

Лабораторная работа № 3

«Детерминированные вычислительные процессы с управлением по аргументу. Численное интегрирование.»

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

Оборудование: ПК, среда разработки «PascalABC»

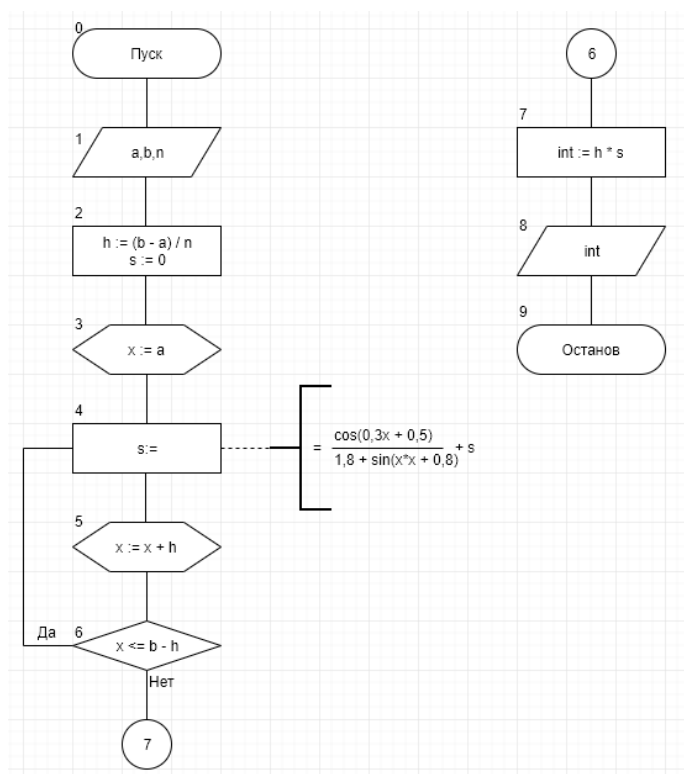
Задание 1

Постановка задачи: написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.

Математическая модель:

$$\int_a^b \frac{\cos(0,3x + 0,5)}{1,8 + \sin(x^2 + 0,8)} dx \approx h * \sum_{i=a}^{b-h} f(i); \quad f(i) = \frac{\cos(0,3x + 0,5)}{1,8 + \sin(x^2 + 0,8)}$$

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
x	real	Управление циклом
int	real	Хранение значения интеграла

Код программы:

```
Program Zadanie1;
Var
  h,x,n,s,int : real;
const
  a = 0.3;
  b = 1.1;
begin
  Writeln('Введите кол-во шагов');
  Readln(n);
  s := 0;
  h := (b - a) / n;
  x := a;
  while x <= b - h do
  begin
    s := s + cos(0.3 * x + 0.5) / (1.8 + sin(x*x + 1.8));
    x := x + h;
  end;
  int := h * s;
  writeln(int);
  readln();
end.
```

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 до $b - h$, и связано это с использованным способом нахождения интеграла (метод левых частей прямоугольников)

