LINEAR PROJECT

YOUSEF HISHAM SHA'BAN

This Project was made through researches on the internet, It contains this document and a program attached in a zip file, feel free to link the program to your IDE and experiment on it.

LinearProject.zip

It was a challenge, Since I couldn't attend the lectures nor use the class materials because of the language barrier.

I choose three topics, since they are considerably easy. Which are:
1- Chap 2.2 Operatii cu vectori liberi// Summation
2- Chap 2.3 Coliniaritate si ciplanaritate // Collinearity
3- Chap 2.5 Produs scalar // Multiplication

Hope it meets your expectations!

Summation
In Vectors

Addition Of Vectors

The term "vector addition" refers to the joining of two or more vectors. In vector addition, we use the addition operation to combine two or more vectors to create a new vector equal to the sum of the two or more vectors. Vector addition is employed in physical quantities where vectors indicate velocity, displacement, and acceleration.

Let's learn about vector addition, its characteristics, and numerous rules with solved cases in this article.

What Is Addiction Of Vectors?

Vectors are written with an alphabet and an arrow over them and are represented as a mix of direction and magnitude.

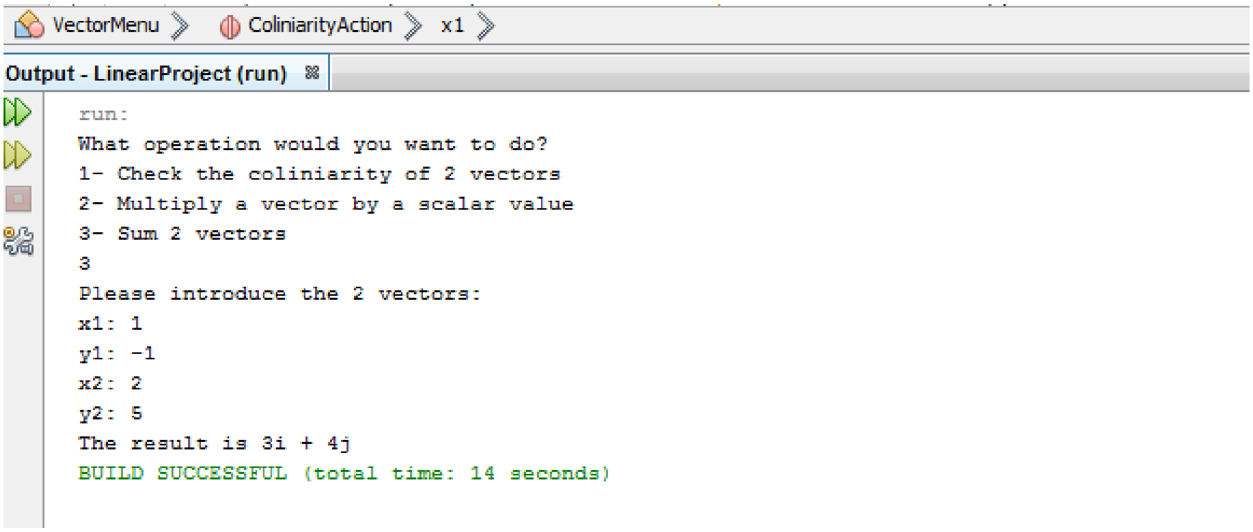
There are two vectors, →aa→ and →bb→, can be combined using vector addition, and the resulting vector can be represented as: →aa→ + →bb→. Before we can learn about the characteristics of vector addition, we must first understand the requirements that must be met while adding vectors. The following are the terms:

- Vectors can only be inserted if they are of the same type. For example, acceleration should be multiplied by acceleration rather than mass.
- We cannot add vectors and scalars together

Consider two vectors C and D. Where, C = Cxi + Cyj + Czk and D = Dxi + Dyj + Dzk. Then, the resultant vector R = C + D = (Cx + Dx)i + (Cy + Dy)j + (Cz + Cz) k

Also, I’ve found this online vectors summation calculator that would be quite convenient:

http://hyperphysics.phy-astr.gsu.edu/hbase/vect.html



Summation Function	Summation Class

public static void SumAction(){
Scanner cin = new Scanner(System.in);
System.out.println("Please introduce the 2 vectors:");
System.out.print("x1: ");
int x1 = cin.nextInt();
System.out.print("y1: ");
int y1 = cin.nextInt();
System.out.print("x2: ");
int x2 = cin.nextInt();
System.out.print("y2: ");
int y2 = cin.nextInt();
Vector v1 = new Vector(x1,y1);
Vector v2 = new Vector(x2,y2);
Vector answer = Sum.Sum(v1,v2);
System.out.println("The result is "+ answer.toString());
}
	public class Sum {
	Vector answer;
	int summationValue;
	public static Vector Sum(Vector v1,Vector v2){
	Vector answer=new Vector((v1.getX()+
	v2.getX()),(v1.getY()+v2.getY()));
	return answer;
	}
	}

The program simply adds a new vector that contains both new vectors lengths.

