

**Министр науки и высшего образования Российской
Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Лабораторная работа №11

Использование заголовочных файлов

Выполнила студентка группы № М3119

Самигуллин Руслан Рустамович

Подпись:

Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2023

1. Заголовочный файл “geometry.h”

```
#ifndef GEOMETRY_H
#define GEOMETRY_H

struct Point {
    double x;
    double y;
};

struct GeometricFigure {
    struct Point vertex1;
    struct Point vertex2;
    struct Point vertex3;
};

double area(const struct GeometricFigure* figure);
double perimeter(const struct GeometricFigure* figure);
void create(struct GeometricFigure* figure);

#endif // GEOMETRY_H
```

2. Файл “geometry.c”

```
#include "geomerty.h"
#include <stdio.h>
#include <math.h>

double distance(const struct Point* p1, const struct Point* p2) {
    return sqrt(pow(p2->x - p1->x, 2) + pow(p2->y - p1->y, 2));
}

double area(const struct GeometricFigure* figure) {
    double a = distance(&(figure->vertex1), &(figure->vertex2));
    double b = distance(&(figure->vertex2), &(figure->vertex3));
    double c = distance(&(figure->vertex3), &(figure->vertex1));

    double s = (a + b + c) / 2.0; // полупериметр

    return sqrt(s * (s - a) * (s - b) * (s - c)); // формула Герона
}

double perimeter(const struct GeometricFigure* figure) {
    double a = distance(&(figure->vertex1), &(figure->vertex2));
    double b = distance(&(figure->vertex2), &(figure->vertex3));
    double c = distance(&(figure->vertex3), &(figure->vertex1));
    return a + b + c;
}

void create(struct GeometricFigure* figure) {
    printf("Введите координаты вершин треугольника (x1 y1 x2 y2 x3 y3): ");
    scanf("%lf %lf %lf %lf %lf %lf",
        &(figure->vertex1.x), &(figure->vertex1.y),
        &(figure->vertex2.x), &(figure->vertex2.y),
        &(figure->vertex3.x), &(figure->vertex3.y));
}
```

3. Файл “main.c”

```
#include <stdio.h>
#include "geomerty.h"

int main() {
    struct GeometricFigure triangle;

    create(&triangle);

    printf("Площадь треугольника: %lf\n", area(&triangle));
    printf("Периметр треугольника: %lf\n", perimeter(&triangle));

    return 0;
}
```

1. Заголовочный файл "geometry.h"

1. Создан структура Point, где находятся координаты X и Y одной точки.
Создан структура GeometricFigure, где находятся вершины.
2. 'double area', 'double perimeter', 'void create' – функции, которые принимают указатели на структуру GeometricFigure, вычисляют, возвращаются определенные результаты.
'double area' – вычисляет площадь произвольного треугольника
'double perimeter' – вычисляет периметр произвольного треугольника
'void create' – заполняет её значениями, полученными от пользователя через консоль. Она используется для ввода координат вершин фигуры.
3. Директивы препроцессора *'#ifndef'*, *'#define'*, *'#endif'* используются для создания механизма защиты от множественного включения заголовочных файлов.

2. Файл "geometry.c"

1. Подключены заголовочные файлы "geomerty.h", "stdio.h", "math.h".
2. 'double distance(...)' – функция, которая принимает два указателя и возвращает расстояние между точками.
3. 'double area(...)' – функция, которая принимает указатель на структуру GeometricFigure и возвращает площадь произвольного треугольника по формуле Герона.
4. 'double perimeter(...)' – функция, которая принимает указатель на структуру GeometricFigure и возвращает периметр треугольника.
5. 'void create(...)' – функция, которая принимает указатель на структуру GeometricFigure и заполняет ее значениями, полученными от пользователя консолью.

3. Файл "main.c"

1. Подключены заголовочные файлы <stdio.h>, "geometry.h".
2. 'int main()' – функция, где находится весь код.
3. 'struct GeometricFigure triangle' – создает новый элемент структуры, т.е. треугольник.
4. 'create(&triangle)' – вызывает функцию, которая запрашивает пользователя ввести координаты вершин произвольного треугольника.
5. 'printf("Площадь треугольника: %lf\n", area(&triangle))' – выводит площадь произвольного треугольника, используя функцию 'area(&triangle)'
6. 'printf("Периметр треугольника: %lf\n", perimeter(&triangle))' – выводит периметр произвольного треугольника, используя функцию 'perimeter(&triangle)'
7. 'return 0' – строка завершает функцию 'main' и возвращает 0 в качестве завершения программы.

INPUT:

Введите координаты вершин треугольника (x1 y1 x2 y2 x3 y3): 2 2 4 6 6 3

OUTPUT:

Площадь треугольника: 7.000000

Периметр треугольника: 12.200793