*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА: КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе (домашнему заданию) № 1**

**Название лабораторной работы: *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Дисциплина:** Основы программирования

Студент гр. ИУ6-12Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В.Астахов**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

**I вариант**

Задание 1.1

Создайте новый проект в отдельной папке и введите программу, представленную ниже, заменив выражения в фигурных скобках соответствующими операторами. Создайте новый проект в отдельной папке и введите программу, представленную ниже, заменив выражения в фигурных скобках соответствующими операторами.

Текст программы:

*program example2;*

*{$APPTYPE CONSOLE}*

*uses*

*SysUtils;*

*Var*

*y: real;*

*begin*

*y := 1;*

*WriteLn('До преобразований y=', y:20:16);*

*y := y / 6000;*

*y := exp(y); { y = ex }*

*y := sqrt(y); { Квадратный корень }*

*y := y / 14;*

*y := 14\*y;*

*y := sqr(y); { Y = y2 }*

*y := ln(y);*

*y := 6000 \* y;*

*WriteLn('После преобразований = ', y:20:16);*

*writeln('абсолютная погрешность = ',#13#10,(abs(y-1) ):20:16);*

*writeln('относительная погрешность = ',#13#10,(abs(y-1) / abs(y) ):20:16);*

*writeln('предельная относительная погрешность = ',(abs(y-1) / abs(y- abs(y-1) ) ):20:16 );*

*readln;*

*end.*

|  |  |  |
| --- | --- | --- |
| Абсолютная погрешность | Относительная погрешность | Предельная относительная погрешность |
| 0.0000000000012632 | 0.0000000000012632 | 0.0000000000012632 |

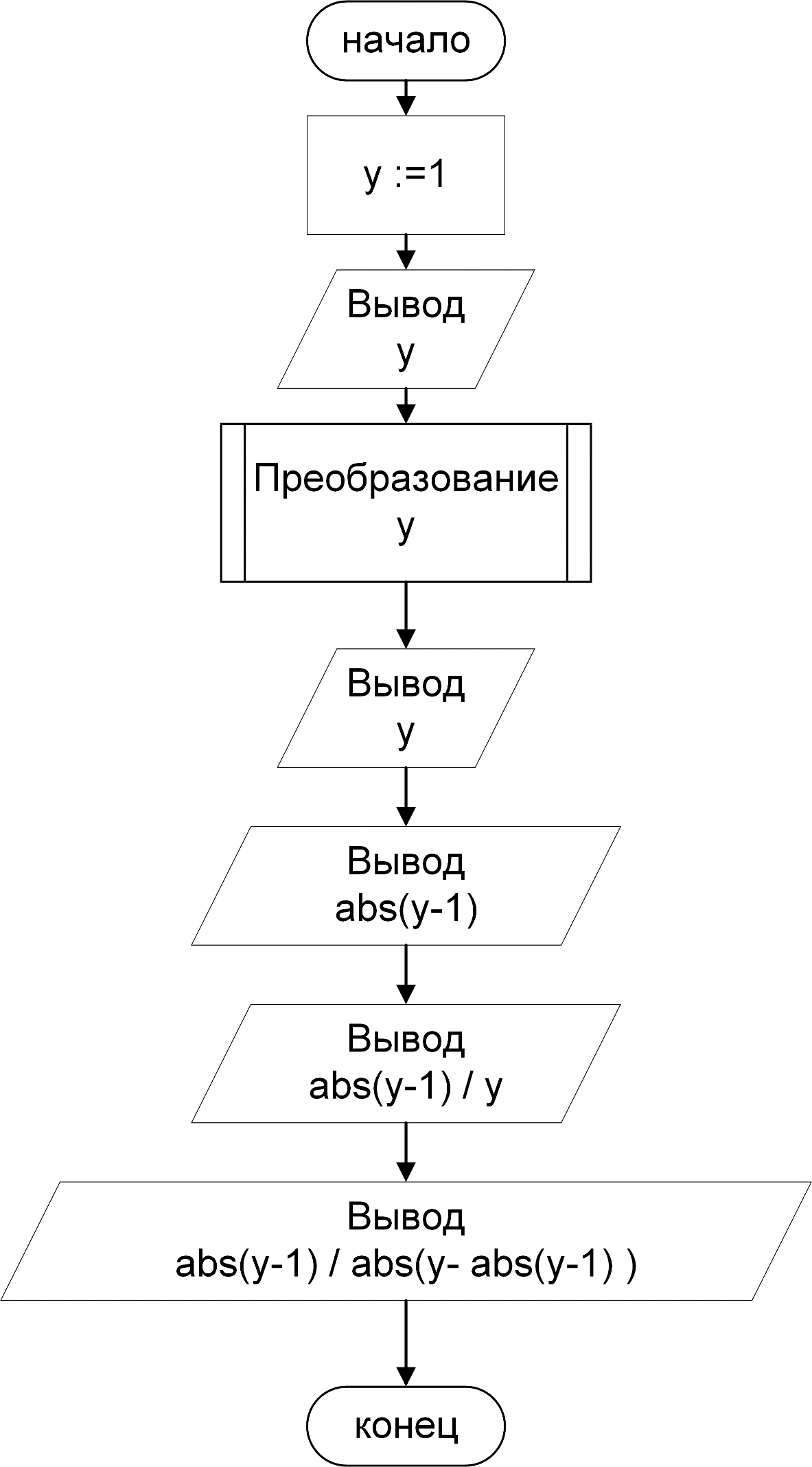
Данные погрешности относятся к типам:

