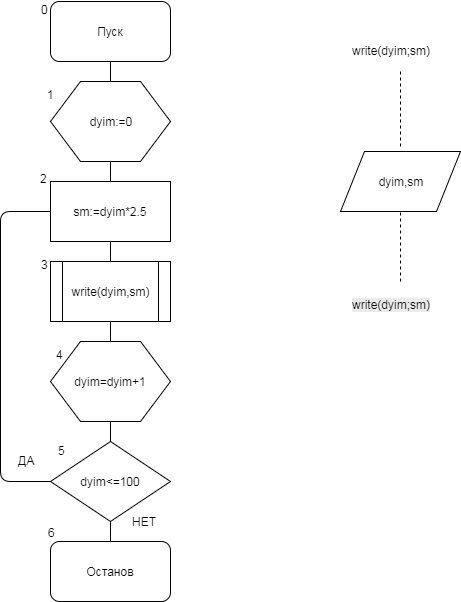
**Лабораторная работа №5**

1. Тема: Процедуры и функции.
2. Цель: научиться реализовывать вычисления при помощи процедур и функций на языке Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.

**Задание 1:**

1. Постановка задачи: Перевести дюймы в сантиметры от 0 до 100 дюймов. Результаты вывести в виде таблицы. Операторы для формирования вывода таблицы оформить в виде пользовательской функции.
2. Математическая модель: 1 дюйм=2.5см
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| Dyim | Integer | Дюймы |
| Sm | Real | Сантиметры |
| write | real | Процедура |

1. Код программы:

program zadanie1;

var sm:real;

dyim:integer;

procedure write(dyim:integer; sm:real);

begin

writeln('dyimy=',dyim, ' sm=',sm:2:1);

end;

begin

for dyim:=0 to 100 do

begin

sm:=dyim\*2.5;

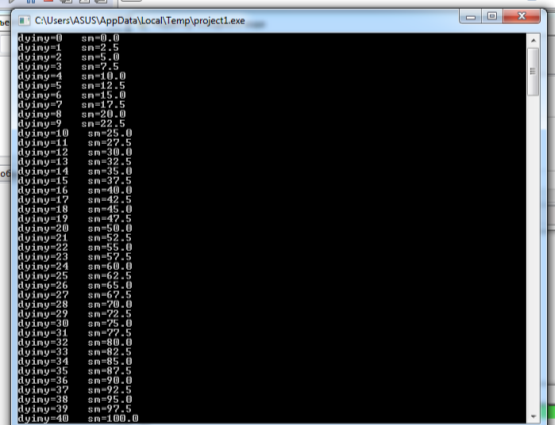
write(dyim,sm);

end;

readln();

end.

1. Результат:

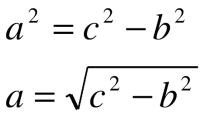


1. Анализ результатов: Программа переводит дюймы от 0 до 100 в сантиметры и выводит соответствующие значения на экран.

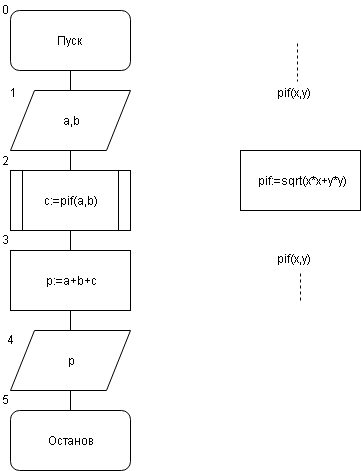
**Задание 2:**

1. Постановка задачи: Напишите программу, которая по двум катетам вычисляет периметр прямоугольного треугольника.
2. Математическая модель:

Теорема Пифагора



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Real | Катет |
| b | Real | Катет |
| c | Real | Гипотенуза |
| p | Real | Периметр |
| Pif | Real | Функция |
| x, y | real | Переменные функции |

1. Код программы:

program zadanie2;

var a,b,c,p: real;

function pif(x,y:real):real;

begin

pif:=sqrt(x\*x+y\*y);

end;

begin

writeln('vvedite a, b');

readln (a,b);

c:=pif(a,b);

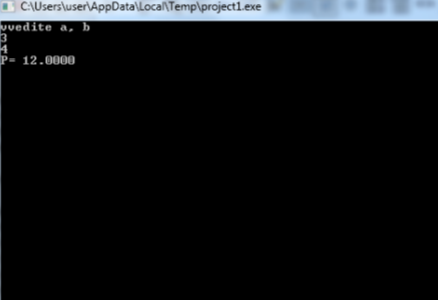
p:=a+b+c;

writeln('P= ',p:5:4);

readln();

end.

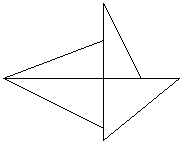
1. Результат:



1. Анализ результатов: На вход поступают значения двух катетов. Вычисление гипотенузы организовано в функцию. Программа выводит на экран периметр прямоугольного треугольника.

**Задание 3:**

1. Постановка задачи: Вычислить площадь фигуры, заданной сторонами. Фигура не является прямоугольником, а треугольники, которые ее составляют, не являются прямоугольными.

