Nombre Alumno: Gomez Victor Agustín LU: 55262

Nombre de Profesor: Medina-Arqueros

Grupo de Laboratorio: Grupo numero 3 TP: 3 Fecha: 12/8/2021

Tipos de conocimiento:

Por variable de instancia: Cuando el objeto principal posee como variable de instancia a otro objeto, como, por ejemplo:  
public class CajaDeAhorro{

private Persona titular; ……. Va a durar hasta que el objeto primario se elimine

Por parámetro de un método: Como su nombre lo indica, sucede cuando un método utiliza como parámetro a otro objeto, por ejemplo:   
public void casarseCon( Mujer p\_mujer ){ ……. Va a durar hasta que el método termine

Por variable temporal: Cuando un objeto colaborador es instanciado dentro de un método, por ejeplo:  
public int antiguedad(){

Calendar fechaHoy = new GregorianCalendar();

int anioHoy = fechaHoy.get(Calendar.YEAR);

return anioHoy - this.getAnioIngreso();

} Dura hasta que la variable temporal culmine

PseudoVariable : Cuando se le asigna el objeto a this en un método, por ejemplo :  
public String toString(){

return this.getNroCuenta() + "\t" + this.getTitular().nomYApe() + "\t" + this.getSaldo();

} Dura mientras se ejecute el método

Se pueden representar gráficamente, con una flecha completa, de un objeto a otro, indicando que este lo conoce por variable de instancia, con flecha punteada, cuando lo conoce por variable temporal, con una sola línea cuando ambos objetos se conocen entre si, o con una línea del objeto a si mismo, cuando se conoce el mismo.

También, cuando el conocimiento es por variable de instancia, se colocan números debajo de las flechas para indicar la cantidad de objetos de ese tipo puede recibir el constructor de ese objeto, por ejemplo, en este practico la clase EMPLEADOCONJEFE, puede recibir 0 o 1 objeto jefe de ese tipo de variable, indicando así que un empleado puede tener jefe o no. También el conocimiento puede ser de a muchos objetos, pero eso ya sería una colección de objetos.

Por otra parte, podemos ver que gracias al encapsulamiento de estructura, nosotros podemos modificar una clase, sin modificar su interfaz publica, esto puede verse claramente en la clase PERSONA, en su versión 2.0, donde modificamos el atributo anioNacimiento, por un objeto tipo Calendar fechaNacimiento, sin modificar la interfaz publica y todavía pudiendo utilizar el constructor que recibía los atributos anioNacimiento.