



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 Тараканов В. Д.

Преподаватель Посевин Д. П.

*Москва 2024*

# 1 Задание

Polygon<T,N> – N-угольник на плоскости, заданный координатами вершин. Координаты вершин представлены числами типа T. Операции: вычисление периметра (возвращает double); добавление новой вершины (в результате формируется новый (N + 1)-угольник). Если N = 3, в классе должна быть дополнительная операция вычисления площади треугольника (возвращает double).

## 2 Результаты

Листинг 1 — Класс Polynomial

```
1
2     #ifndef declaration_h
3     #define declaration_h
4
5     #include <cmath>
6
7     template<typename T, int N>
8     class Polygon {
9     private:
10        std::pair<T, T>* vertices;
11
12    public:
13        Polygon(const std::pair<T, T>* verts) {
14            vertices = new std::pair<T, T>[N];
15            for (size_t i = 0; i < N; ++i) {
16                vertices[i] = verts[i];
17            }
18        }
19
20
21
22        double perimeter() const {
23            double perimeter = 0.0;
24            for (size_t i = 0; i < N - 1; ++i) {
25                perimeter += distance(vertices[i], vertices[i + 1]);
26            }
27            perimeter += distance(vertices[N - 1], vertices[0]);
28            return perimeter;
29        }
30
31
32
33
```

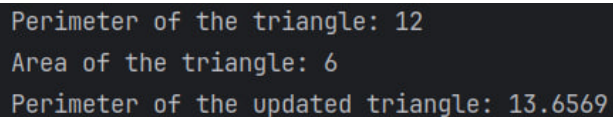
## Листинг 2 — Класс Polygon(продолжение)

```
1
2     Polygon<T, N+1> addVertex(const std::pair<T, T>& vertex) const {
3         std::pair<T, T>* newVertices = new std::pair<T, T>[N + 1];
4         for (size_t i = 0; i < N; ++i) {
5             newVertices[i] = vertices[i];
6         }
7         newVertices[N] = vertex;
8         return Polygon<T, N + 1>(newVertices);
9     }
10
11     template<int M = N>
12     typename std::enable_if<M == 3, double>::type
13     area() const {
14         double a = distance(vertices[0], vertices[1]);
15         double b = distance(vertices[1], vertices[2]);
16         double c = distance(vertices[2], vertices[0]);
17         double s = (a + b + c) / 2.0;
18         return sqrt(s * (s - a) * (s - b) * (s - c));
19     }
20
21     double distance(const std::pair<T, T>& p1, const std::pair<T, T>&
22     p2) const {
23         return sqrt(pow(p2.first - p1.first, 2) + pow(p2.second - p1.
24     second, 2));
25     }
26
27     };
28
29     #endif
```

### Листинг 3 — Функция main для проверки работы класса Polygon

```
1      #include <iostream>
2      #include "declaration.h"
3
4      int main() {
5          std::pair<double, double> vertices[3] = { {0.0, 0.0}, {3.0,
6              0.0}, {0.0, 4.0} };
7          Polygon<double, 3> triangle(vertices);
8
9          std::cout << "Perimeter of the triangle: " << triangle.perimeter
10             () << std::endl;
11          std::cout << "Area of the triangle: " << triangle.area() << std
12             ::endl;
13
14          std::pair<double, double> newVertex(2.0, 2.0);
15          Polygon<double, 4> updatedTriangle = triangle.addVertex(
16              newVertex);
17
18          std::cout << "Perimeter of the updated triangle: " <<
19              updatedTriangle.perimeter() << std::endl;
20
21          return 0;
22      }
```

Результат запуска представлен на рисунках 1.



```
Perimeter of the triangle: 12
Area of the triangle: 6
Perimeter of the updated triangle: 13.6569
```

Рис. 1 — Результат