Taller #7

1.

a) Se debe definir como clase abstracta porque posee métodos sin código, que sin embargo todas sus subclases deben tener, la diferencia con una clase normal, es que no se pueden crear objetos de únicamente clase Instrumento, sino que, para crear un objeto de esta clase, debe crearse como objeto de alguna de sus subclases, y los métodos definidos como abstractos en esta clase deben ser reescritos en sus subclases.

b)
```
public class Piano extends Instrumento {
    public Piano(String tipo) { super(tipo); }
    public void Tocar() {System.out.println("Tocando piano"); }
    public void Afinar() {System.out.println("Afinando piano"); }
}
```

c) Todas se ejecutan con normalidad.

d) Tocando saxofon
Tocando guitarra

2.

a) Se generaría un error pues, al ser un método abstracto, este no debe tener código en la clase abstracta, si no en la sobreescritura de las subclases.

b) No sería un error, pues al no ser definidos como abstractos, no es necesario sobreescribirlos en las subclases, y simplemente pueden ser heredados.

c) No habría error, pues no es necesario que una clase abstracta tenga métodos abstractos, sin embargo, en la clase objtaller7, se intenta acceder al método descripcion con un apuntador de clase ObjetoAstronomicoExtraSolar, y con el código presentado, este método no está definido para esta clase, y ahí se presentaría un error.

d) Porque los objetos a los que apuntan las variables de tipo ObjetoAstronomicoExtraSolar, también pertenecen a dicha clase, pero como el método está definido como abstracto no puede ser convocado por ligadura dinámica, y en su defecto, llama al método redefinido en las respectivas subclases.

e) Porque en la línea anterior pasó a apuntar al mismo objeto al que apunta la posición 2 del arreglo oa, el cuál es de clase SuperNova, y como me método de la clase del apuntador está definido como abstracto, el método al que se llama es el de la clase del objeto apuntado.

f) Porque la clase Estrella también es abstracta, por lo que no es necesario redefinir los métodos abstractos de la clase superior.

g) Debido a que la ligadura dinámica no puede llamar a un método abstracto, esta mediante referencia desde la clase abstracta llama al método del objeto apuntado, cosa que no se podría hacer si el método se redefine como privado, pues no sería accesible desde la clase superior.

h) Porque su clase superior estrella ya está definiendo este método, por lo que este es heredado, pero si podría definirse, pues sería una sobreescritura, lo que significa que el método desde la clase Estrella ya no sería abstracto.

i) Imprimiría la dirección del espacio de memoria del objeto apuntado, pues, al n,o estar definido el método toString en estas clases, el método será heredado de la clase object, que devuelve la dirección si se imprime un objeto.
j) Porque sí se pueden crear variables apuntadoras de clases abstractas, lo que no se uede hacer es crear objetos de estas clases.
k) Hay un error en la línea B, debido a que se está intentando crear un objeto de clase ObjetoAstronomicoExtraSolar, mientras que la instrucción en la línea C si es factible.
l) La línea B es correcta porque Nova es una subclase de ObjetoAstronomicoExtraSolar, por lo que es posible que un objeto de dicha clase sea apuntado por una variable de clase superior, la línea C está mal porque debido a la ligadura dinámica, se busca llamar al método explosión de la clase de la variable, sin embargo, en esta clase no está definido este método, ni de forma convencional, ni abstracta, por lo que no se puede realizar el llamado al método; e ignorando la línea C, se imprime:
   Boom!
m) Imprime true, debido a que cualquier objeto no nulo es de clase Object, por lo que cualquier objeto que sea no nulo imprimirá esto, por otro lado, lo que imprimiría las líneas mostradas sería:
   false
   true
   El objeto al que se está apuntando en ambos casos es nulo, pero en la segunda instrucción se le concatena un String vacío, y los Strings son de clase Object, por esto, en el primer caso si imprime que es falso, y en el segundo verdadero.
n) No se genera error, porque a pesar de que no se pueden crear objetos de una clase abstracta, sí se pueden crear objetos de sus subclases, y este constructor puede ser usado para ser llamado por un eventual constructor de estas subclases para asignarle un valor a su atributo ID, el cual es privado, y no accesible para sus subclases.
o) Se generarían errores, porque no se está definiendo loas métodos descripcion y explotar que se solicitan en las dos clases superiores abstractas