Лабораторная работа № 8 «Итерационные циклические вычислительные процессы с управлением по аргументу и функции»

Цель: разработать и научиться использовать алгоритмы, основанные на итерационных циклических вычислительных процессах, управление которыми осуществляется по аргументу и функции.

Оборудование: ПК, среда разработки «PascalABC»

Задание 1

Постановка задачи:

Дан процесс, связанный с изменением выходного напряжения Uвых на обкладках конденсатора электрической цепи, которая включает активное сопротивление R = 2 Ом и конденсатор с емкостью C=0.01 Ф. Построить переходную характеристику заряда конденсатора по схеме RC цепочки с заданной точностью ε = 10-3, Uвх = 50 В:

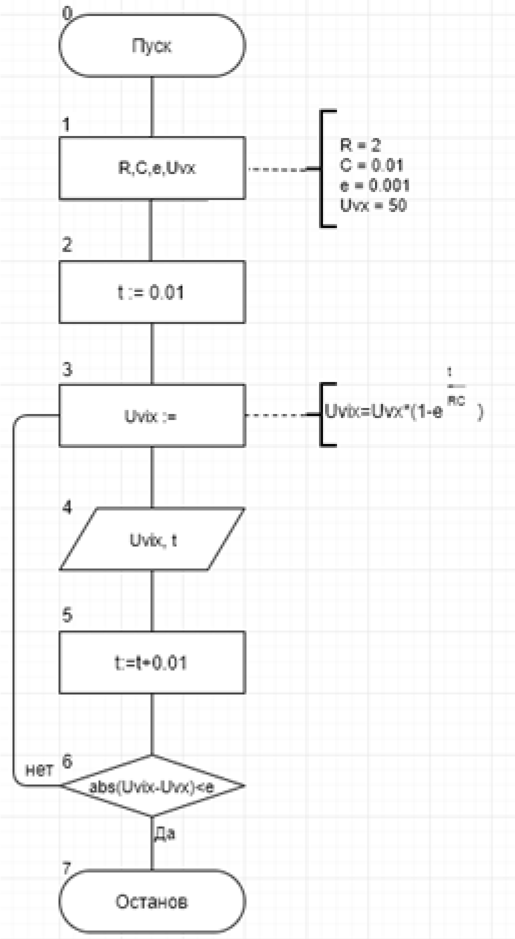
$$U_{вых} = U_{вх}\left(1 - e^{-\frac{t}{RC}}\right),$$

начальное значение t = 0.01, с шагом 0.01

Математическая модель:

$$U_{вых} = U_{вх}\left(1 - e^{-\frac{t}{RC}}\right),$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
R	Integer	Хранение значения сопротивления
C	Real	Хранение значения емкости
e	Real	Хранение значения точности
t	Real	Хранение значения времени
Uvx	Integer	Хранение значения входного напряжения
Uvix	Real	Хранение значения выходного напряжения

Код программы:

```
program zadaniel;
Const
R = 2;
C = 0.01;
e = 0.001;
Uvx = 50;
Var
Uvix, t :real;
begin
  t := 0.01;
  repeat
   Uvix := Uvx*(1 - exp(-t/(R*C)));
   Writeln('Uvix = ', Uvix, '  t = ', t);
   t := t + 0.01;
   Until abs(Uvix - Uvx) < e;
end.
```

Результаты вычислений:

