

Міністерство освіти і науки України  
КПІ ім. Ігоря Сікорського  
Кафедра ІІІ

ЗВІТ  
з виконання лабораторної роботи № 5  
з кредитного модуля  
“Основи програмування-2. Методології програмування”

Варіант № 21

Виконав:  
студент 1-го курсу  
гр. ІІ-22 ФІОТ  
Патріюк Юрій Олексійович

Київ 2023

## ПОСТАНОВКА ЗАДАЧІ

21. Створити клас TVector, який представляє вектор і містить методи для визначення того, чи є інший вектор паралельним / перпендикулярним до нього та метод знаходження довжини вектора. На основі цього класу створити класи-нащадки, які представляють вектори з просторів  $R^2$  та  $R^3$ . Створити з двовимірні та 4 три-вимірні вектори. Знайти суму довжин векторів, паралельних до першого по порядку двовимірного вектора, та суму векторів, перпендикулярних до першого по порядку тривимірного вектора.

### ТЕКСТ ПРОГРАМИ

#### Вміст файлу main.cpp

```
#include "MainFunctions.h"
int main() {
    //Two-dimensional vectors
    TwoDimensionalVector two_dim_vectors[] = {
        TwoDimensionalVector(2,1),
        TwoDimensionalVector(4,2),
        TwoDimensionalVector(6,1)
    };
    sumOfParallelTwoDimensionalVectors(two_dim_vectors,3);

    //Three-dimensional vectors
    ThreeDimensionalVector three_dim_vectors[] = {
        ThreeDimensionalVector(1, 2, 0),
        ThreeDimensionalVector(2, -1, 10),
        ThreeDimensionalVector(4, -2, 3),
        ThreeDimensionalVector(2, 4, 7)
    };
    sumOfPerpendicularThreeDimensionalVectors(three_dim_vectors,4);

    return 0;
}
```

#### Вміст файлу MainFunctions.h

```
#include "Classes.h"

void sumOfParallelTwoDimensionalVectors(TwoDimensionalVector* vectors, int size);
void sumOfPerpendicularThreeDimensionalVectors(ThreeDimensionalVector* vectors, int size);
```

#### Вміст файлу MainFunctions.cpp

```
#include "MainFunctions.h"

void sumOfParallelTwoDimensionalVectors(TwoDimensionalVector* vectors, int size)
{
    double two_dim_sum = 0;
    for(int i=1; i<size; i++)
    {
        if(vectors[0].isParallel(vectors[i])) {
```

```

        cout<<"vectors[0] is parallel to vectors["<<i<<']"<<endl;
        two_dim_sum += vectors[i].getLength();
    }
}
cout<<"Sum: "<<two_dim_sum<<endl<<endl;
}
void sumOfPerpendicularThreeDimensionalVectors(ThreeDimensionalVector* vectors, int
size)
{
    double three_dim_sum = 0;
    for(int i=1; i<size; i++)
    {
        if(vectors[0].isPerpendicular(vectors[i])) {
            cout<<"vectors[0] is perpendicular to vectors["<<i<<']"<<endl;
            three_dim_sum += vectors[i].getLength();
        }
    }
    cout<<"Sum: "<<three_dim_sum<<endl;
}

```

## Вміст файл Classes.h

```

#include <iostream>
#include <cmath>
using namespace std;

class TVector
{
public:
    bool isParallel() const;
    bool isPerpendicular() const;
    virtual double getLength() const = 0;
    virtual bool isZero() const = 0;
};

class TwoDimensionalVector : public TVector
{
    double X;
    double Y;
public:
    TwoDimensionalVector();
    TwoDimensionalVector(double, double);
    bool isParallel(const TwoDimensionalVector&) const;
    bool isPerpendicular(const TwoDimensionalVector&) const;
    double getLength() const override;
    bool isZero() const override;
};

class ThreeDimensionalVector : public TVector
{
    double X;
    double Y;
    double Z;
public:
    ThreeDimensionalVector();
    ThreeDimensionalVector(double, double, double);
    bool isParallel(const ThreeDimensionalVector&) const;
    bool isPerpendicular(const ThreeDimensionalVector&) const;
    double getLength() const override;
    bool isZero() const override;
};

