



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
GUIA DE PRACTICA DE LABORATORIO /TALLER

CARRERA: Ingeniería Electrónica	GUÍA No. 01	TIEMPO ESTIMADO: 1h y 20 min.
ASIGNATURA: Programación Orientada a Objetos	FECHA DE ELABORACION: 26/07/2021 SEMESTRE: Mayo – Septiembre 2021	
TÍTULO: Composición En Java	DOCENTE: Ing. César Osorio Integrantes: <ul style="list-style-type: none">– Samuel Aldaz– Jean Jácome– Steven López– Jennifer Tipan	

OBJETIVO GENERAL

- Construir un programa capaz de crear una clase nueva agrupando objetos de clases que ya existen y Analizar la relación dependiente en donde un objeto más complejo es conformado por objeto más pequeños.

OBJETIVO ESPECIFICO

- Desarrollar un programa sobre creación y una clase nueva agrupando objetos, generar un link para subirlo al repositorio de GitHub.
- Analizar paso a paso la composición que 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.
- Demostrar el funcionamiento del programa con pruebas de compilación.



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

GUIA DE PRACTICA DE LABORATORIO /TALLER

INTRODUCCIÓN.

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.

En una clase se pueden definir uno o más métodos constructores para inicializar los atributos de un objeto con valores distintos de los valores por defecto de Java. Para instanciar un objeto es necesario indicar los valores iniciales de sus atributos cuando se ejecuta el método constructos. Java diferencia los métodos sobrecargados por el número y el tipo de los argumentos que tiene el método que debe ejecutar por el número y el tipo de argumentos que recibe.

MARCO TEÓRICO.

¿Qué es Composición En Java?

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. Los Constructores de una clase permiten dar un valor inicial a los atributos del objeto. La composición significa utilizar objetos dentro de otros objetos. Por ejemplo, un applet es un objeto que contiene en su interior otros objetos como botones, etiquetas, etc. Cada uno de los controles está descrito por una clase.

Para crear un objeto se utiliza el operador `new`. Si no se ha definido un método constructor para la clase, entonces el objeto se instancia indicando el nombre de la clase y a continuación un paréntesis abierto y otro cerrado. Si ya se ha definido un método constructor, entonces no es posible instanciar un objeto utilizando el constructor.



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN GUIA DE PRACTICA DE LABORATORIO /TALLER

- Prueba de escritorio del código.

Sentencia	i	j	arreglo.length	menor	cual	temp	arreglo			
Inicialización				4			48	45	14	71
i = 0	0									
i < n - 1 → 0 < 4 - 1 → 0 < 3 → true										
menor = arreglo[i] → menor = arreglo[0] → menor = 48					48					
cual = i → cual = 0						0				
j = i + 1 → j = 0 + 1 → j = 1		1								
j < n → 1 < 4 → true										
arreglo[j] < menor → arreglo[1] < 48 → 45 < 48 → true										
menor = arreglo[i] → menor = arreglo[0] → menor = 45					45					
cual = j → cual = 1						1				
j++		2								
arreglo[j] < menor → arreglo[2] < 45 → 14 < 45 → true										
menor = arreglo[i] → menor = arreglo[0] → menor = 14					14					
cual = j → cual = 2						2				
j++		3								
arreglo[j] < menor → arreglo[3] < 14 → 71 < 14 → false										
j++		4								
j < n → 4 < 4 → false										
temp = arreglo[i] → temp = arreglo[0] → temp = 48						48				
arreglo[i] = menor → arreglo[0] = 14							14			
arreglo[cual] = temp → arreglo[2] = 48								48		
i++	1									
i < n - 1 → 1 < 4 - 1 → 1 < 3 → true										
menor = arreglo[i] → menor = arreglo[1] → menor = 45					45					
cual = i → cual = 1						1				
j = i + 1 → j = 1 + 1 → j = 2		2								
j < n → 2 < 4 → true										
arreglo[j] < menor → arreglo[2] < 45 → 48 < 45 → false										
j++		3								

Con la prueba de escritorio hecha en el programa podemos notar el funcionamiento de este. Logramos ver con éxito como se inicializaba sin problema y se podía ejecutar los códigos predichos. La función de este programa fue exitosa. La introducción de la cuenta y la identificación de los usuarios. Pudimos observar la separación por paquetes de cada disposición dando al programa una mejor optimización y sobre todo una mejor visualización para el programador. Dando un aprecio de orden.

Se comprobó el programa de acuerdo a sus códigos como se ve en la estructura anterior. Determinando una cantidad y un nombre de cuenta. Todos los términos booleanos fueron aceptados al igual que todos los caracteres. Realmente el programa no presente afectaciones, con la excepción de regresiones, ya que el programa debía ser reiniciado para poder volver a usarlo. Se usa con claridad las herramientas aprendidas y estas fueron incorporadas sin causar un problema.



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

GUIA DE PRACTICA DE LABORATORIO /TALLER

En si el programa funciono con naturalidad. Si bien con la prueba de escritorio se pudo notar algunas formas de mejora como la organización de la salida de datos, es algo que realmente no afecta en sí al programa o a la funcionalidad. Se pueden crear versiones futuras que logren mostrar un avance notorio en el programa y que pueda incorporar algunos otros códigos que puedan servir para su mejora.

Conclusiones.

Se llegó a la conclusión después de haber realizado el programa, donde la composición consiste en crear una clase nueva agrupando objetos de clases que ya existen. el uso será de las composiciones serán de manera optimizada con la estructura que se encuentra en el programa, su funcionamiento los distintos tipos de clases por objetos se diferencian por las clases que usaremos para representar y manipularlos.

Recomendaciones.

Antes de finalizar, se sugiere que se tenga una estructura planificada de las clases que agrupan los objetos en diferentes clases a crear el Programa en Java, así mismo analizar la relación dependiente en donde un objeto más complejo es conformado por objeto más pequeños.

Bibliografía.

Elorduy, V. T. L. E. D. G. (2017, 23 septiembre). *Composición de clases*. Java a tu alcance.

<https://javaparajavatos.wordpress.com/2016/03/21/composicion-de-clases/>



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN GUIA DE PRACTICA DE LABORATORIO /TALLER

Composición. (s. f.). Composición. Recuperado 27 de julio de 2021, de

<http://www.sc.edu.es/sbweb/fisica/cursosJava/fundamentos/clases1/composicion.htm>

Martinez, J. (2016). *Fundamentos de programación* (1.^a ed., Vol. 1). De Eenhoorn.

<http://190.57.147.202:90/jspui/bitstream/123456789/1401/1/Fundamentos%20de%20programacion%20en%20Java.pdf>

DOCENTE RESPONSABLE
Ing. César O. Osorio A

COORDINADOR DE ÁREA
Ing. Silvia Arévalo