

```

#include <iostream>
using namespace std;
class Mnozhenie
{
    int a[100], n;
public: void vvod();
    void vychisl();
    void vichisl(Mnozhenie m, Mnozhenie m2);
    void vichisl(Mnozhenie m, Mnozhenie m2);
    void umnozhenie(Mnozhenie m, Mnozhenie m2);
    void vivod();
};

void main()
{
    Mnozhenie a, b, c;
    a.vvod();
    //a.vivod();
    //a.vichisl(0);
    //a.vvod();
    b.vvod();
    c.umnozhenie(a, b);
    cout << "(a) * b = ";
    c.vivod();
}

void Mnozhenie::vvod()
{
    cout << endl;
    cout << "vvedite stepen' mnozheniya: "; cin >> n;
    //cout<<endl;
    cout << "vvedite koefitsienty:\n";
    for (int k = 0; k <= 99; k++) { a[k] = 0; }
    for (int i = 0; i <= n; i++) { cout << "a" << i << " = "; cin >> a[i]; }
    //cout<<endl;
}

void Mnozhenie::vivod()
{
    int i;
    f = 0;
    //cout<<endl;
    for (int i = n; i >= 0; i--)
        if (a[i] != 0)
        {
            if (f == 0) {
                if (i != 0) cout << a[i] << "x" << i; else cout << a[i];
                f++;
            }
            else {
                if (i != 0)
                {
                    if (a[i] > 0) cout << "+" << a[i] << "x" << i;
                    else cout << a[i] << "x" << i;
                }
                else
                {
                    if (a[i] > 0) cout << "+" << a[i];
                    else cout << a[i];
                }
                f++;
            }
        }
    if (f == 0) { cout << 0; }
    cout << endl;
}

void Mnozhenie::vichisl()
{
    cout << endl;
    int x, v, vx;
    v = 0;
    cout << "vvedite x: "; cin >> x;
    vx = 1;
    for (int i = 0; i <= n; i++) { v = v + a[i] * vx; vx = vx * x; }
}

```

```

40 for (int i = n; i >= 0; i--)
41     if (a[i] != 0)
42     {
43         if (i == 0) {
44             if (i != 0) cout << a[i] << "x" << i; else cout << a[i];
45             i++;
46         }
47         else {
48             if (i != 0)
49                 if (a[i] > 0) cout << "+" << a[i] << "x" << i;
50                 else cout << a[i] << "x" << i;
51             else
52                 if (a[i] > 0) cout << "+" << a[i];
53                 else cout << a[i];
54             i++;
55         }
56     }
57 if (i == 0) { cout << 0; }
58 cout << endl;
59
60 void Mngachlan::vichisl()
61 {
62     cout << endl;
63     int n, v, vx;
64     v = 0;
65     cout << "vvedite x: "; cin >> x;
66     vx = 1;
67     for (int i = 0; i <= n; i++) { v = v + a[i] * vx; vx = vx * x; }
68     cout << "f(x) = a << x) = " << v;
69     cout << endl;
70 }
71
72 void Mngachlan::stozh(Mngachlan m1, Mngachlan m2)
73 {
74     for (int k = 0; k <= 99; k++) { a[k] = 0; }
75     if (m1.n > m2.n) n = m1.n; else n = m2.n;
76     for (int i = 0; i <= n; i++) { a[i] = m1.a[i] + m2.a[i]; }
77 }
78
79 void Mngachlan::vichit(Mngachlan m1, Mngachlan m2)
80 {
81     for (int k = 0; k <= 99; k++) { a[k] = 0; }
82     if (m1.n > m2.n) n = m1.n; else n = m2.n;
83     for (int i = 0; i <= n; i++) { a[i] = m1.a[i] - m2.a[i]; }
84 }
85
86 void Mngachlan::sumozh(Mngachlan m1, Mngachlan m2)
87 {
88     for (int k = 0; k <= 99; k++) { a[k] = 0; }
89     n = m1.n + m2.n;
90     for (int i = 0; i <= m1.n; i++) for (int j = 0; j <= m2.n; j++) { a[i + j] = m1.a[i] + m2.a[j]; }
91 }