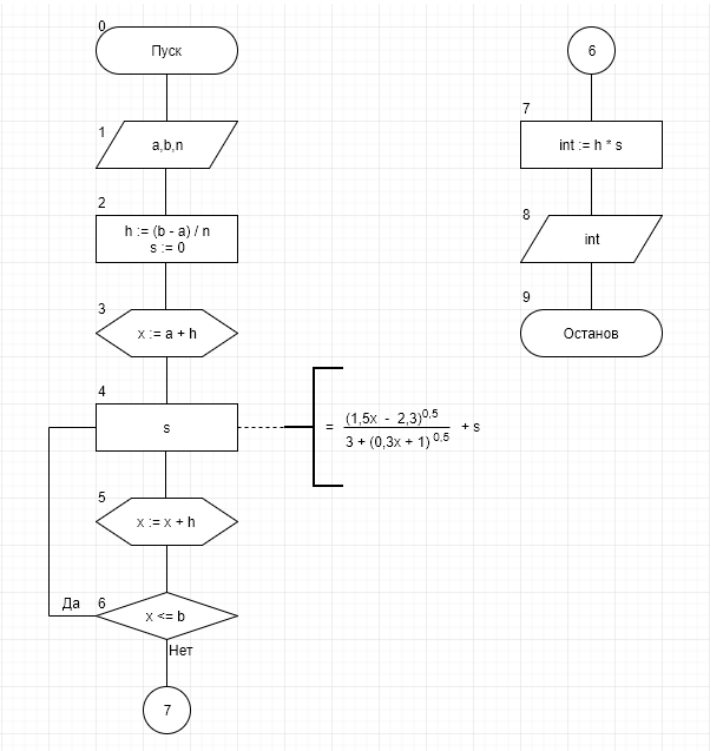
Задание 2

Постановка задачи: написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника правых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.

Математическая модель:

$$\int_a^b \frac{\cos(0,3x + 0,5) \, dx}{1,8 + \sin(x^2 + 0,8)} \approx h * \sum_{i=a}^{b-h} f(i); \quad f(i) = \frac{\cos(0,3x + 0,5) \, dx}{1,8 + \sin(x^2 + 0,8)}$$

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
x	real	Управление циклом
int	real	Хранение значения интеграла

Код программы:

```
program zadanie2;
  Var
    a,b,h,x,n,s,int : real;
begin
  Writeln('Введите нижний предел интегрирования');
  Readln(a);
  Writeln('Введите верхний предел интегрирования');
  Readln(b);