1. связанных с использованием приближенных значений параметров, например, любых физических констант (начальные погрешности);
2. связанных с ограниченным количеством разрядов, используемых для представления чисел (погрешности округления);

Вывод:

* При программировании вычислений возникают погрешности, полученный результат отличается от истинного.
* Существует 3 основных метода оценки погрешностей.
* Существует множество факторов, влияющих на величину погрешности.
* Погрешности в вычислении чисел типа real незначительны для большинства задач, однако для высокоточных операций их разрядной сетки может быть недостаточно.

Схема алгоритма:



Задание 1.2

Из математики известно, что ch2x – sh2x = 1. Разработайте программy, которая вычисляет левую часть этого равенства.

Указание. Программа должна реализовывать следующую последовательность вычислений: y1=sh x, y2 =ch x, y = y22 – y12, где x, y, y1, y2 – переменные типа real. Полученные значения y1, y2 и y вывести на экран, указав ширину поля вывода не менее 20 и количество дробных цифр не менее 16. Заполните таблицу и объясните результат. Измените в программе типы переменных на double. Объясните полученный результат. Типы каких переменных реально влияют на точность результата и почему?

Текст программы:

*program Project1;*

*{$APPTYPE CONSOLE}*

*uses*

*System.SysUtils;*

*var*

*x, y1, y2, y, delta, sigma: real;*

*begin*

*writeln('Enter X');*

*readln(x);*

*y1 := (exp(x) + exp(-x)) / 2;*

*y2 := (exp(x) - exp(-x)) / 2;*

*y := sqr(y1) - sqr(y2);*

*delta := abs(y - 1);*

*sigma := delta / abs(y);*

*writeln('y1, y2,y : ',y1:28:20,' ',y2:28:20,' ',y:28:20);*

*writeln('абсолютная погрешность: ', delta:22:20);*