```

Консоль отладки Microsoft Visual Studio

```
vvedite stepen' mnogochlena: 1
vvedite koeficienti
a0 = 1
a1 = 2
```

```
vvedite stepen' mnogochlena: 2
vvedite koeficienti
a0 = 1
a1 = 2
a2 = 0
C(x) = 4*x^2+4*x^1+1
```

F:\Учеба Олега\labs\laba1-2\x64\Debug\laba1-2.exe (процесс 13740) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно:

```
1 #include <iostream>
2 #include <vector>
3 #include <iomanip>
4
5 using namespace std;
6
7 template<class T>
8 class vector3d // 3D вектор
9 {
10 public:
11     vector3d();
12     vector3d(T);
13
14     void SetVector(const char* const caption); // функция назначения значений вектору
15     void display(const char* const caption) const; // функция вывода вектора на экран
16     void sum(vector3d* vec1, vector3d* vec2); // функция сложения векторов
17     void razn(vector3d* vec1, vector3d* vec2); // функция разности векторов
18     void dlinu(vector3d* vec1); // функция длины вектора
19     void skalar(vector3d* vec1, vector3d* vec2); // скалярное произведение векторов
20     void kosinus(vector3d* vec1, vector3d* vec2); // косинус угла между векторами
21 private:
22     T x, y, z;
23 };
24
25 template<class T>
26 vector3d(T)::vector3d()
27 {
28     x = T(0);
29     y = T(0);
30     z = T(0);
31 }
32
33 template<class T>
34 vector3d(T)::vector3d(T)
35 {
36 }
37
38 template<class T>
39 void vector3d(T)::SetVector(const char* const caption)
40 {
41     cout << caption << endl;
42     cout << "x = ";
43     cin >> x;
44     cout << "y = ";
45     cin >> y;
46     cout << "z = ";
47     cin >> z;
48 }
49
50 template<class T>
51 void vector3d(T)::sum(vector3d* vec1, vector3d* vec2)
52 {
53     x = vec1->x + vec2->x;
54     y = vec1->y + vec2->y;
55     z = vec1->z + vec2->z;
56 }
57
58 template<class T>
59 void vector3d(T)::razn(vector3d* vec1, vector3d* vec2)
60 {
61     x = vec1->x - vec2->x;
62     y = vec1->y - vec2->y;
63     z = vec1->z - vec2->z;
64 }
65
66 template<class T>
67 void vector3d(T)::dlinu(vector3d* vec1)
68 {
69     x = vec1->x;
70     y = vec1->y;
71     z = vec1->z;
72 }
```

```

64     z = vec1->z - vec2->z;
65 }
66
67 template<class T>
68 void vector3d<T>::dline(vector3d* vec1)
69 {
70     cout << "Длина вектора:" << endl;
71     float a = sqrt((vec1->x + vec1->x) + (vec1->y + vec1->y) + (vec1->z + vec1->z));
72     cout << a << endl;
73 }
74
75 template<class T>
76 void vector3d<T>::scalar(vector3d* vec1, vector3d* vec2)
77 {
78     float det = (vec1->x + vec2->x) + (vec1->y + vec2->y) + (vec1->z + vec2->z);
79     cout << "Результат скалярного произведения векторов:" << endl;
80     det << endl;
81 }
82
83 template<class T>
84 void vector3d<T>::cosinus(vector3d* vec1, vector3d* vec2)
85 {
86     float det = (vec1->x + vec2->x) + (vec1->y + vec2->y) + (vec1->z + vec2->z);
87     float a = sqrt((vec1->x + vec1->x) + (vec1->y + vec1->y) + (vec1->z + vec1->z));
88     float a2 = sqrt((vec2->x + vec2->x) + (vec2->y + vec2->y) + (vec2->z + vec2->z));
89     // по формуле нахождения косинуса угла между векторами
90     float cos = (det / (a * a2));
91
92     cout << "Косинус угла между векторами:" << endl;
93     cos << endl;
94 }
95
96 template<class T>
97 void vector3d<T>::display(const char* const caption)
98 {
99     cout << caption << endl;
100     cout << "Вектор с координатами:" << endl;
101     cout << "x = " << x << endl;
102     cout << "y = " << y << endl;
103     cout << "z = " << z << endl;
104 }
105
106 int main(int argc, char* argv[])
107 {
108     locale::global(locale(""));
109
110     vector3d<float>* vector1 = new vector3d<float>;
111     vector3d<float>* vector2 = new vector3d<float>;
112     vector3d<float>* result_vector = new vector3d<float>;
113
114     int p;
115     while (true)
116     {
117         cout
118             << "1 - Создать два вектора" << endl
119             << "2 - Показать два вектора" << endl
120             << "3 - Векторная сумма вектора" << endl
121             << "4 - Скалярное произведение векторов" << endl
122             << "5 - Косинус угла между векторами" << endl
123             << "6 - Выход из программы" << endl;
124
125         cin >> p;
126
127         switch (p)
128         {
129             case 1:
130                 vector1->SetVector("Вектор с координатами вектора 1.");
131

