



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления» (ИУ)
КАФЕДРА «Информационная безопасность» (ИУ8)

Лабораторная работа №2
ПО КУРСУ
«Алгоритмические языки»
на тему
«Изучение перегрузки стандартных
операций в языке Си++»

Студент

ИУ8-22
(Группа)

Г. А. Карев
(И. О. Фамилия)

Преподаватель:

В. В. Соболева
(И.О. Фамилия)

Цель работы:

Цель работы состоит в овладении навыками разработки программ на языке Си++, использующих перегрузку стандартных операций

Вариант 12

Задача:

* скалярное произведение векторов (одинаковой размерности), на выходе значение этого произведения	double	Vector	Vector	double	11
		Vector	double *	double	12
		double *	Vector	double	13

Код:

Vector.h

```
#pragma once
#include <sstream>;
using namespace std;

class Vector {
    double* p = nullptr;
    int n = 0;
public:
    Vector();
    Vector(double* p, int n);
    Vector(const Vector& vcopy);
    Vector(Vector&& vmove);
    double& operator[](int index);
    Vector& operator=(const Vector& vgive_copy);
    Vector& operator=(Vector&& vgive_move);
    ~Vector();
    double operator*(const double* arr) const;
    friend ostream& operator<<(ostream& outS, const Vector& V);
    friend istream& operator>>(istream& inS, const Vector& V);
};
```

Vector.cpp

```
#include "Vector.h"
#include <iostream>
using namespace std;

Vector::Vector()
{
    p = nullptr;
    n = 0;
    cout << "Vector()" << endl;
}

Vector::Vector(double* p, int n)
{
    this->n = n;
    this->p = new double[n];
    for (int i = 0; i < n; i++) this->p[i] = p[i];
    cout << "Vector(double *p, int n)" << endl;
}
```

```

Vector::Vector(const Vector& vcopy)
{
    n = vcopy.n;
    p = new double[n];
    for (int i = 0; i < n; i++)
        p[i] = vcopy.p[i];
    cout << "Vector(const Vector & vcopy)" << endl;
}

Vector::Vector(Vector&& vmove)
{
    swap(p, vmove.p);
    swap(n, vmove.n);
    cout << "Vector(Vector&& vmove)" << endl;
}

double& Vector::operator[](int index)
{
    return p[index];
}

Vector& Vector::operator=(const Vector& vgive_copy)
{
    if (this != &vgive_copy)
    {
        n = vgive_copy.n;
        if (p != nullptr)
        {
            delete[] p;
            p = new double[n];
            for (int i = 0; i < n; i++) p[i] = vgive_copy.p[i];
        }
        cout << "Vector & operator=(const Vector& vgive_copy)" << endl;
        return *this;
    }
}

Vector& Vector::operator=(Vector&& vgive_move)
{
    if (this != &vgive_move)
    {
        delete[] p;
        p = vgive_move.p;
        n = vgive_move.n;
        vgive_move.p = nullptr;
        vgive_move.n = 0;
    }
    cout << "Vector & operator =(Vector&& vgive_move)" << endl;
    return *this;
}

Vector::~Vector()
{
    cout << "~Vector()" << endl;
    if (p != nullptr) delete[] p;
}

double Vector::operator*(const double* arr) const
{
    if (!arr) {
        throw invalid_argument("Передан нулевой указатель (nullptr)");
    }

    double result = 0.0;

```

```

        for (int i = 0; i < n; ++i) {
            result += p[i] * arr[i];
        }
        return result;
    }

ostream& operator<<(ostream& outS, const Vector& v)
{
    for (int i = 0; i < v.n; ++i) {
        outS << v.p[i] << " ";
    }
    return outS;
}

istream& operator>>(istream& inS, const Vector& v)
{
    for (int i = 0; i < v.n; ++i) {
        inS >> v.p[i];
    }
    return inS;
}

```

main.cpp

```

#include <Windows.h>
#include "Vector.h"
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    setlocale(LC_ALL, "RUS");

    ifstream inF("input.txt");
    if (!inF.is_open()) {
        cerr << "input.txt не найден(" << endl;
        return 1;
    }

    int n;
    inF >> n;

    double* arr1 = new double[n];
    double* arr2 = new double[n];
    for (int i = 0; i < n; ++i) {
        inF >> arr1[i];
    }
    for (int i = 0; i < n; ++i) {
        inF >> arr2[i];
    }
    inF.close();

    Vector vec1(arr1, n);
    cout << "Объект 'Вектор': " << vec1 << endl;

    double result = vec1 * arr2;

    ofstream outF("output.txt");
    if (!outF.is_open()) {
        cerr << "Ошибка открытия файла output.txt!" << endl;
        return 1;
    }
    outF << "Результат скалярного произведения векторов: " << result << endl;
    outF.close();
}