*writeln('относительная погрешность: ', sigma:22:20);*

*readln;*

*end.*

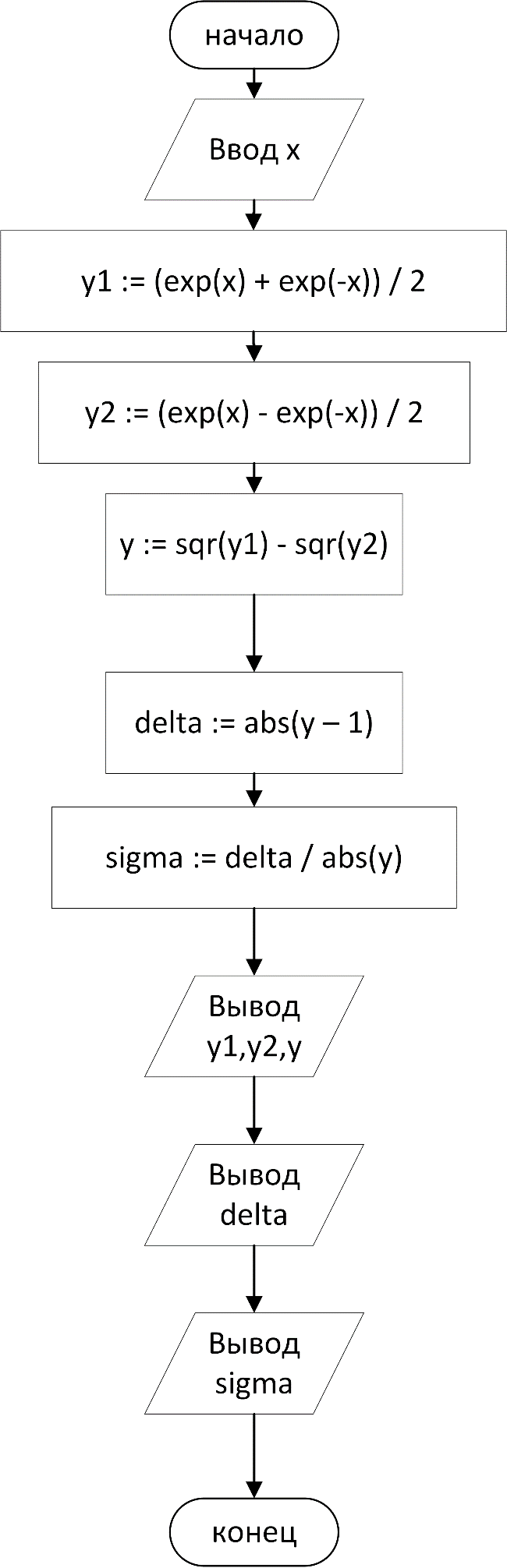
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y1 | y2 | y | Δ | δ |
| 8,25 | 1913.81304134923175000000 | 1913.812  78009067455000000 | 0.99999999962005858800 | 0.00000000037994141167 | 0.00000000037994141181 |
| 8,32 | 2052.58012593502781000000 | 2052.579  88233916376000000 | 0.99999999927467797500 | 0.00000000072532202466 | 0.00000000072532202519 |
| 8,45 | 2337.53647470609940000000 | 2337.536  26080568438000000 | 0.99999999837200448400 | 0.00000000162799551617 | 0.00000000162799551882 |
| 8,55 | 2583.37731036054720000000 2583.377 | 2583.377  11681544761000000 | 1.00000000017280399000 | 0.00000000017280399334 | 0.00000000017280399331 |
| 8,65 | 2855.07345444017847000000 | 2855.073  27931333066000000 | 0.99999999800002115100 | 0.00000000199997884920 | 0.00000000199997885320 |
| 8,75 | 3155.34413327517450000000 | 3155.343  97481384940000000 | 0.99999999989904608800 | 0.00000000010095391190 | 0.00000000010095391191 |

Вывод:

* Данная программа демонстрирует, что при операциях над близкими по значению числами погрешность вычисления возрастает (для x=8.25, 8.32, 8.75 y1 и y2 более далеки по значению, чем для других входных данных и при этом погрешности при подсчете результата значительно меньше).
* При использовании типа double для переменных y, y1, y2 точность возрастает, т.к. над ними совершаются действия с экспонентой, а число Эйлера иррационально (и должно быть представлено в функции exp() с высокой точность) => так или иначе в ходе вычисления происходит потеря точности, но за счет большей разрядной сетки у double она уменьшается.