  Writeln('Введите кол-во шагов');
  Readln(n);
  s := 0;
  h := (b - a) / n;
  x := a + h;
  while x <= b do
  begin
    s := s + ((sqrt(1.5 * x + 2.3)) / (3 + sqrt(0.3 * x + 1)));
    x := x + h;
  end;
  int := h * 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
0.820051982192323
```

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 , и связано это с использованным способом нахождения интеграла (метод правых частей прямоугольников)

Задание 3

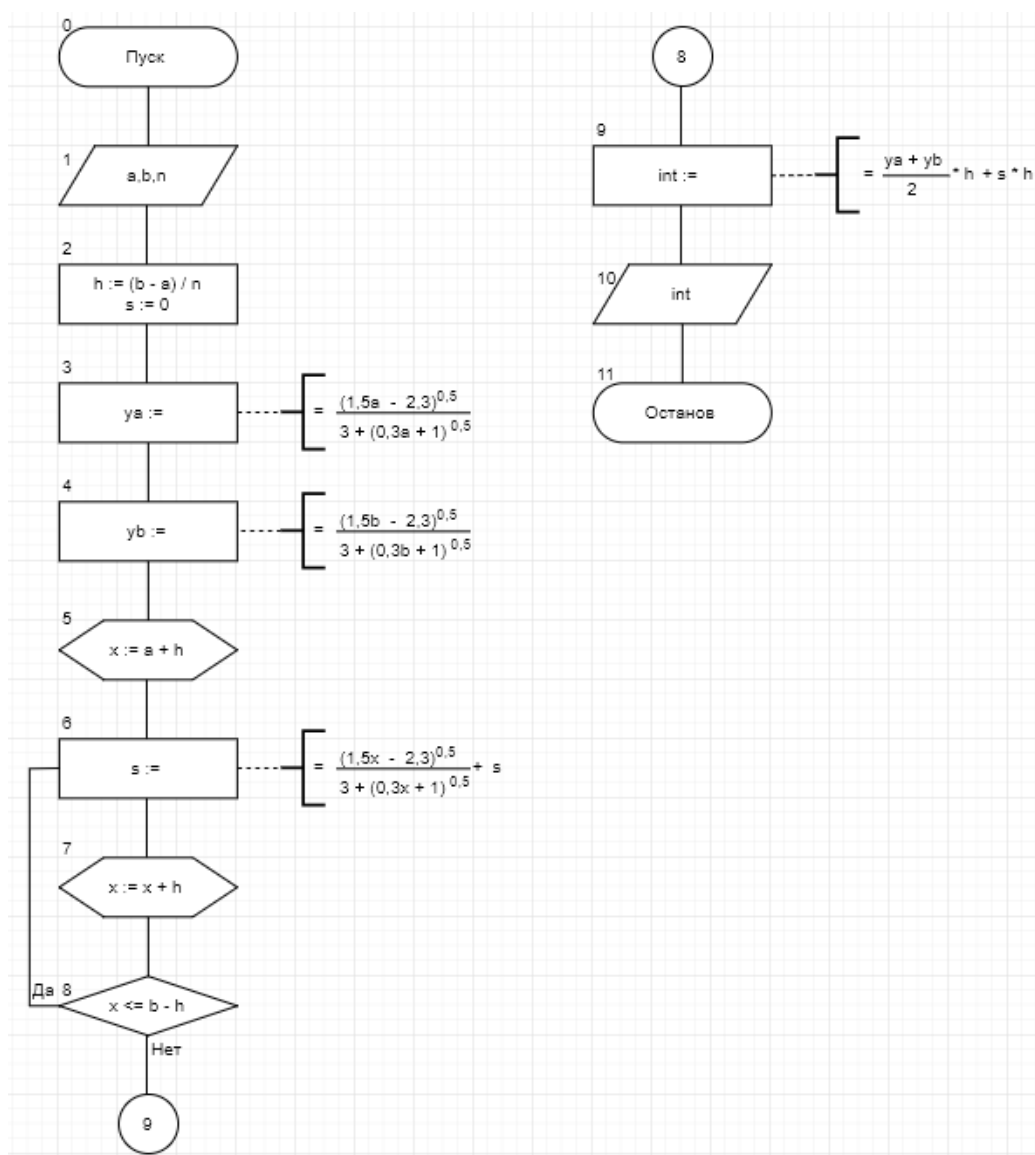
Постановка задачи: написать программу для вычисления определенного интеграла из индивидуального задания методом трапеций. Протестировать программу на определенном интеграле, вычисленном в ходе выполнения самостоятельной работы 3

