

Ejercicio 2 – Taller 7 – Java – Python

1.

a) La clase Instrumento debe definirse como abstracta, lo cual permite que las clases derivadas como Saxofon y Guitarra implementen los métodos Tocar y Afinar. A diferencia de una clase convencional, una clase abstracta no puede ser instanciada, pero actúa como un modelo para las subclases que la extienden.

b)

```
public class Piano extends Instrumento {

    public Piano(String tipo) {

        // Se utiliza el constructor de la clase base para inicializar el tipo de
instrumento.

        super(tipo);

    }

    public void Tocar() {

        // Implementación del método Tocar, que permite ejecutar acciones
propias de tocar un piano.

    }

    public void Afinar() {

        // Implementación del método Afinar, necesario para preparar el piano
para su correcto funcionamiento.

    }

}
```

c) El código se ejecuta sin inconvenientes, creando instancias de Saxofon y Guitarra. Se imprime:

- "Tocando Saxofon"
- "Tocando Guitarra"

2.

a) Un método abstracto no debe contener implementación dentro de su definición, ya que generaría un error al intentar compilar.

b) En este contexto no es un error, ya que los métodos abstractos están diseñados para que cada subclase proporcione su propia implementación, lo cual no aplica en este caso específico.

c) Aunque no presenta problemas de ejecución, no es apropiado declarar una clase como abstracta si no se está utilizando como base para heredar características en subclases.

d) El arreglo "oa" se relaciona con una clase abstracta, que actúa como base para almacenar objetos de sus subclases. Gracias a la ligadura dinámica, en la línea 25 se puede invocar el método abstracto "descripción".

e) La salida es "soy una super nova", debido a la reasignación de "oa[0]" a "oa[2]".

f) La clase Estrella, al ser abstracta, no obliga a que se implemente el método "descripción" en sí misma.

g) Si un método sobrescribe a uno de su clase padre, su nivel de visibilidad no puede ser más restrictivo que el del método original.

h) El método heredado de la clase Estrella no requiere definición adicional a menos que se desee sobrescribirlo con una nueva implementación.

i) Imprime la referencia de memoria de la instancia de "g", en este caso algo como "Galaxia@identificadorUnicodelObjeto". Esto es posible gracias a la herencia del método "toString" de la clase Object.

j) Es posible crear un puntero porque se realiza un casteo explícito hacia la clase Nova, que no es abstracta y, por ende, puede ser apuntada sin problemas.

k) La opción B no es válida porque ObjetoAstronomicoExtraSolar es una clase abstracta. La opción C es válida ya que Nova se instanció a partir de la clase Nova, que hereda de ObjetoAstronomicoExtraSolar.

l) La opción B es correcta porque Nova es una subclase de ObjetoAstronomicoExtrasolar, aplicándose el principio de generalización (una subclase puede tratarse como una superclase). La opción C es incorrecta porque, aunque se realiza un casteo explícito, la clase ObjetoAstronomicoExtrasolar no define dicho método.

m) La salida es "True", ya que todos los objetos heredan de la clase Object. Siempre se imprimirá "True" para cualquier objeto que no apunte a null. Por otro lado:

- Imprime "False" porque apunta a null.
- Imprime "False" porque instanceof evalúa cadenas y no objetos.

n) No genera error declarar constructores en una clase abstracta, ya que estos pueden inicializar atributos comunes para las subclases.

o) Si se agrega una nueva clase sin declarar los métodos abstractos heredados de Estrella, se generarán errores. Para corregirlo, simplemente se deben implementar dichos métodos en la nueva clase.