**Тема лабораторной работы:** Детерминированные циклические вычислительные процессы с управлением по аргументу

**Цель л/р:**

Реализовать алгоритмы ДЦВП по аргументу с помощью Pascal

**Используемое оборудование:** draw.io(блок-схемы), PascalABC(код программы)

**Задача 1**

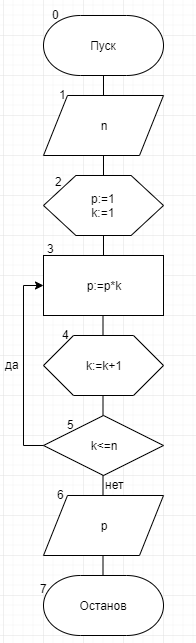
**Постановка задачи:**

Вычислить n!, где n вводится с клавиатуры.

**Математическая модель:**

p(n)=1\*2\*3\*…\*n

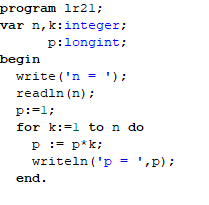
**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| n | Вводимая переменная | integer |
| k | Параметр цикла | integer |
| p | Накопительная переменная | longint |

**Код программы:**



**Результаты выполненной работы:**



**Анализ результатов вычисления:**

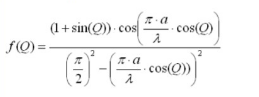
Чтобы найти факториал умножаем накопительную переменную на параметр цикла, который на каждом цикле увеличивается на единицу. Цикл завершится когда параметр цикла будет больше вводимой нами переменной на единицу.

**Задача 2**

**Постановка задачи:**

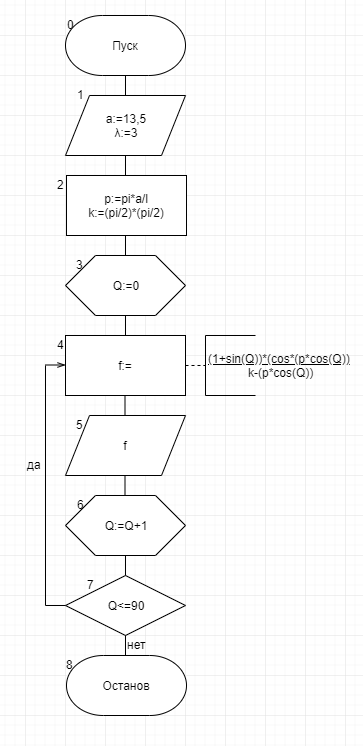
Рассчитать значения для построения диаграммы направленности антенны в вертикальной плоскости.Q меняется в диапазоне от 0 до 90 градусов с шагом в 1 градус , a=13.5 , λ=3 см

**Математическая модель:**



p=

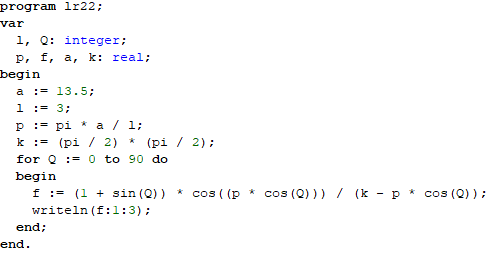
**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | константа | real |
| l | константа | integer |
| Q | параметр цикла | integer |
| f | результат вычислений | real |
| p | промежуточная переменная | real |
| k | промежуточная переменная | real |

**Код программы:**



**Результаты выполненной работы:**

0.000

-0.076

0.211

0.010

-0.020

0.017

-0.035

0.067

-0.205

0.087

0.024

0.000

-0.039

-0.132

-1.320

-0.032

0.025

-0.004

0.036

-0.016

-0.504

0.020

0.000

0.005

-0.026

-0.009

0.254

-0.163

0.040

-0.011

-0.024

-0.053

-0.119

0.740

0.089

0.037

0.000

0.007

-0.070

1.220

-0.147

0.009

0.008

0.000

0.000

-0.155

0.218

0.007

-0.019

0.012

-0.031

0.100

-0.279

0.083

0.021

0.000

-0.044

-0.138

-0.290

-0.012

0.027

-0.005

0.037

-0.020

-0.460

-0.017

0.000

0.008

-0.027

-0.007

0.244

-0.096

0.035

-0.014

-0.282

-0.052

-0.104

0.623

0.094

0.037

0.000

0.001

-0.078

1.720

-0.141

0.011

0.006

0.006

-0.001

-0.235

0.215

**Анализ результатов вычисления:**

Для упрощения вычислений выводим промежуточные переменные p и k, равные и ()2 соответственно . Чтобы вывести несколько результатов вычисления с разным шагом , нужно выводить результирующую в самом цикле до изменения параметра цикла.

