

**Лабораторная работа № 4**  
**«Детерминированные вычислительные**  
**процессы с управлением по аргументу.**  
**Численное интегрирование с использованием**  
**пользовательских функций.»**

Цель: разработать и научиться использовать алгоритмы, основанные на детерминированных вычислительных процессах, управление которыми осуществляется по аргументу для вычисления определенного интеграла и рационализировать вычисления путем применения пользовательских функций.

Оборудование: ПК, среда разработки «PascalABC»

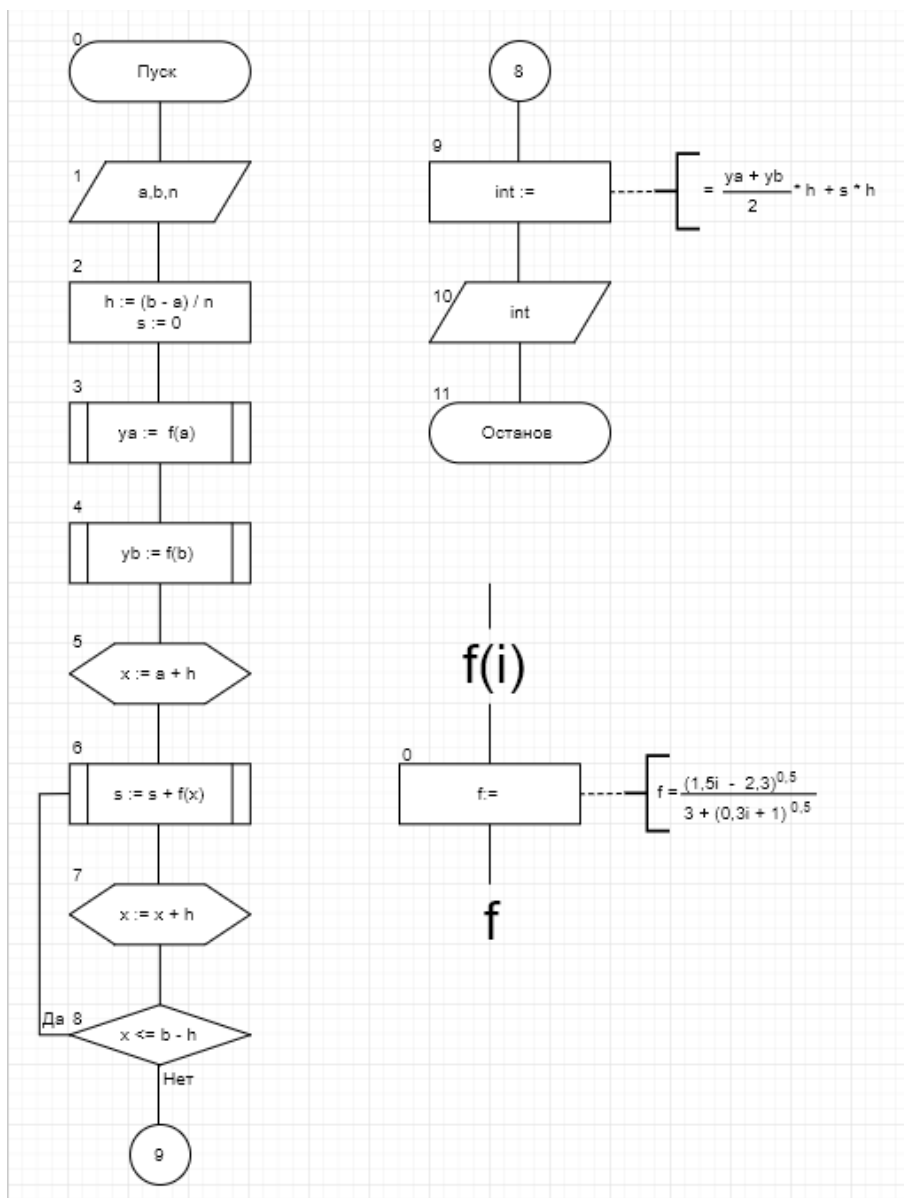
# Задание 1

**Постановка задачи:** Реализовать вычисление определенного интеграла из индивидуального задания методом трапеций с использованием пользовательской функции.

