**Лабораторная работа №3**

**Детерминированные вычислительные процессы**

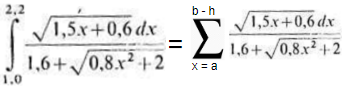
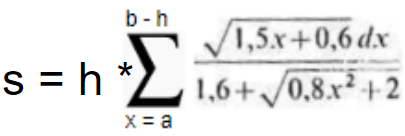
**с управлением по аргументу. Численное интегрирование.**

**Цель:** научиться численному интегрированию с помощью алгоритмов детерминированных циклических вычислительных процессов с управлением по аргументу средствами компилятора free Pascal.

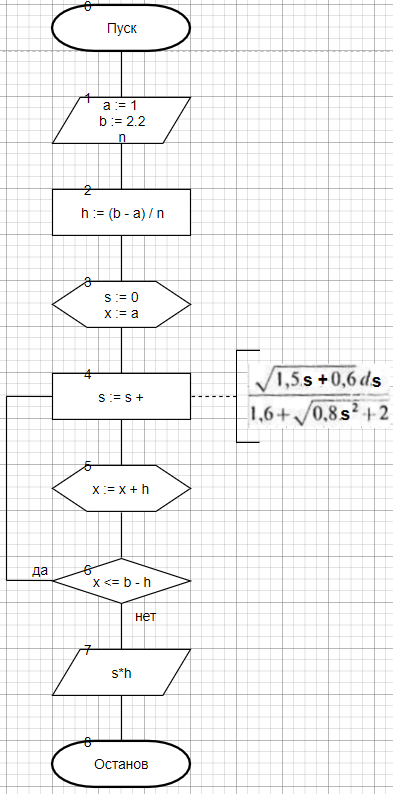
**Используемое оборудование:** пк, PascalABC.

**Задание 1:** метод левых частей прямоугольников.

**4.1)** Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей.

**5.1)****,** 

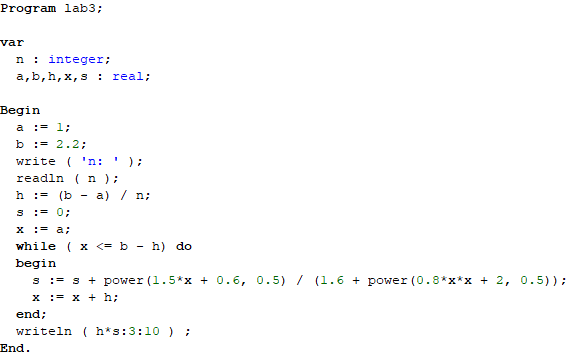
**6.1)**



**7.1)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **n** | **Переменная, на сколько частей разбит отрезок [a;b]** | **Int** |
| **a** | **Переменная, левая граница криволинейной функции)** | **Real** |
| **b** | **Переменная, правая криволинейной функции** | **Real** |
| **h** | **Переменная, шаг** | **Real** |
| **x** | **Переменная, параметр цикла** | **Real** |
| **s** | **Переменная, площадь** | **Real** |

**8.1)**



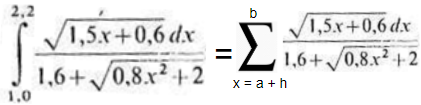
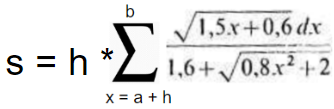
**9.1)**

|  |  |  |
| --- | --- | --- |
| **n**  **Количество**  **разбиений** | **h**  **Шаг** | **I**  **Метод левых**  **частей**  **прямоугольников** |
| **10** | **0,12** |  |
| **100** | **0,012** |  |
| **1000** | **0,0012** |  |
| **10000** | **0,00012** |  |