Примечание: использование double для x в данном конкретном тесте незначительно влияет на точность, т.к. все входные данные умещаются в разрядную сетку real.

Схема алгоритма:



Задание 1.3

Разработайте программу, которая проверяет равенство sin2 x + cos2 x = 1. Убедитесь, что погрешность достаточно мала. Поясните полученный результат.

Текст программы:

*program Project1;*

*{$APPTYPE CONSOLE}*

*uses*

*System.SysUtils;*

*var*

*x, leftside: real;*

*eps: double;*

*begin*

*writeln('Enter X, Eps');*

*readln(x, eps);*

*leftside := sqr(sin(x)) + sqr(cos(x));*

*if (abs(leftside - 1) < eps) then*

*writeln('sin^2(x)+cos^2(x) equals 1')*

*else*

*writeln('sin^2(x)+cos^2(x) DOES NOT equal 1');*

*readln;*

*end.*

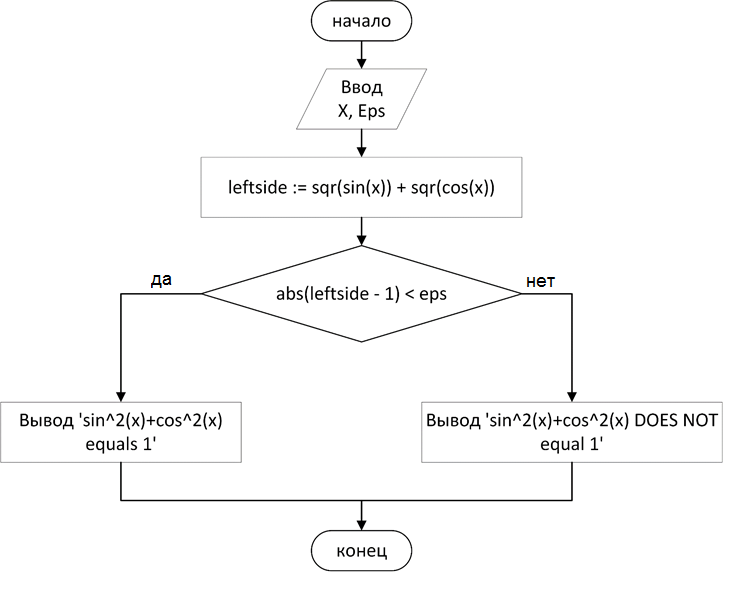
Тесты:

|  |  |
| --- | --- |
| Ввод | Вывод |
| 1 1e-10 | sin^2(x)+cos^2(x) equals 1 |
| 5 1e-50 | sin^2(x)+cos^2(x) equals 1 |
| 9 1e-200 | sin^2(x)+cos^2(x) equals 1 |
| 7 1e-500 | sin^2(x)+cos^2(x) DOES NOT equal 1 |
| 11 1e-1000 | sin^2(x)+cos^2(x) DOES NOT equal 1 |

Вывод:

Погрешность очень мала, т.к. из всех возможных факторов в данной программе на точность влияет лишь ограниченность разрядной сетки. Особенно важным является тот факт, что не происходит вычитания близких чисел.

Схема алгоритма:



Задание 2

Ввести действительные числа X, Y и W. Определить max2(X+Y+W, X\*Y\*W)

Текст программы:

