

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

Функции

Вариант 8

Выполнил студент группы № М3111

Гонтарь Тимур Сергеевич

Подпись:



Санкт-Петербург
2022

Условие ЛР:

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.

Создать файл исходного текста для размещения в нём точки входа. Разработать программный код функции main, создающий экземпляр структуры моделируемой фигуры, а также осуществляющий вызовы созданных функций расчёта параметров этой фигуры. Входные данные для моделирования фигуры вводить с консоли, результаты расчёта параметров фигуры также выводить на консоль.

Решение с комментариями:

Мой вариант №8 – Прямоугольный треугольник

Я использовал 3 файла – один заголовочный с структурами и объявлением функций, другой с функциями и основной, который я использую для ввода данных и вызова всех функций.

В gho1.h я объявляю структуру точки, структуру треугольника а также нужные для расчёта функции. В raschet у меня находятся сами функции, для расчёта длины отрезка, для периметра, площади а так же для преобразования входных данных в поля структуры. В main я ввожу координаты треугольника с помощью клавиатуры, создаю экземпляр структуры и заполняю его введёнными данными с помощью соответствующих функций. Затем я таким же образом рассчитываю нужные параметры фигуры – периметр и площадь.

```
CMakeLists.txt x main.c x raschet.c x ghou.h x
1 #ifndef ELEVENTH_GHOUL_H
2 #define ELEVENTH_GHOUL_H
3
4 struct Point {
5     double x;
6     double y;
7 };
8
9 struct Triangle {
10     struct Point p1;
11     struct Point p2;
12     struct Point p3;
13     double len1;
14     double len2;
15     double perimeter;
16     double square;
17 };
18
19 double length(struct Point p1, struct Point p2);
20 double trperimetr(struct Triangle tr);
21 double trsquare(struct Triangle tr);
22 void triangalise(double arr[], struct Triangle *tr);
23
24 #endif //ELEVENTH_GHOUL_H
25

CMakeLists.txt x main.c x raschet.c x ghou.h x
1 #include "stdio.h"
2 #include "locale.h"
3 #include "ghoul.h"
4
5
6 int main() {
7     setlocale( Category: LC_ALL, Locale: "Russian");
8     double points[6];
9     for (int i = 0; i < 6; i++) {
10         scanf( format: "%lf", &points[i]);
11     }
12
13     struct Triangle tr1;
14     triangalise( arr: points, tr: &tr1);
15     tr1.len1 = length( p1: tr1.p1, p2: tr1.p2);
16     tr1.len2 = length( p1: tr1.p3, p2: tr1.p2);
17     tr1.perimetr = trperimetr( tr: tr1);
18     tr1.square = trsquare( tr: tr1);
19
20     printf( format: "Периметр равен: %lf\n", tr1.perimetr);
21     printf( format: "Площадь равна: %lf", tr1.square);
22
23
24     return 0;
25 }
26
```

```
CMakelists.txt x main.c x raschet.c x ghouh.h x
1 #include "ghoul.h"
2 #include "math.h"
3
4 double length(struct Point p1, struct Point p2) {
5     double ans = sqrt( pow( (p1.x - p2.x), 2) + pow( (p2.y - p1.y), 2));
6     return ans;
7 }
8
9 double trperimetr(struct Triangle tr) {
10     double len3 = sqrt( pow( tr.len1, 2) + pow( tr.len2, 2));
11     double p = tr.len1 + tr.len2 + len3;
12     return p;
13 }
14
15 double trsquare(struct Triangle tr) {
16     double s = (tr.len1 * tr.len2) / 2.0;
17     return s;
18 }
19
20 void triangalise(double arr[], struct Triangle *tr) {
21     tr->p1.x = arr[0];
22     tr->p1.y = arr[1];
23     tr->p2.x = arr[2];
24     tr->p2.y = arr[3];
25     tr->p3.x = arr[4];
26     tr->p3.y = arr[5];
27 }
```

Вывод: В ходе данной лабораторной работы я хранил все функции и структуры программы не в 1 файле как обычно, а в 3 разных файлах – один заголовочный, один для функций и один для их вызова. Так же я использовал защиту от повторного включения, чтобы 1 структура не создавалась несколько раз.