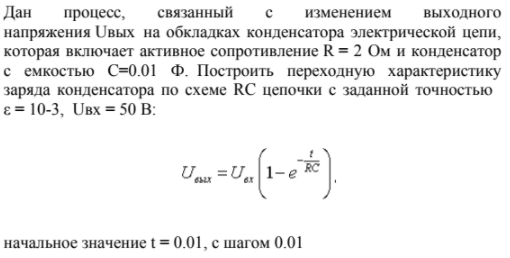
Лабораторная работа №8. Итерационные циклические вычислительные процессы с управлением по индексу/аргументу и функции.

Цель л/р: рассмотрение способов организации ИЦВП с управлением по аргументу и функции средствами Lazarus и Free Pascal.

Оборудование: ПК, Lazarus, Word, Draw.io

Задание 1.

1. Математическая модель:
2. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| r | Активное сопротивление, заданная величина | real |
| c | Ёмкость конденсатора, заданная величина | real |
| uin | Входное напряжение, заданная величина | real |
| eps | Точность вычисления, заданная величина | real |
| t | Время(с) | real |
| step | Шаг изменения времени | real |
| uout2 | Выходное напряжение на обкладках конденсатора в момент времени t | real |
| uout1 | Переменная, хранящая предыдущее значение выходного напряжения | real |

1. Код программы:

program characteristic;

var r,c,uin,eps,t,step,uout1,uout2:real;

begin

r:=2;

c:=0.01;

uin:=50;

eps:=1e-3;

t:=0.01;

uout2:=0;

step:=0.01;

repeat

uout1:=uout2;

uout2:=uin\*(1-exp(-t/(r\*c)));

writeln('t = ',t:0:2);

writeln('Output voltage = ',uout2:0:3);

writeln();

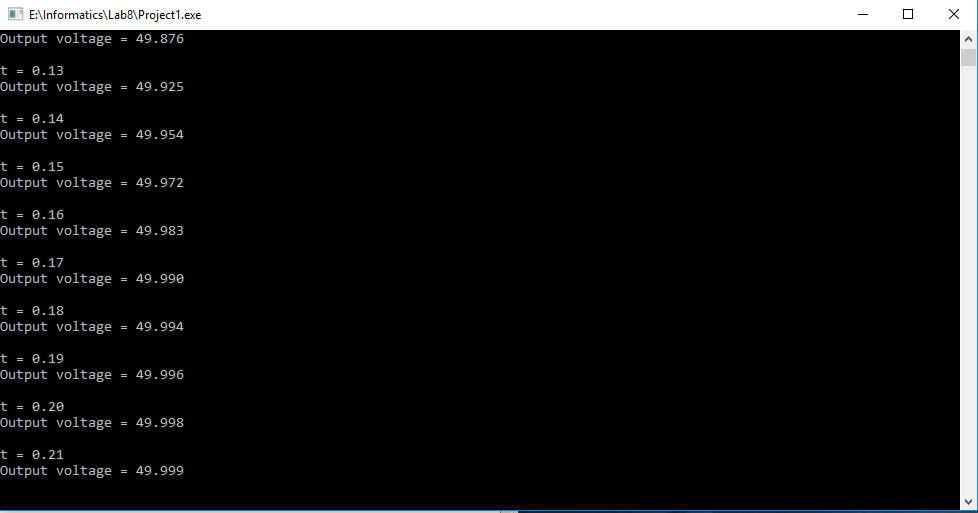
t:=t+step;

until uout2-uout1<=eps;

readln();

end.

