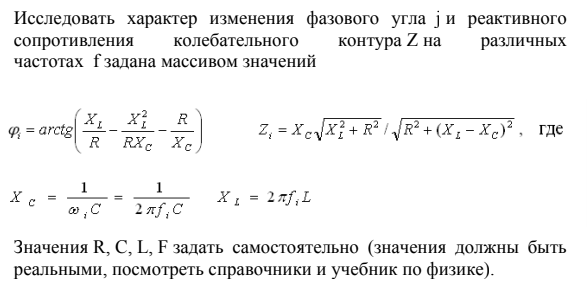
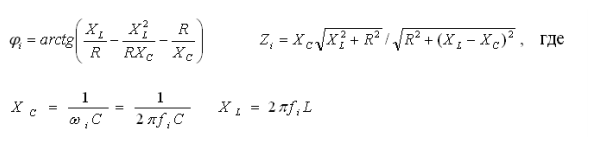
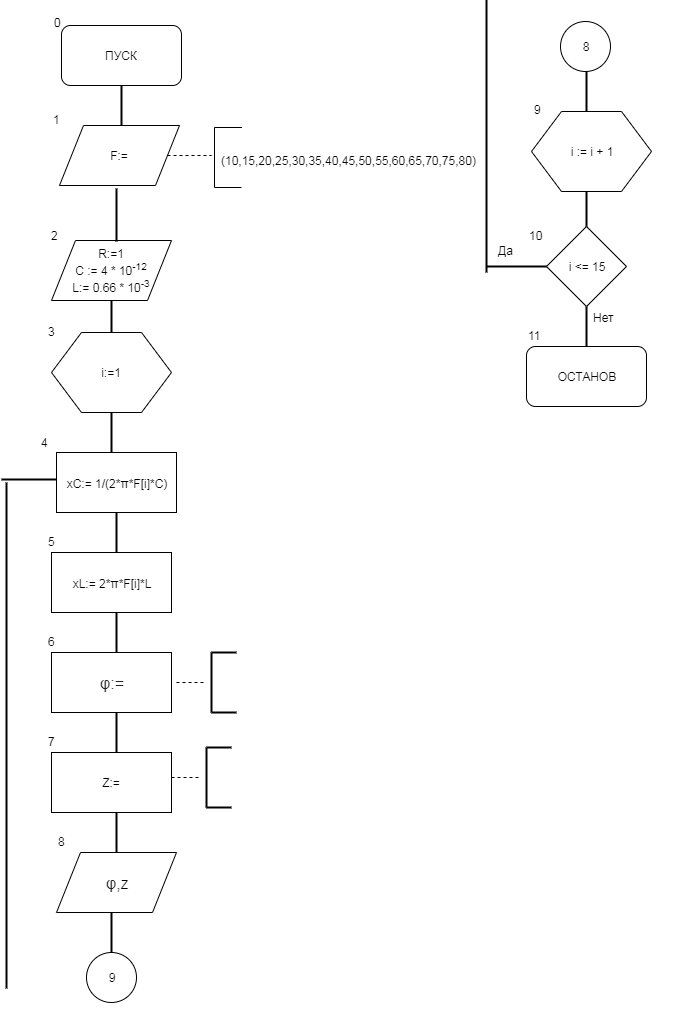
1. Тема: ДЦВП с управлением по индексу. Одномерные массивы.
2. Цель: реализовать одномерные массивы с помощью ДЦВП с управлением по индексу средствами PascalABCNET.
3. Оборудование: персональный компьютер, PascalABC.net, draw.io.

*Задание 1.*

4)

5)

6) 

7)

|  |  |  |
| --- | --- | --- |
| Значение | Смысл | Тип |
| F | Массив | integer |
| i | Параметр цикла и индекс |
| R | Сопротивление | real |
| C | Электроемкость |
| L | Индуктивность |
| xL | Промежуточные переменные |
| xC |
| fi | Фазовый угол |
| Z | Реактивное сопротивление |

8) **program** pr1;

**const**

F:**array**[1..15] **of** integer = (10,15,20,25,30,35,40,45,50,55,60,65,70,75,80);

**var**

i:integer;

R,C,L,xL,xC,fi,Z:real;

**begin**

R := 1;

C := 0.000000000004;

L := 0.00066;

**for** i:=1 **to** 15 **do**

**begin**

xC := 1/(2\*3.1415926\*F[i]\*C);

xL := 2\*3.1415926\*F[i]\*L;

fi := ArcTan(xL/R - xL\*xL/(R\*xC) -R/xC);

Z := xC\*sqrt(xL\*xL+R\*R)/sqrt(R\*R+(xL-xC)\*(xL-xC));

writeln(i,') При частоте = ',F[i] );

writeln('Фазовый угол: ',fi);

writeln('Реактивное сопротивление: ',Z);