```

```

        cout << "Результат скалярного произведенич векторов: " << result << endl;

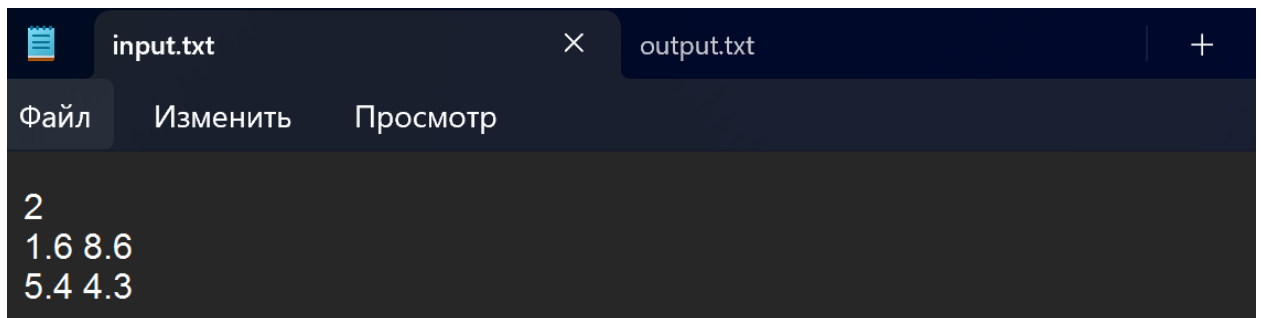
        delete[] arr1;
        delete[] arr2;

        return 0;
    }

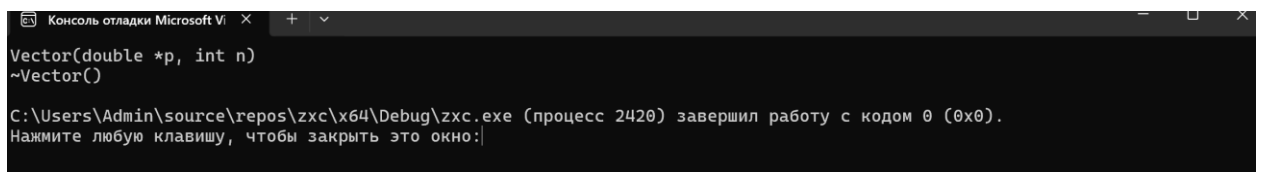
```

Результаты вывода программы:

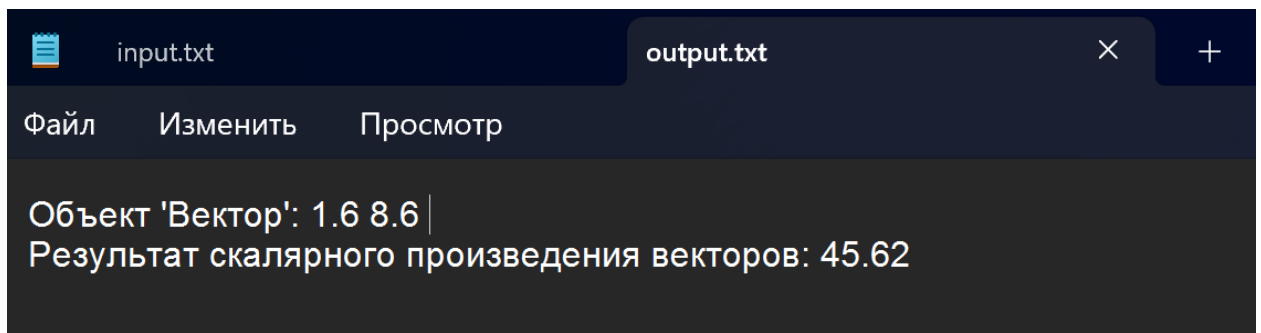
Входные данные из input.txt:



Вывод в консоль:



Вывод в output.txt:



Док-во правильности подсчета скалярного произведения:

Скалярное произведение векторов a и b , заданных своими координатам, находится по формуле: $a \cdot b = x_1 \cdot x_2 + y_1 \cdot y_2$.

Найдем скалярное произведение векторов $a=(1.6;8.6)$ и $b(5.4;4.3)$.

По формуле находим:

$$a \cdot b = 1.6 \cdot 5.4 + 8.6 \cdot 4.3 = 45.62$$

Вывод:

Овладел навыками разработки программ на C++, использующих перегрузку. Скомпилировал программу и получил верный ответ.