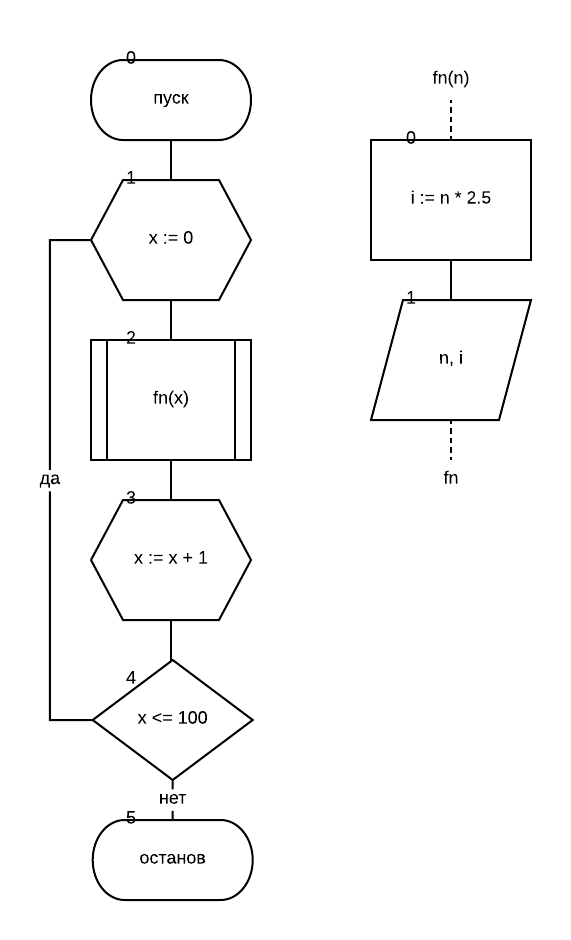
# Лабораторная работа № 5

## 1 задание

1. Вычисление определенного интеграла.
2. Научится реализовать алгоритм для вычисление определенного интеграла средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. Перевести дюймы в сантиметры от 0 до 100 дюймов (1 дюйм=2.5 см). Результаты вывести в виде таблицы. Операторы для формирования вывода таблицы оформить в виде пользовательской процедуры.
5. .
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i | Сантиметры в функции | real |
| n | Дюймы в функции | integer |
| x | Сантиметры | integer |

**program** pr5;

**function** fn(n: integer): real;

**var** i: real;

**begin**

i := n \* 2.5;

writeln(n, ' ', i)

**end**;

**var** x:integer;

**begin**

writeln('in cm');

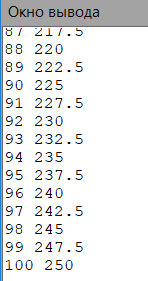
**for** x:= 0 **to** 100 **do**

**begin**

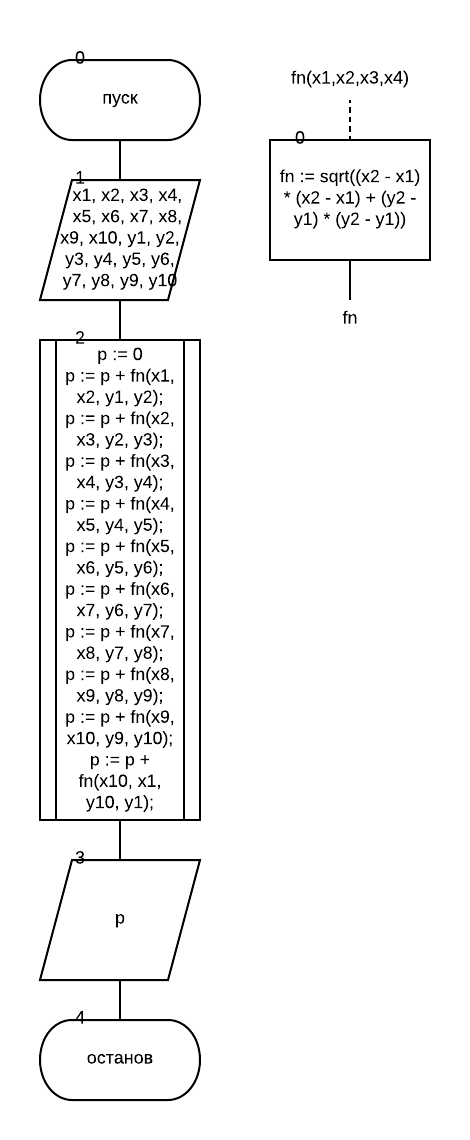
fn(x)