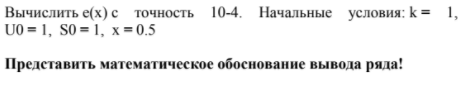
1. Результаты выполненной работы:



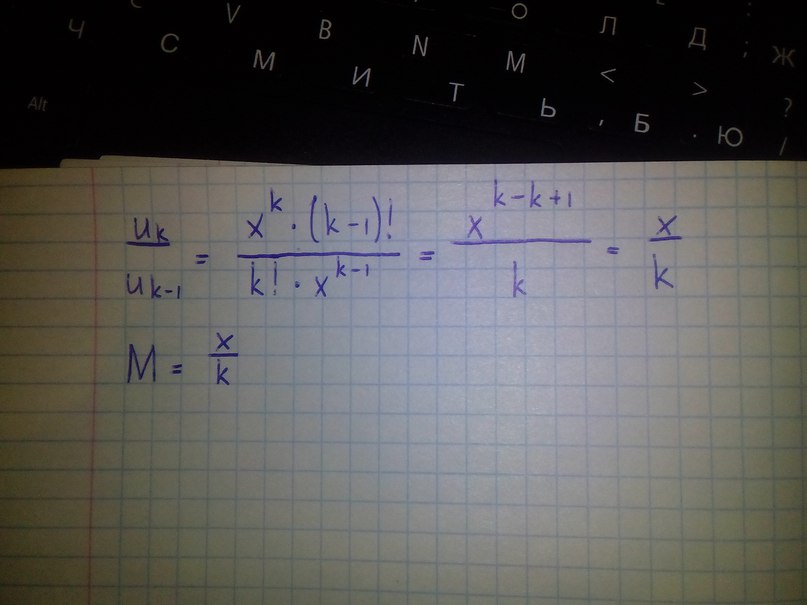
1. Анализ результатов вычисления:

При выполнении программы задаются все входные параметры, обнуляется переменная uout2, т.к. далее в теле цикла repeat..until происходит присвоение её значения переменной uout1, после чего расчитывается само значение выходного напряжения на обкладках конденсатора в данный момент времени. Время t и выходное напряжение uout2 выводятся на экран, переменная t увеличивается на значение step. Цикл продолжается до тех пор, пока условие uout2-uou1<=eps не обратится в истину, т.е. пока не будет достигнута необходимая точность.

Задание 2.



1. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| eps | Точность вычисления, заданная величина | real |
| k | Индекс членов ряда | byte |
| u0 | Значение дроби под индексом k-1 | real |
| u1 | Значение дроби под индексом k | real |
| s0 | Сумма дробей | real |
| x | Показатель степени экспоненты, заданное значение | real |

1. Код программы:

program Project2;

var eps,u0,u1,s0,x:real;

k:byte;

begin

eps:=1e-4;

k:=1;

u1:=1;

s0:=1;

x:=0.5;

repeat

u0:=u1;

u1:=u0\*x/k;

s0:=s0+u1;

k:=k+1;

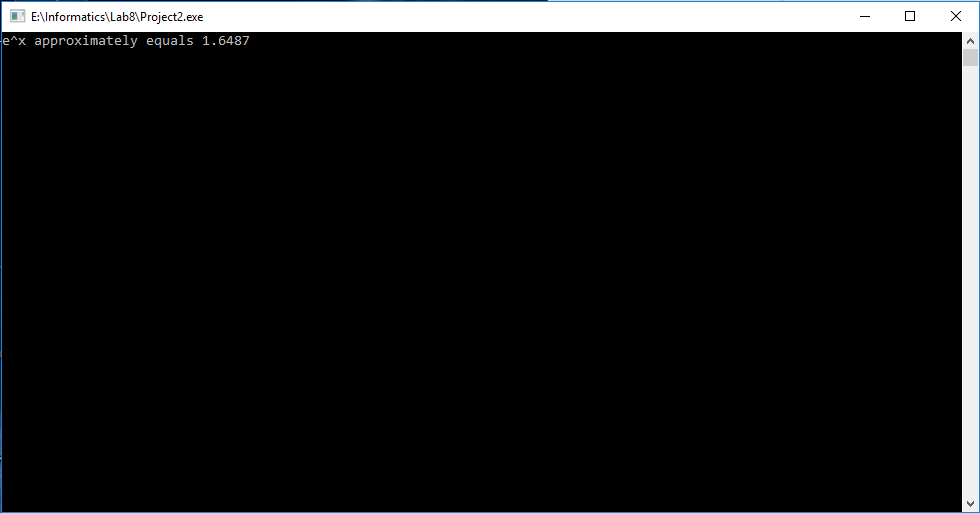
until u0-u1<=eps;

writeln('e^x approximately equals ',s0:0:4);

readln();

end.