```

```
lab01-2.cpp  в X: "no name"  + (Глобальная область)  + main(1st arg: char* argv[])
107 void vector3d<T>::display(const char* const caption)
108 {
109     cout << caption << endl
110         << "Вектор с координатами:" << endl
111         << "x = " << x << endl
112         << "y = " << y << endl
113         << "z = " << z << endl;
114 }
115
116 int main(int argc, char* argv[])
117 {
118     locale::global(locale(""));
119
120     vector3d<float> vector1 = new vector3d<float>;
121     vector3d<float> vector2 = new vector3d<float>;
122     vector3d<float> result_vector = new vector3d<float>;
123
124     int p;
125     while (true)
126     {
127         cout
128             << "1 - Создать два вектора" << endl
129             << "2 - Разность двух векторов" << endl
130             << "3 - Векторы двух векторов" << endl
131             << "4 - Скалярное произведение векторов" << endl
132             << "5 - Нормировать вектор" << endl
133             << "6 - Выход из программы" << endl;
134
135         cin >> p;
136
137         switch (p)
138         {
139             case 1:
140                 vector1->SetVector("Вектор координаты вектора 1:");
141                 vector2->SetVector("Вектор координаты вектора 2:");
142                 result_vector->sum(vector1, vector2);
143                 result_vector->display("-----РЕЗУЛЬТАТ-----");
144                 break;
145             case 2:
146                 vector1->SetVector("Вектор координаты вектора 1:");
147                 vector2->SetVector("Вектор координаты вектора 2:");
148                 result_vector->sub(vector1, vector2);
149                 result_vector->display("-----РЕЗУЛЬТАТ-----");
150                 break;
151             case 3:
142                 vector1->SetVector("Вектор координаты вектора 1:");
143                 result_vector->dLine(vector1);
144                 break;
155             case 4:
146                 vector1->SetVector("Вектор координаты вектора 1:");
147                 vector2->SetVector("Вектор координаты вектора 2:");
148                 result_vector->scalr(vector1, vector2);
149                 break;
150             case 5:
141                 vector1->SetVector("Вектор координаты вектора 1:");
142                 vector2->SetVector("Вектор координаты вектора 2:");
143                 result_vector->cosinus(vector1, vector2);
144                 break;
155             case 6:
146                 exit(0);
147                 break;
158         }
159     }
160 }
```

```

1 - Сложение двух векторов
2 - Разность двух векторов
3 - Вычислить длину вектра
4 - Скалярное произведение векторов
5 - Косинус угла между векторами
0 - Выход из программы
3

```

Введите координаты вектора:

```

X = 3
Y = 4
Z = 5

```

ДЛИНА ВЕКТОРА:

```

7.07107

```

```

1 - Сложение двух векторов
2 - Разность двух векторов
3 - Вычислить длину вектра
4 - Скалярное произведение векторов
5 - Косинус угла между векторами
0 - Выход из программы
1

```

Введите координаты вектора 1:

```

X = 1
Y = 2
Z = 3

```

Введите координаты вектора 2:

```

X = 3
Y = 4
Z = 5

```

-----РЕЗУЛЬТАТ-----

Вектор с координатами:

```

X = 4
Y = 6
Z = 8

```

```

1 - Сложение двух векторов
2 - Разность двух векторов
3 - Вычислить длину вектра
4 - Скалярное произведение векторов
5 - Косинус угла между векторами
0 - Выход из программы
0

```

F:\Учеба Олега\labs\laba1-2\x64\Debug\laba1-2.exe (процесс 3036) завершил работу с кодом 0.

