

UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”

INTEGRANTES:

FECHA: 26/07/2021

- *Bonifaz Christian*
- *Calderon Mateo*
- *Guano Ariel*
- *Mera Erick*
- *Rosas Mario*

CARRERA:

Electrónica y Automatización

NRC:

3725

ASIGNATURA:

Programación Orientada a Objetos

TEMA:

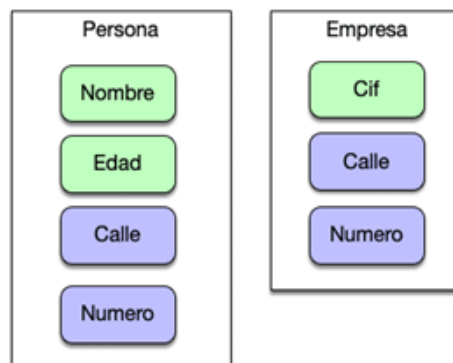
Composición En Java

OBJETIVOS:

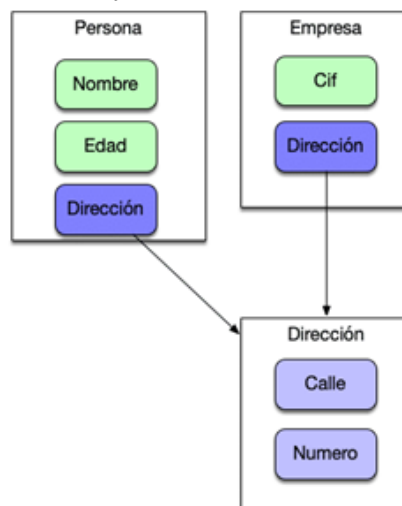
- Seguir avanzando dentro del campo de la programación orientada a objetos con el estudio y aplicación de la composición en Java.
- Analizar y comprender la composición en Java con el agrupamiento de uno o varios objetos y valores como atributos.
- Crear un código haciendo uso de lo aprendido en clase en conjunto con lo investigado con respecto a la composición en Java.

MARCO TEÓRICO:

La composición es el agrupamiento de uno o varios objetos y valores, como atributos, que conforman el valor de los distintos objetos de una clase. Normalmente, los atributos contenidos se declaran con acceso privado (private) y se inicializan en el constructor de la nueva clase.



Este enfoque no favorece la reutilización ya que ambas clases comparten las propiedades y nos encontramos ante una situación sencilla de repetición de código. Para solventar este problema debemos apostar por un enfoque de Java Composición. En donde tanto Persona como Empresa se apoyan en la clase Dirección para gestionar el concepto de calle y número.



Uso e implementación de composición en Java

- **Uso**

El uso de Java Composición en el trabajo diario es muy habitual. Cuando uno comienza a programar siempre parece que la herencia es la característica de un lenguaje de programación orientado a objetos que aporten mayor reutilización. Sin embargo, esto no es cierto ya que la herencia se usa normalmente para categorizar es decir por ejemplo Persona y Deportista un Deportista es una Persona. Fuera de este tipo de opciones el uso de herencia no suele encajar y se opta por un enfoque de composición una clase usa otras clases. Un ejemplo cotidiano podría ser el diseñar las Clases Persona, Empresa y Dirección. En un ejemplo muy básico tanto la Persona como la Empresa contienen ellas mismas la información de la dirección como propiedades individuales.



- **Implementación**

La composición consiste en crear una clase nueva agrupando objetos de clases que ya existen. Una composición agrupa uno o más objetos para construir una clase, de manera que las instancias de esta nueva clase contienen uno o más objetos de otras clases. Normalmente los objetos contenidos se declaran con acceso private y se inicializan en el constructor de la clase.

Ejemplo de implementación de composición

```
public class Vehículo {  
  
    private String matricula;  
  
    private String marca;  
  
    private String modelo;  
  
    private String color;  
  
    private double tarifa;  
  
    private boolean disponible;
```

// se omiten los métodos 'get' y 'set' de la clase

```
public Vehiculo (String matricula,
```

```
String marca,
```

```
String modelo,
```

```
String color,
```

```
double tarifa) {
```

```
this.matricula = matricula;
```

```
this.marca = marca;
```

```
this.modelo = modelo;
```

```
this.color = color;
```

```
this.tarifa = tarifa;
```

```
this.disponible = false;
```

```
}}
```

CONCLUSIONES Y RECOMENDACIONES:

- La composición nos sirve para crear una clase nueva agrupar los objetos de clases que ya existen. Una composición agrupa uno o más objetos para construir una clase, de manera que las instancias de esta nueva clase contienen uno o más objetos de otras clases.
- Es una relación que representa a objetos compuestos por otros objetos. El objeto en el nivel superior de la jerarquía es el todo y los que están en los niveles inferiores son sus partes o componentes. Un componente es parte esencial de una entidad. La relación es fuerte al punto que, si el componente es eliminado o desaparece, la clase mayor (el todo) deja de existir.
- Permite dinamizar y reciclar el código haciendo que el programa sea más ordenado y dinámico a la hora de poder trabajar haciendo entendible el programa.

BIBLIOGRAFÍA:

Martínez, G. (2016). Composición en Java. Obtenido de:

<http://www.ingenieriasystems.com/2016/03/composicion-en-java.html>

López, E. (2014). Java Composición y la reutilización de Código. Obtenido de:

<https://www.arquitecturajava.com/java-composicion-y-la-reutilizacion-del-codigo/>

Schildt, Helbert (2007). *Java. Manual de referencia* (7 edición). Mc Graw Hill. pp.

105. ISBN 0-07-226385-7. Consultado el 21 de enero de 2016.