Окно вывода

```
Uvix = 19.6734670143683  t = 0.01
Uvix = 31.6060279414279  t = 0.02
Uvix = 38.8434919925785  t = 0.03
Uvix = 43.2332358381694  t = 0.04
Uvix = 45.8957500688051  t = 0.05
Uvix = 47.5106465816068  t = 0.06
Uvix = 48.4901308288841  t = 0.07
Uvix = 49.0842180555633  t = 0.08
Uvix = 49.4445501730879  t = 0.09
Uvix = 49.6631026500457  t = 0.1
Uvix = 49.7956614280768  t = 0.11
Uvix = 49.8760623911667  t = 0.12
Uvix = 49.9248280403511  t = 0.13
Uvix = 49.9544059017223  t = 0.14
Uvix = 49.9723457814926  t = 0.15
Uvix = 49.9832268686049  t = 0.16
Uvix = 49.9898265815495  t = 0.17
Uvix = 49.9938295097957  t = 0.18
Uvix = 49.9962574085056  t = 0.19
Uvix = 49.9977300035119  t = 0.2
Uvix = 49.9986231775325  t = 0.21
Uvix = 49.9991649149605  t = 0.22
```

Анализ результатов вычисления: Решение данной задачи было получено путем применения алгоритма, основанного на итерационном вычислительном процессе, управление которым осуществляется по аргументу и функции. Основной особенностью этой программы является большое кол-во констант. «R» типа «integer», хранящая в себе значение сопротивления, «C» типа «real», хранящая в себе значение емкости, «e» типа «real», хранящая в себе значение точности и «Uvx» типа «integer», хранящая в себе значение входного напряжения. Перед входом в цикл переменной «t» типа «real» присваивается начальное значение. В самом цикле это переменная будет меняться по рекуррентной зависимости (t = t + 0.01). В самом цикле высчитывается значения функции, которое хранится в переменной «Uvix» типа «real». Выход из цикла осуществляется по разницы в значениях функции.

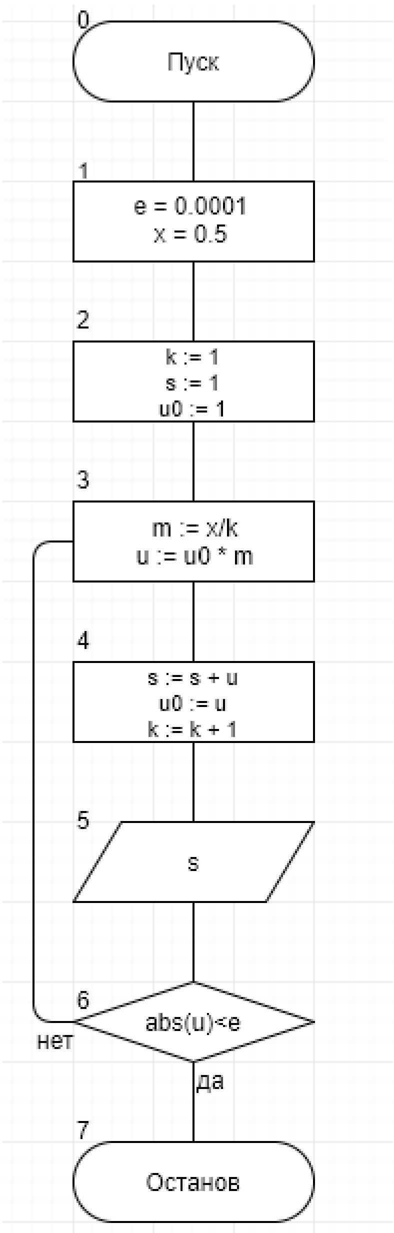
Задание 2

Постановка задачи: Вычислить e(x) с точность 10-4. Начальные условия: k = 1, U0 = 1, S0 = 1, x = 0.5

Математическая модель:

$$\boldsymbol{e^{x} \approx \sum_{k=1}^{\infty} \frac{x^{k}}{k!}, |x| < \infty}$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
e	Real	Хранение значения точности
m	Real	Хранение значения множителя
x	Real	Хранение значения x
k	Integer	Хранение значения порядка
s	Real	Хранение значения функции
u0	Real	Хранение значения предыдущего члена суммы
u	Real	Хранение значения последующего члена суммы

Код программы:

```
Program zadanie2;
Const
x = 0.5;
e = 0.0001;
Var
u,u0, m, s : real;
k : integer;
begin
 u0 := 1;
 s := 1;
 k := 1;
 repeat
  m := x/k;
  u := u0* m;
  k := k + 1;
  s := s + u;
  u0 := u;
  writeln('e(x) = ', s);
  Until abs(u) < e;
end.
```