*program Project1;*

*{$APPTYPE CONSOLE}*

*uses*

*System.SysUtils;*

*var*

*x, y, w, r1, r2: real;*

*begin*

*writeln('Enter x,y,w');*

*readln(x, y, w);*

*r1 := x + y + w;*

*r2 := x \* y \* w;*

*if (sqr(r1) > sqr(r2)) then*

*begin*

*writeln('(x+y+w)^2 > (x\*y\*w)^2')*

*end*

*else*

*begin*

*if (sqr(r1) < sqr(r2)) then*

*writeln('(x+y+w)^2 < (x\*y\*w)^2')*

*else*

*writeln('(x+y+w)^2 = (x\*y\*w)^2');*

*end;*

*readln;*

*end.*

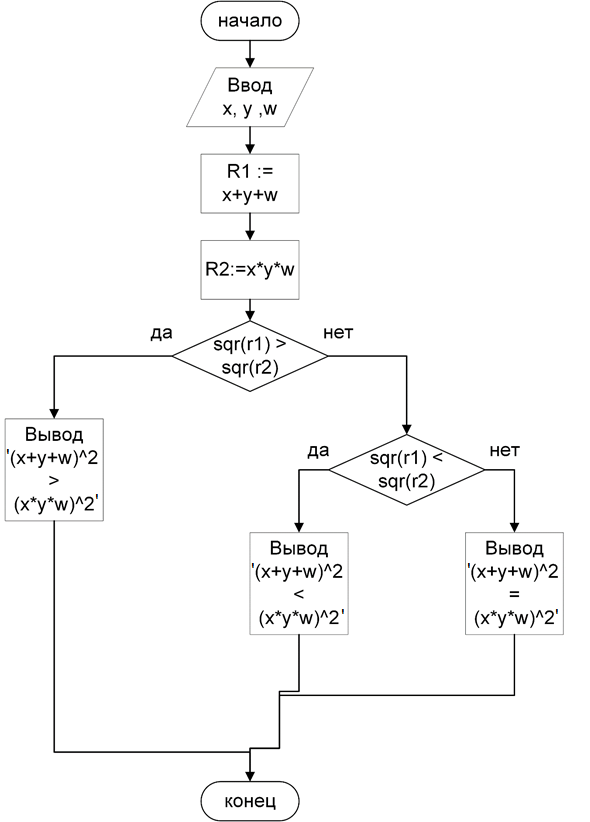
Тесты

|  |  |  |
| --- | --- | --- |
| Ввод | Ожидаемый вывод | Вывод |
| 0 1 -1 | (x+y+w)^2 = (x\*y\*w)^2 | (x+y+w)^2 = (x\*y\*w)^2 |
| 2 2 2 | (x+y+w)^2 < (x\*y\*w)^2 | (x+y+w)^2 < (x\*y\*w)^2 |
| 1 1 1 | (x+y+w)^2 > (x\*y\*w)^2 | (x+y+w)^2 > (x\*y\*w)^2 |
| 5 3 8 | (x+y+w)^2 < (x\*y\*w)^2 | (x+y+w)^2 < (x\*y\*w)^2 |

Вывод:

Turbo Delphi позволяет программировать ветвящиеся процессы. В том числе процессы со вложенными операторами ветвления.

Схема алгоритма:



Задание 3

Решить задачу с точностью ξ, организовав итерационный цикл. Значение точности вводится с клавиатуры. Найти первый член последовательности y=n/(n2+2), для которого y ≤ ξ. Определить, как изменяется число итераций при изменении точности. Проверить программу при ξ=10-2, 10-4.

Текст программы:

*program Project1;*

*{$APPTYPE CONSOLE}*

*uses*

*System.SysUtils;*

*var*

*eps, y: real;*

*n: integer;*

*begin*

*writeln('Введите точность: ');*

*readln(eps);*

*n := 1;*

*y := n / (sqr(n) + 2);*

*while not(y <= eps) do*

*begin*

*n:=n+1;*

*y := n / (sqr(n) + 2);*

*end;*

*writeln('y <= eps: ',y:20:16);*

*writeln('n= ',n);*

*readln;*

*end.*

Тесты

|  |  |  |
| --- | --- | --- |
| Ввод | Ожидаемый вывод (для n) | Вывод (для n) |
| 0.1 | 10 | 10 |
| 1e-2 | 100 | 100 |
| 1e-4 | 10000 | 10000 |

Ответ:

В данной программе число итераций обратно точности (n = 1/eps).

Вывод:

* Число итераций при расчетах с погрешностью напрямую связано с заданной точностью.
* Пределы последовательностей в программировании вычисляются не точно, а с задаваемой пользователем погрешностью.

Схема алгоритма:

