Estudiantes

12-10561 - Pablo Maldonado

12-10645 - Alejandra Cordero

Respuestas

1. Los lenguajes de programación orientados a objetos que poseen herencia simple están limitados a incorporar comportamientos de un solo ancestro al momento de definir una clase.

***No**, una de las características principales de los lenguajes orientados a objetos es que estos pueden heredar comportamientos de sus ancestros. Es decir, las instancias que se heredan no sólo se limitan a la de su padre, ya que una clase puede heredar tanto de su abuelo, del papá de su abuelo, y así sucesivamente.*

2. Lenguajes de POO con un sistemas de tipos estático (C++, Java, C#) no tienen la posibilidades de elegir la implementación de un método a tiempo de ejecución (despacho dinámico).

***No**, estos lenguajes sí tienen la posibilidad de elegir qué método se usará a tiempo de ejecución, sólo que para hacerlo deben construir una tabla en tiempo de compilación llamada "Tabla virtual" (Virtual Table). En ella se almacenan todos los métodos y variables que una clase puede alcanzar, bien sea por posesión, o por herencia. De esta forma, se permite el despacho dinámico en estos tipos de lenguajes.*

3. La introspección y reflexividad son conceptos que se manejan en la POO pero no guardan ninguna relación entre sí.

***No**, la instrospección y la reflexividad se relacionan de la siguiente manera. La primera de ellas, permite conocer la clase de un objeto determinado. Para ello, es necesario que este tenga definido un método que devuelva el tipo de la clase, y que además pueda referirse a sí mismo. En este punto es necesaria la reflexividad. Cuando se desee conocer el tipo de un objeto, este llamará el método que devuelve su clase, pasándose como argumento a sí mismo.*

4. En un lenguaje con un sistema de tipos dinámico la sobrecarga de métodos es innata y representa una comodidad dado que permite implementar un mismo método para distintos tipos.

***No**, existen lenguajes con sistema de tipos dinámicos como Python que no sobrecargan los métodos, sólo los sobreescriben. Por lo general, en lenguajes que siguen este esquema no es necesario sobrecargar, ya que se pueden usar técnicas como el duck-typing debido a la flexibilidad que brinda el sistema de tipos.*

5. En los lenguajes POO existen los términos interfaz, módulo, clase abstracta, rol, etc; definidos como objetos que pueden encapsular definiciones de clases o implementaciones concretas de métodos.(redactar mejor)

***Sí**, la función de estos es encapsular bien sea definiciones, implementaciones o definiciones de clase.*

6. Los métodos virtuales permiten asociar, al momento de compilar, una implementación de un método sobrecargado con una llamada del mismo; eliminando el **overhead** del despacho dinámico.

***No,** dado que la información almacenada por las Tablas Virtuales es básicamente las firmas de los métodos, entonces a tiempo de ejecución de igual forma sería necesario hacer una búsqueda de la firma correspondiente.*

7. Cuando un lenguaje de POO tiene herencia simple no ocurre el problema del diamante, pero de igual forma pueden existir llamadas ambiguas de métodos, dado que incorporar interfaces, módulos, protocolos, etc, no evita colisión de nombres.

***Sí**, a pesar de que con herencia simple se elimina el problema del diamante, de igual forma pueden existir choques de nombres. Por ejemplo, si se crean los módulos A y B, que definen un método "imprimir", y ambos son incluidos a una clase X, entonces habrá un choque de nombres en ella. Al momento de la llamada no se sabe si se tiene que utilizar la versión de A o la de B.*

8. El paso de mensaje es un término que se maneja en modelos concurrentes, también de POO y es equivalente a la llamada de una función.

***Sí**, el paso de mensajes en los POO por lo general se implementa como llamadas a los métodos de otros objetos, y las llamadas a métodos son principalmente llamadas a funciones con un objeto como parámetro.*

9. Sin importar la herencia del lenguaje de POO, una clase podría tener más de un ancestro.

***Sí**, puede tener más de un ancestro ya que los ancestros de una clase están conformados por su padre, la clase papá de su padre, la clase papá de su abuelo y así sucesivamente. Si una clase está debajo de una cadena de herencia de más de dos niveles, entonces tendrá varios ancestros independientemente de si la herencia del lenguaje es simple o múltiple.*