Нажмите любую клавишу, чтобы закрыть это окно:

```

1 #include <iostream>
2
3 using namespace std;
4
5 struct Train {
6     string namePunkt;
7     int numTrain;
8     string timeArrive;
9 };
10
11 Train putTrain(Train t) {
12     cout << "\n Enter namePunkt - ";
13     cin >> t.namePunkt;
14     cout << "\n Enter numTrain - ";
15     cin >> t.numTrain;
16     cout << "\n Enter timeArrive - ";
17     cin >> t.timeArrive;
18     return t;
19 }
20
21 void outTrain(Train t) {
22     cout << "Name: " << t.numTrain << " Destination Route: " << t.namePunkt << " Arrive: " << t.timeArrive;
23 }
24
25 void outTrainList(Train t[], int size) {
26     for (int i = 0; i < size; i++) {
27         outTrain(t[i]);
28     }
29 }
30
31 void swap(Train& x, Train& y)
32 {
33     Train temp = x;
34     x = y;
35     y = temp;
36 }
37
38 void showInfoSort(Train t[], int size) {
39     cout << "\n _____ " << "Full timetable train" << " _____ ";
40     for (int i = 0; i < size; i++) {
41         for (int j = 0; j < size - 1; j++) {
42             if (t[i].numTrain < t[j].numTrain) swap(t[i], t[j]);
43         }
44     }
45     outTrainList(t, size);
46 }
47
48
49 int main()
50 {
51     int size = 5, numTrainEnter;
52     bool found = false;
53     Train TrainList[5];
54     for (int i = 0; i < size; i++) {
55         TrainList[i] = putTrain(TrainList[i]);
56     }
57     cout << "Enter num train:";
58     cin >> numTrainEnter;
59     for (int i = 0; i < size; i++) {
60         if (numTrainEnter == TrainList[i].numTrain) {
61             outTrain(TrainList[i]);
62             found = true;
63         }
64     }
65     if (!found) {
66         cout << "\n Information about train not found";
67     }
68 }

```



```
16     cout << "Enter timeArrive - ";
17     cin >> t.timeArrive;
18
19     return t;
20 }
21
22 void outTrainList(Train t[], int size) {
23     cout << "Index: " << t.numTrain << " Destination Route: " << t.nameRoute << " Arrive: " << t.timeArrive;
24 }
25
26 void swap(Train x, Train y) {
27     Train temp = x;
28     x = y;
29     y = temp;
30 }
31
32 void showInfoSort(Train t[], int size) {
33     cout << "Print out full timetable train" << endl;
34     for (int i = 0; i < size; i++) {
35         for (int j = 0; j < size - 1; j++) {
36             if (t[i].numTrain < t[j].numTrain) swap(t[i], t[j]);
37         }
38     }
39     outTrainList(t, size);
40 }
41
42 int main() {
43     int size = 5, numTrainEnter;
44     bool found = false;
45     Train TrainList[5];
46     for (int i = 0; i < size; i++) {
47         TrainList[i] = getTrain(TrainList[i]);
48     }
49     cout << "Enter num train: ";
50     cin >> numTrainEnter;
51     for (int i = 0; i < size; i++) {
52         if (numTrainEnter == TrainList[i].numTrain) {
53             outTrain(TrainList[i]);
54             found = true;
55         }
56     }
57     if (!found) {
58         cout << "No information about train not found";
59     }
60     showInfoSort(TrainList, size);
61     return 0;
62 }
```

Enter namePynkt - Tombov

Enter numTrain - 6

Enter timeArrive - 9.00

Enter namePynkt - Kirov

Enter numTrain - 4

Enter timeArrive - 19.30

Enter num train:12

Information about train not found

Full timetable train

Num: 1 Destination Route: Moskva Arrive: 2.00

Num: 2 Destination Route: Samara Arrive: 3.00

Num: 4 Destination Route: Kirov Arrive: 19.30

Num: 5 Destination Route: Kazan Arrive: 12.00

Num: 6 Destination Route: Tombov Arrive: 9.00

F:\Учеба Олега\labs\laba1-2\x64\Debug\laba1-2.exe (процесс 29244) завершил работу с кодом 0.

Нажмите любую клавишу, чтобы закрыть это окно: