



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 и системы управления»

КАФЕДРА \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

**Лабораторная работа №8**  
**по курсу «Языки и методы программирования»**  
**«Разработка шаблона класса»**

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

*Москва 2025*

# 1 Задание

Выриант 2: `Polygon<T, N>` – N-угольник на плоскости, заданный координатами вершин. Координаты вершин представлены числами типа `T`. Операции: вычисление периметра (возвращает `double`); вставка новой вершины после *i*-той вершины (в результате формируется новый  $(N + 1)$ -угольник). В случае, если `T` – `double`, в классе должна быть дополнительная операция поворота N-угольника относительно его первой вершины на заданный угол.

Выриант 4: `IntVector<L, H, N>` – целочисленный вектор размера `N` с элементами из диапазона от `L` до `H`, имеющий следующие операции:

1. сложение с другим вектором типа `IntVector<L2, H2, N>`, в результате которого формируется новый вектор типа `IntVector<L+L2, H+H2, N>`;
2. скалярное умножение на вектор типа `IntVector<L2, H2, N>`.

Если размер диапазона не превышает 256, для представления вектора должен использоваться массив `char`'ов.

# 2 Результаты

Исходный код 3– 4.

## Листинг 1 — Polygon.cpp (2 вариант)

```

1 #include <cmath>
2
3 using namespace std;
4
5 template<int N, typename T>
6 class Polygon {
7 private:
8     pair<T, T> data[N];
9 public:
10     Polygon(pair<T, T> (&vertices)[N]) {
11         for (int i = 0; i < N; ++i) {
12             data[i] = vertices[i];
13         }
14     }
15
16     double perimeter() {
17         int ans = 0;
18         for (int i = 0; i < N; i++) {
19             double x = data[i].first - data[(i+1) % N].first;
20             double y = data[i].second - data[(i+1) % N].second;
21             ans += sqrt(x*x + y*y);
22         }
23         return ans;
24     }
25
26     void addNode(int i, pair<T, T> node) {
27         pair<T, T> new_data[N + 1];
28         for (int j = 0; j < N; j++) {
29             if (j < i) {
30                 new_data[j] = data[j];
31             } else if (j == i) {
32                 new_data[j] = node;
33             } else {
34                 new_data[j] = data[j-1];
35             }
36         }
37         data = new_data;
38     }
39
40     template <typename U = T>
41     typename enable_if<is_same<U, double>::value>::type
42     rotate(double angle) {
43         const double x0 = data[0].first;
44         const double y0 = data[0].second;
45         const double cos_a = cos(angle);
46         const double sin_a = sin(angle);
47
48         for (size_t i = 1; i < N; ++i) {
49             const double dx = data[i].first - x0;
50             const double dy = data[i].second - y0;
51             data[i].first = x0 + dx * cos_a - dy * sin_a;
52             data[i].second = y0 + dx * sin_a + dy * cos_a;
53         }
54     }
55
56     const pair<T, T>& getVertex(int i) const {
57         return data[i];
58     }
59 };

```

## Листинг 2 — main.cpp (2 вариант)

```
1 #include <iostream>
2 #include "Polygon.cpp"
3
4 int main() {
5     std::pair<double, double> square[4] = {
6         {0.0, 0.0},
7         {1.0, 0.0},
8         {1.0, 1.0},
9         {0.0, 1.0}
10    };
11    Polygon<4, double> poly(square);
12
13    std::cout << "Perimeter: " << poly.perimeter() << std::endl;
14
15    double pi = 2 * acos(0.0);
16    poly.rotate(pi / 2);
17
18    for (int i = 0; i < 4; ++i) {
19        const auto& vertex = poly.getVertex(i);
20        std::cout << "Vertex " << i << ": (" << vertex.first << ", " << vertex.second << ")"
21        << std::endl;
22    }
23    return 0;
24 }
```

## Листинг 3 — main.cpp (4 вариант)

```
1 #include <iostream>
2 #include "IntVector.cpp"
3
4 int main() {
5     IntVector<-10, 10, 3> v1 = {1, 2, 3};
6     IntVector<0, 1000, 3> v2 = {100, 200, 300};
7
8     auto v3 = v1 + v2;
9     std::cout << "Sum: ";
10    v3.print();
11
12    std::cout << "product: " << v1.dot(v2) << std::endl;
13    return 0;
14 }
```

## Листинг 4 — IntVector.cpp (4 вариант)

```

1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 template<int L, int H, size_t N>
7 class IntVector {
8 private:
9     using StorageType = typename std::conditional<(H - L <= 256), char, int>::type;
10    std::vector<StorageType> data;
11
12 public:
13    IntVector() : data(N, static_cast<StorageType>(0 - L)) {}
14
15    IntVector( initializer_list<int> values) : data(N) {
16        size_t i = 0;
17        for (int val : values) {
18            data[i++] = static_cast<StorageType>(val - L);
19        }
20    }
21
22    int operator[](size_t index) const {
23        return static_cast<int>(data[index]) + L;
24    }
25
26    void set(size_t index, int value) {
27        data[index] = static_cast<StorageType>(value - L);
28    }
29
30    template<int L2, int H2>
31    auto operator+(const IntVector<L2, H2, N>& other) const {
32        IntVector<L + L2, H + H2, N> result;
33        for (size_t i = 0; i < N; ++i) {
34            result.set(i, (*this)[i] + other[i]);
35        }
36        return result;
37    }
38
39    template<int L2, int H2>
40    int dot(const IntVector<L2, H2, N>& other) const {
41        int result = 0;
42        for (size_t i = 0; i < N; ++i) {
43            result += (*this)[i] * other[i];
44        }
45        return result;
46    }
47
48    void print() const {
49        cout << "[ ";
50        for (size_t i = 0; i < N; ++i) {
51            cout << (*this)[i] << " ";
52        }
53        cout << "]" << std::endl;
54    }
55
56    constexpr size_t size() const { return N; }
57 };

```

```
● veter.ok77@MacBook-Pro-Rodion var1 % ./program
Perimeter: 4
Vertex 0: (0, 0)
Vertex 1: (6.12323e-17, 1)
Vertex 2: (-1, 1)
Vertex 3: (-1, 6.12323e-17)
```

Рис. 1 — Результат работы

```
● veter.ok77@MacBook-Pro-Rodion var2 % ./program
Vector sizes: 3, 3
Sum: [ 101 202 303 ]
Dot product: 1400
```

Рис. 2 — Результат работы

### 3 Вывод

Я научился создавать шаблоны класса