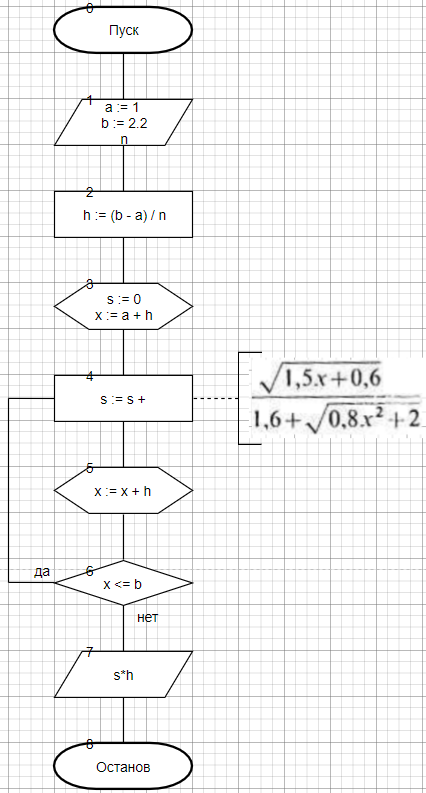
**10.1)** В ходе решения поставленной задачи использовались одни и те же математические модели. Переменная “x” типа real - параметр цикла. Переменная “s” типа real содержит вычисленное выражение для последующего вывода на экран.

**Задание 2:** метод правых частей прямоугольников.

**4.2)** Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника правых частей.

**5.2)** , 

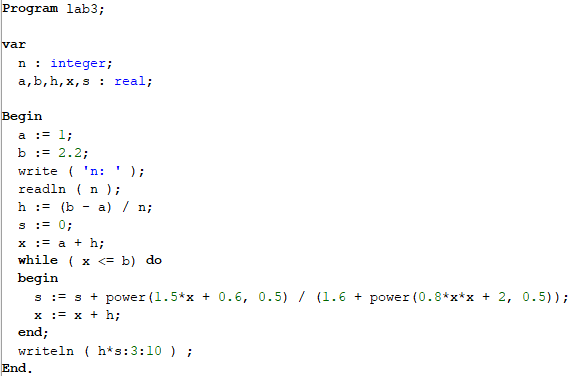
**6.2)**



**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **n** | **Переменная, на сколько частей разбит отрезок [a;b]** | **Int** |
| **a** | **Переменная, левая граница криволинейной функции** | **Real** |
| **b** | **Переменная, правая криволинейной функции** | **Real** |
| **h** | **Переменная, шаг** | **Real** |
| **x** | **Переменная, параметр цикла** | **Real** |
| **s** | **Переменная, площадь** | **Real** |

**8.2)**



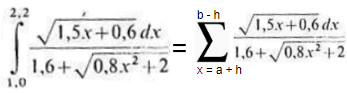
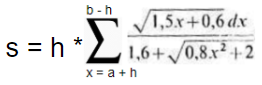
**9.2)**

|  |  |  |
| --- | --- | --- |
| **n**  **Количество**  **разбиений** | **h**  **Шаг** | **I**  **Метод правых**  **частей**  **прямоугольников** |
| **10** | **0,12** |  |
| **100** | **0,012** |  |
| **1000** | **0,0012** |  |
| **10000** | **0,00012** |  |

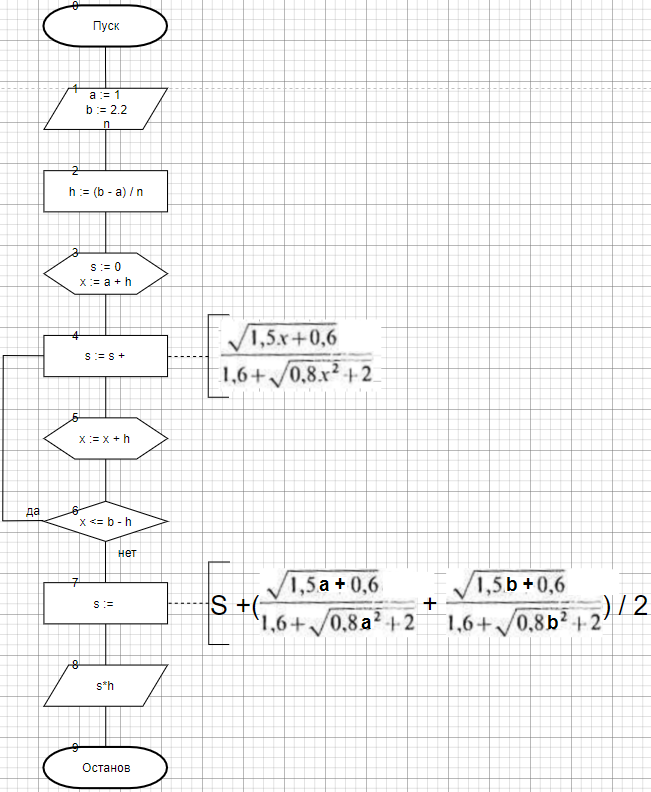
**10.2)** В ходе решения поставленной задачи использовались одни и те же математические модели, а также для возведения в степень помогла функция power . Переменная “x” типа real - параметр цикла, где его начальное значение равняется a+h. Шаг “h” типа int выражается (b – a) /n. Переменная “s” типа real содержит вычисленное выражение для последующего вывода на экран. Для получения более точного результата увеличивалось количество разбиений “n” области [a,b].

