, b (a,b,d,e)

Por alguna razon pense que si se podria llamar métodos de variables primitivas, o bien que solo estarian contando los primeros errores. Ademas de que no habia visto la falta de parentesis en la linea 7.

15) c, e, f (c,e,f)

Me base en lo que hemos visto del garbage collection y lo que decia el libro para mis respuestas

16) c (a,d)

Me confundi con el inicio del triple "", pero ya entendi cómo pasa, el primer salto de linea en blanco no cuenta para la impresión, ademas de que una diagonal invertida simple cancela ese salto de linea

17) d, f, g (d,f,g)

Me base en los valores default que hemos visto a lo largo de las sesiones y lo que mencionaba el mismo libro

18) b, c, e, f, g (b,c,f)

Elegi e porque pense que el valor no podria cambiar, pero me he dado cuenta que si puede cambiar al momento de ejecución, lo que no puede cambiar es el tipo de valor que es. La g pense que seria correcta porque me sonaba correcta, pero ya me di cuenta que var no es palabra reservada

19) a (a,d)

No tome en cuenta que parseLong devolvía el valor primitivo.

20) d (c)

Completo descuido mio, por alguna razon pense que decia q=p en vez de p=q y andaba tratandolo cómo tal

21) b (d)

No conte lo que sucedio al instanciar el objeto salmon, que tanto imprime un 0 y declara en 4 count para luego imprimir un 2

22) a, c, f, g (c,f,g)

No me di cuenta que la a tenia el amount en mayuscula

23) a (a,d)

No me di cuenta que depth estaba fuera de scope

chapter 2

1) a, d, f (adg)

pense que g seria la otra opcion porque no pense que el casteo se considerara operacion ni que el operador ~ funcionaria solo con enteros, pense que seria cómo otra version del operador ! para booleanos

2) a, b, d (a,b,d)

Con lo que lei del libro, ya le ando agarrando un poco mas el hilo a conversiones implicitas entre valores

3) b, c, d, f (b,c,d,f)

Mismo que la pregunta de arriba, ya anda cobrando mas sentido el casteo intrínseco que tiene java

4) e (b)

pense que no se podria asignar un valor a una variable mientras se usa cómo operando y que causaria error al no ser algo que devolviera un valor booleano, se me olvido que se puede y que de hecho devuelve el mismo valor asignado

5) a (a,c)

No considere el c, que tiene sentido por ir de menor a mayor prioridad

6) f (f)

no compila debido al valor que regresa, se estaría regresando un float el cual no es posible volver long, ya que no es error en línea 9 fue la única respuesta que tenía sentido

7) d (d)

por el && de la declaración de safe, no se toca ese ph++, que si se llegara a él volvería el último valor un true

8) c (a)

se me sigue dificultando un poco el operador post al ser usado en declaraciones, pero ya me anda quedando un poco más claro que solo modifica el valor sin guardarlo si se encuentra en alguna operación

9) c (ade)

mismo problema que arriba, me cuesta un poco ver el operador post

10) g (g)

el código no compila por el casteo de weight y height a byte, estos hacen que se vuelva int al momento de ser operados, y cómo int no cabe dentro de short lanza error, se podría castear la operación a un int o bien cambiar el valor de zebra a int

11) d (d)

aquí era simplemente un orden de operaciones

12) d (d)

creo no hay mucho que pueda explicar sobre mi razonamiento en esta pregunta

13) f (f)

es una pregunta más sobre orden de operadores, pero cómo todos tienen la misma prioridad se sigue de izq a derecha, ya que goingtothestore sería la única variable que podría afectar el resultado

14) e, g (beg)

se me pasó que los operadores de igualdad y desigualdad si pueden ser usados en objetos, el que comparen referencias a memoria es otra cosa

15) d (d)

el único operador con tres operadores es el ternario, de ahí su nombre

16) b (b)

es el mismo problema de casteo a valores con mayor tamaño e intentando meterlas en variables de menor tamaño, en este caso todo se vuelve long por el casteo del 3 para ser operado e intenta guardarlo en int

17) g (cf)

pense que por la línea 3, el código no compilaría, pero en retrospectiva tiene sentido que compile ya que el += es técnicamente un valor=valor+ haciendo que los valores finales sean tickets sold 6 y tickets taken 4

18) c (bien)

solo los parentesis son validos para añadir prioridad, lo demás tiene diferentes usos dentro de java

19) b, f (bf)

al añadirle un uno al valor máximo que puede admitir byte, se crea un overflow corriendo al mínimo valor, de end es solo otro ejercicio con pre que ya ando agarrándole la onda

20) d, e (ade)

por alguna razón no contemple lo de los unarios, pero tiene bastante sentido ya que estos alteran el valor de la variable y se ocupa saber este valor antes de hacer algo, cómo en el caso de --i, su valor no es el que ya tiene asignado sino uno menos por el pre, o bien un operador !, su valor no es el que tiene ya que este está negado