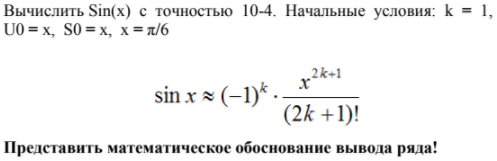
1. Результаты выполненной работы:

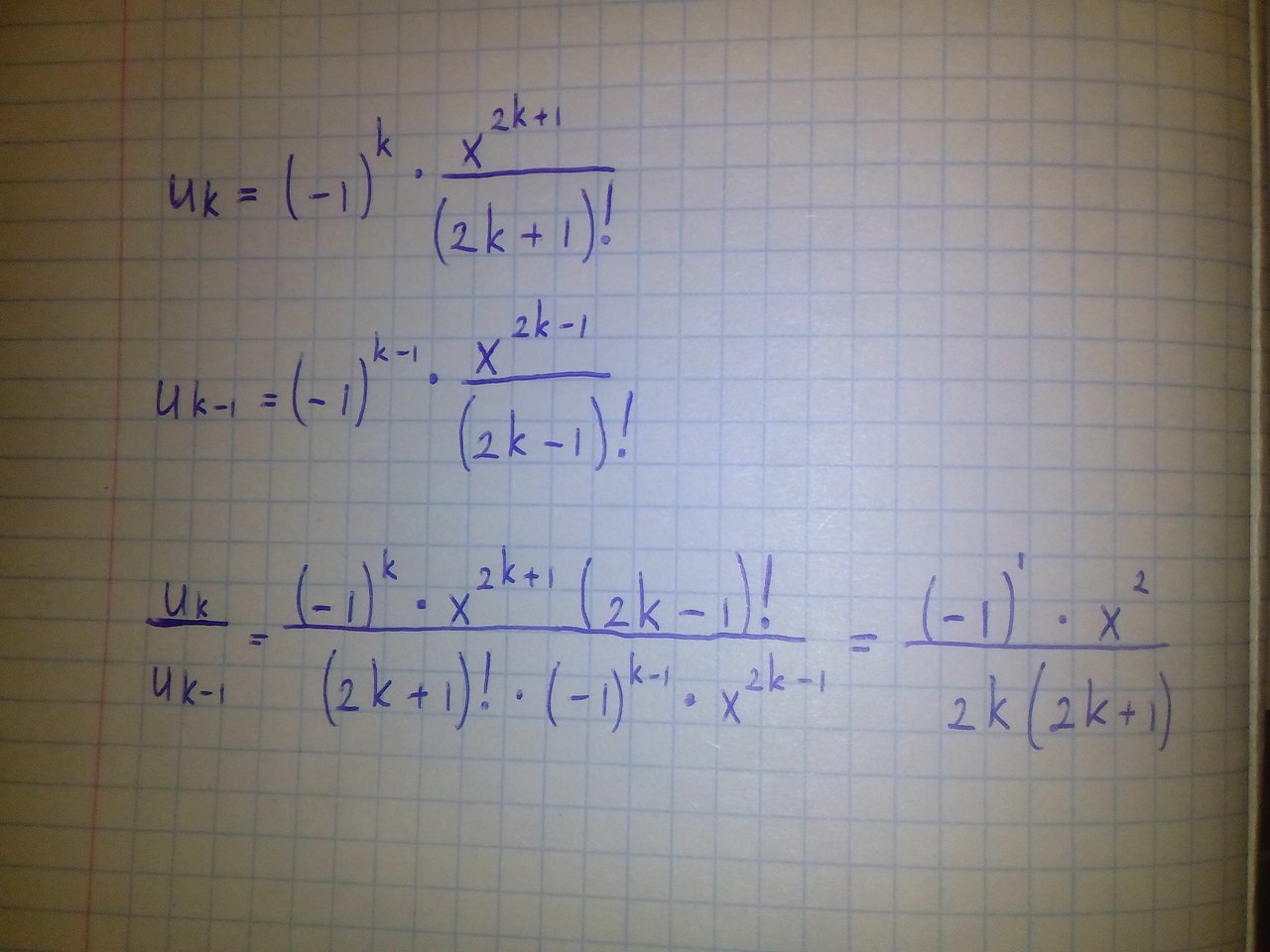


1. Анализ результатов вычисления:

При старте программы производится присвоение переменным значений заданных парметров, переменной u1 присваивается значение 1, т.к. в теле цикла происходит присвоение u0:=u1, далее в цикле вычисляется значение члена ряда с индексом k, переменная k увеличивается на 1 – осуществляется сдвиг по элементам ряда. Цикл выполняется до тех пор, пока условие u0-u1<=eps не станет истинным, т.е. пока не будет достигнута необходимая точность.

Задание 3.

1. 
2. Математическая модель:



1. ­­­­Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| eps | Точность вычисления, заданная величина | real |
| k | Индекс члена ряда | byte |
| u0 | Значение члена под индексом k-1 | real |
| s0 | Сумма членов ряда | real |
| x | Аргумент функции sin(), заданная величина | real |
| u1 | Значение члена под индексом k | real |

1. Код программы:

program project3;

var eps,u0,s0,x,u1:real;

k:byte;

begin

eps:=1e-4;

k:=1;

x:=pi()/6;

u1:=x;

s0:=x;

repeat

u0:=u1;

u1:=u0\*(-x)\*x/(2\*k\*(2\*k+1));

s0:=s0+u1;

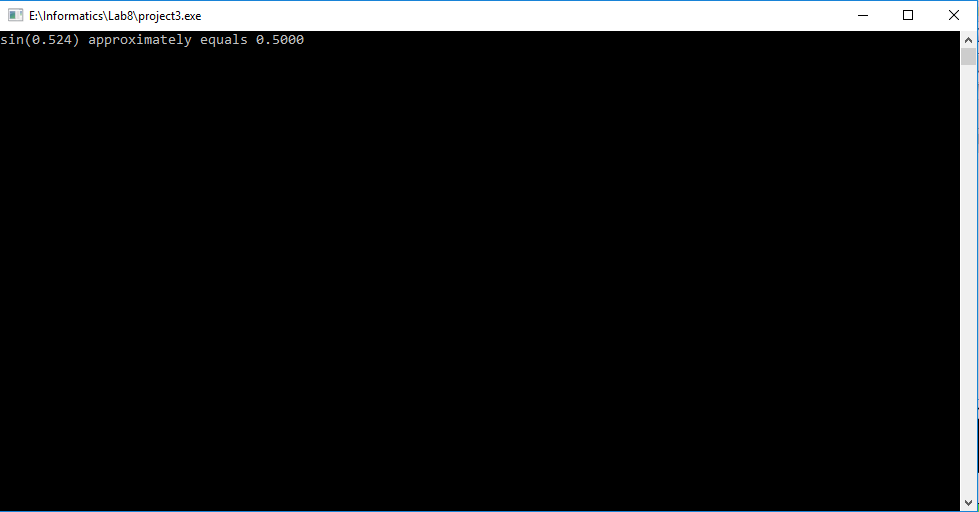
k:=k+1;

until abs(u1-u0)<=eps;

writeln('sin(',x:0:3,') approximately equals ',s0:0:4);

readln();

end.

