

```

1 #include <iostream>
2 #include <string.h>
3 #include <iomanip>
4 using namespace std;
5
6 class Vector {
7 public:
8     Vector() {}
9     Vector(x, y, z, point[3]) {
10
11         point[double x, double y, double z] {
12
13             this->x = x;
14             this->y = y;
15             this->z = z;
16             point[0] = x;
17             point[1] = y;
18             point[2] = z;
19
20         }
21
22         Vector() {
23             x = y = z = 0;
24             point[2] = { 0 };
25
26         }
27
28         void print() {
29             cout << "Point (x = " << x << ", y = " << y << ", z = " << z << ", " << endl;
30
31         }
32
33         Point operator+(const Vector v);
34         Point operator-(const Vector v);
35         double operator*(int i) {
36             return point[i];
37
38         }
39
40         double operator[](char a) {
41             int i = (a - 'a') * 220;
42             return point[i];
43
44         }
45
46 };
47
48 class Vector {
49 public:
50     friend Point Point::operator+(const Vector v);
51     friend Point Point::operator-(const Vector v);
52     double x, y, z, vector[3];
53
54     Vector(double x, double y, double z) {
55
56         this->x = x;
57         this->y = y;
58         this->z = z;
59         vector[0] = x;
60         vector[1] = y;
61         vector[2] = z;
62
63     }
64
65     Vector() {
66         x = y = z = 0;
67         vector[2] = { 0 };
68
69     }
70
71     void print() {
72         cout << "Vector (x = " << x << ", y = " << y << ", z = " << z << ", " << endl;
73
74     }
75
76     double operator*(int i) {
77         return vector[i];
78
79     }
80
81     double operator[](char a) {
82         return vector[a - 'A' * 220];
83
84     }
85
86     Point operator+(const Vector v) {
87         return Point(x + v.x, y + v.y, z + v.z);
88
89     }
90
91     Point operator-(const Vector v) {
92         return Point(x - v.x, y - v.y, z - v.z);
93
94     }
95
96     friend Point operator+(const Vector v, const Vector v);
97     friend Point operator-(const Vector v, const Vector v);
98
99 };

```

```
File Edit View Compiler Tools Help Window
C:\Users\user\Documents\Vector.cpp
Print

20 x = y = z = 0;
21 point[2] = { 0 };
22
23 void print() {
24     cout << "point: \nx = " << x << "\ny = " << y << "\nz = " << z << endl;
25 }
26
27 Point operator+(const Vector& v);
28 Point operator-(const Vector& v);
29 double operator()(int i) {
30     return point[i];
31 }
32
33 double operator[](char a) {
34     int i = (int)a - 128;
35     return point[i];
36 }
37
38 class Vector {
39     friend Point Point::operator+(const Vector& v);
40     friend Point Point::operator-(const Vector& v);
41     double x, y, z, vector[3];
42 public:
43     Vector(double x, double y, double z) {
44         this->vx = x;
45         this->vy = y;
46         this->vz = z;
47         vector[0] = x;
48         vector[1] = y;
49         vector[2] = z;
50     }
51     Vector() {
52         x = y = z = 0;
53         vector[2] = { 0 };
54     }
55     void print() {
56         cout << "vector: \nx = " << x << "\ny = " << y << "\nz = " << z << endl;
57     }
58     double operator()(int i) {
59         return vector[i];
60     }
61     double operator[](char a) {
62         return vector[a - '0' - 128];
63     }
64 };
65
66 Point Point::operator+(const Vector& v) {
67     return Point(x + v.x, y + v.y, z + v.z);
68 }
69
70 Point Point::operator-(const Vector& v) {
71     return Point(x - v.x, y - v.y, z - v.z);
72 }
73
74 int main() {
75     Point a(2, 3, 0.5);
76     Vector d(2, 0, 1.0, 0.5);
77     (a + d).print();
78     (a - d).print();
79
80     cout << a["x"] << endl;
81     cout << d["y"] << endl;
82     return 0;
83 }
```