These rules for summing vectors were applied to free-body diagrams in order to determine the net force (sum of all the individual forces).

Collinearity
In Vectors

Collinear Vectors

Collinear vectors are regarded as an essential subject in vector algebra. Collinear vectors are defined as two or more provided vectors that lie along the same given line. Because two parallel vectors point in the same or opposite direction, we can consider them collinear vectors.

Let's learn about collinear vectors, their definition, vector collinearity requirements, and solved instances in this article.

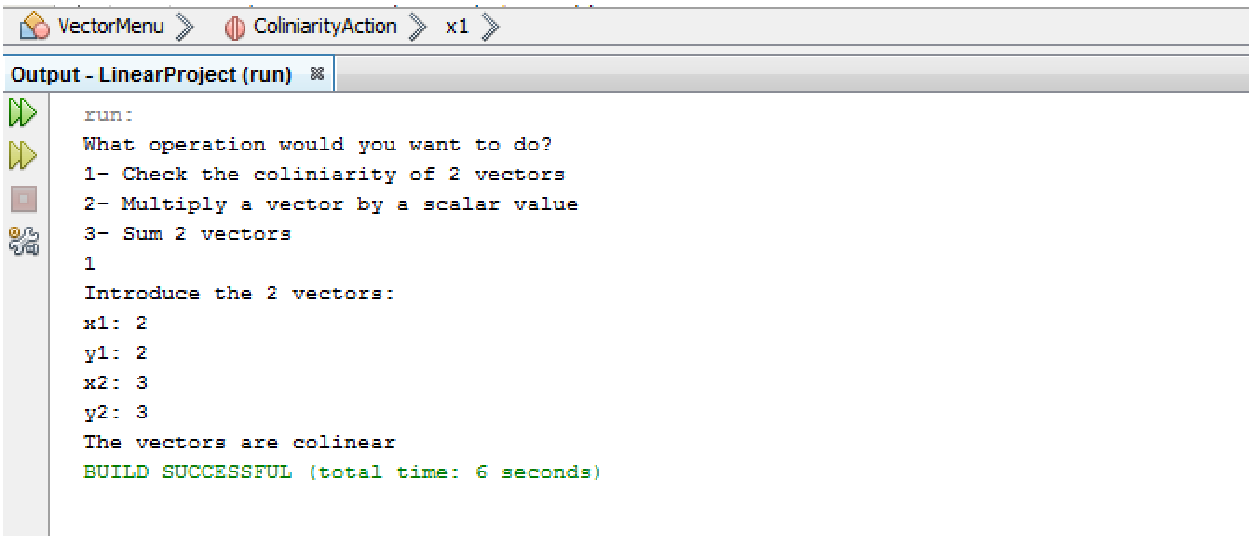
What Are Collinear Vectors?

Any two provided vectors can be termed collinear if they are parallel to the same given line. As a result, we may regard any two vectors to be collinear if and only if they are either along the same line or parallel to each other. The criterion for any two vectors to be parallel to one another is that one of the vectors is a scalar multiple of the other vector.

Conditions Of Collinear Vectors

Certain requirements must be met by any two vectors in order for them to be collinear. The following are the critical vector collinearity conditions:

- **Condition 1**: Two vectors →pp→ and →qq→ are termed collinear vectors if a scalar 'n' exists such that pp = n qq
- **Condition 2**: Two vectors →pp→ and →qq→ Collinear vectors are defined as such if and only if the ratio of their respective coordinates is identical. If one of the components of the supplied vector is equal to zero, this condition is invalid.
- **Condition 3**: Two vectors →pp→ and →qq→ If their cross product equals the zero vector, they are termed collinear vectors. Only three-dimensional or spatial difficulties can be affected by this condition.



Collinearity Function	Collinearity Class

public static void ColiniarityAction(){
Scanner cin = new Scanner(System.in);
System.out.println("Introduce the 2 vectors: ");
System.out.print("x1: ");
int x1 = cin.nextInt();
System.out.print("y1: ");
int y1 = cin.nextInt();
System.out.print("x2: ");
int x2 = cin.nextInt();
System.out.print("y2: ");
int y2 = cin.nextInt();
Vector v1 = new Vector(x1,y1);
Vector v2 = new Vector(x2,y2);
if(Coliniarity.IsColinear(v1,v2)) System.out.println("The vectors are colinear");
else System.out.println("The vectors aren't colinear");
}
	public class Coliniarity {
	public static boolean IsColinear(Vector v1, Vector v2){
	if(v1.getX()/ v2.getX() == v1.getY()/ v2.getY())
	return true;
	else
	return false;
	}
	}

This program simply does the following process:

1) divide x1 on x2.

2) divide y1 on y2.

3) if the value of the division in the past two steps equal the same, then the vectors are collinear.

Multiplication
In Vectors

Dot Product

Dot products are one method of multiplying two or more vectors. The dot product of vectors produces a scalar number. As a result, the dot product is often referred to as a scalar product. It is the sum of the products of the corresponding elements of two number sequences. Geometrically, it is the sum of two vectors' Euclidean magnitudes and the cosine of the angle between them. The vector dot product has several uses in geometry, mechanics, engineering, and astronomy. In the next sections, we will go through the dot product in further depth.