```

## Вміст файл Classes.cpp

```
#include "Classes.h"

//Two-dimensional vector
TwoDimensionalVector::TwoDimensionalVector()
{
    X=Y=0;
}
TwoDimensionalVector::TwoDimensionalVector(double x, double y)
{
    X = x;
    Y = y;
}
bool TwoDimensionalVector::isParallel (const TwoDimensionalVector& vector) const{
    if(this->isZero() || vector.isZero())
    {
        cout<<"One of the vectors is zero!"<<endl;
        return false;
    }
    double n;
    if(X!=0) n = vector.X / X;
    else n = vector.Y / Y;

    if(X*n == vector.X && Y*n == vector.Y) return true;
    return false;
}
bool TwoDimensionalVector::isPerpendicular(const TwoDimensionalVector& vector) const
{
    return vector.X*X + vector.Y*Y == 0;
}
double TwoDimensionalVector::getLength() const{
    return sqrt(pow(X,2)+pow(Y,2));
}
bool TwoDimensionalVector::isZero() const
{
    return X==0 && Y==0;
}

//Three-dimensional vector
ThreeDimensionalVector::ThreeDimensionalVector() {
    X=Y=Z=0;
}
ThreeDimensionalVector::ThreeDimensionalVector(double x, double y, double z)
{
    X = x;
    Y = y;
    Z = z;
}
bool ThreeDimensionalVector::isParallel (const ThreeDimensionalVector& vector) const{
    if(this->isZero() || vector.isZero())
    {
        cout<<"One of the vectors is zero!"<<endl;
        return false;
    }
    double n;
    if(X!=0) n = vector.X / X;
    else if(Y!=0) n = vector.Y / Y;
    else n = vector.Z / Z;

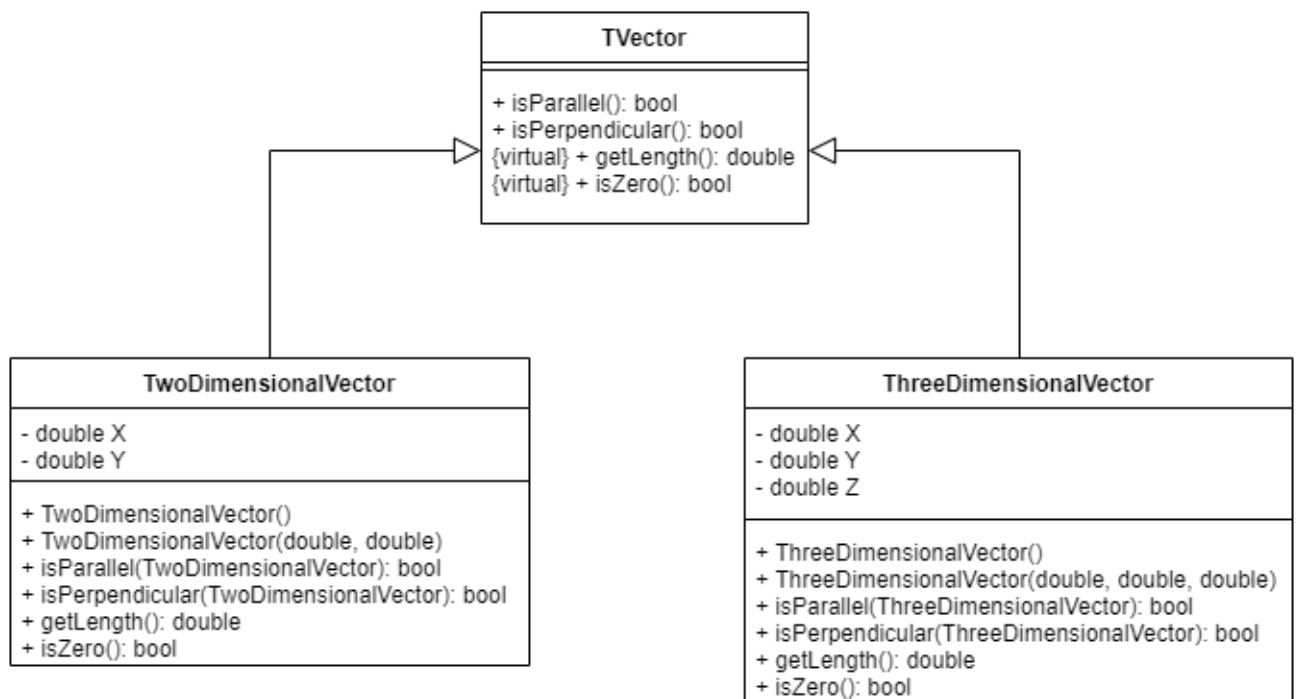
    if(X*n == vector.X && Y*n == vector.Y && Z*n == vector.Z) return true;
    return false;
}
```

```

}
bool ThreeDimensionalVector::isPerpendicular(const ThreeDimensionalVector& vector)
const {
    return vector.X*X + vector.Y*Y + vector.Z*Z == 0;
}
double ThreeDimensionalVector::getLength() const{
    return sqrt(pow(X,2)+pow(Y,2)+pow(Z,2));
}
bool ThreeDimensionalVector::isZero() const
{
    return X==0 && Y==0 && Z==0;
}

```

## ДІАГРАМА КЛАСІВ



## РЕЗУЛЬТАТ ТЕСТУВАННЯ

```

vectors[0] is parallel to vectors[1]
Sum: 4.47214

vectors[0] is perpendicular to vectors[1]
vectors[0] is perpendicular to vectors[2]
Sum: 15.6321

Process finished with exit code 0

```

Посилання на GitHub: <https://github.com/YuriiPatriuk12/Labs.git>