Консоль отладки Microsoft Visual Studio

```
point:
x - 5.4
y - 1
z - 2.5
point:
x - 0.6
y - -1
z - 1.5
3
1

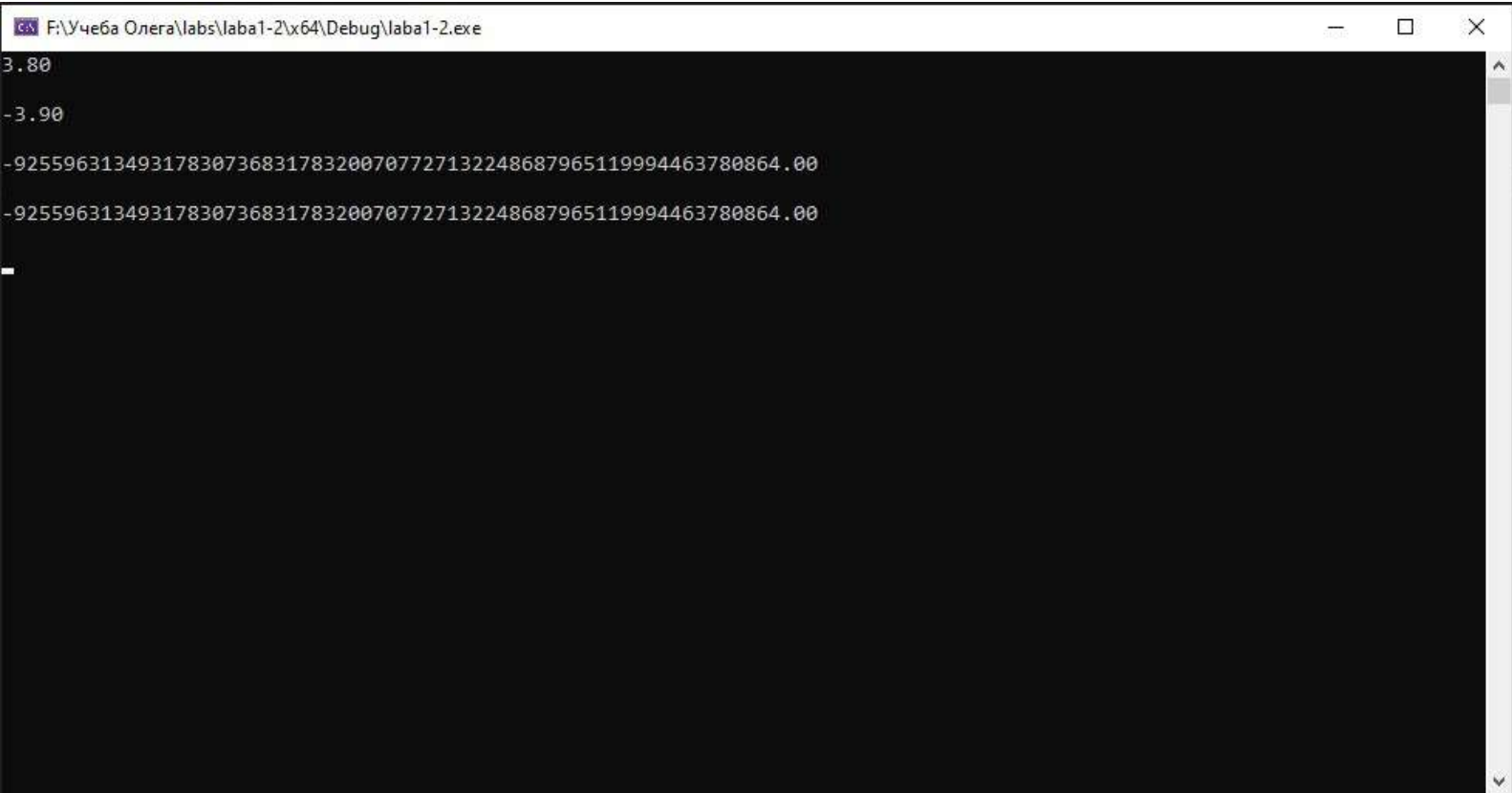
F:\Учеба Олега\labs\laba1-2\x64\Debug\laba1-2.exe (процесс 29092) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:■
```

```
1 #include <iostream>
2 #include <string.h>
3 #include <string>
4 #include <iomanip>
5 using namespace std;
6
7 class Float {
8 private:
9     int a, b;
10    double c;
11 public:
12    Float() {
13        a = b = c = 0;
14    }
15    Float(int a, int b) {
16        if ((-9999 < a && a < 9999) && (-99 < b && b < 99)) {
17            this->a = a;
18            this->b = b * 10;
19        }
20        else {
21            cout << "bezumnaya sz zapovedat" << endl;
22            exit(1);
23        }
24        c = 0;
25    }
26    Float(double nc) {
27        double f;
28        f = modf(nc, &i);
29        if (f > 99) {
30            this->c = round(nc);
31        }
32        else this->c = nc;
33        a = b = 0;
34    }
35    Float& operator=(const Float& f) {
36        string str1 = to_string(a) + "." + to_string(b);
37        string str2 = to_string(f.a) + "." + to_string(f.b);
38        double x1 = stod(str1);
39        double x2 = stod(str2);
40
41        double t = round(((x1 + x2) * 100) / 100);
42        Float v(t);
43        return v;
44    }
45    Float& operator=(const Float& f) {
46        string str1 = to_string(a) + "." + to_string(b);
47        string str2 = to_string(f.a) + "." + to_string(f.b);
48        double x1 = stod(str1);
49        double x2 = stod(str2);
50        double t = round((x1 - x2) * 100) / 100;
51        Float v(t);
52        return v;
53    }
54    friend ostream& operator<<(ostream& os, const Float& f);
55    friend istream& operator>>(istream& in, Float& f);
56 };
57 ostream& operator<<(ostream& os, const Float& f) {
58     string str = "";
59     if (f.c == 0 && (f.a < 0 || f.b < 0)) {
60         str << "-" + to_string(abs(f.a)) + "." + to_string(abs(f.b));
61         os << str << endl;
62     }
63     else if (f.c == 0 && (f.a > 0 || f.b > 0)) {
64         str << to_string(abs(f.a)) + "." + to_string(abs(f.b));
65         os << str << endl;
66     }
67     else if (f.c != 0) {
68         str << to_string(f.c);