Математическая модель:

$$\int_a^b \frac{\cos(0,3x + 0,5) dx}{1,8 + \sin(x^2 + 0,8)} \approx$$

$$\approx h * \left(\frac{f(a) + f(b)}{2} + \sum_{i=a+h}^{b-h} f(i) \right); \quad f(i) = \frac{\cos(0,3x + 0,5) dx}{1,8 + \sin(x^2 + 0,8)}$$

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

Код программы:

```
Program Zadanie3;
Var
  a,b,n,h,s,x,int,ya,yb : real;
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 := (sqrt(1.5 * a + 2.3)) / (3 + sqrt(0.3 * a + 1));
  yb := (sqrt(1.5 * b + 2.3)) / (3 + sqrt(0.3 * b + 1));
  While x <= b - h do
  begin
    s:= s + ((sqrt(1.5 * x + 2.3)) / (3 + sqrt(0.3 * x + 1))) ;
    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 «ya» и «yb» типа «real», которые в свою очередь хранили в себе значений функции в точках a и b соответственно.

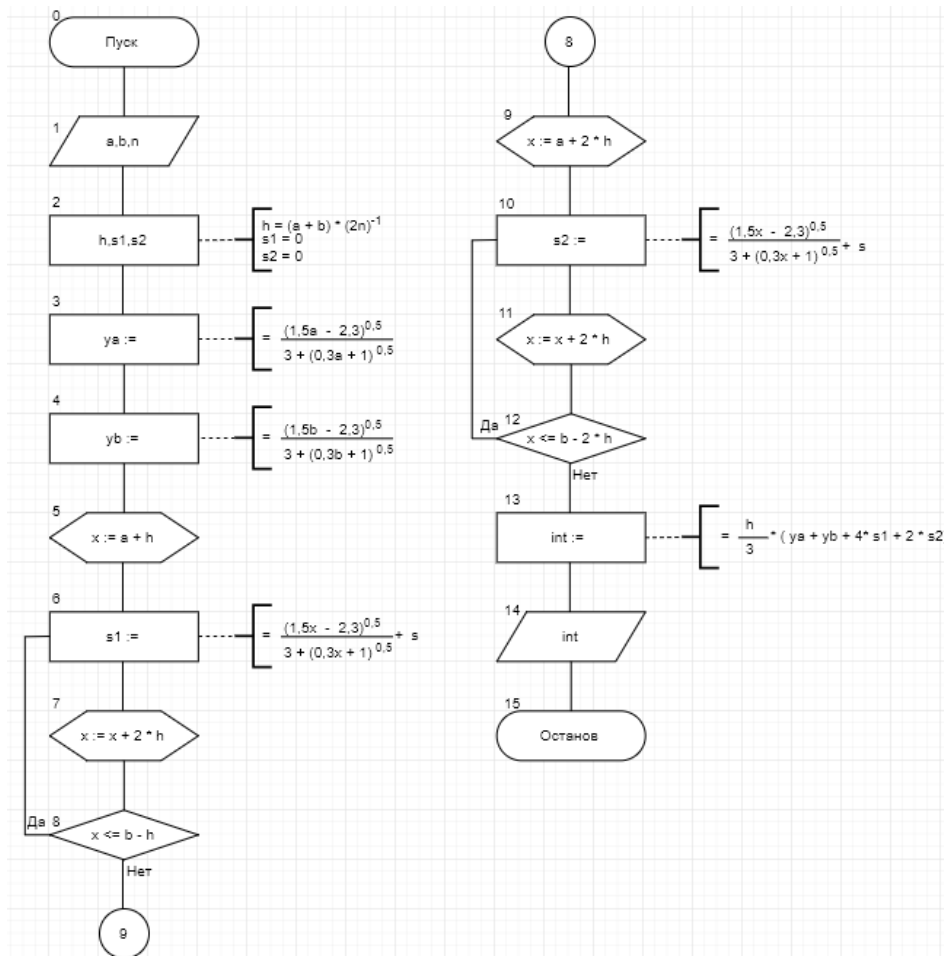
Задание 4

Постановка задачи: написать программу для вычисления определенного интеграла из индивидуального задания методом парабол. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3

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

Код программы:

```
Program Zadanie4;
Var
  a,b,n,x,s1,s2,ya,yb,int, h : real;
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 := (sqrt(1.5 * a + 2.3)) / (3 + sqrt(0.3 * a + 1));
  yb := (sqrt(1.5 * b + 2.3)) / (3 + sqrt(0.3 * b + 1));
  x := a + h;
  while x <= b - h do
  begin
    s1 := s1 + ((sqrt(1.5 * x + 2.3)) / (3 + sqrt(0.3 * x + 1)));
    x := x + 2 * h;
  end;
  x := a + 2 * h;
  while x <= b - 2 * h do
  begin
    s2 := s2 + ((sqrt(1.5 * x + 2.3)) / (3 + sqrt(0.3 * x + 1)));
    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.

Таблица сравнения результатов вычисления различных способов:

N Количество разбиений	h шаг	I Метод левых частей прямоугольника	I Метод правых частей прямоугольника	I Метод трапеций	I Метод парабол
10	0,16	0.8112234	0.8285963	0.8199098	0.8199651
100	0,016	0.8100964	0.8118191	0.8109650	0.8109607
1000	0,0016	0.8198782	0.8200519	0.8199651	0.8199651
10000	0,00016	0.8198662	0.8199569	0.8198749	0.8198749

Вывод: Наиболее точным способом вычисления определенного интеграла является метод Симпсона (метод парабол) и его использования, как впрочем и любого другого, было бы невозможно без использования ДЦВП. Все 4 рассматриваемых метода, в результате вычислений при минимальном шаге, выдают абсолютно одинакового значения до 3 знака после запятой, это говорит о том, что в случае не указания точности измерений или же в случае работы с точность до 10^{-3} разумнее будет использовать один из первых методов, тк они являются более эффективными по времени и памяти. Увеличить точность измерения можно уменьшая шаг, и в результате этого получая большее кол-во точек разбиения, и в следствии этого более точный результат.