

Часть 1. Вычисления. Погрешности вычислений

Цель работы

Убедиться в наличии погрешности вычислений при использовании компьютера, определить их типы, установить их абсолютные и относительные значения, оценить связь между типами данных переменных и погрешностями значений, хранимых в этих переменных.

Задание

Задание 1

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

```
program example2;
```

```
{ $APPTYPE CONSOLE }
```

```
uses
```

```
  SysUtils;
```

```
Var
```

```
  y:real;
```

```
begin
```

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

```
    { y=1 }
```

```
    { y= y/6000 }
```

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

```
  y:= sqrt(y);          { Квадратный корень }
```

```
    { y = y / 14 }
```

```
    { y = 14y }
```

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

```
    { y = ln y }
```

```
{ y = 6000y }
```

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

```
end.
```

2. Выполните оценку абсолютной и относительной погрешности представления числа 1 и вычислений над числами типа *real*. К каким типам относятся данные погрешности (см. список типов погрешностей на предыдущей странице)?
3. Текст программы и результаты занесите в отчет.

Задание 2

Из математики известно, что $\operatorname{ch}^2 x - \operatorname{sh}^2 x = 1$, где $\operatorname{ch} x = \frac{e^x + e^{-x}}{2}$, $\operatorname{sh} x = \frac{e^x - e^{-x}}{2}$.

Разработайте программу, которая вычисляет левую часть этого равенства.

Указание. Программа должна реализовывать следующую последовательность вычислений: $y_1 = \operatorname{sh} x$, $y_2 = \operatorname{ch} x$, $y = y_2^2 - y_1^2$, где x , y , y_1 , y_2 – переменные типа *real*.

Полученные значения y_1 , y_2 и y вывести на экран, указав ширину поля вывода не менее 20 и количество дробных цифр не менее 16.

2. Текст программы и ее результаты занесите в отчет.
3. Последовательно вводя указанные значения аргумента и рассчитывая погрешности вычислений, заполните таблицу.

x	y1	y2	y	Δ	δ
5					
10					
15					
20					
25					

4. Поясните полученный результат и объяснения включите в отчет.
5. Измените в программе типы переменных x , y , y_1 , y_2 на *double*. Повторите опыт и заполните аналогичную таблицу. Повторите опыт с типами *single* и *extended*. Сравните четыре таблицы и объясните полученные результаты.

6. Ответьте на вопрос: изменение типа данных каких переменных (из x , y , y_1 , y_2) реально влияет на точность результата и почему?

Задание 3

Разработайте программу, которая проверяет равенство $\sin^2 x + \cos^2 x = 1$.

Убедитесь, что погрешность достаточно мала. Поясните полученный результат.

