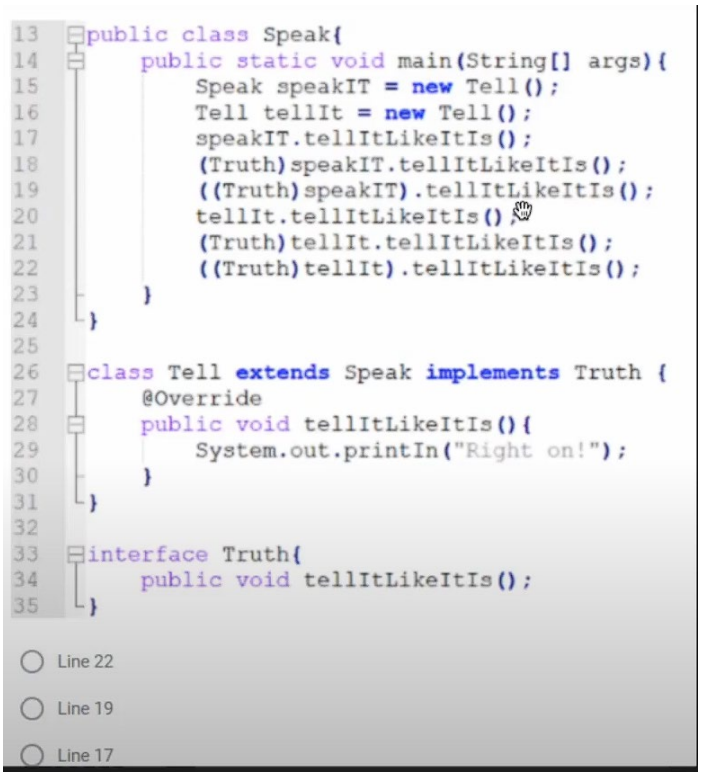
PREGUNTAS VIDEO 20230930_01

1. ¿QUÉ LÍNEAS IMPRIMIRÁN “RIGHT ON!”?



```
13  public class Speak{
14      public static void main(String[] args){
15          Speak speakIT = new Tell();
16          Tell tellIt = new Tell();
17          speakIT.tellItLikeItIs();
18          (Truth)speakIT.tellItLikeItIs();
19          ((Truth)speakIT).tellItLikeItIs();
20          tellIt.tellItLikeItIs()
21          (Truth)tellIt.tellItLikeItIs();
22          ((Truth)tellIt).tellItLikeItIs();
23      }
24  }
25
26  class Tell extends Speak implements Truth {
27      @Override
28      public void tellItLikeItIs(){
29          System.out.println("Right on!");
30      }
31  }
32
33  interface Truth{
34      public void tellItLikeItIs();
35  }
```

- ○ Line 22
- ○ Line 19
- ○ Line 17

EXPLICACIÓN:

Observamos que “Tell” es una clase que extiende “Speak” e implementa la interfaz “Truth”. Truth es una interfaz que declara el método “tellItLikeItIs”.

La línea: **speakIT.tellItLikeItIs();** causará un error de compilación porque el método tellItLikeItIs no está definido en la clase Speak. Por lo tanto, no imprimirá nada.