**end**;

**end**.

9)

1) При частоте = 10Гц

Фазовый угол: 0.0414452754190036

Реактивное сопротивление: 1.0008594705717

2) При частоте = 15Гц

Фазовый угол: 0.0621234912230677

Реактивное сопротивление: 1.00193277200842

3) При частоте = 20Гц

Фазовый угол: 0.0827486559990059

Реактивное сопротивление: 1.00343346532426

4) При частоте = 25Гц

Фазовый угол: 0.103303508006041

Реактивное сопротивление: 1.00535963662651

5) При частоте = 30Гц

Фазовый угол: 0.123771140258372

Реактивное сопротивление: 1.00770884609598

6) При частоте = 35Гц

Фазовый угол: 0.144135080776423

Реактивное сопротивление: 1.0104781432371

7) При частоте = 40Гц

Фазовый угол: 0.164379367504057

Реактивное сопротивление: 1.01366408505852

8) При частоте = 45Гц

Фазовый угол: 0.184488617101699

Реактивное сопротивление: 1.01726275692831

9) При частоте = 50Гц

Фазовый угол: 0.204448086959184

Реактивное сопротивление: 1.02126979581692

10) При частоте = 55Гц

Фазовый угол: 0.224243729915319

Реактивное сопротивление: 1.0256804156177

11) При частоте = 60Гц

Фазовый угол: 0.243862241319083

Реактивное сопротивление: 1.03048943421814

12) При частоте = 65Гц

Фазовый угол: 0.263291098215403

Реактивное сопротивление: 1.03569130198585

13) При частоте = 70Гц

Фазовый угол: 0.282518590582385

Реактивное сопротивление: 1.04128013133071

14) При частоте = 75Гц

Фазовый угол: 0.301533844682856

Реактивное сопротивление: 1.04724972700921

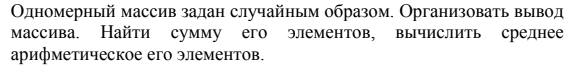
15) При частоте = 80Гц

Фазовый угол: 0.320326838718006

Реактивное сопротивление: 1.05359361684674

10) Так как в задании было указано, что нужно самостоятельно задать частоту, в const был объявлен массив. С помощью цикла каждый элемент массива подставлялся в формулу и выводился на экран вместе со значение фазового угла и реактивного сопротивления.

*Задание 2.*

4)

5) s=

m = =

7)

|  |  |  |
| --- | --- | --- |
| Значение | Смысл | Тип |
| F | Массив | integer |
| i | Параметр цикла и индекс |
| s | Сумма элементов |
| sr | Среднее арифметическое | real |

8) **program** pr2;

**uses** crt;

**var**

sr: real;

i,s:integer;

F:**array** [1..10] **of** integer;

**begin**

s:=0;

write('Элементы массива: ');

randomize;

**for** i:=1 **to** 10 **do**

**begin**

F[i]:=random (1,100);

write (F[i],' ');

s:=s+F[i];

**end**;

writeln();

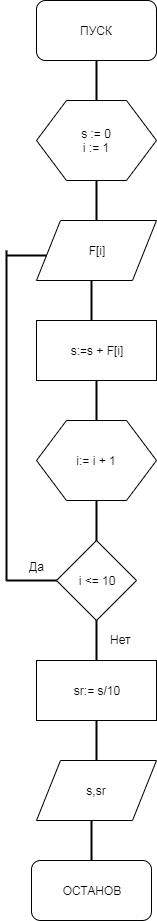
writeln('Сумма элементов массива: ',s);

sr:= s/10;

writeln('Среднее арифметическое элементов: ',sr);

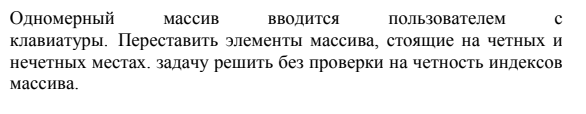
**end**.

9)

6)

10) В данном задании я подключала модуль srt, чтобы получить матрицу, состоящую из сгенерированных рандомно чисел. Я сгенерировала натуральные числа, но с помощью random можно генерировать и другие числа (например, действительные).

