МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет КНТ

Кафедра КИ

Лабораторная работа №2

по курсу «ООП»

по теме «Одномерный массив (вектор)»

Выполнил:

ст. гр. КН-16

\_\_\_\_\_\_\_\_\_\_\_ Семеренко В.К.

Проверила:

\_\_\_\_\_\_\_\_\_\_\_\_ Дикова Ю. Л.

Покровск 2017

Цель:

Освоение основ технологии объектно-ориентированного программирования на примере создания класса «вектор» для описания одномерного массива

Задание:

В вещественном массиве X=(x1,x2,...,xn) каждую пару x[i] и x[j] (i,j=1,...,n; i<>j) отрицательных элементов преобразовать в положительные элементы по формуле x[i],x[j]:=sqrt(x[i]\*x[j]). Пары элементов (i,j) выбирать в порядке их следования в массиве X. Определить, как при этом изменилось среднее арифметическое значение элементов массива X. Сформировать массив Y, включив в него все неповторяющиеся элементы из массива X.

Файл Vector.h:

#pragma once

#include <stdio.h>

class MyVector

{

public:

MyVector();

MyVector(int);

void SetVector(int);

float MiddleCount();

void Compare(float,float);

void SubstitutionNegativ();

void PrintVector();

void AddNewVector();

~MyVector();

private:

float \*X, \*Y;

int size;

};

Файл Vector-methods.cpp:

#include "Vector.h"

#include <iostream>

using namespace std;

MyVector::MyVector() {

size = 0;

X = 0;

}

MyVector::MyVector(int MySize) {

SetVector(MySize);

}

void MyVector::SetVector(int MySize) {

if (MySize <= 0) {

cout << "You enter incorrect size. Pls enter correct size(>0)" << endl;

cin >> MySize;

SetVector(MySize);

}

else

{

size = MySize;

X = new float[size];

cout << "Enter count vector" << endl;

for (int i = 0; i < size; i++) {

cin >> X[i];

}

}

}

float MyVector::MiddleCount() {

float count = 0;

for(int i=0;i<size;i++){

count += X[i];

}

return (count / size);

}

void MyVector::PrintVector() {

for (int i = 0; i < size; i++) {

cout << "X = " << X[i]<<endl;

}

}

void MyVector::SubstitutionNegativ() {

for (int i = 0; i < size; i++) {

if (X[i] < 0) {

int j = i;

for (j; j < size; j++) {

if (X[j] < 0) {

X[i]/= sqrt(X[i] \* X[j]);

X[j] /= sqrt(X[i] \* X[j]);

i = j;

continue;

}

}

}

}

}

void MyVector::Compare(float MiddleBefore, float MiddleAfter) {

if (MiddleBefore > MiddleAfter)

cout << "Middle count non has not changed" << endl;

else if (MiddleAfter > MiddleBefore)

cout << "Middle after became more" << endl;

else

cout << "Middle after and middle before are equal" << endl;

}

void MyVector::AddNewVector() {

int count = 0;

int sizeY = 0;

for (int i = 0; i < size; i++)

{

int j = i + 1;

for (j; j < size; j++) {

if (X[i] == X[j]) {

count++;

}

}

if (count == 0) {

Y = new float[sizeY];

int k = i;

Y[k]=X[i];

sizeY++;

count = 0;

}

else

count = 0;

}

if (sizeY > 0) {

cout << "Second vector: " << endl;

for (int n = 0; n < sizeY; n++)

cout << "Y= " << Y[n] << endl;

//delete Y;

}

else

{

cout << "Secont vector does not exist" << endl;

}

}

MyVector::~MyVector() {

delete X;

}

Файл Main.cpp:

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include "Vector.h"

using namespace std;

int main() {

int Size;

cout << "Enter size vector" << endl;

cin >> Size;

MyVector Vector(Size);

Vector.PrintVector();

float MiddleCountBefore = Vector.MiddleCount();

Vector.SubstitutionNegativ();

Vector.PrintVector();

float MiddleCountAfter = Vector.MiddleCount();

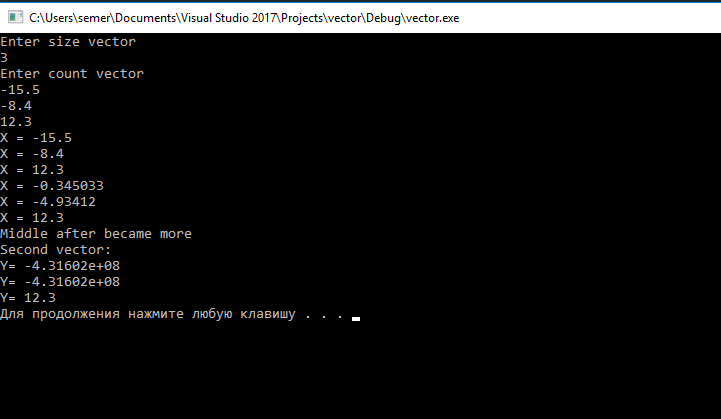
Vector.Compare(MiddleCountBefore, MiddleCountAfter);

Vector.AddNewVector();

system("pause");

}

Листинг:



Вывод:

В ходу выполнения лабораторной работы я выучил основы технологии объектно-ориентированного программирования на примере создания класса «вектор» для описания одномерного массива