1. F -respjavi
2. V -respjavi
3. F -respjavi
4. V -respjavi

No se comparte el mismo objeto LineaTicket entre Tickets, cada ticket tiene su propio array con objetos de tipo LineaTicket, objetos diferentes e independientes entre cada uno de los tickets que haya. Si es verdad, que siendo objetos distintos los datos almacenados PUEDEN coincidir. Una composición no admite partes compartidas

1. V

Es compatible, FileNotFoundException es una IOException.

1. F

Exception no es una IOException.

1. V

Si son derivadas de las que aparecen en la base.

1. F

Siempre privado, si lo ponemos público podríamos crear cualquier número de instancias.

1. F

No podría crear la primera instancia porque necesitaría una instancia para crearla.

1. F

Es un singleton solo puede haber una instancia.

1. V
2. V
3. F
4. F

Como mínimo se tiene que crear con un empleado.

1. V

El \* implica 0 o muchos.

1. F
2. F

Es una relación entre objetos (Composición)

1. F

La única relación que puede ser bidireccional es la asociación.

1. V
2. Falso (no se pueden crear objetos de tipo Articulo)
3. V
4. Falso. (porque no se puede Agregar A con B y a la vez B con A, una podría ser una agregación y otra podría ser una asociación)
5. Falso
6. Falso. Cuando refinamos un método en la sobrescritura incluimos una llamada al método de la base, en este caso no se puede porque es abstracto).
7. Verdadero (no tienen otra opción)
8. Falso, porque es abstracto y tiene que poder ser sobrescrito.
9. Falso, es un enlace dinámico.
10. Verdadero, el método esta definido en la clase base por lo tanto existe en todas las derivadas nazis.
11. Falso, no esta en la base, hay que bajar hacia producto para buscarlo.
12. Verdadero, esta definido en la base y por enlace dinámico invocará al que este sobrescrito en su derivada.
13. Verdadero.
14. Verdadero.
15. F – Los métodos virtuales PUROS
16. F
17. V -respjavi
18. F -respjavi
19. ni puta idea -respjavi