

**TALLER No. 7**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

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

**abstract class Estrella extends ObjetoAstronomicoExtraSolar { abstract void explotar() {**

**System.out.println("Estrella explotar");**

**}**

**int a = super.getID(); public void tipoCuerpo1() {**

**System.out.println("Simple " + a);**

**}**

**}**

1. ¿Qué significa que los métodos *tipoCuerpo2()* y *getID()* de la clase *ObjetoAstronomicoExtraSolar*, no se definan como *abstract*? ¿Podría considerarse esta situación un error? Explique.
2. 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.

**abstract class** ObjetoAstronomicoExtraSolar {

**private int** ID;

**public void** tipoCuerpo2() { System.***out***.println("Extrasolar");

}

**public int** getID() {

**return this**.ID;

}

}

1. 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.
2. ¿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*?
3. ¿Por qué en la clase *Estrella* no se define el método *descripcion()* si la superclase lo está solicitando, ya que en este método descripción en abstracto?
4. ¿Qué sucede si el método *tipoCuerpo1()* de la clase *Galaxia* se define como privado?

¿Por qué se genera error?

1. ¿Por qué la clase *Nova* no define el método *tipoCuerpo1()*? ¿Se podría definir? Si lo define, ¿qué interpreta de esta situación?
2. ¿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?

1. ¿Por qué en la línea 11 se puede crear un puntero obN de tipo

*ObjetoAstronomicoExtraSolar* si esta es una clase abstracta?

1. ¿Las siguientes instrucciones (instrucciones en las líneas B y C) son válidas? Explique.
   1. Nova nova = **new** Nova();
   2. ObjetoAstronomicoExtraSolar ob = **new** ObjetoAstronomicoExtraSolar();
   3. ObjetoAstronomicoExtraSolar oa = nova;
2. 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.
   1. Nova nova = new Nova();
   2. ObjetoAstronomicoExtraSolar oa = nova;
   3. oa.explotar();
   4. ((Nova) oa).explotar();
3. ¿Por qué la línea 15 imprime *true*? ¿Para cualquier objeto que se cree siempre imprimirá lo mismo? ¿Qué imprimen las siguientes líneas? ¿Por qué?

obN = **null**;

System.***out***.println(obN **instanceof** Object); System.***out***.println("" + obN **instanceof** Object);

1. Agregue el siguiente constructor, ¿Se genera error? ¿Se puede colocar constructores a clases abstractas? ¿Qué sentido tiene?

**public** ObjetoAstronomicoExtraSolar () {

**this**.ID = 4;

**this**.tipoCuerpo2();

}

1. 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?

**class** EnanaBlanca **extends** Estrella

{ **void** agotarCombustible() {

System.***out***.println("Enana blanca muere");

}

}

1. ¿Por qué las líneas 16 y 17 imprimen *false* y *false* respectivamente?