1. Результат выполненной работы:
2. Анализ результатов вычисления:

При запуске программы происходит присвоение переменным заданных значений параметров, u1 задаётся значение x, т.к. далее в теле цикла происходит присвоение u0:=u1, после чего происходит расчёт значения члена ряда с индексом k, данный элемент прибавляется к сумме членов s0, переменная k увеличивается на 1(осуществляется переход к следующему элементу). Цикл выполняется до тех пор, пока не выполнится условие abs(u1-u0)<=eps. При выходе из цикла на экран выводится значение s0, работа программы завершается.

Задание 4.

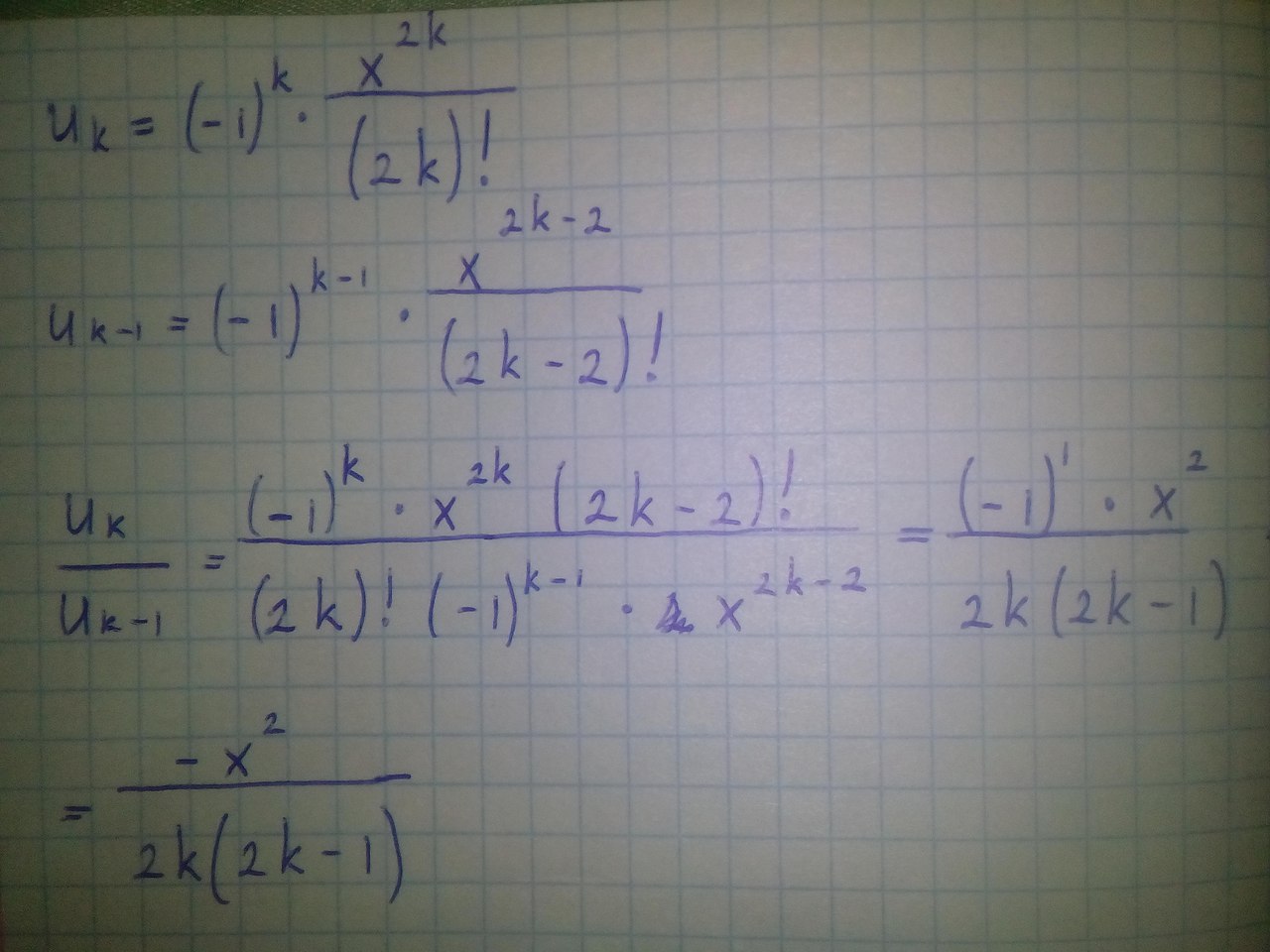
1. Вычислить Cos(x) с точностью 10-4. Начальные условия:

k = 1, U0 = 1, S0 = 1, x = π/6



Представить математическое обоснование вывода ряда!