**Задание 3:** метод трапеций.

**4.3)** Написать программу для вычисления определенного интеграла из индивидуального задания методом трапеций.

**5.3)** ,

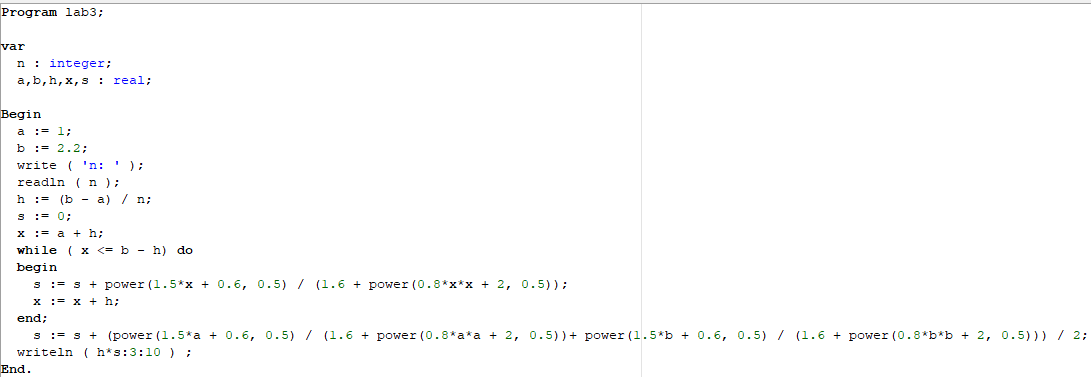
**6.3)**



**7.3)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| n | Переменная, на сколько частей разбит отрезок [a;b] | Int |
| a | Переменная, левая граница криволинейной функции | Real |
| b | Переменная, правая граница криволинейной функции | Real |
| h | Переменная, шаг | Real |
| x | Переменная, параметр цикла | Real |
| s | Переменная, площадь | Real |

**8.3)**



**9.3)**

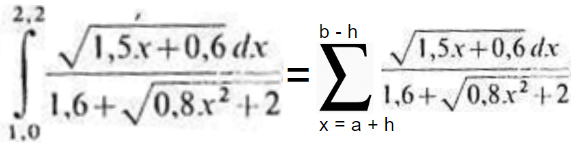
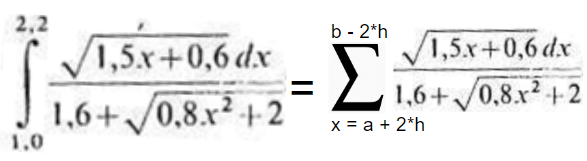
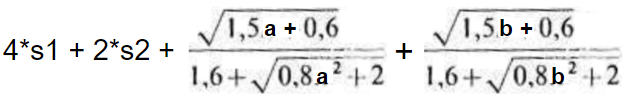
|  |  |  |
| --- | --- | --- |
| **n**  **Количество**  **разбиений** | **h**  **Шаг** | **I**  **Метод**  **трапеций** |
| **10** | **0,12** |  |
| **100** | **0,012** |  |
| **1000** | **0,0012** |  |
| **10000** | **0,00012** |  |

**10.3)** В ходе решения поставленной задачи методом трапеций использовались одни и те же математические модели, а также для возведения в степень потребовалась функция power. Переменная “x” типа real - параметр цикла, где его начальное значение равняется a + h. Шаг “h” типа int выражается (b – a) / n. Переменная “s” типа real содержит вычисленное выражение для последующего вывода на экран. При увеличении количества дроблений “n” отрезка [a, b] точность результатов возрастала.