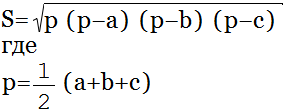


1. Математическая модель:

Длины сторон каждого треугольника находятся по формуле:

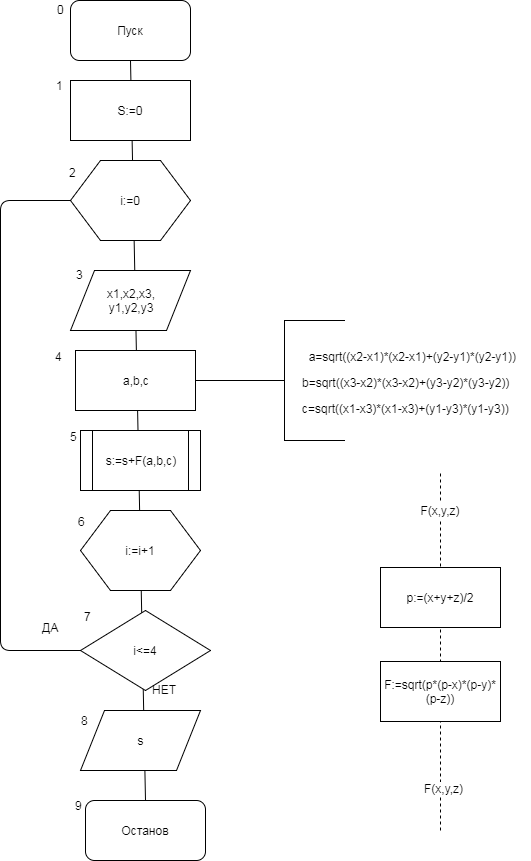


Площадь каждого треугольника находится по формуле Герона:



Найденные площади суммируются.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| S | Real | Площадь |
| F | Real | Функция |
| i | Integer | Номер треугольника |
| x1,x2,x3,y1,y2,y3 | Real | Координаты вершин |
| a,b,c | Real | Стороны треугольников |
| x,y,z | Real | Переменные функции |
| p | real | Полупериметр |

1. Код программы:

program zadanie3;

var x1,x2,x3,y1,y2,y3,a,b,c,p,s:real;

i:integer;

function F(x,y,z:real):real;

begin

p:=(x+y+z)/2;

F:=sqrt(p\*(p-x)\*(p-y)\*(p-z));

writeln('S=', F:5:4);

end;

begin

s:=0;

for i:=1 to 4 do

begin

writeln ('treygolnik ', i);

writeln ('vvedite koordinaty vershin');

readln (x1,y1,x2,y2,x3,y3);

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

b:=sqrt((x3-x2)\*(x3-x2)+(y3-y2)\*(y3-y2));

c:=sqrt((x1-x3)\*(x1-x3)+(y1-y3)\*(y1-y3));

s:=s+F(a,b,c);

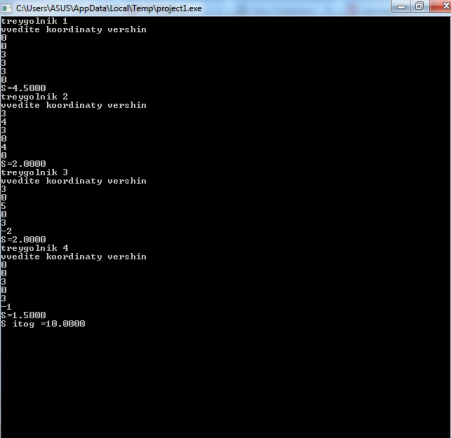
end;

writeln ('S itog =',s:5:4);

readln();

end.

1. Результат:



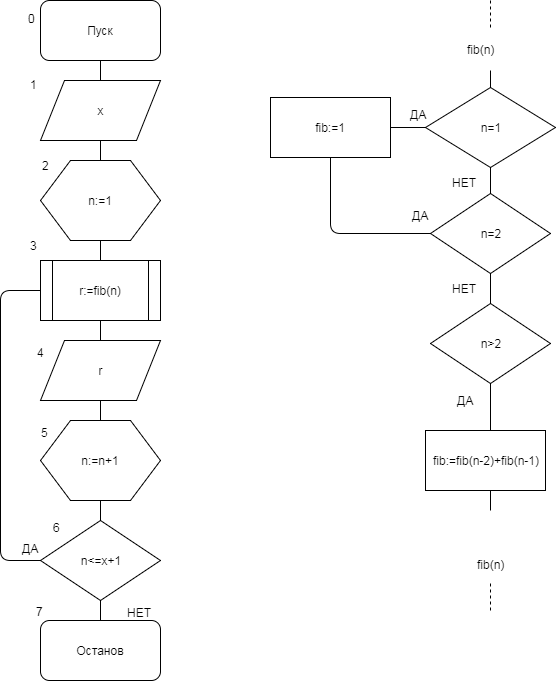
1. Анализ результатов: На ввод поступают координаты вершин каждого из 4 данных треугольников. Программа высчитывает длины сторон треугольников, их полупериметры и площади. Полученные площади складываются и на экран выводится конечный результат – площадь заданной фигуры.

**Задание 4:**

1. Постановка задачи: 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.
2. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| x | integer | Вводимое число |
| n | Integer | Переменная функции |
| r | Integer | Результат |
| fib | longint | Функция |

1. Код программы:

program zadanie4;

var x,n,r: integer;

function fib(n:integer):longint;

begin

if n=1 then fib:=1;

if n=2 then fib:=1;

if n>2 then fib:=fib(n-2)+fib(n-1);

end;

begin

writeln('vvedite chislo');

readln(x);

for n:=1 to x+1 do begin

r:=fib(n);

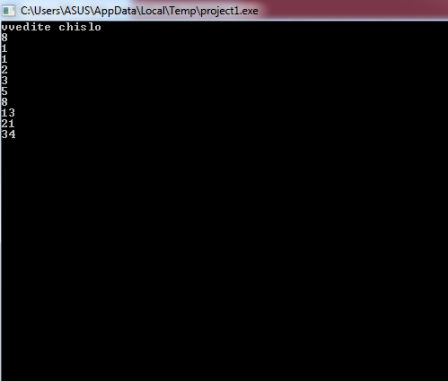
writeln (r);

end;

readln();

end.

1. Результат:



1. Анализ результатов: Программа выводит на экран заданное количество элементов ряда Фибоначчи.