

**Министерство 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

Выполнил студент группы № М3109

Бабич Артём Антонович

Подпись:



Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2020

Текст задания для 1 варианта

Задания:

Разработать структуру, описывающую геометрическую фигуру на плоскости с помощью её координат, а также функции расчёта площади этой геометрической фигуры и её периметра (длины окружности). Дополнительно разработать функцию, позволяющую создавать структуру моделируемой геометрической фигуры на основании данных, введённых с консоли (самостоятельно определить формат этих вводимых данных). Вид геометрической фигуры и набор её рассчитываемых параметров определяются по номеру варианта.

Разместить программный код функций, рассчитывающих параметры фигуры в отдельном файле исходного текста. Создать заголовочный файл, позволяющий вызывать функции созданного файла исходного текста из других файлов. Разместить в созданном заголовочном файле структуру, описывающую моделируемую геометрическую фигуру. Предусмотреть в созданном заголовочном файле защиту от повторного включения с помощью Include Guard.

Вариант геометрической фигуры: **прямоугольник**.

Решение с комментариями

```
// файл main.c, содержит функцию main, в которой запрашиваются с консоли
координаты
// и на экран площадь и периметр прямоугольника

#include <stdio.h>
#include <locale.h>
#include "rectangle.h"

int main() {
    setlocale(LC_ALL, "Rus");

    struct Rectangle rect;
    printf("Введите координаты первой точки через пробел: ");
    scanf("%lf %lf", &rect.x1, &rect.y1);

    printf("Введите координаты второй точки через пробел: ");
    scanf("%lf %lf", &rect.x2, &rect.y2);

    printf("Введите координаты третьей точки через пробел: ");
    scanf("%lf %lf", &rect.x3, &rect.y3);

    printf("Периметр = %.2lf \n", Perimeter(rect));
    printf("Площадь = %.2lf \n", Area(rect));

    return 0;
}
```

```
// файл "func.c", содержит функции для расчета периметра и площади

#include <math.h>
#include "rectangle.h"

double Perimeter(struct Rectangle rect){
    return 2 * (sqrt(pow(rect.x2 - rect.x1, 2) + pow(rect.y2 - rect.y1, 2)) +
sqrt(pow(rect.x3 -
    rect.x1, 2) + pow(rect.y3 - rect.y1, 2)));
}

double Area(struct Rectangle rect){
    return sqrt(pow(rect.x2 - rect.x1, 2) + pow(rect.y2 - rect.y1, 2)) *
sqrt(pow(rect.x3 - rect.x1, 2)
    + pow(rect.y3 - rect.y1, 2));
}
```

```
// заголовочный файл "rectangle.h", содержит структуру координат
прямоугольника
// и прототипы функций

#ifndef LAB11_RECTANGLE_H
#define LAB11_RECTANGLE_H

struct Rectangle{
    double x1, y1;
    double x2, y2;
    double x3, y3;
};

double Perimeter(struct Rectangle);
double Area(struct Rectangle);

#endif //LAB11_RECTANGLE_H
```

Пошаговое решение:

1. Создаем три файла: “*main.c*” – файл исполняемого кода самой программы, “*rectangle.h*” – заголовочный файл, “*func.c*” – файл исполняемого кода, предназначенный для записи в него функций.
2. В заголовочном файле включаем защиту от повторного включения, прописываем структуру координат прямоугольника и прототипы функций для расчета периметра и площади прямоугольника (получают на вход структуру).
3. В файле “*func.c*” подключаем библиотеку $\langle math.h \rangle$ для математических расчетов и записываем функцию для расчета периметра прямоугольника: сумма длин векторов (находится, поскольку на вход мы получаем координаты, а не длины сторон) умноженная на 2;
Также пишем функцию для расчета площади: произведение длин векторов (сторон прямоугольника);
4. В главном файле исходного кода подключаем библиотеку ввода-вывода, языковую библиотеку и заголовочный файл;
5. Запрашиваем координаты и выводим на экран периметр и площадь прямоугольника;