*Задание 3.*

4)

5) t = an

an = an+1

an+1 = t

7)

|  |  |  |
| --- | --- | --- |
| Значение | Смысл | Тип |
| A | Массив | real |
| t | Промежуточная переменная |
| i | Параметр цикла и индекс | integer |

8) **program** pr3;

**var**

i:integer;

t:real;

A:**array** [1..10] **of** real;

**begin**

writeln('Введите элементы массива: ');

**for** i:=1 **to** 10 **do**

read(A[i]);

i:=1;

**while** i<10 **do**

**begin**

t:= A[i];

A[i]:= A[i+1];

A[i+1]:=t;

i:=i+2;

**end**;

writeln('Массив с переставленными элементами попарно:',A);

**end**.

9) Введите элементы массива:

1.6

4.7

2.35

1

5

7.089

3

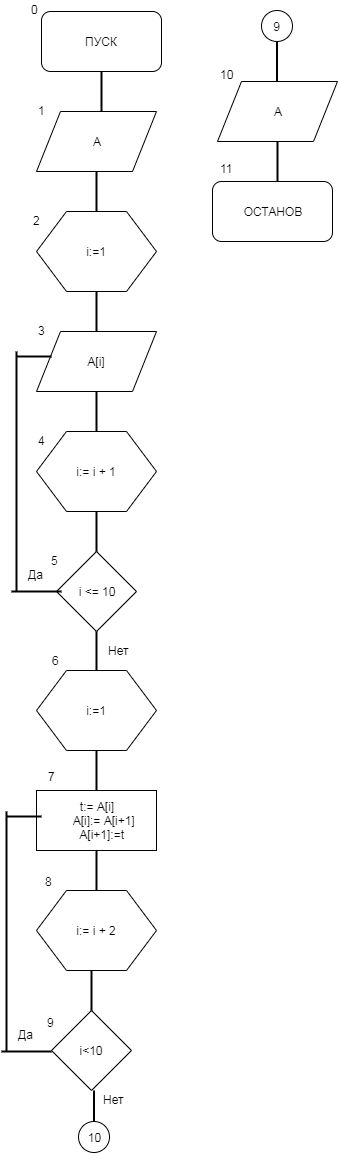
21

22.1

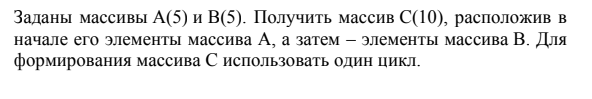
2

Массив с переставленными элементами попарно:[4.7,1.6,1,2.35,7.089,5,21,3,2,22.1]

10) В данной задаче я использовала два цикла: один запрашивал у пользователя элементы массива, второй – менял местами элементы соответственно попарно.

6)

*Задание 4.*

4)

5) С1 X 10 = (a11 ,a12 , a13 , a14 , a15 , b11 ,b12 , b13 , b14 , b15)

7)

|  |  |  |
| --- | --- | --- |
| Значение | Смысл | Тип |
| A | Массив | real |
| B | Массив |
| C | Результат (массив) |
| i | Параметр цикла и индекс | integer |

8) **program** pr4;

**var**

i:integer;

A:**array** [1..5] **of** real;

B:**array** [1..5] **of** real;

С:**array** [1..10] **of** real;

**begin**

writeln ('Введите массив A: ');

**for** i:=1 **to** 5 **do**

read(A[i]);

writeln ('Введите массив B: ');

**for** i:=1 **to** 5 **do**

read(B[i]);

**for** i:=1 **to** 5 **do**

**begin**

С[i] := A[i];

С[i+5] := B[i];

**end**;

writeln ('Полученный массив C: ', С);

**end**.

9) Введите массив A:

1

2

3

4.7

2.6

Введите массив B:

23

1.2

22.3

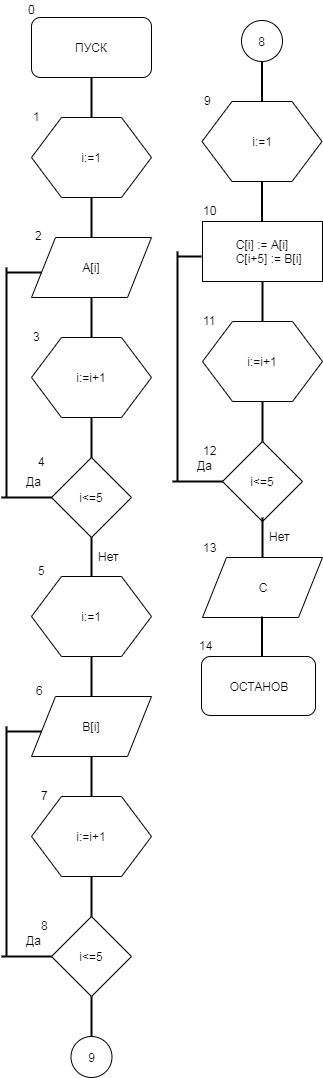
1

7

Полученный массив C: [1,2,3,4.7,2.6,23,1.2,22.3,1,7]

10) В данной программе я использовала 2 цикла, чтобы ввести массивы A и B, и 1 цикл, чтобы получить массив C (элементы массива A присваивались с индекса 1 по 5, элементы B присваивались с индекса 6 по 10).

6)



11) Вывод: с помощью данной работы я научилась реализовывать одномерные массивы с помощью ДЦВП с управлением по индексу средствами PascalABCNET.