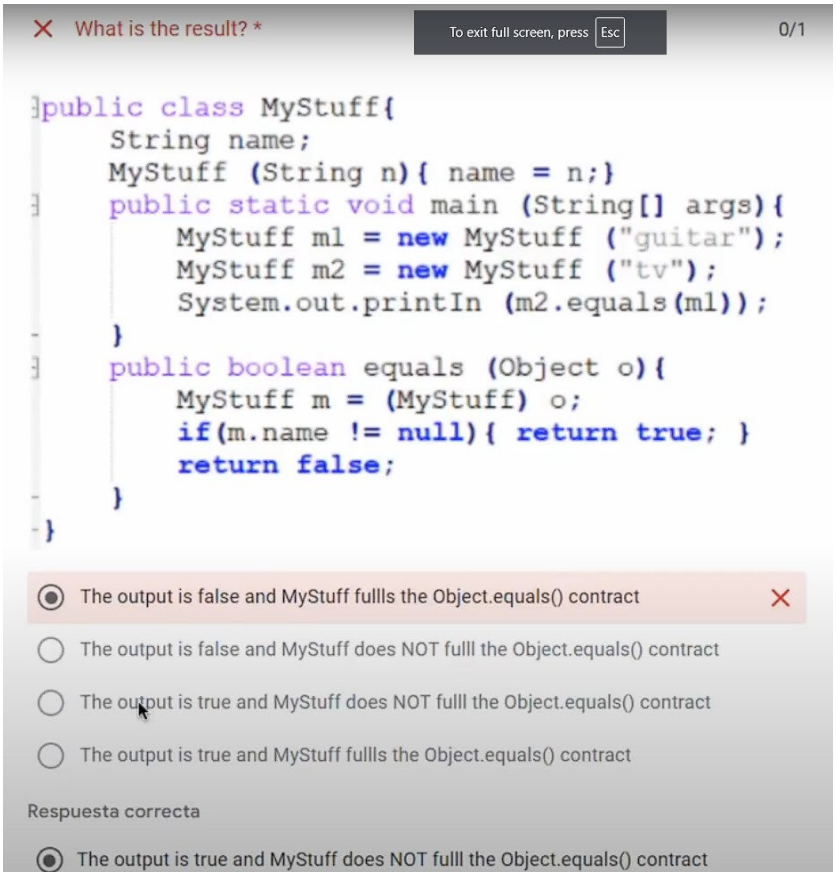
La línea: **(Truth)speakIT.tellItLikeItIs();** causará un error de compilación porque el cast debería aplicarse antes al speakIT porque de esta manera es como querer hacer el cast a lo que devuelva **speakIT.tellItLikeItIs();**

La línea: **((Truth)speakIT).tellItLikeItIs();** imprime “Right on!” satisfactoriamente porque speakIT se convierte a Truth (que Tell implementa) y luego se llama al método tellItLikeItIs. Dado que la clase Tell sobrescribe el método tellItLikeItIs, esta línea imprimirá "Right on!".

La línea: tellIt.tellItLikeItIs(); imprime “Right on!” satisfactoriamente porque tellIt es de tipo Tell, que implementa directamente tellItLikeItIs.

La línea: ((Truth)tellIt).tellItLikeItIs(); imprime “Right on!” satisfactoriamente porque tellIt se convierte a Truth (que Tell implementa) y se llama al método tellItLikeItIs.

2. ¿CUÁL ES EL RESULTADO?



EXPLICACIÓN

La respuesta correcta es:

“The ouput is true and MyStuff does NOT full the Object.equals() contract”.

Este método sobrescribe el método equals de la clase Object. Intenta convertir el objeto o a un objeto de tipo MyStuff. En el método equals, se está comprobando si el nombre (name) del objeto pasado como argumento no es null. Dado que m1 tiene el nombre "guitar", que no es null, la condición se cumple y el método devuelve true. Por lo tanto, System.out.println(m2.equals(m1)); imprimirá true.

Ahora no cumple con el contrato porque la implementación actual solo verifica si el campo name no es null, lo cual no garantiza la simetría. El contrato del método equals en la clase Object establece varias propiedades que deben cumplirse:

1. Reflexividad: Para cualquier referencia no nula x, x.equals(x) debe devolver true.
2. Simetría: Para cualquier referencia no nula x y y, x.equals(y) debe devolver true si y solo si y.equals(x) devuelve true.
3. Transitividad: Para cualquier referencia no nula x, y y z, si x.equals(y) devuelve true y y.equals(z) devuelve true, entonces x.equals(z) debe devolver true.
4. Consistencia: Para cualquier referencia no nula x y y, múltiples invocaciones a x.equals(y) deben devolver consistentemente true o false.
5. No nulidad: Para cualquier referencia no nula x, x.equals(null) debe devolver false.