Практическая работа №7 Асылбек уулу Бакыта ИСП-308

Тема: «Структурное тестирование. Тестирование «белым ящиком»»

Цель работы: изучить метод тестирования «Белым ящиком»

1. Разработка программы, для определения площади и вида треугольника по его сторонам.

#include <stdio.h>

#include <math.h>

int main()

{

int a = 0, b = 0, c = 0, p = 0, s = 0;

printf("Введите длины сторон треугольника: ");

scanf("%i%i%i", &a, &b, &c);

if ((a > 0) && (b > 0) && (c > 0))

{

if (((a + b) > c) && ((a + c) > b) && ((b + c) > a))

{

p = (a + b + c)/2;

s = sqrt(p \* (p - a) \* (p - b) \* (p - c));

printf("\nПолупериметр %i\nПлощадь %i", p, s);

if ((a == b) && (a == c) && (b == c))

{

printf("\nТреугольник равносторонний");

}

else

{

if (((a != b) && (a == c) && (b != c)) || ((a != c) && (b == c) && (b != a)) || ((b != c) && (a == b) && (c != a)))

{

printf("\nТреугольник равнобедренный");

}

else

{

if ((a != b) && (a != c) && (b != c))

{

printf("\nТреугольник разносторонний");

}

}

}

}

else printf("Сумма двух сторон не должно быть меньше третьего");

}

else

{

printf("Длина сторон треугольника должны быть больше 0");

}

return 0;

}

1. Набор тестовых вариантов для обнаружения ошибок в программе.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | Ожидаемый результат | Объект проверки |
| -1 | 5 | -5 | Длина сторон треугольника должны быть больше 0 | Ожидаемый результат |
| 0 | 0 | 0 | Длина сторон треугольника должны быть больше 0 | Ожидаемый результат |
| 1 | 2 | 6 | Сумма двух сторон должно быть больше третьего | Ожидаемый результат |
| 5 | 5 | 5 | Треугольник равносторонний | Ожидаемый результат |
| 5 | 5 | 5 | Площадь равна 10,825 (7) | Ожидаемый результат |
| 4 | 5 | 5 | Треугольник равнобедренный | Ожидаемый результат |
| 4 | 5 | 5 | Площадь равна 9,165 (9) | Ожидаемый результат |
| 5 | 4 | 5 | Треугольник равнобедренный | Ожидаемый результат |
| 5 | 4 | 5 | Площадь равна 9,165 (9) | Ожидаемый результат |
| 5 | 5 | 4 | Треугольник равнобедренный | Ожидаемый результат |
| 5 | 5 | 4 | Площадь равна 9,165 (9) | Ожидаемый результат |
| 5 | 6 | 7 | Треугольник разносторонний | Ожидаемый результат |
| 5 | 6 | 7 | Площадь равна 14,697 (14) | Ожидаемый результат |

1. Разработать программу для определения площади и вида треугольника по его углам.

#include <stdio.h>

#include <math.h>

int main()

{

int a = 0, b = 0, c = 0, p = 0, s = 0;

printf("Введите длины сторон треугольника: ");

scanf("%i%i%i", &a, &b, &c);

if ((a > 0) && (b > 0) && (c > 0))

{

if (((a + b) > c) && ((a + c) > b) && ((b + c) > a))

{

p = (a + b + c) / 2;

s = sqrt(p \* (p - a) \* (p - b) \* (p - c));

printf("\nПолупериметр %i\nПлощадь %i", p, s);

if ((a > b) && (a > c))

{

if (pow(a, 2) > (pow(b, 2) + pow(c, 2))) printf("\nТреугольник тупоугольный");

else

{

if (pow(a, 2) == (pow(b, 2) + pow(c, 2))) printf("\nТреугольник прямоугольный");

else

{

if (pow(a, 2) < (pow(b, 2) + pow(c, 2))) printf("\nТреугольник остроугольный");

}

}

}

else

{

if ((b > a) && (b > c))

{

if (pow(b, 2) > (pow(a, 2) + pow(c, 2))) printf("\nТреугольник тупоугольный");

else

{

if (pow(b, 2) == (pow(a, 2) + pow(c, 2))) printf("\nТреугольник прямоугольный");

else

{

if (pow(b, 2) < (pow(a, 2) + pow(c, 2))) printf("\nТреугольник остроугольный");

}

}

}

else

{

if ((c > a) && (c > b))

{

if (pow(c, 2) > (pow(b, 2) + pow(a, 2))) printf("\nТреугольник тупоугольный");

else

{

if (pow(c, 2) == (pow(b, 2) + pow(a, 2))) printf("\nТреугольник прямоугольный");

else

{

if (pow(c, 2) < (pow(b, 2) + pow(a, 2))) printf("\nТреугольник остроугольный");

}

}

}

}

}

}

else printf("Сумма двух сторон не должно быть меньше третьего");

}

else printf("Длина сторон треугольника должны быть больше 0");

return 0;

}

Набор тестовых данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | Ожидаемый результат | Объект проверки |
| 1 | 5 | -5 | Длина сторон треугольника должны быть больше 0 | Ожидаемый результат |
| 0 | 0 | 0 | Длина сторон треугольника должны быть больше 0 | Ожидаемый результат |
| 1 | 2 | 5 | Сумма двух сторон не должно быть меньше третьего | Ожидаемый результат |
| 6 | 6 | 6 | Сумма двух сторон не должно быть меньше третьего | Ожидаемый результат |
| 6 | 6 | 6 | Площадь равна 15,59 (15) | Ожидаемый результат |
| 3 | 4 | 3 | Треугольник остроугольный | Ожидаемый результат |
| 3 | 4 | 3 | Площадь равна 4,47 (4) | Ожидаемый результат |
| 3 | 4 | 5 | Треугольник прямоугольный | Ожидаемый результат |
| 3 | 4 | 5 | Площадь равна 6 (6) | Ожидаемый результат |
| 3 | 4 | 6 | Треугольник тупоугольный | Ожидаемый результат |
| 3 | 4 | 6 | Площадь равна 5,33 (0) | Ожидаемый результат |

1. Рекомендации по исправлению ошибок, выявленных в ходе тестирования.

2 задание.

В ходе выполнения тестов 4, 6, 8, 10, 12 было обнаружено, что площадь вычисляется неточно.

Рекомендация: Поменять тип данных переменных, чтобы программа вычисляла точно и без ошибок.

3 задание.

В ходе выполнения 4 теста было обнаружено, что можно вводить 3 одинаковых числа, хоть и присутствует условие «Сумма двух сторон больше третьего».

Рекомендация: Дополнить условие так, чтобы программа следовало необходимым ограничениям.

В ходе выполнения

В ходе выполнения тестов 5, 7 было обнаружено, что площадь вычисляется неточно.

Рекомендация: Поменять тип данных переменных, чтобы программа вычисляла точно и без ошибок.

В ходе выполнения 11 теста было обнаружено, что площадь треугольника вычисляется неправильно.

Рекомендация: Поменять тип данных переменных, чтобы программа вычисляла точно и без ошибок.