Результаты вычислений:

```
Окно вывода
e(x) = 1.5
e(x) = 1.625
e(x) = 1.64583333333333
e(x) = 1.6484375
e(x) = 1.64869791666667
e(x) = 1.64871961805556
```

Анализ результатов вычисления: Результатом выполнения данной программы является примерное значение элементарной функции e^x полученное благодаря алгоритму, основанному на ИЦВП управление которым осуществляется по аргументу и функции. Перед началом цикла мы задаем переменным «s», «x», «e» и «u0» типа «real» начальные значения, а также переменной «k» типа «integer». В самом цикле меняется лишь аргумент «k» по рекуррентной зависимости (k = k +1). После каждого выполнения цикла, выводится промежуточное примерное значение функции. Выход из цикла осуществляется в том случае если последующий член суммы меньше точности.

Задание 3

Постановка задачи:

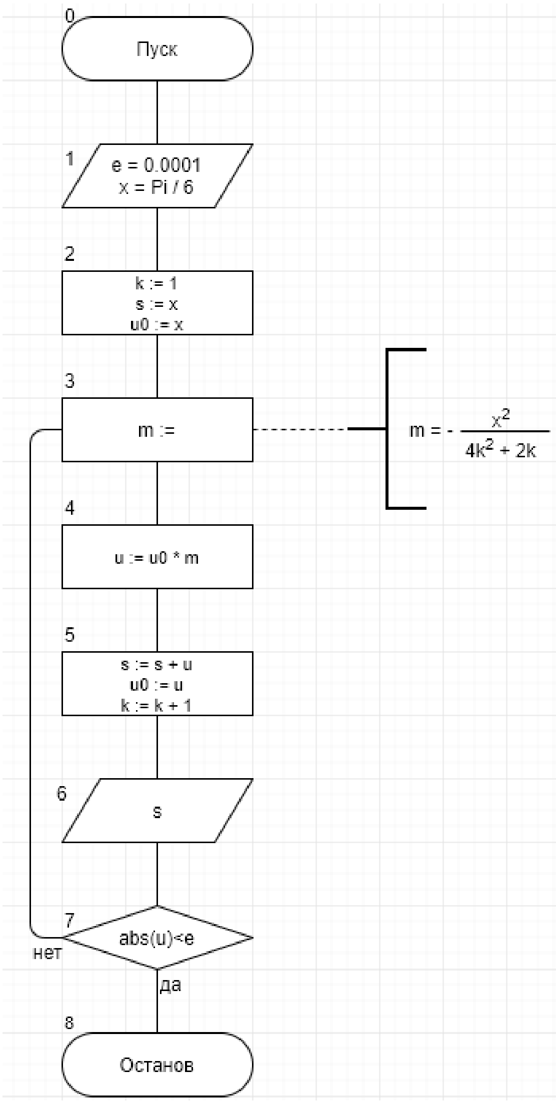
Вычислить Sin(x) с точностью 10-4. Начальные условия: k = 1, U0 = x, S0 = x, x = π/6

$$\sin x \approx (-1)^k \cdot \frac{x^{2k+1}}{(2k+1)!}$$

Математическая модель:

$$\sin x \approx (-1)^k \cdot \frac{x^{2k+1}}{(2k+1)!}$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
e	Real	Хранение значения точности
m	Real	Хранение значения множителя
x	Real	Хранение значения x
k	Integer	Хранение значения порядка
s	Real	Хранение значения функции
u0	Real	Хранение значения предыдущего члена суммы
u	Real	Хранение значения последующего члена суммы