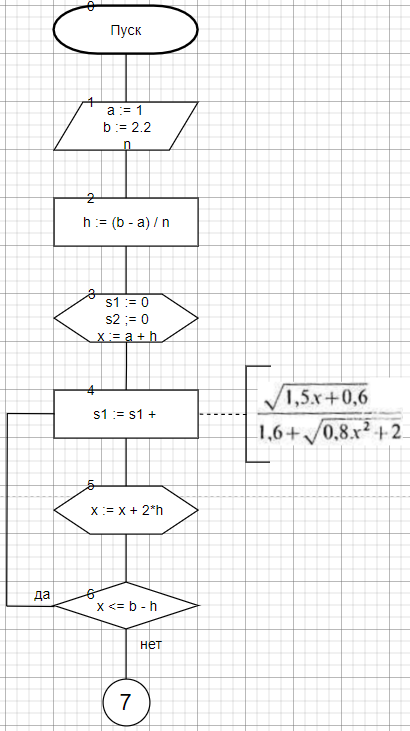
**Задание 4:** метод парабол.

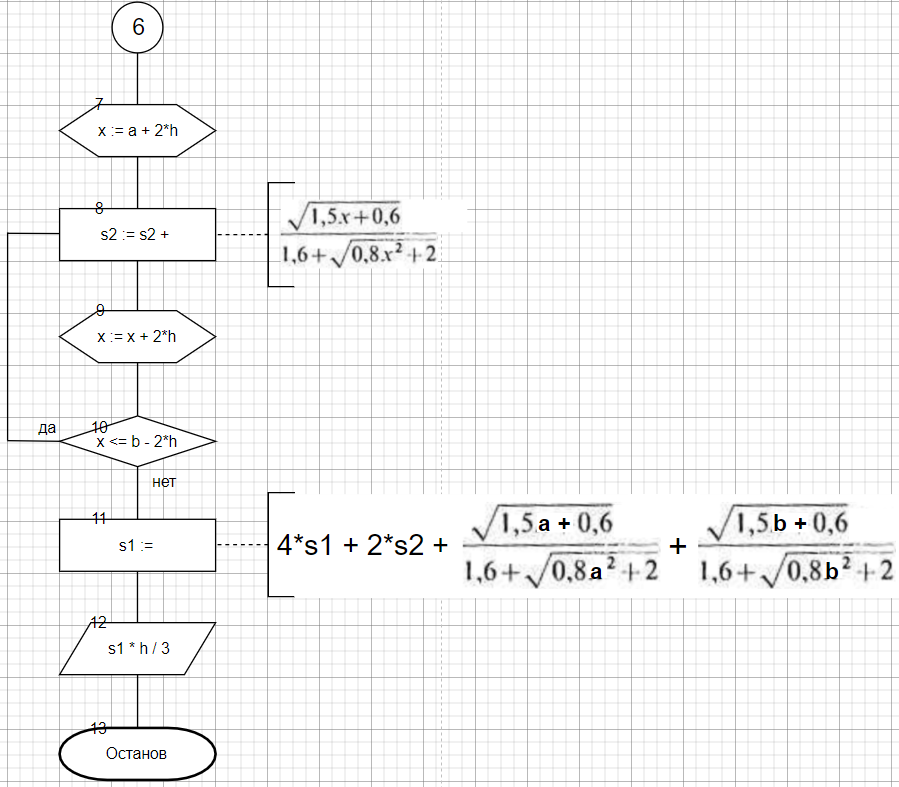
**4.4)** Написать программу для вычисления определенного интеграла из индивидуального задания методом парабол.

