



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 № 7
по курсу «Языки и методы программирования»
«Разработка простейшего класса на C++»
Вариант 25

Студент группы ИУ9-21Б Шиятов Н.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Необходимо разработать на языке Java класс ломаной линии с операциями:

1. получение количества точек;
2. получение ссылки на i -тую точку;
3. конкатенация двух ломаных (формируется новая ломаная);
4. выделение части ломаной с i -той по j -тую точки (формируется новая ломаная).

Точки на плоскости должны быть представлены структурами с целочисленными полями x и y .

2 Результаты

Исходный код программы представлен в листингах 1– 3.

Результат запуска представлен на рисунке 1.

Листинг 1 — Объявление класса Polyline

```
1 #ifndef DECLARATION_H
2 #define DECLARATION_H
3
4 #include <vector>
5
6 using namespace std;
7
8 struct Point {
9     int x;
10    int y;
11 };
12
13 class Polyline {
14 public:
15     Polyline();
16     Polyline(const vector<Point>& points);
17     int size() const;
18     Point& operator[] (int index);
19     const Point& operator[] (int index) const;
20     Polyline operator+(const Polyline& other) const;
21     Polyline subPolyline(int i, int j) const;
22
23 private:
24     vector<Point> m_points;
25 };
26
27 #endif
```

Листинг 2 — Определения методов класса

```
1 #include "declaration.h"
2
3 using namespace std;
4
5 Polyline::Polyline(const vector<Point>& points) : m_points(points) {}
6
7 int Polyline::size() const {
8     return m_points.size();
9 }
10
11 Point& Polyline::operator[](int index) {
12     return m_points[index];
13 }
14
15 const Point& Polyline::operator[](int index) const {
16     return m_points[index];
17 }
18
19 Polyline Polyline::operator+(const Polyline& other) const {
20     vector<Point> new_points(m_points);
21     new_points.insert(new_points.end(), other.m_points.begin(), other.
22 m_points.end());
23     return Polyline(new_points);
24 }
25
26 Polyline Polyline::subPolyline(int i, int j) const {
27     return Polyline(vector<Point>(m_points.begin() + i, m_points.begin()
28 + j + 1));
29 }
```

Листинг 3 — Проверка работоспособности

```
1 #include <iostream>
2 #include "declaration.h"
3
4 using namespace std;
5
6 int main() {
7     vector<Point> points1 = {{1, 2}, {3, 4}, {5, 6}};
8     vector<Point> points2 = {{7, 8}, {9, 10}};
9
10     Polyline p1(points1);
11     Polyline p2(points2);
12
13     cout << "p1.size() = " << p1.size() << std::endl;
14     cout << "p1[1].x = " << p1[1].x << ", p1[1].y = " << p1[1].y << std
15 ::endl;
16
17     Polyline p3 = p1 + p2;
18     cout << "p3.size() = " << p3.size() << std::endl;
19
20     Polyline p4 = p1.subPolyline(1, 2);
21     cout << "p4.size() = " << p4.size() << std::endl;
22
23     return 0;
24 }
```

```
nailorsh@nail-IdeaPad-L340-15IRH-Gaming:/media/hdc  
emester/PLaM/CPP/lab7/SimpleClass$ ./program  
p1.size() = 3  
p1[1].x = 3, p1[1].y = 4  
p3.size() = 5  
p4.size() = 2
```

Рис. 1 — Результат