

## Taller 7-ejercicio 2 Andrés Jacobo Leal Aguirre

1.

a) Explique por qué la clase Instrumento debe definirse como abstracta y qué la diferencia de una clase normal, en el ejemplo siguiente:

Una clase abstracta como Instrumento tiene como objetivo principal establecer una estructura común para las clases derivadas como Saxofón, por esto debe definirse como abstracta. A diferencia de una clase normal, esta puede contener métodos abstractos, que son declarados pero no implementados, obligando a las subclases a proporcionar su propia implementación.

b) Elabore una clase denominada Piano heredada de Instrumento, añada un constructor y redefina para ella los métodos Tocar y Afinar:

```
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) ¿Señale cuál de las siguientes líneas no se pueden ejecutar y por qué?

Todas las líneas se pueden ejecutar

**d) ¿Qué imprime el siguiente programa?**

El programa imprime lo siguiente:

Tocando saxofón

Tocando guitarra

Esto por ligadura dinamica

**2.**

**a. ¿Qué sucede si se define el método explotar() de la clase Estrella como se indica a continuación? Explique su respuesta.**

Los métodos abstractos no deben tener cuerpo

**b. ¿Qué significa que los métodos tipoCuerpo2() y getID() de la clase ObjetoAstronomicoExtraSolar no se definan como abstract? ¿Podría considerarse esta situación un error? Explique.**

Significa que estos métodos tienen una implementación por defecto en la clase base. No se considera un error, ya que no es obligatorio que una clase abstracta contenga exclusivamente métodos abstractos. Sin embargo, si el comportamiento de estos métodos debe variar según las subclases, podría ser más adecuado declararlos como abstractos.

**c. Si se define como abstracta la clase ObjetoAstronomicoExtraSolar, como se indica a continuación, ¿puede considerarse un error definir una clase abstracta sin métodos abstractos? Explique.**

No es un error. Una clase abstracta puede utilizarse simplemente para evitar la instanciación directa y proporcionar una estructura común para las subclases.

**d. Explique por qué el arreglo oa (línea 19) hace referencia a una clase abstracta y sin embargo, en la línea 25 se invoca el método correspondiente a cada clase derivada.**

Esto es posible gracias al polimorfismo. Aunque el arreglo está declarado con el tipo de la clase abstracta, las instancias en él son de clases derivadas, por lo que los métodos llamados corresponden a las implementaciones específicas de esas subclases.

**e. ¿Por qué la línea 29 imprime “Soy una Super Nova” sabiendo que el arreglo oa en esa posición fue inicializado con un objeto de tipo Galaxia?**

Porque en la línea 28 se pone al espacio 0 del arreglo a apuntar al mismo objeto que al espacio 2 que es una supernova

**f. ¿Por qué en la clase Estrella no se define el método descripcion() si la superclase lo está solicitando, ya que este método descripción en abstracto?**

Porque la clase Estrella es una clase abstracta, entonces no necesariamente debe implementar los métodos abstractos de la clase abstracta de la que hereda

**g. ¿Qué sucede si el método tipoCuerpo1() de la clase Galaxia se define como privado? ¿Por qué se genera error?**

Porque los métodos que se sobrescriben en clases hijas no pueden reducir su nivel de accesibilidad

**h. ¿Por qué la clase Nova no define el método tipoCuerpo1()? ¿Se podría definir? Si lo define, ¿qué interpreta de esta situación?**

Si Nova hereda de una clase que ya define el método tipoCuerpo1(), no es obligatorio redefinirlo, pero podría hacerlo para proporcionar un comportamiento específico.

**i. ¿Qué imprime la línea 9? ¿Por qué se puede llamar al método toString() si la clase Galaxia no lo define y su papá ObjetoAstronomicoExtraSolar tampoco?**

imprime Galaxia@3af49f1c

El método toString() es heredado de la clase Object, por lo que no es necesario definirla para usarla

**j. ¿Por qué en la línea 11 se puede crear un puntero obN de tipo ObjetoAstronomicoExtraSolar si esta es una clase abstracta?**

Porque los punteros o referencias pueden declararse del tipo de una clase abstracta. Lo que no se puede hacer es instanciar objetos directamente de esta clase.

**k. ¿Las siguientes instrucciones (instrucciones en las líneas B y C) son válidas? Explique.**

La B es invalida porque intenta instanciar un objeto de una clase abstracta, mientras que la C es válida porque, aunque asigne un objeto de un tipo a un apuntador de otro, si la clase del apuntador hereda o está en algún nivel superior a la del objeto no hay problema, como es el caso.

**l. Explique por qué (ver código a continuación) la siguiente instrucción en la línea B es correcta y la instrucción en la línea C es incorrecta. Omitiendo la instrucción en la línea C, ¿qué se imprime por pantalla? Explique su respuesta.**

La B es correcta porque, aunque asigne un objeto de un tipo a un apuntador de otro, si la clase del apuntador hereda o está en algún nivel superior a la del objeto no hay problema, como es el caso. Y la C es incorrecta porque el método explotar no se encuentra definido en la clase del apuntador. Sin tomar en cuenta la instrucción C, el programa imprime Boom!

**m. ¿Por qué la línea 15 imprime true? ¿Para cualquier objeto que se cree siempre imprimirá lo mismo? ¿Qué imprimen las siguientes líneas? ¿Por qué?**

Imprime true porque lo que se le pasa es un objeto y sabemos que todos heredan de la clase Object. Las siguientes líneas imprimen false porque la clase Nova no es instancia ni de la clase SuperNova ni de la clase Galaxia.

**n. Agregue el siguiente constructor. ¿Se genera error? ¿Se pueden colocar constructores a clases abstractas? ¿Qué sentido tiene?**

No genera error. Las clases abstractas pueden tener constructores, los cuales se utilizan para inicializar variables comunes cuando son invocados por las subclases.

**o. Suponga que agrega la clase EnanaBlanca como se indica a continuación. ¿Qué se puede concluir de esta nueva situación? ¿Qué errores se presentan y cómo podría corregirlos?**

La clase EnanaBlanca no define el método abstracto explotar() de la clase padre que es estrella, lo cual generaría error porque las clases hijas que no son abstractas deben definir todos los métodos abstractos de la clase padre.