**Математическая модель:**

$$\int_a^b \frac{\sqrt{1,5x + 2,3} dx}{3 + \sqrt{0,3x + 1}} \approx h * \left( \frac{f(a) + f(b)}{2} + \sum_{i=a+h}^{b-h} f(i) \right); \quad f(i) = \frac{\sqrt{1,5i + 2,3}}{3 + \sqrt{0,3i + 1}}$$

**Блок-схема:**



### Список идентификаторов:

Название	Тип	Функция
a	real	Хранение значения нижнего предела интегрирования
b	real	Хранение значения верхнего предела интегрирования
n	real	Хранение значения кол-ва шагов
h	real	Хранение значения величины шага
s	real	Хранение значения суммы значений функции
ya	real	Хранение значения функции при аргументе равном a
yb	real	Хранение значения функции при аргументе равном b
x	real	Управление циклом
int	real	Хранение значения интеграла
i	real	Переменная необходимая для работы польз. функции

### Код программы:

```
Program Zadanie3;
Var
  a,b,n,h,s,x,int,ya,yb : real;
Function f(i :real) :real;
begin
  f := (sqrt(1.5 * i + 2.3)) / (3 + sqrt(0.3 * i + 1));
end;
begin
  writeln('Введите нижний предел интегрирования');
  readln(a);
  writeln('Введите верхний предел интегрирования');
  readln(b);
  writeln('Введите кол-во шагов');
  readln(n);
  h := (b - a) / n;
  s := 0;
  x := a + h;
  ya := f(a);
  yb := f(b);
  While x <= b - h do
  begin
    s := s + f(x);
    x := x + h;
  end;
  int := h * (((ya + yb) / 2) + s);
  writeln(int);
  readln();
end.
```

### Результаты вычислений:

#### Окно вывода

```
Введите нижний предел интегрирования
0.8
Введите верхний предел интегрирования
2.4
Введите кол-во шагов
1000
0.819965117914123
```

### **Анализ результатов вычисления:**

Решение данной задачи было получено путем использования детерминированного циклического процесса, управляемого аргументом (переменная  $x$  типа «real»). Основная особенность этой программы состоит в том, что из-за дробного шага (переменная  $h$  типа «real»), пришлось использовать конструкция `while`. Вводимые с клавиатуры значения (переменные «a», «b» и «n» типа «real») как-раз и позволяли установить значения шага. Задачей цикла являлось накопление суммы (переменная «s» типа «real») значений данной функции в каждой точке разбиения. Изменение аргумента происходило по рекуррентной формуле  $x = x + h$ , а сумма накапливалась путем последовательного суммирования значений функции. Диапазон значений аргумента был от  $a + h$  до  $b - h$ , и связано это с использованным способом нахождения интеграла (метод правых частей прямоугольников). Также из-за особенностей этого метода, понадобился ввод двух дополнительных переменных «ua» и «ub» типа «real», которые в свою очередь хранили в себе значений функции в точках  $a$  и  $b$  соответственно. В целях рационализации вычислений, мы используем функцию «f» которая возвращает значение типа `real`, и необходимая для вычисления значения функции.

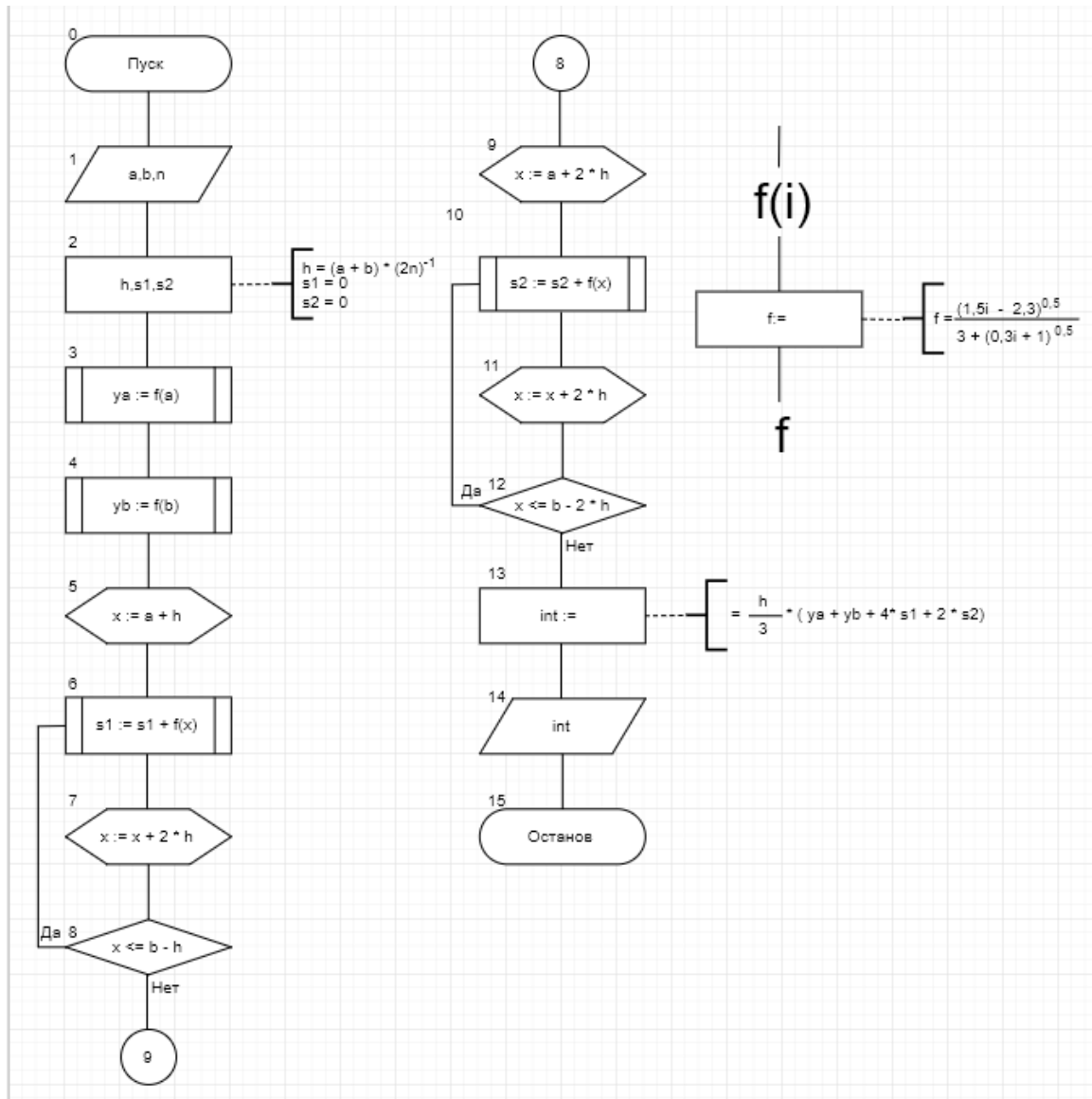
## **Задание 2**

**Постановка задачи:** Реализовать вычисление определенного интеграла из индивидуального задания методом парабол с использованием пользовательской функции.

**Математическая модель:**

$$\int_a^b f(x)dx \approx \frac{h}{3} * (f(a) + 4 * (f(a + h) + f(a + 3h) + \dots + f(b - h)) + 2 * (f(a + 2h) + f(a + 4h) + \dots + f(b - 2h)) + f(b))$$

## Блок-схема:



## Список идентификаторов:

Название	Тип	Функция
a	real	Хранение значения нижнего предела интегрирования
b	real	Хранение значения верхнего предела интегрирования
n	real	Хранение значения кол-ва шагов
h	real	Хранение значения величины шага
s1	real	Хранение значения суммы значений функции в нечетных шагах
s2	real	Хранение значения суммы значений функции в четных шагах
ya	real	Хранение значения функции при аргументе равном a
yb	real	Хранение значения функции при аргументе равном b
x	real	Управление циклом
int	real	Хранение значения интеграла
i	real	Переменная необходимая для работы польз. функции

## Код программы:

```
Program Zadanie2;
Var
  a,b,n,x,s1,s2,ya,yb,int, h : real;
Function f( i: real) : real;
begin
  f := (sqrt(1.5 * i + 2.3)) / (3 + sqrt(0.3 * i + 1))
end;
begin
  Writeln('Введите нижний предел интегрирования');
  Readln(a);
  Writeln('Введите верхний предел интегрирования');
  Readln(b);
  Writeln('Введите кол-во шагов');
  Readln(n);
  s1 := 0;
  s2 := 0;
  h := (b - a) / (2*n);
  ya := f(a);
  yb := f(b);
  x := a + h;
  while x <= b - h do
  begin
    s1 := s1 + f(x);
    x := x + 2 * h;
  end;
  x := a + 2 * h;
  while x <= b - 2 * h do
  begin
    s2 := s2 + f(x);
    x := x + 2 * h;
  end;
  int := (h / 3) * (ya + 4 * s1 + 2 * s2 + yb);
  writeln(int);
  readln();
end.
```