What Is Dot Product Of Two Vectors?

The vector dot product equals the product of the magnitudes of the two vectors and the cosine of the angle between the two vectors. The dot product of two vectors lies in the same plane as the two vectors. The dot product might be a positive or negative actual value.

Dot Product Formula For Vectors

Let a and b be two non-zero vectors, and let be the vectors' included angle. The scalar or dot product is then indicated by a.b, which is defined as:

a→.b→ **=** |→a||→b||a→||b→| **cos θ**.

Here, |→a||a→| is the magnitude of →aa→,

|→b||b→| is the magnitude of →bb→, and θ is the angle between them.

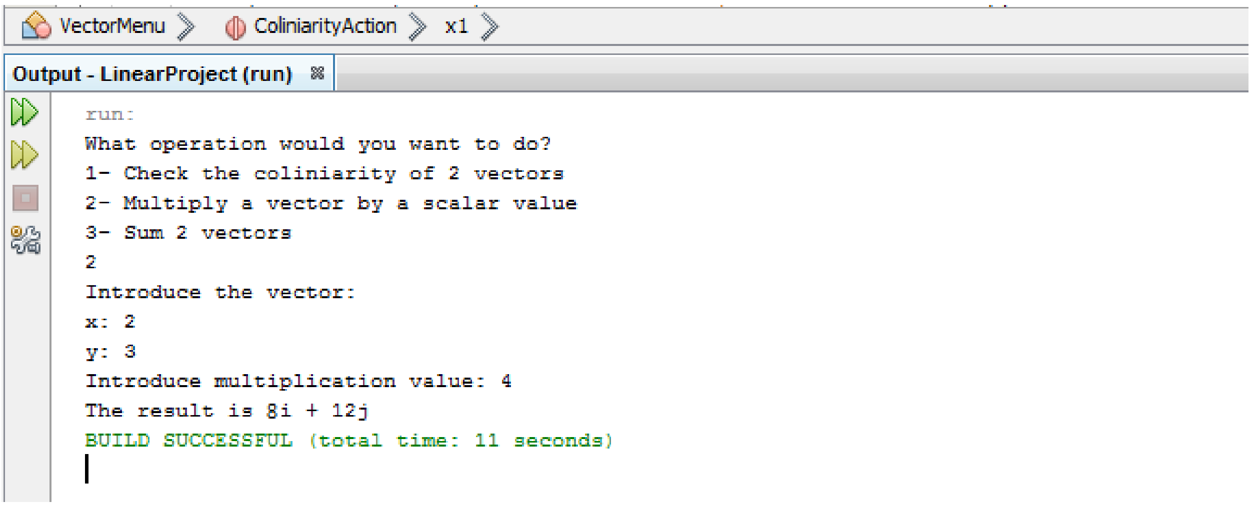
Note: θ is not defined if either →aa→ = 0 or →bb→ = 0.

Angle Between Two Vectors Using Dot Product

The angle between two vectors is calculated as the cosine of the angle between the two vectors. The cosine of the angle between two vectors is equal to the sum of the product of the individual constituents of the two vectors, divided by the product of the magnitude of the two vectors. The formula for the angle between the two vectors is as follows.

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{|a| \cdot |b|}$$

$$\cos\theta = \frac{a_1 . b_1 + a_2 . b_2 + a_3 . b_3}{\sqrt{a_1^2 + a_2^2 + a_3^3} . \sqrt{b_1^2 + b_2^2 + b_3^2}}$$



Multiplication Function	Multiplication Class

public static void MultiplicationAction(){
Scanner cin = new Scanner(System.in);
System.out.println("Introduce the vector: ");
System.out.print("x: ");
int x = cin.nextInt();
System.out.print("y: ");
int y = cin.nextInt();
System.out.print("Introduce multiplication value: ");
int m = cin.nextInt();
Vector v = new Vector(x,y);
Vector result = Multiplication.Multiplication(v,m);
System.out.println("The result is "+ result.toString());
 	
}
	public class Multiplication {
	Vector answer;
	int multiplicationValue;
	public static Vector Multiplication(Vector v,int multiplicationValue){
	Vector answer=new Vector((v.getX()*multiplicationValue),(v.getY()*multiplicationValue));
	return answer;
	}
	}

This program simply does just multiply the indices of the vector.

Main Function	Scalar Class

public static void main(String[] args)
{
int vect_A[] = { 3, -5, 4 };
int vect_B[] = { 2, 6, 5 };
int cross_P[] = new int[n];
 	
System.out.print("Dot product:");
System.out.println(dotProduct(vect_A, vect_B));
}
}
	static int dotProduct(int vect_A[], int vect_B[])
	{
	
	int product = 0;
	for (int i = 0; i < n; i++)
	product = product + vect_A[i] * vect_B[i];
	return product;
	}

This program simply does the Scalar Product implementation to multiple vectors.