Tipo Abstracto de Datos

Equipo:

* Mora Zavala Enrique
* Cámara Torres Cesar Augusto
* Diaz Bautista Rodrigo Andrawi

Materia: Estructura de Datos

Profesor: Esquivel Pat Agustín

Fecha: 27/10/2023

Una fuerza es una acción tal que aplicada sobre un cuerpo es capaz de modificar su velocidad (provocando una aceleración) o su forma, y las fuerzas son representadas por vectores.

El programa que se desarrollo tiene el propósito de representar los vectores de fuerza usando clases y métodos para calcular estos mismos vectores.

El código contiene dos clases, la clase Main y la clase Vector.

* La clase Main se encarga de insertar los componentes de los vectores de fuerza, y de imprimir la magnitud dirección y suma de los vectores.
* La clase Vector se encarga de calcular la magnitud y dirección de los vectores de fuerza, al igual que la suma de estos.
* La clase Vector tiene los métodos getMagnitud, getDireccion y addVectores.
* El método getMagnitud se encarga de calcular la magnitud de los vectores de fuerza.
* El método getDireccion se encarga de calcular la dirección de los vectores de fuerza.
* El método addVectores se encarga de calcular la suma de los vectores de fuerza.

**Diagrama de clases**

**Main**

**- main (args : String [ ])**

**1**

**Vector**

**- x: double**

**- y: double**

**+ Vector(x:double, y: double)**

**+ getMagnitud(): double**

**+ getDireccion(): double**

**+ addVectores(v1: Vector, v2: Vector)**

**+ toString(): String**

**Código:**

**// Se crea un objeto tipo Scanner, se usa para leer la entrada del texto desde la consola.**

import java.util.Scanner;

**// se crea la clase para main**

public class Main {

public static void main(String[] args) {

**//preparar escáner para lectura de datos**

Scanner scanner = new Scanner(System.in);

**// se piden los elementos para establecer el primer vector**

**// se solicita al usuario ingresar las componentes “X” y “Y” del primer vector de fuerza y las almacena en las variables “x1” y “y1”, respectivamente. Luego, se crea un objeto Vector llamado “vector1” con las informacion ingresada.**

System.out.println("Ingresa los componentes del primer vector de fuerza:");

System.out.print("Componente X: ");

double x1 = scanner.nextDouble();

System.out.print("Componente Y: ");

double y1 = scanner.nextDouble();

Vector vector1 = new Vector(x1, y1);

**// se piden los elementos para establecer el segundo vector**

**// se solicita al usuario ingresar las componentes “X” y “Y” del primer vector de fuerza y las almacena en las variables “x2” y “y2”, respectivamente. Luego, se crea un objeto Vector llamado “vector2” con las informacion ingresada**

System.out.println("Ingresa los componentes del segundo vector de fuerza:");

System.out.print("Componente X: ");

double x2 = scanner.nextDouble();

System.out.print("Componente Y: ");

double y2 = scanner.nextDouble();

Vector vector2 = new Vector(x2, y2);

**// Calcular la magnitud y dirección de los de los objetos vector1 y vector2**

System.out.println("Magnitud del primer vector: " + vector1.getMagnitud());

System.out.println("Dirección del primer vector: " + vector1.getDireccion());

System.out.println("Magnitud del segundo vector: " + vector2.getMagnitud());

System.out.println("Dirección del segundo vector: " + vector2.getDireccion());

**// Calcular la suma de los vectores y almacena el resultado en el objeto “sumaVectores”**

Vector sumaVectores = Vector.addVectores(vector1, vector2);

System.out.println("Suma de vectores: " + sumaVectores);

scanner.close();

}

}

**// se crea la clase para Vector**

class Vector {

**// se declaran las variables**

private double x;

private double y;

**// se crea un constructor de tipo “Vector”. Se asignan dos variables “x” y “y” respectivamente**

public Vector(double x, double y) {

this.x = x;

this.y = y;

}

**// se crea el método “getMagnitud” calcula y devuelve la magnitud del vector utilizando el teorema de Pitágoras**

public double getMagnitud() {

return Math.sqrt(x \* x + y \* y);

}

**// se crea el método “getDireccion” calcula y devuelve la dirección (ángulo) del vector en radianes, el resultado representa el ángulo entre el vector y el eje “X”**

public double getDireccion() {

return Math.atan2(y, x);

}

**// se crea el método “addVectores” y suma los vectores “v1” y “v2” creando un nuevo vector cuyas componentes son la suma de las componentes “X” y “Y” de los vectores originales**

public static Vector addVectores(Vector v1, Vector v2) {

double newX = v1.x + v2.x;

double newY = v1.y + v2.y;

return new Vector(newX, newY);

}

**// regresa el valor de “Vector” en forma de una cadena de texto**

public String toString() {

return String.format("Vector(%.2f, %.2f)", x, y);

}

}