## Результаты вычислений:

Окно вывода

```
Введите нижний предел интегрирования
0.8
Введите верхний предел интегрирования
2.4
Введите кол-во шагов
1000
0.819965123437928
```

**Анализ результатов вычисления:** Решение данной задачи было получено путем использования детерминированного циклического процесса, управляемого аргументом (переменная  $x$  типа «real»). Основная особенность этой программы состоит в том, что из-за дробного шага (переменная  $h$  типа «real»), пришлось использовать конструкция `while`. Вводимые с клавиатуры значения (переменные «a», «b» и «n» типа «real») как-раз и позволяли установить значения шага. Задачей циклов являлось накопление суммы (переменная «s1» типа «real» значений данной функции в каждой нечетной точке разбиения, а переменная «s2» типа «real» значений данной функции в каждой четной точке разбиения). Изменение аргумента происходило по рекуррентной формуле  $x = x + 2h$ , а суммы накапливались путем последовательного суммирования значений функции. Оба цикла управлялись одной и той-же переменной, но вот только диапазон значений аргумента первого цикла был от  $a + h$  до  $b - h$ , а второго от  $a + 2h$  до  $b - 2h$ , и связано это с использованным способом нахождения интеграла (метод правых частей прямоугольников). Также из-за особенностей этого метода, понадобился ввод двух дополнительных переменных «ya» и «yb» типа «real», которые в свою очередь хранили в себе значений функции в точках  $a$  и  $b$  соответственно. Использование функции в данной задаче, позволяет рационализировать вычисления, а также представить код в более читаемом виде. За счет обращения к функции «f», возвращающей значение функции типа «real» при заданном аргументе, хранящемся в переменной «i» типа «real», подсчет суммы будет происходить рациональнее.

## Задание 3

**Постановка задачи: Вычислить:**

$$y = \frac{\sum_{i=1}^n \left( \frac{1}{(i+1)!} \cdot \frac{x^{2i+1}}{2i+1} \right)}{5.5 + x^2 + (3n)!}$$