1. Математическая модель:



1. **Блок-схема:
2. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| eps | Точность вычисления, заданная величина | real |
| k | Индекс члена ряда | byte |
| u1 | Значение члена под индексом k | real |
| s0 | Сумма членов ряда | real |
| x | Аргумент функции cos(), заданная величина | real |
| u0 | Значение члена под индексом k-1 | real |

1. Код программы:

program Project4;

var eps,k,u0,s0,x,u1:real;

begin

eps:=1e-4;

k:=1;

u1:=1;

s0:=1;

x:=pi()/6;

repeat

u0:=u1;

u1:=u0\*(-x)\*x/(2\*k\*(2\*k-1));

s0:=s0+u1;

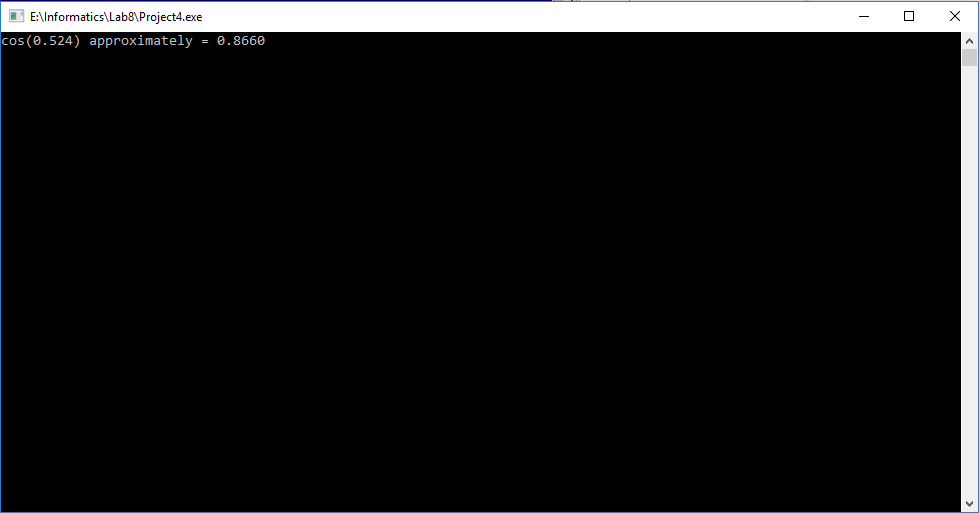
k:=k+1;

until abs(u1-u0)<=eps;

writeln('cos(',x:0:3,') approximately = ',s0:0:4);

readln();

end.

1. Результаты выполненной работы:
2. Анализ результатов вычисления:  
   При запуске программы происходит присвоение перменным заданных значений параметров, в u1 устанавливается 1, т.к. далее в теле цикла repeat..until происходит присваивание u0:=u1. Далее в цикле происходит вычисление значения члена с индексом k, полученный результат прибавляется к s0 – сумме членов. Перменная k увеличивается на 1(происходит сдвиг по индексу элемента). Цикл Продолжается до тех пор, пока не выполнится условие abs(u1-u0)<=eps. После выхода из цикла на экран выводится значение s0. Программа завершает работу.

Вывод:

ИЦВП с управлением по аргументу и функции находят применение в задачах на построение переходных харакетристик, при приближенном вычислении элементарных функций численными методами разложения в ряд. Условием для завершения цикла в данных случаях является достижение точности ε в модуле разности элементов Во Free Pascal данный процесс организуется при помощи конструкций while..do и repeat..until.