21) c (e)

no vi el detalle de que era un operador bitwise, pense que solamente era un negativo, son muy parecidos

chapter 3

1) a, b, d, e, f, h, g (a,b,c,e,f,g)

pense se podia admitir punto-flotante, pero no es asi, ademas de que pense que no se podia admitir wrappers de primitivas pero si se puede

2) b (b)

en este caso, humedad no es menor que 6, haciendo que entre al primer else asignado al segundo if

3) c,d,e,f,h (adfh)

se me paso elegir también a, pensé map y string se podrían, map por ser parte de collections pero no tiene un iterable, y string por ser una especie de array pero en retrospectiva no tiene sentido alguno esto último.

4) f (f)

al no tener un default y querer asignar valor a la variable, corre en un error de compilacion. ya que hay un caso donde la variable podria no tener algo asignado y eso no es valido

5) f (e)

no compila por el unreachable line of code por el print despues del continue, pense que habria otro error por iterar con un object c pero en retrospectiva me di cuenta que no es asi

6) a, c, d, e (cde) (map no es iterable)

mismo problema que el 3, pense que todo collections era iterable, pero solo algunos tienen el iterable

7) b, d (bd)

b y d son las dos diferentes formas de iterar, uno desde el length completo y otro llegando al valor de length -1, solo que uno invierte el orden de impresion

8) f (g)

al momento de decir que si es un instanceof Long, queda fuera de scope, esto no lo habia tomado en cuenta

9) b, c, e (bce)

ya que se quiere que se imprima 2, solo hay 3 formas de llegar a ese, break rabbit y break son identicos en funcionalidad, ya que el ciclo que se maneja seria el de rabbit, continue bunny es otra forma de hacer un break a rabbit

10) c (e)

contemple el caso del continue y el caso del dayofweek.monday, pero no habia contemplado tambien el case con sunday no siendo final ni el caso de thursday no siendo una variable conocida al momento de compilar, pense que solo siendo final seria suficiente

11) a (a)

cómo se tienen contemplados todos los casos donde la misma variable pueda tener un valor, y hasta donde no pueda tener un valor, compila y el resultado es el valor que marca mammal

12) f (c)

mismo problema que tuve en la unidad dos, pense que al ser += no se podia meter otra operacion pero no es el caso

13) d (g)

aqui mi error fue suponer que correria por no tener parentesis la linea 8

14) b, d, e (bdf)

mi error fue pensar que la lista tendria prioridad en su tipo en vez del valor dentro de ella

15) b (f)

no me di cuenta que en la linea del case b estaba mal declarado c, confundi los dos puntos con una coma al analizar rapidamente

16) a, b, c, d (abd)

no analice bien el c, no me di cuenta que da un `indexOutOfBounds` en la primera corrida

17) b, e (b,e)

en este caso, cómo es pura condicion de salida de los bucles, el unico que no hace nada a la variable es el `for`

18) c, d (c,e)

pense que tendria mas sentido la d por encima de la e, pero no es el caso ya que la e tiene mas sentido, la d podria referirse a dentro del `if` por lo ambiguo de la misma

19) e (e)

`snake` no anda dentro del scope del `while`, por eso el programa no compila

20) a (ae)

solo contemple la a, no contemple si habria mas opciones que podrian correr el programa

21) c (e)

no contemple el `long` ni el case repetido, no registre que habia un case repetido y se me paso por completo el hecho de que era un `long`

22) e (e)

al ser 3 la variable que entra al `switch`, se imprime un 5 y un espacio, luego se reduce en uno para luego imprimir el valor de `tailfeathers`, dando un 2 y luego un 1 ya que es un ciclo `while`

23) d (f)

no me di cuenta del doble `else` asociado a un solo `if`

24) a,b,e (g)

no vi el `in`, mi mente lo sustituyo por dos puntos, es la tercera vez que me pasa (primera contestando ciegamente, segunda checando resultados y tercera analizando estos resultados para este documento)

25) d (d)

el código compila ya que se respetan todas las variables dentro siendo `final`, de ahí entra a `default instrument` por no coincidir con `VIOLIN`, esto porque `string` es case sensitive

26) f (f)

se hace un ciclo infinito en el `if`, ya que siempre terminara siendo mayor `name.length` que cero ya que se le anda agregando a `name`

27) f (f)

no compila ya que el case 2 no contempla que pasa si el `if` no se cumple

28) b (f)

no tome en cuenta que `guppy` se usó dos veces para declarar el mismo `instanceof`, haciendo un error en el scope

29) d (c)

se me paso contemplar el caso donde `y=5`, ya que es aqui donde entraria de nuevo al `do-while` para imprimir el 6