

Programación Orientada a Objetos. Taller 7 Java

David Fernando Cerón Quintero

1.

a) Por 2 motivos esenciales. El más pragmático consiste en el hecho de la obligación de tener que definirse los métodos `Tocar()` y `Afinar()`. Una perspectiva más idealista destacaría el hecho de la existencia de métodos abstractos como entidades que todo objeto metafísico concebido por "Instrumento" debería poseer.

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 pueden ejecutar.

d) Se imprimiría "Tocando Saxofon" y "Tocando Guitarra" en ese orden, porque son los métodos que posee el objeto al que apunta x en cada momento que se invoca el método `tocar()`.

2.

a) Se genera un error, ya que los métodos abstractos no deben contener ningún tipo de código en su interior.

b) Que serán heredados como si se tratase de un método de una clase no abstracta. No es un error, ya que podría interpretarse como que se busca que estos métodos se hereden con la estructura establecida, sin buscar redefinirse en las clases hijas.

c) Pragmáticamente, el hecho de construir una clase abstracta sin métodos abstractos iría en contra de su finalidad, la cual es crear una estructura preestablecida sin la necesidad de dar explícitamente la sintaxis de la misma; esta postura consideraría el hecho de crearla como un error, porque no definirla como clase abstracta haría que sigan cumpliendo sus funcionalidades, y estaría más en concordancia con las buenas prácticas.

d) Esto es debido a lo siguiente, cada objeto en la lista no es estrictamente un objeto de la clase ObjetoAstronomico, sino que se especifica (El primero es un objeto de la clase Galaxia, el segundo Nova y el tercero SuperNova), los cuales sí tienen un método (No abstracto) descripcion definido.

e) Porque el objeto en la posición es uno de la clase SuperNova.

f) Esto es consecuencia de que la clase Estrella es igualmente abstracta, por lo cual hereda este método abstracto.

g) Se generaría un error, porque no se puede cambiar el nivel de acceso de un método heredado por uno más restringido.

h) No es necesario, ya que la clase Estrella ya lo define y la clase Nova lo hereda. Sí, se puede redefinir el método como si se tratase de una herencia habitual. Una interpretación es que la herencia de métodos no abstractos ocurre de igual manera tanto en clases no abstractas como en clases abstractas.

i) Galaxia@3af49f1c. Porque toda clase es hija de la clase Object, la cual tiene definida el método toString, por ende, es heredada por toda clase y no es necesario definirlo para que pueda ser invocado.

j) Porque Java asumirá que este objeto se instanciará como alguna de las clases hijas no abstractas de ObjetoAstronomicoExtraSolar. Lo cual puede respaldarse con el hecho de que el objeto nova es una instancia de la clase Nova.

k) La de la línea B no es válida, porque no se pueden instanciar clases abstractas. Por otra parte, la línea C es válida, porque se instanció una clase no abstracta y se está haciendo una especificación correcta.

l) Porque, a pesar de ser un objeto de clase Nova, este apuntador lo generaliza a uno de clase ObjetoAstronomicoExtraSolar, y esta clase no tiene definida dicho método.

m) Porque toda clase es hija de la clase Object. Imprimen "true" y "false" en ese orden, porque null no es una instancia de una clase, mientras que la sentencia ""+null retorna un objeto de clase String (El cual sí es un objeto).

n) No se genera ningún error, indicando que en esencia se pueden asociar constructores a clases abstractas. El sentido que se le puede dar es el siguiente, este constructor está definido no para la clase abstracta, sino para las clases hijas, ya que lo heredan de manera habitual.

o) Se concluye que la clase EnanaBlanca hereda todas las cualidades que posee la clase Estrella (Y por lo tanto, las que esta hereda de la clase ObjetoAstronomicoExtraSolar). Los errores se generan precisamente por este hecho, la clase Estrella es abstracta y requiere que se definan los métodos explotar() (Abstracto en Estrella), y descripcion() (Abstracto por herencia de ObjetoAstronomicoExtraSolar).