

Respuestas Ejercicio 2 Taller 7

1.

a) Se debe establecer como abstracta porque tiene métodos abstractos, por lo tanto, no se están definiendo, y generarían error en caso de que la clase no fuera abstracta y se diferencia de una clase normal por su public abstract al definir la clase, y sus métodos abstract.

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) Ninguna línea tiene error, ya que se puede tener apuntadores de una clase abstracta.

d) Por ligadura dinámica imprime “Tocando saxofón” y “Tocando guitarra”, respectivamente

2.

a) No se puede definir de esa manera porque un método abstracto sólo puede definirse en las subclases de la clase abstracta.

b) Significa que esos métodos se heredarán como métodos “normales” a las clases hijas, y no es motivo de error ya que una clase abstracta puede tener atributos y métodos que no sean abstract.

c) Sí, es posible, sólo que las clases hijas no tendrían obligación de definir métodos obligatoriamente, ya que la clase abstracta no tiene métodos abstractos por definir.

d) Por polimorfismo los objetos creados de las subclases, pueden ejecutar sus métodos sobrescritos del método descripcion() que inicialmente se encontraba en la clase abstracta ObjetoAstronomicoExtraSolar.

e) Porque en la línea 28 se estableció que ahora en oa[0] va a estar un objeto de clase SuperNova y por lo tanto imprimirá “Soy una Super Nova”

f) Porque estrella también es una clase abstracta, entonces no está obligada a definir los métodos de la clase padre.

- g)** Como la clase padre tiene definido ese método por defecto (package), la clase Galaxia, que es hija, no puede establecer el método de manera más privada que package.
- h)** No define el método porque su clase padre ya lo está definiendo como un método normal, no abstracto, si se define, se estaría sobrescribiendo el método.
- i)** Está heredando el método de la clase Object, por lo tanto, imprime el nombre de la clase su código asociado.
- j)** Porque si se pueden crear apuntadores de tipo de clases abstractas, lo que no se pueden crear son instancias(objetos) de clases abstractas.
- k)** La línea B no es válida porque claramente se está creando una instancia de tipo ObjetoAstronomicoExtraSolar, que es una clase abstracta, y eso genera error y la línea C si es correcta.
- l)** La línea B es correcta porque el apuntador está apuntando a un objeto de menor rango, y la línea C es incorrecta porque al ser el apuntador oa de tipo ObjetoAstronomicoExtraSolar se asegura que esa clase tenga el método, como no lo tiene, lanza error. Omitiendo la línea C se imprime Boom!, ya que ahora le estamos diciendo al apuntador que juegue a ser de tipo Nova y para esa clase si está definido el método explotar().
- m)** Imprime true porque hereda de Object, todos los objetos imprimirán true, las líneas de el PDF imprimen “false” y un error porque la última línea está comparando en el instanceof una cadena de texto, (en este caso “null”) con Object.
- n)** No genera error y es útil para que las clases hijas por medio de super() puedan acceder a un constructor que reciba sus parámetros.
- o)** Faltaría definir el método descripción() heredado desde la clase abuelo y que no fue definido en la clase abstracta padre y el método explotar() un método abstracto de la clase padre.