Код программы:

```
Program zadanie3;
Const
e = 0.0001;
x = pi / 6;
Var
u,u0, m, s : real;
k : integer;
begin
 u0 := x;
 s := x;
 k := 1;
 repeat
  m := x*x/(4*k*k + 2* k)*-1;
  u := u0* m;
  k := k + 1;
  s := s + u;
  u0 := u;
  writeln('sin(x) = ', s);
  Until abs(u) < e;
end.
```

Результаты вычислений:

```
Окно вывода
sin(x) = 0.499674179394364
sin(x) = 0.500002132588792
sin(x) = 0.499999991869023
```

Анализ результатов вычисления: Результатом выполнения данной программы является примерное значение элементарной функции sin(x) полученное благодаря алгоритму, основанному на ИЦВП управление которым осуществляется по аргументу и функции. Перед началом цикла мы задаем переменным «s», «x», «e» и «u0» типа «real» начальные значения, а также переменной «k» типа «integer». В самом цикле меняется лишь аргумент «k» по рекуррентной зависимости (k = k +1). После каждого выполнения цикла, выводится промежуточное примерное значение функции. Выход из цикла осуществляется в том случае если последующий член суммы меньше точности.

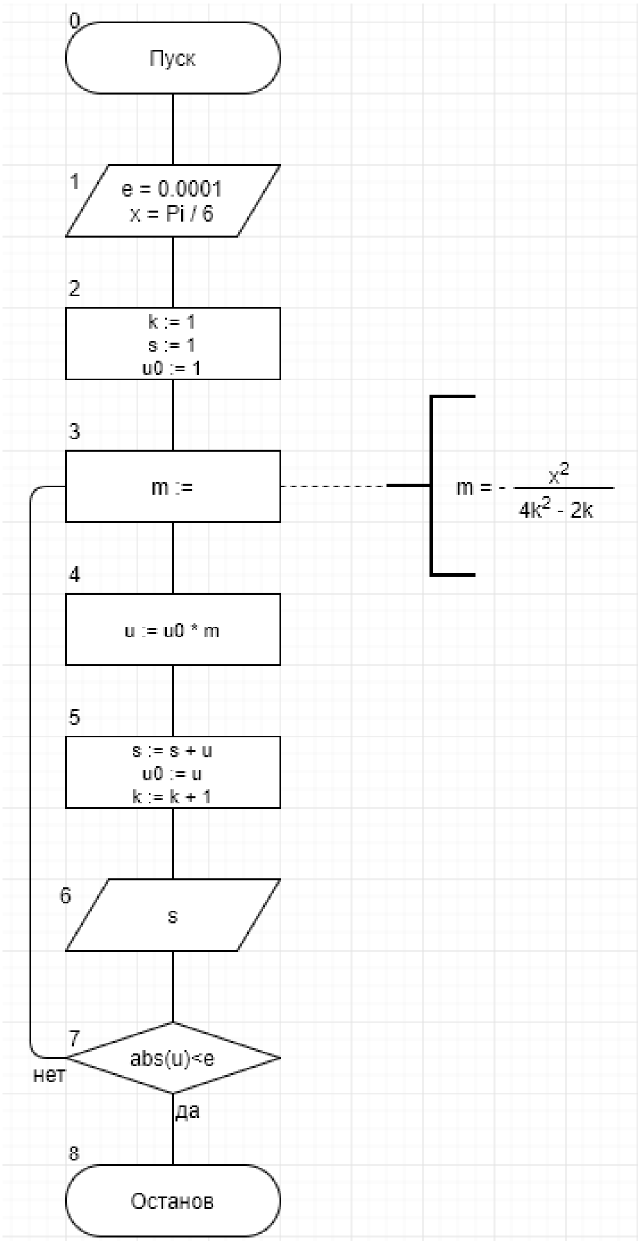
Задание 4

Постановка задачи: Вычислить Cos(x) с точность 10-4. Начальные условия: k =1, U0 = 1, S0 =1, x = π / 6