**Задача 3**

**Постановка задачи:**

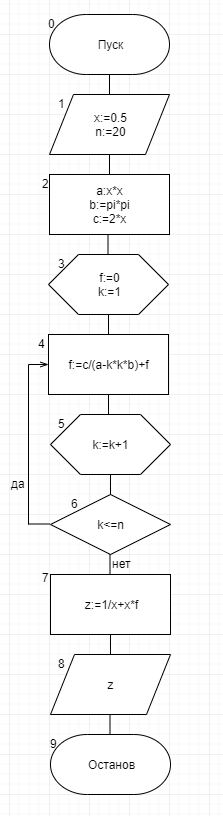
Вычислить значение выражения

**Математическая модель:**



p=

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| x | константа | real |
| k | параметр цикла | integer |
| n | верхняя граница вычислений | integer |
| p | промежуточная переменная | real |
| z | результат | real |
| f | накопительная переменная | real |
| a | промежуточная переменная | real |
| b | промежуточная переменная | real |
| c | промежуточная переменная | real |

**Код программы:**

**program** lr3;

**var** z,f,a,b,c:real;k:integer;

**const**

x=0.5;n=20;

**begin**

a:=x\*x;

b:=pi\*pi;

c:=2\*x;

f:=0;

**for** k:=1 **to** n **do begin**

f:=c/(a-k\*k\*b)+f;

**end**;

z:=1/x+x\*f;

writeln(z:1:3);

**end**.

**Результат выполненной работы:**

1.918

**Анализ результатов вычисления:**

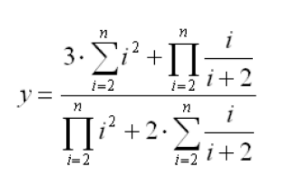
Результат **f** является значением сигмы , которая 20 раз увеличивается на своё предыдущее значение , которое отличается по причине параметра цикла, участвующего в вычислении.

**Задача 4**

**Постановка задачи:**

Вычислить

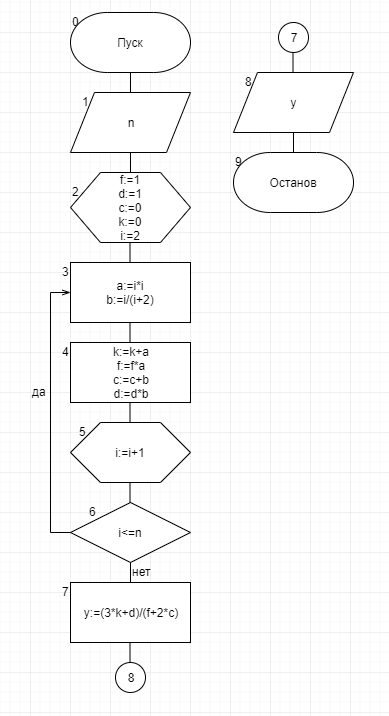
**Математическая модель:**



b=

a=i2

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| n | вводимая переменная | integer |
| i | параметр цикла | integer |
| a | промежуточная переменная | integer |
| b | промежуточная переменная | real |
| k | накопительная переменная | integer |
| f | накопительная переменная | integer |
| c | накопительная переменная | real |
| d | накопительная переменная | real |
| y | результат | real |

**Код программы:**

**program** lr4;

**var** n , i , a , k , f :integer;

b , c , d , y :real;

**begin**

write('n = ');

readln(n);

f:=1;

d:=1;

c:=0;

k:=0;

**for** i:=2 **to** n **do begin**

a:=i\*i;

b:=i/(i+2);

k:=k+a;

f:=f\*a;

c:=c+b;

d:=d\*b;

**end**;

y := (3\*k + d)/(f+2\*c);

writeln(y:1:3);

**end**.

**Результат выполненной работы:**

n = 3

1.029

**Анализ результатов вычисления:**

Мы присваиваем **f** и **d** значение единицы , так как в противном случае результат при умножении всегда будет равен нулю . Значения всех сигм находим в цикле с параметром цикла от **i=2** до значения вводимой переменной.

**Вывод:**

Была изучена реализация алгоритмов ДЦВП по аргументу с помощью Pascal