#include <iostream>

#include <math.h>

#include <conio.h>

#include <vector>

using namespace std;

class Vector

{

private:

double\* vector;

int size;

public:

Vector()

{

size = 1;

vector = new double[size];

for (int index = 0; index != size; index++)

vector[index] = 0;

}

Vector(int size)

{

if (size > 0)

{

this->size = size;

vector = new double[size];

for (int index = 0; index != size; index++)

vector[index] = 0;

}

else cout << "Размер не подходящий\n";

}

double& operator[](int index)

{

return vector[index];

}

~Vector()

{

delete[] vector;

}

Vector(Vector& other)

{

this->size = other.size;

vector = new double[size];

for (int index = 0; index != size; index++)

vector[index] = other.vector[index];

}

void operator =(Vector& other)

{

for (int index = 0; index != other.size; index++)

this->vector[index] = other.vector[index];

}

Vector operator +(Vector& other)

{

if (size == other.size)

{

for (int index = 0; index != other.size; index++)

this->vector[index] = this->vector[index] + other.vector[index];

return \*this;

}

else cout << "Операция сложения не осуществима из-за разных размеров векторов\n";

}

Vector operator -(Vector& other)

{

if (size == other.size)

{

//Vector temp(size);

for (int index = 0; index != other.size; index++)

this->vector[index] = this->vector[index] - other.vector[index];

return \*this;

}

else cout << "Операция разности не осуществима из-за заных размеров векторов\n";

}

bool operator ==(Vector& other)

{

for (int index = 0; index != size; index++)

{

if (vector[index] != other.vector[index])

{

return 0; break;

}

}

return 1;

}

bool operator !=(Vector& other)

{

for (int index = 0; index != size; index++)

{

if (vector[index] == other.vector[index])

{

return 0; break;

}

}

return 1;

}

bool operator <(Vector& other)

{

if (size == other.size)

{

int a1 = 0, a2 = 0;

for (int index = 0; index != other.size; index++)

{

a1 += vector[index] \* vector[index];

a2 += other.vector[index] \* other.vector[index];

}

return(sqrt(a1) < sqrt(a2));

}

else cout << "Операция разности не осуществима из-за разных размеров векторов\n";

}

bool operator >(Vector& other)

{

if (size == other.size)

{

int a1 = 0, a2 = 0;

for (int index = 0; index != other.size; index++)

{

a1 += vector[index] \* vector[index];

a2 += other.vector[index] \* other.vector[index];

}

return(sqrt(a1) > sqrt(a2));

}

else cout << "Операция разности не осуществима из-за разных размеров векторов\n";

}

void scalar(Vector& other)

{

double sc = 0;

for (int index = 0; index != other.size; index++)

sc += vector[index] \* other.vector[index];

cout << "Скалярное произведение векторов: " << sc << endl;

}

friend ostream& operator<<(ostream& output, const Vector& other);

friend istream& operator>>(istream& input, const Vector& other);

};

ostream& operator<<(ostream& output, const Vector& other)

{

cout << "{ ";

for (int index = 0; index != other.size; index++)

output << other.vector[index] << " ";

cout << "}";

return output;

}

istream& operator>>(istream& input, const Vector& other)

{

for (int index = 0; index != other.size; index++)

{

input >> other.vector[index];

}

return input;

}

///////////////////////////////////////////////////////////

int main()

{

setlocale(LC\_ALL, "");

cout << "\n";

int size\_v;

cout << "Введите размер векторов: "; cin >> size\_v;

Vector a(size\_v);

Vector b(size\_v);

cout << "\nВведите вектор a: ";

cin >> a;

cout << "\nВведите вектор b: ";

cin >> b;

cout << "a: " << a;

cout << "\nb: " << b;

if (a == b)cout << "\nВекторы равны\n";

if (a != b)cout << "\nВекторы не равны\n";

if (a > b)cout << "\na < b\n";

if (a < b)cout << "\na > b\n";

cout << "\na + b: " << a + b;

cout << "\na - b: " << a - b << endl;

a.scalar(b);

return 0;

}