Математическая модель:

$$\cos x \approx (-1)^k \frac{x^{2k}}{(2k)!}$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
e	Real	Хранение значения точности
m	Real	Хранение значения множителя
x	Real	Хранение значения x
k	Integer	Хранение значения порядка
s	Real	Хранение значения функции
u0	Real	Хранение значения предыдущего члена суммы
u	Real	Хранение значения последующего члена суммы

Код программы:

```
Program zadanie3;
Const
e = 0.0001;
x = pi / 6;
Var
u,u0, m, s : real;
k : integer;
begin
 u0 := 1;
 s := 1;
 k := 1;
 repeat
  m := -1*x*x/(4*k*k -2 * k);
  u := u0* m;
  k := k + 1;
  s := s + u;
  u0 := u;
  writeln('cos(x) = ', s);
  Until abs(u) < e;
end.
```

Результаты вычислений:

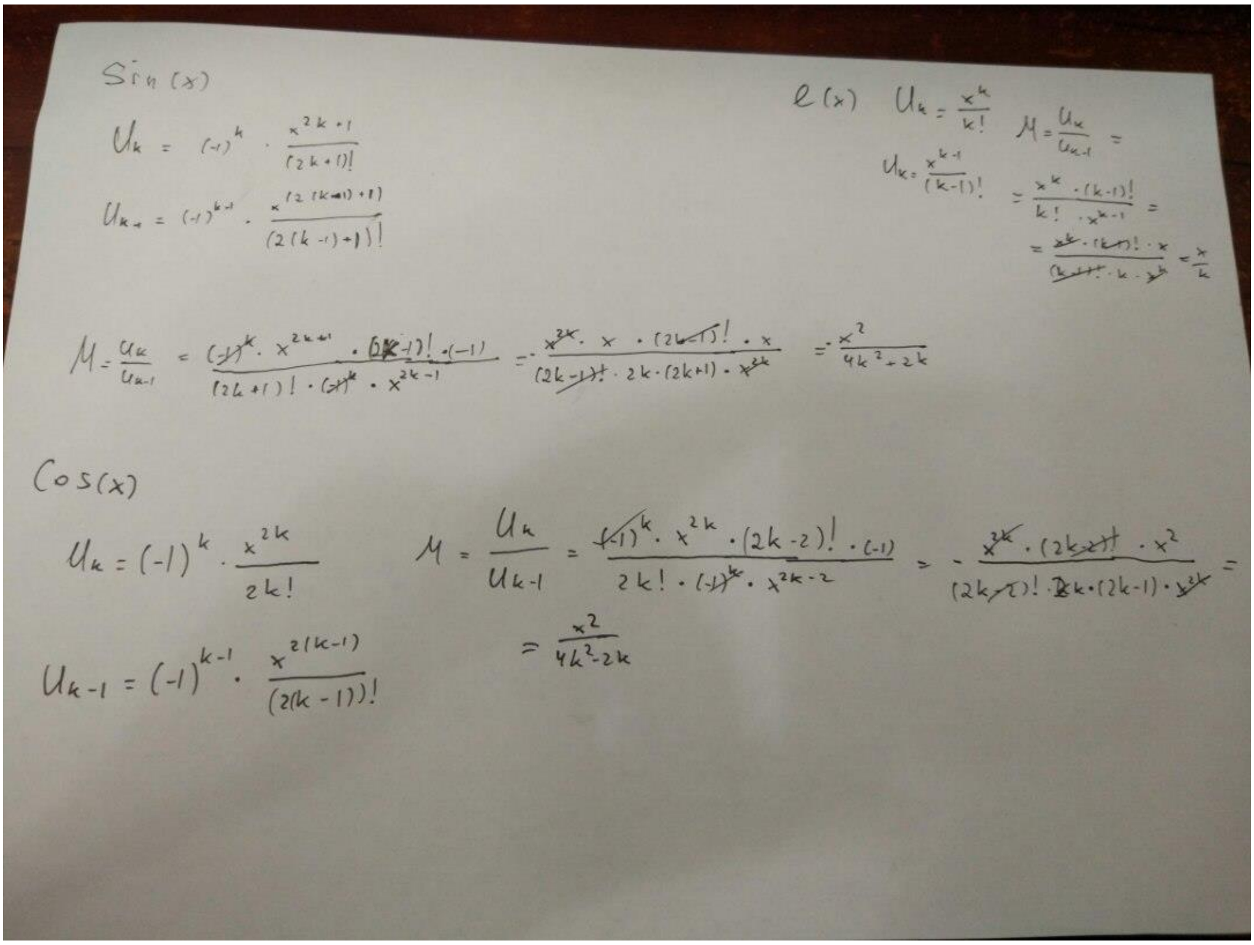
```
Окно вывода
cos(x) = 0.862922161095981
cos(x) = 0.866053883415747
cos(x) = 0.866025264100571
```

