**Análisis de las diferentes relaciones de Herencia e Interface.**

**Relaciones de Interface (Vehicle):**

La interface de Vehicle, es una interface que posee métodos como el de incrementar velocidad y disminuirla, métodos que se van a implementar en la clase abstracta de Car, para definir la velocidad de un vehículo. Es por eso, que la flecha de implementación apunta hacia ella.

**Relación de Has entre Car (abstract) y CarType (enumeration):**

La relación de Has entre Car y CarType, significa que en la clase abstracta Car, su atributo CarType type necesita un tipo de modelo que obtiene mediante CarType. Es decir, Car tiene un CarType, (O sea, el auto tiene un tipo de auto), una relación de Has de 1.1 ya que cada uno de los autos solo puede tener un tipo, sea COMPACT, SPORT, entre otros.

**Relaciones de Herencia de Car (Abstract) con FORD y BMW:**

La relación de Herencia que existe entre estas clases significa que FORD y BMW son clases hijas de Car, donde heredan sus atributos para poder desarrollarse, es decir, Car siendo una super clase que se extiende en FORD y BMW. Es por eso que la flecha apunta de las clases hijas a la madre.

**Describa como aplicar el Polimorfismo al proyecto:**

Para iniciar, lo desarrollaría siguiendo los pasos de la guía, haría que procese objetos que compartan la misma superclase como si todos fueran objetos de la superclase, para que cada objeto sepa como “hacer lo correcto” en respuesta a la llamada del mismo método., para que en tiempo de ejecución se pueda accionar el método mas conveniente para la acción, y así en la hora de ejecutarlo, cada objeto sepa que hacer, volviendo el código mas completo.

Referencia Bibliográfica:

Ramírez Gonzales, W. Clases Abstractas, Interfaces y Polimorfismo.