**end**

**end**.

1. 
2. Операторы для формирования вывода таблицы оформил в виде пользовательской процедуры.
3. Перевёл дюймы в сантиметры от 0 до 100 дюймов (1 дюйм=2.5 см). Результаты вывел в виде таблицы.

## 2 задание

1. Вычисление определенного интеграла.
2. Научится реализовать алгоритм для вычисление определенного интеграла средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. Даны действительные числа x1, y1, x2, y2, ..., x10, y10. Найти периметр десятиугольника, вершины которого имеют соответственно координаты (x1, y1), (x2, y2), ..., (x10, y10). (Определить функцию вычисления расстояния между двумя точками, заданными своими координатами.)
5. .
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, | Координаты | real |
| p | Периметр фигуры | real |

**program** pr5;

**var** x1, x2, x3, x4, x5, x6, x7, x8, x9, x10,

y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, p: real;

**function** fn(x1, x2, y1, y2: real): real;

**begin**

fn := sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

**end**;

**begin**

readln(x1, x2, x3, x4, x5, x6, x7, x8, x9, x10,

y1, y2, y3, y4, y5, y6, y7, y8, y9, y10);

p := 0;

p := p + fn(x1, x2, y1, y2);

p := p + fn(x2, x3, y2, y3);

p := p + fn(x3, x4, y3, y4);

p := p + fn(x4, x5, y4, y5);

p := p + fn(x5, x6, y5, y6);

p := p + fn(x6, x7, y6, y7);

p := p + fn(x7, x8, y7, y8);

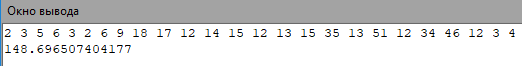
p := p + fn(x8, x9, y8, y9);

p := p + fn(x9, x10, y9, y10);

p := p + fn(x10, x1, y10, y1);

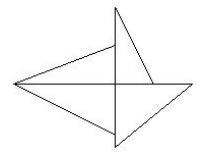
writeln(p)

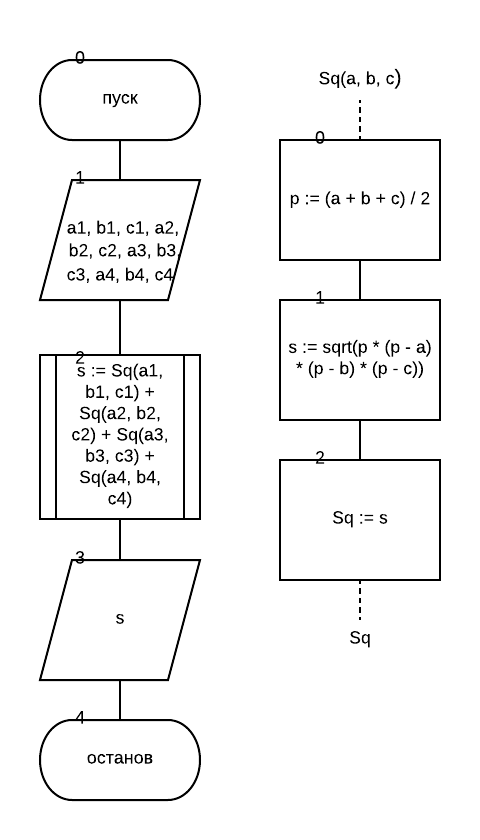
**end**.

1. 
2. В функции считается длина отрезка.
3. Даны действительные числа x1, y1, x2, y2, ..., x10, y10. Нашёл периметр десятиугольника, вершины которого имеют соответственно координаты (x1, y1), (x2, y2), ..., (x10, y10).