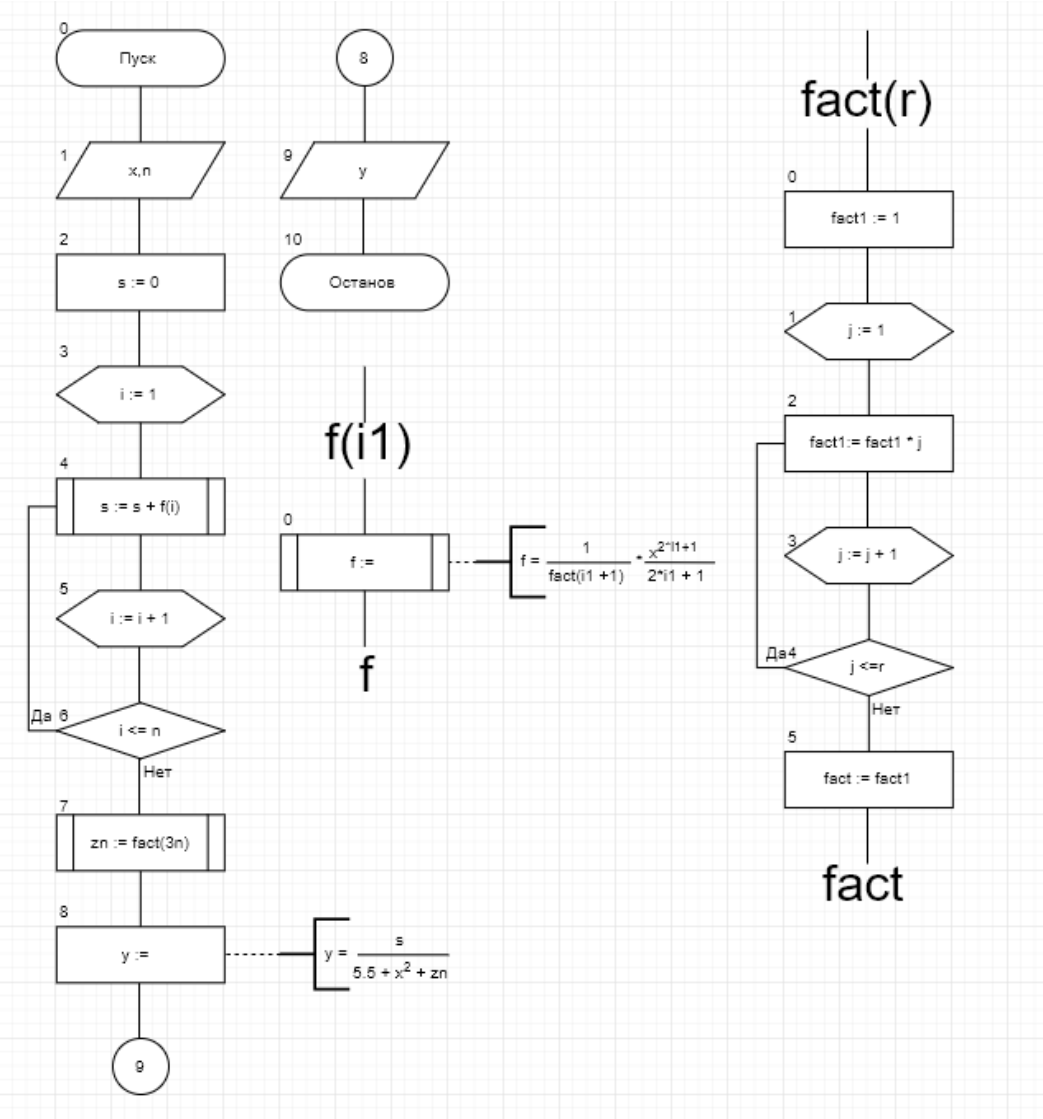
$$x = 1, n = 5$$

**Математическая модель:**

$$y = \frac{\sum_{i=1}^n \left( \frac{1}{(i+1)!} \cdot \frac{x^{2i+1}}{2i+1} \right)}{5.5 + x^2 + zn}$$

$$zn = (3n)!$$

**Блок-схема:**



**Список идентификаторов:**

Название	Тип	Функция
x	integer	Хранение значения «x»
n	integer	Хранение значения «n»
s	real	Хранение суммы значений функций
i	integer	Управление циклом подсчета суммы
zn	longint	Хранение значения 3n!
y	real	Хранение итогового значения
i1	integer	Хранение значения, с которым работает функция «f»
j	integer	Управление циклом в функции подсчета факториала
fact1	integer	Промежуточное хранение значения факториала
r	real	Хранение значения, с которым работает функция «fact»



## Код программы:

```
Program zadanie3;
Var
  zn : longint;
  s,y : real;
  i,n,il,x : integer;
function fact(r : integer) : longint;
Var factl, j : integer;
begin
  factl := 1;
  for j := 1 to r do
    Begin
      factl := factl * j;
    End;
  fact := factl;
end;
function f (il : integer) : real;
Begin
  f := ( 1 / fact( il + 1)) * ((power(x, 2* il + 1)) / (2 * il + 1))
end;
begin
  writeln('Введите x');
  readln(x);
  writeln('Введите n');
  readln(n);
  s := 0;
  For i := 1 to n do
    begin
      s := s + f(i);
    end;
  zn := fact(3*n);
  y := s / (5.5 + x*x + zn);
  writeln('y = ', y);
end.
```

## Результаты вычислений:

Окно вывода

```
Введите x
1
Введите n
5
y = 1.03279715802833E-10
```

## Анализ результатов вычисления:

Выполнение данных вычислений стало возможным благодаря использованию детерминированных циклических вычислительных процессов управление которыми осуществляется аргументом. Входные параметры хранятся в переменных «x» и «n» типа «integer». С целью рационализации вычислений, нам понадобится использовать две пользовательские функции: «fact» возвращающая значение типа «integer», и функция «f» возвращающая значение типа «real». Первая функция «fact» вычисляет значение факториала,

делается это путем циклических вычислений. Циклом управляет переменная «j» типа «integer». А число, для которого нужно взять интеграл хранится в переменной «r» типа «integer». Вторая функция «f» вычисляет значение функции. Аргумент, при котором нужно сосчитать значение передается в переменную «i1» типа «integer». Итоговое значение записывается в переменную «u» типа «real» и выводится на экран. Избавляться от экспоненциальной формы числа в этой задаче не является рациональным.

**Вывод:** Использование пользовательских функций является неотъемлемой частью программирования. Веди они позволяют рационализировать необходимые вычисления, а также делают код куда более понятным. Вдобавок к функциям, можно еще использовать процедуру, разновидностью которых и является функция. В данных задачах использовать процедуру не являлось рациональным, так как нам необходимо возвращать вычисляемое значение. В связке с ДЦВП, использование пользовательских функций и процедур может помочь в решении довольно обширного пласта задач.