**5.4)**

,,

**6.4)**

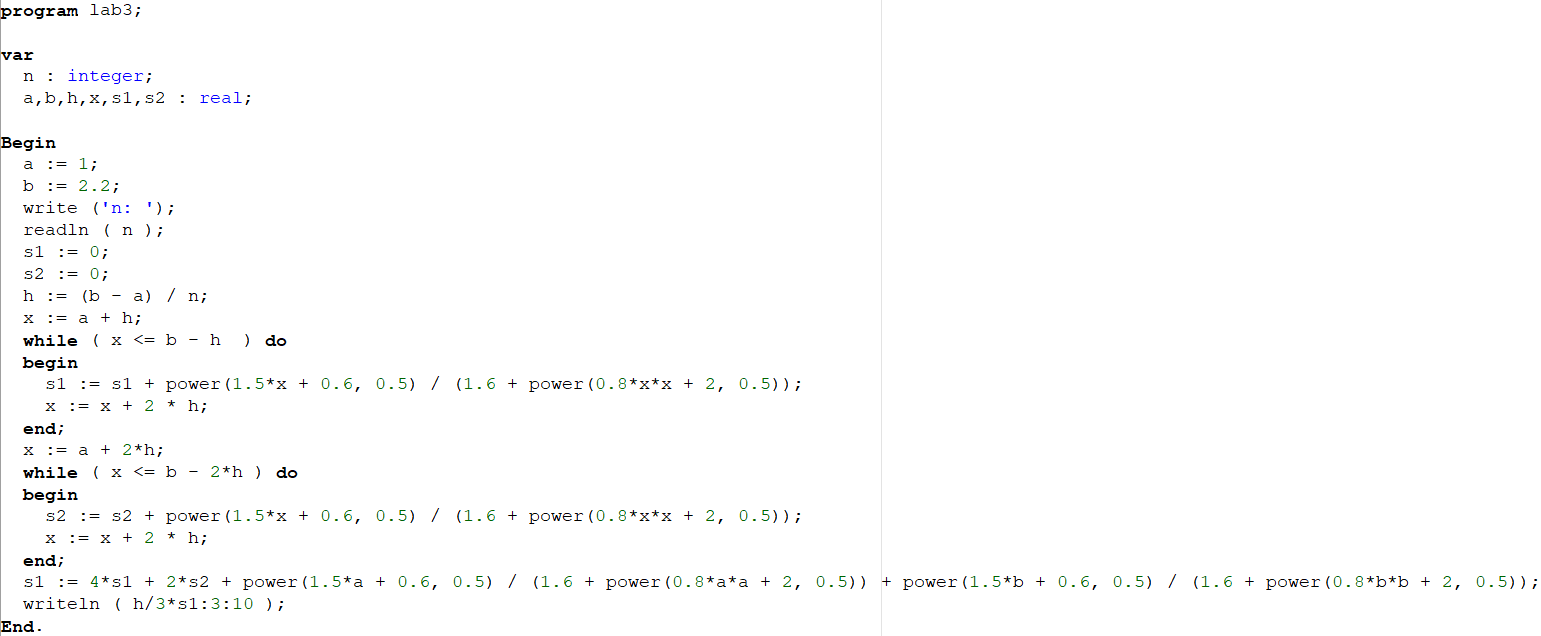




**7.4)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| n | Переменная, на сколько частей разбит отрезок [a;b] | Int |
| a | Переменная, левая граница криволинейной функции | Real |
| b | Переменная, правая граница криволинейной функции | Real |
| h | Переменная, шаг | Real |
| x | Переменная, параметр цикла | Real |
| s1 | Переменная, сумма некой части, а после итоговая сумма | Real |
| s2 | Переменная, сумма некой части | Real |

**8.4)**



**9.4)**

|  |  |  |
| --- | --- | --- |
| **n**  **Количество**  **разбиений** | **h**  **Шаг** | **I**  **Метод парабол** |
| **10** | **0,12** |  |
| **100** | **0,012** |  |
| **1000** | **0,0012** |  |
| **10000** | **0,00012** |  |

**10.4)** В ходе решения задачи методом парабол использовались одни и те же математические модели, а также для возведения в степень потребовалась функция power. Для реализации алгоритма понадобились два цикла с разными условиями, так для первого цикла условие остановки при x <= b – h, а у второго при x <= b – 2\*h. Переменная “x” типа real - параметр цикла, где его начальное значение равняется a + h для первого цикла, для второго же a + 2\*h. Шаг “h” типа int выражается (b – a) / n. Переменная “s” типа real содержит вычисленное выражение для последующего вывода на экран. Можно заметить, что точность результатов не всегда возрастает при увеличении количества дроблений отрезка [a, b].

**11.1)** Мы научились численному интегрированию с помощью алгоритмов детерминированных циклических вычислительных процессов с управлением по аргументу средствами компилятора free Pascal.

**11.2)** Проанализировав результаты каждого из методов, можно сказать, что самый точный результат получился у метода правых частей прямоугольников, а то есть , при количестве разбиений равном 10000.

**11.3)** Чтобы увеличить точность вычисления каждого из методов, нужно разбивать участок [a; b] на большее количество n равных частей, это можно увидеть из результатов вычислений каждым методом при изменении n. Также чем больше знаков после запятой, тем точнее получается результат. Можно еще добавить, что у метода парабол при увеличении n не всегда получается результат точнее, что наглядно представлено в таблице метода парабол.