## 3 задание

1. Вычисление определенного интеграла.
2. Научится реализовать алгоритм для вычисление определенного интеграла средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. Вычислить площадь фигуры, заданной сторонами. Фигура не является прямоугольником, а треугольники, которые ее составляют, не являются прямоугольными.



1. .
2. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| a1, b1, c1, a2, b2, c2, a3, b3, c3, a4, b4, c4, a, b, c | Длины сторон треугольников | real |
| s | Площадь фигуры | real |

**program** pr5;

**var**

a1, b1, c1, a2, b2, c2, a3, b3, c3, a4, b4, c4, s: real;

**function** Sq(a, b, c: real): real;

**var** p, s: real;

**begin**

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

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

Sq := s

**end**;

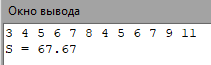
**begin**

readln(a1, b1, c1, a2, b2, c2, a3, b3, c3, a4, b4, c4);

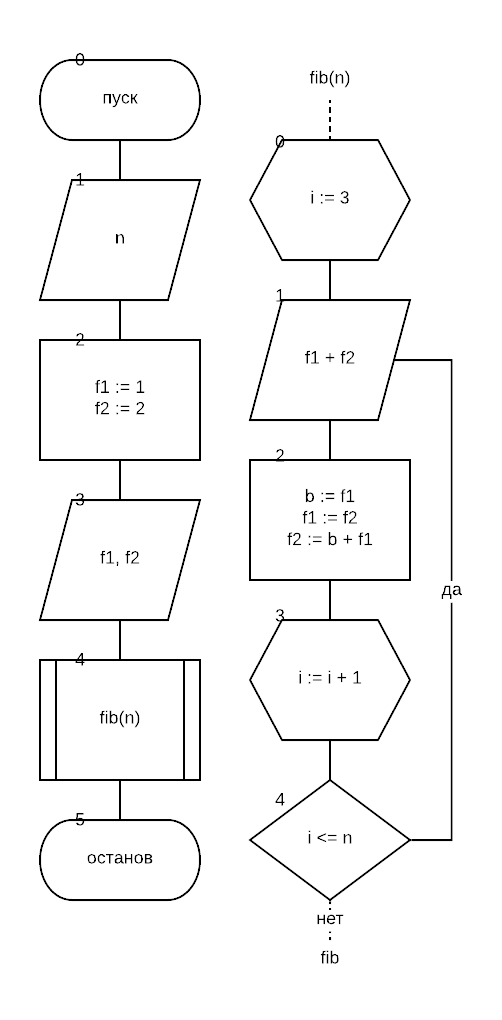
s := Sq(a1, b1, c1) + Sq(a2, b2, c2) + Sq(a3, b3, c3) + Sq(a4, b4, c4);

write('S = ', s:0:2);

**end**.

1. 
2. В функции высчитывается площадь по формуле Герона.
3. Вычислил площадь фигуры, заданной сторонами.

## 4 задание

1. Вычисление определенного интеграла.
2. Научится реализовать алгоритм для вычисление определенного интеграла средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. С клавиатуры вводится число. Вывести на экран столько элементов ряда Фибоначчи, сколько указал пользователь. Вычисление ряда организовать в функцию. Например, если на ввод поступило число 6, то вывод должен содержать шесть первых чисел ряда Фибоначчи: 1 2 3 5 8 13.
5. .
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i | Счётчик цикла | integer |
| n | Кол-во чисел Фибоначчи, кот. нужно вывести | integer |
| f1 | Значение предпоследнего элемента ряда | integer |
| f2 | Значение последнего элемента ряда | integer |
| b | Запасная переменная | integer |

**program** pr5;

**var** f1, f2, n: integer;

**function** fib(n: integer): integer;

**var** i, b: integer;

**begin**

**for** i := 3 **to** n **do**

**begin**

write(f1 + f2, ' ');

b := f1;

f1 := f2;

f2 := b + f1

**end**

**end**;

**begin**

readln(n);

f1 := 1; f2 := 2;

write(f1, ' ', f2, ' ');

fib(n)

**end**.

1. 
2. Для правильного вычисления пришлось весь цикл убрать в функцию.
3. Вывел на экран столько элементов ряда Фибоначчи, сколько указал пользователь.