

NOMBRE DE LA ASIGNATURA	Programación de Computadores II						
NOMBRE DE LA ACTIVIDAD	Quizz. Clases y objetos con java						
TIPO DE ACTIVIDAD	Sincrónica		Asincrónica	x	Individual	x	Grupal
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD			OBJETIVOS				
Unidad 1. Introducción a la POO Unidad 2. Programación basada en objetos			Desarrollo etapa de análisis, diseño e implementación de programa básico				
COMPETENCIAS			INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS				
<ul style="list-style-type: none"> Identificación de clases con sus atributos requeridas en el dominio de solución de un problema Representación de clases mediante UML Identificación de relaciones de asociación (agregación y composición) entre clases Representación de relaciones entre clases mediante UML Diseño de diagramas de clases Implementación de clases mediante Java 			<ul style="list-style-type: none"> Material educativo y material complementario de la asignatura “Unidad 1.” Material educativo y material complementario de la asignatura “Unidad 2.” Talleres y código elaborado en el desarrollo de la asignatura. 				
CONOCIMIENTOS PREVIOS REQUERIDOS							
Conceptos fundamentales de POO - Estructura básica de clases – Sobrecarga de métodos, clases api de java							
ESPECIFICACIONES DE LA ACTIVIDAD							
Se requiere desarrollar un programa orientado objetos en Java que de solución a la problemática planteada.							
Los principales requerimientos que debe cumplir la aplicación son los siguientes: <ul style="list-style-type: none"> Se debe diseñar el respectivo diagrama de clases, el cual debe incluir Clases, atributos, y metodos El proyecto debe ser desarrollado en NetBeans, y debe mantener la siguiente estructura: 							
<pre> graph LR subgraph Source_Packages [Source Packages] Modelo Problema Uml Vista end subgraph Libraries [Libraries] direction TB L1[+] L2[+] end </pre>							

El código fuente debe cumplir con las siguientes prestaciones:

- El código fuente debe ser legible, bien indentado y sin errores de sintaxis. Debe seguir las convenciones de Java en lo que se refiere a nombres de las clases, nombres de métodos, atributos y constantes.
- Código que no compile, no es calificado, es decir, debe ser funcional el aplicativo desarrollado.

Otras consideraciones:

- El quizz es individual
- El código implementado debe ser consistente con el diagrama de clases diseñado (con los atributos definidos, métodos)

Problema planteado:

En física y matemáticas, un vector es un segmento de línea recta que permite representar magnitudes físicas como la fuerza, la velocidad o el desplazamiento.

Los vectores se pueden representar numéricamente mediante las componentes (o coordenadas) de un vector. Por ejemplo:

$\vec{u} = (2, 3)$, representa el vector **U**, con valor 2 en la componente **X**, y 3 para la componente **Y**, en un plano 2d.

$\vec{A} = (4, 4, 4)$, representa el vector **A**, con 4 para sus componentes en **X, Y, Z**, en un plano 3d.

Sobre los vectores se pueden realizar operaciones como la suma de vectores, el módulo de un vector y la multiplicación por un escalar. A continuación, se explican estas operaciones:

Suma de vectores	Módulo de un vector	Multiplicación por un scalar
$\vec{u} + \vec{v} = (2, 3) + (4, -1)$ $= (2 + 4, 3 + (-1))$ $= (6, 2)$	$\vec{u} = (3, -4)$ $ \vec{u} = \sqrt{3^2 + (-4)^2} = \sqrt{9 + 16} = \sqrt{25} = 5$	$\vec{u} = (3, -2)$ $4 \cdot \vec{u} = (4 \cdot 3, 4 \cdot (-2)) = (12, -8)$

Se desea un programa en Java que modele los vectores en un plano 3d. Por lo cual se solicita una clase Vector3D, que incluya los atributos que usted considere necesarios y los siguientes métodos:

Constructor por defecto	Debe asignar un estado inicial al objeto de 0 en sus coordenadas
Constructor sobrecargado	Constructor sobrecargado con las tres coordenadas de un vector 3d, permite inicializar las coordenadas del vector 3d con los valores de x, y, z recibidos como parámetro.
Getter y Setter	Getter y setter por cada atributo de la clase
getVector ()	Debe retornar un String con los valores del vector, de la forma "(x, y, z)", por ejemplo: "(4, 6, 8)"
sumVector ()	Debe aplicar la suma de vectores y retornar su resultado. Recibe como parámetros las coordenadas x, y, z de un segundo vector.
moduleVector ()	Debe retornar el cálculo del módulo del vector
scalarMultiplication ()	Recibe como parámetro un valor entero y retorna el cálculo de su multiplicación por el vector.

Una clase TestVector con un método principal, en el cual, se deberá crear dos objetos de tipo Vector3D. Uno de ellos con el constructor por defecto y el otro con el constructor sobrecargado. El programa deberá imprimirlos en la forma "(x, y, z)", mostrar el resultado de su suma, su módulo y su multiplicación por un valor escalar.

En su estructura, el programa a construir debe contener un paquete modelo, que incluya la clase Vector3D, un paquete vista, que incluya la clase TestVector, y un paquete uml, que incluya el diseño Uml de la clase Vector3D implementada.

- **Modo de entrega**

Se debe montar en aula web enlace a repositorio GitHub del código elaborado.

RECOMENDACIONES / OBSERVACIONES	Sin observaciones
--------------------------------------------	-------------------