69         os << str << endl;
70     }
71 }
```

```

37 string str2 = to_string(f.a) + "." + to_string(f.b);
38 double t1 = stod(str1);
39 double t2 = stod(str2);
40
41 double t = round((t1 + t2) + 100) / 100;
42
43 float v(t);
44 return v;
45 }
46
47 float operator-(const Float& f) {
48     string str1 = to_string(a) + "." + to_string(b);
49     string str2 = to_string(f.a) + "." + to_string(f.b);
50     double t1 = stod(str1);
51     double t2 = stod(str2);
52     double t = round((t1 - t2) + 100) / 100;
53     float v(t);
54     return v;
55 }
56
57 friend ostream& operator<< (ostream& os, const Float& f) {
58     friend istream& operator>> (istream& is, Float& f);
59 };
60
61 ostream& operator<< (ostream& os, const Float& f) {
62     string str = "";
63     if (f.c == 0 && (f.a < 0 || f.b < 0)) {
64         str += "-" + to_string(abs(f.a)) + "." + to_string(abs(f.b));
65         os << str << endl;
66     }
67     else if (f.c == 0 && (f.a > 0 || f.b > 0)) {
68         str += to_string(abs(f.a)) + "." + to_string(abs(f.b));
69         os << str << endl;
70     }
71     else if (f.c != 0) {
72         os << fixed << setprecision(2) << f.c << endl;
73     }
74     return os;
75 }
76
77 istream& operator>> (istream& is, Float& f) {
78     int n = f.c;
79     return is;
80 }
81
82 int main() {
83     float a(3, 8);
84     cout << a << "\n"; // 3.80
85     float b(-3, 9);
86     cout << b << "\n"; // -3.90
87     cout << a + b << "\n"; // -0.10
88     cout << a - b << "\n"; // 7.70
89     float c;
90     cin >> b; // 11.999999
91     cout << b << "\n"; // 12.00
92 }

```



The image shows a Windows command prompt window with a black background and white text. The title bar at the top reads "F:\Учеба Олега\labs\laba1-2\x64\Debug\laba1-2.exe". The command prompt displays the following output:

```
3.80  
-3.90  
-92559631349317830736831783200707727132248687965119994463780864.00  
-92559631349317830736831783200707727132248687965119994463780864.00  
_
```

The output consists of four lines of text. The first two lines are "3.80" and "-3.90". The next two lines are long decimal numbers: "-92559631349317830736831783200707727132248687965119994463780864.00". The final line is a single underscore character "_".