Анализ результатов вычисления: Результатом выполнения данной программы является примерное значение элементарной функции cos(x) полученное благодаря алгоритму, основанному на ИЦВП управление которым осуществляется по аргументу и функции. Перед началом цикла мы задаем переменным «s», «x», «e» и «u0» типа «real» начальные значения, а также переменной «k» типа «integer». В самом цикле меняется лишь аргумент «k» по рекуррентной зависимости (k = k +1). После каждого выполнения цикла, выводится промежуточное примерное значение функции.

Выход из цикла осуществляется в том случае если последующий член суммы меньше точности.

Вывод: Использование ИЦВП с управлением по аргументу и функции, является обязательным умением для любого программиста. Ведь только благодаря алгоритмам, основанным на нем, стал возможен подсчет элементарных функций на ЭВМ в разумное время, и при разумных затратах памяти. И за счет подсчета элементарных функций уже можно решать довольно обширный спектр задач.

Вывод формул:



Sin(x)

$$U_k = (-1)^k \cdot \frac{x^{2k+1}}{(2k+1)!}$$

$$U_{k-1} = (-1)^{k-1} \cdot \frac{x^{(2(k-1)+1)}}{(2(k-1)+1)!}$$

$$M = \frac{U_k}{U_{k-1}} = \frac{\cancel{(-1)^k} \cdot x^{2k+1} \cdot \cancel{(2k-1)!} \cdot (-1)}{(2k+1)! \cdot \cancel{(-1)^k} \cdot x^{2k-1}} = -\frac{\cancel{x^{2k}} \cdot x \cdot \cancel{(2k-1)!} \cdot x}{\cancel{(2k-1)!} \cdot 2k \cdot (2k+1) \cdot \cancel{x^{2k}}} = -\frac{x^2}{4k^2+2k}$$

e(x)

$$U_k = \frac{x^k}{k!} \quad M = \frac{U_k}{U_{k-1}} =$$

$$U_{k-1} = \frac{x^{k-1}}{(k-1)!}$$

$$= \frac{x^k \cdot (k-1)!}{k! \cdot x^{k-1}} = \frac{\cancel{x^k} \cdot \cancel{(k-1)!} \cdot x}{\cancel{(k-1)!} \cdot k \cdot \cancel{x^k}} = \frac{x}{k}$$

Cos(x)

$$U_k = (-1)^k \cdot \frac{x^{2k}}{2k!}$$

$$M = \frac{U_k}{U_{k-1}} = \frac{\cancel{(-1)^k} \cdot x^{2k} \cdot (2k-2)! \cdot (-1)}{2k! \cdot \cancel{(-1)^k} \cdot x^{2k-2}} = -\frac{\cancel{x^{2k}} \cdot \cancel{(2k-2)!} \cdot x^2}{\cancel{(2k-2)!} \cdot 2k \cdot (2k-1) \cdot \cancel{x^{2k}}} =$$

$$= \frac{x^2}{4k^2-2k}$$

$$U_{k-1} = (-1)^{k-1} \cdot \frac{x^{2(k-1)}}{(2(k-1))!}$$