Ejercicio 2 java taller 7

Juan Pablo Mira Cañas

A)

En la clase instrumento se están definiendo 2 métodos abstractos para que luego sean implementados por sus clases hojas; lo que la diferencia de una clase normal es que de esta no pueden existir instancias, es decir, no puede crear objetos

B)

```
class Pianno extends Instrumento{
    Piano(String tipo){
        this.tipo=tipo;
    }

    public void Tocar(){
        System.out.println("Tocando Piano");
    }
    public void Afinar(){
        System.out.println("Afinando Piano");
    }
}
```

C)

En la clase Test se esta declarando la referencia x de tipo Instrumento de manera correcta y además se está creando también los objetos Saxofón y Guitarra

D)

En la primera línea se crea la referencia de tipo Instrumento que apunta a un objeto de tipo Saxofón, para luego llamar al método Tocar(), se resolvería por ligadura dinámica en tiempo de ejecución, imprimiendo “ Tocando Saxofón”, luego línea se crea la referencia de tipo Instrumento que apunta a un objeto de tipo Guitarra ,para luego llamar al método Tocar(), se resolvería por ligadura dinámica en tiempo de ejecución, imprimiendo “ Tocando Guitarra]”

2.

A)

Los métodos abstractos solo deben de definirse mas no implementarse, ya que este trabajo va en cada sub clase, por ende, este no puede tener un cuerpo.

B)

Estos 2 métodos al no ser abstractos, no serán una obligación para sus sub clases de implementarlos, lo único que pasara es que estos lo heredaran normalmente.

C)

No habría error de compilación, pero al no tener métodos abstractos no tendría sentido definirla como una clase abstracta

D)

Lo que ocurre aquí es simple polimorfismo, Aunque la referencia es de la clase base abstracta, una vez se hace el llamado al método, este reaccionara diferente dependiendo del tipo del objeto

E)

Lo que ocurre es que en el arreglo se le reasigna el objeto al que apunta esa posición

F)

Es cierto que no se implementa el método en la clase estrella, y es por eso mismo que esta clase debe ser también abstracta

G)

En la herencia una sobreescritura de un método no puede reducir la visibilidad del método

H)

Esta aunque también hereda de la clase objeto astronómico, la clase estrella implementa este método abstracto, y luego la clase nova lo hereda de este

I)

Se puede llamar al método toString(), ya que todas las clases heredan de Object, donde si se implementa este método

J)

Lo que no se puede hacer con una clase abstracta es instanciarla, pero si se pueden usar para referenciar objetos

K)

La instrucción B no es válida ya que se estas intentando instanciar un objeto de la clase abstracta, y en la instrucción C si es valido ya que solo se le esta asignando a la referencia oa a el objeto a el que apunta nova

L)

En la instrucción B se le esta asignando correctamente a la referencia oa el objeto a el que apunta nova, Mientras que en la instrucción C, se esta tratando de usar polimorfismo, pero como el método Explotar no esta en las clases padres entonces este no compilara

M)

- Como todas las clases heredan de la clase object, por ende instanceof dará siempre true

- Imprimira en este caso false ya que obN tiene a null el cual no es una instancia de la clase object

N)

Se puede agregar este constructor a la clase abstracta y además agregarle este método, como sus subclases no lo sobre escriben por ende siempre se ejecutará el método de esta clase;

O)

No implementa los métodos que hereda de estrella y por ende debe ser clase abstracta, lo que no es, por ende, aquí habría un error