**1.- ¿Qué es una clase?** Una clase es una definición de objetos que comparten el mismo tipo de propiedades, relaciones y comportamientos.

**2.- ¿Qué es la clase de modelaje?** Modelado de clases es la tarea de especificar estas clases utilizando un lenguaje específico en el que las propiedades se denominan atributos, las relaciones se denominan asociaciones y el comportamiento se define como operaciones.

**3.- ¿Cómo es la estructura de una clase?** Nombre de la clase, atributos y funcionalidades o métodos.

**4.- Describe la generalización** Una generalización es una relación entre una clase más genérica, por ejemplo Animal, y una clase más específica, por ejemplo el perro. La clase genérica generaliza el específico, así como la específica se especializa el genérico. Una generalización puede ser interpretado como un "es-un" relación.

**5.- ¿A qué se le llama superclase y subclase?** La clase genérica se denomina superclase mientras que la clase específica se denomina la subclase.

**6.- ¿Cómo se indica una generalización en UML?** En UML una generalización se indica dibujando una flecha abierta de la subclase a la superclase. La subclase se pone preferiblemente por debajo de la superclase para expresar la jerarquía.

**7.- ¿Qué es una clase abstracta?** Es una superclase en donde no se pueden crear instancias.

**8.- ¿Qué es asociación?** Una asociación es una relación estructural entre las clases que especifica que los objetos de una clase están conectados a los objetos de otra clase, como los empleados están conectados a un departamento.

**9.- ¿Cómo se crean las asociaciones en UML?** Asociaciones se dibujan en UML como líneas continuas. En ambos extremos de la línea de la multiplicidad de la asociación se indica, que expresa el número de objetos de una clase están asociados con el número de objetos de la otra clase.

**10.- ¿Qué es agregación y composición?** Agregación y composición son a la vez una relación "todo-parte". En caso de una composición, la parte no puede existir sin el conjunto mientras que en caso de una agregación que puede. Agregación, por tanto, es a menudo llamado "la agregación débil" mientras que la composición se llama "agregación fuerte".

**11.- ¿En qué se diferencian la agregación y composición? Explique con un ejemplo:** Para entender la diferencia entre la composición y la agregación, revise los ejemplos de la figura anterior. Un grupo de alumnos se compone de estudiantes. Un estudiante no deja de existir cuando se desmanteló el grupo de alumnos. Por lo tanto, el modelo sea una agregación.

**12.- ¿Qué es una interfaz?** Una interfaz es una colección de atributos relacionados y firmas de operación. Considere una interfaz como una especie de un contrato para implementar el comportamiento estándar específico, al tiempo que oculta la implementación específica de este comportamiento. Nombres de interfaz a menudo terminan con "-able" o "ble".

**13.- Define una restricción:** En el contexto de la modelización de clase una restricción puede ser definida como una restricción que se aplica al estado de un objeto o de un conjunto de objetos (no necesariamente de la misma clase) pueden estar en, o la transición de un estado a otro.

**14.- ¿Qué es un modelo de análisis?** El lenguaje utilizado para describir el modelo de análisis es el lenguaje de los desarrolladores. Pero, al ser un modelo conceptual, el modelo de análisis se supone que abstraer de tecnologías o lenguajes específicos.

**15.- ¿Qué es el diseño de modelo?** El modelo de diseño consiste en clases de diseño e incorpora decisiones que se toman para hacer frente a los requerimientos no funcionales, como el rendimiento, la distribución y así sucesivamente.

**Diferencia entre clase abstracta e interfaz Interfaz:** Define unos métodos generales -nunca los implementa- de obligada implementación en las clases que la implementen. Las interfaces permiten la Herencia Múltiple, funcionalidad por otra parte NO contemplada en PHP (y en los lenguajes de programación en general). Las clases afectadas no tienen por qué tener parentesco (Herencia), al contrario que ocurre con las clases abstractas.

Abstracta: Una clase abstracta puede definir un interface de programación que deben implementar sus clases hijas; es decir, proporciona a sus subclases la declaración de todos los métodos que deben implementar para cumplir el interface de programación (Contrato de Implementación). La diferencia estriba en que las clases afectadas tienen parentesco (Herencia); es decir, cuando se desarrolla una jerarquía de clases en que algún comportamiento está presente en todas ellas pero que se materializa de distintas formas para cada una.

Integrantes:

Jorge Luis Hernández Falfán

José Luis Rueda Martínez

Francisco Tejeda Lucio

Cristopher Alexis Díaz Esparza