Проект программы

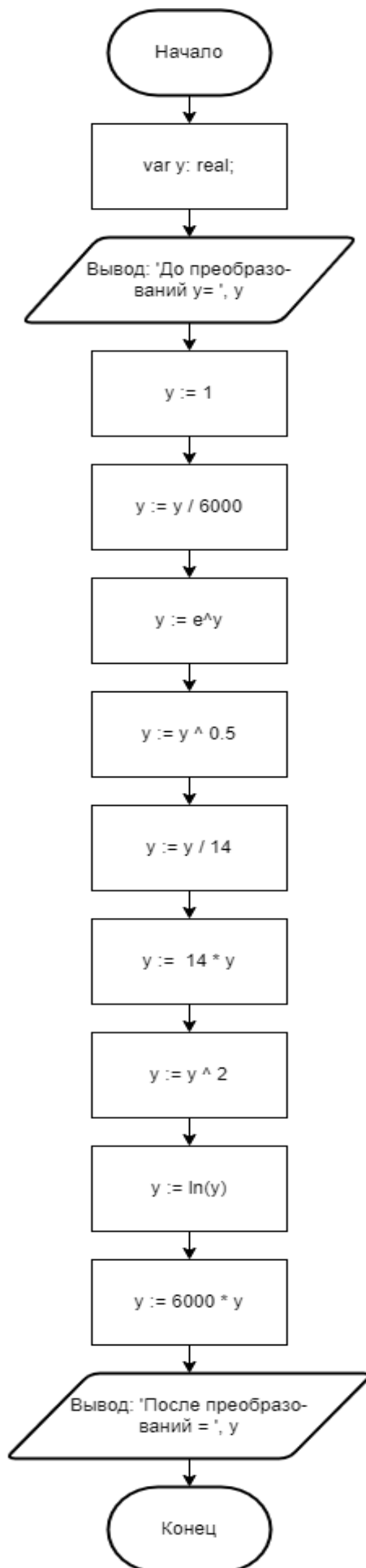


Рис. 1. Проект программы задания 1

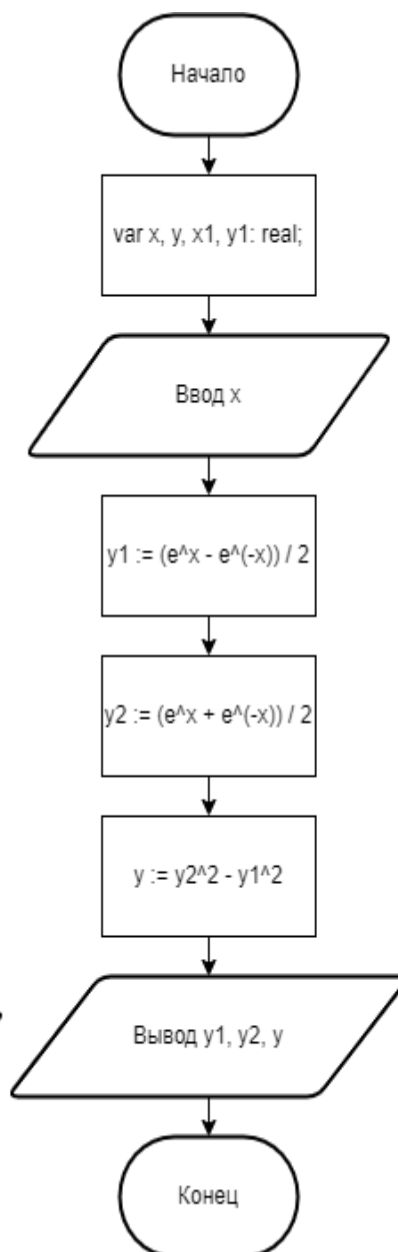


Рис. 2. -//- 2

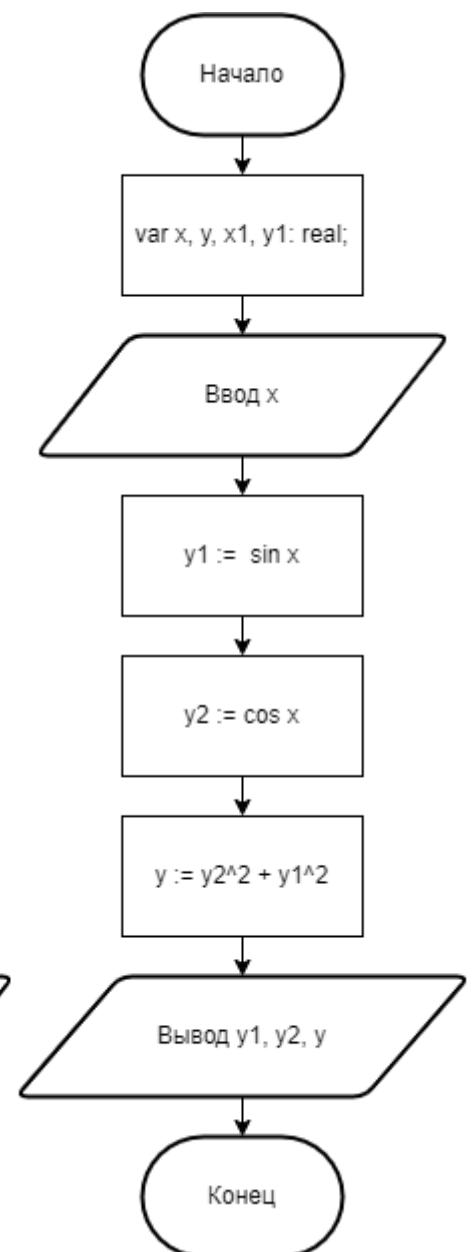
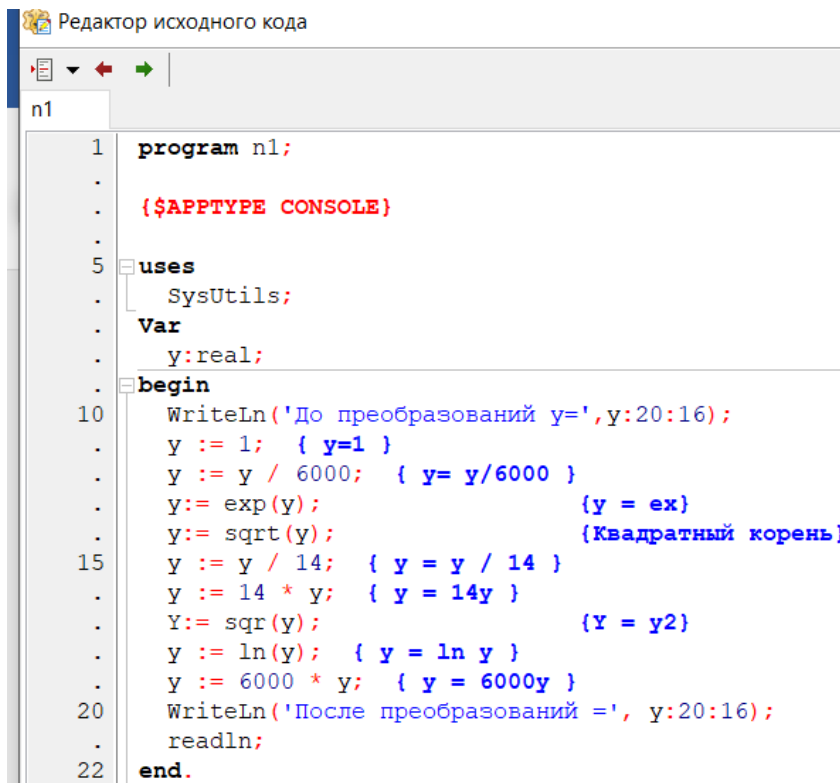


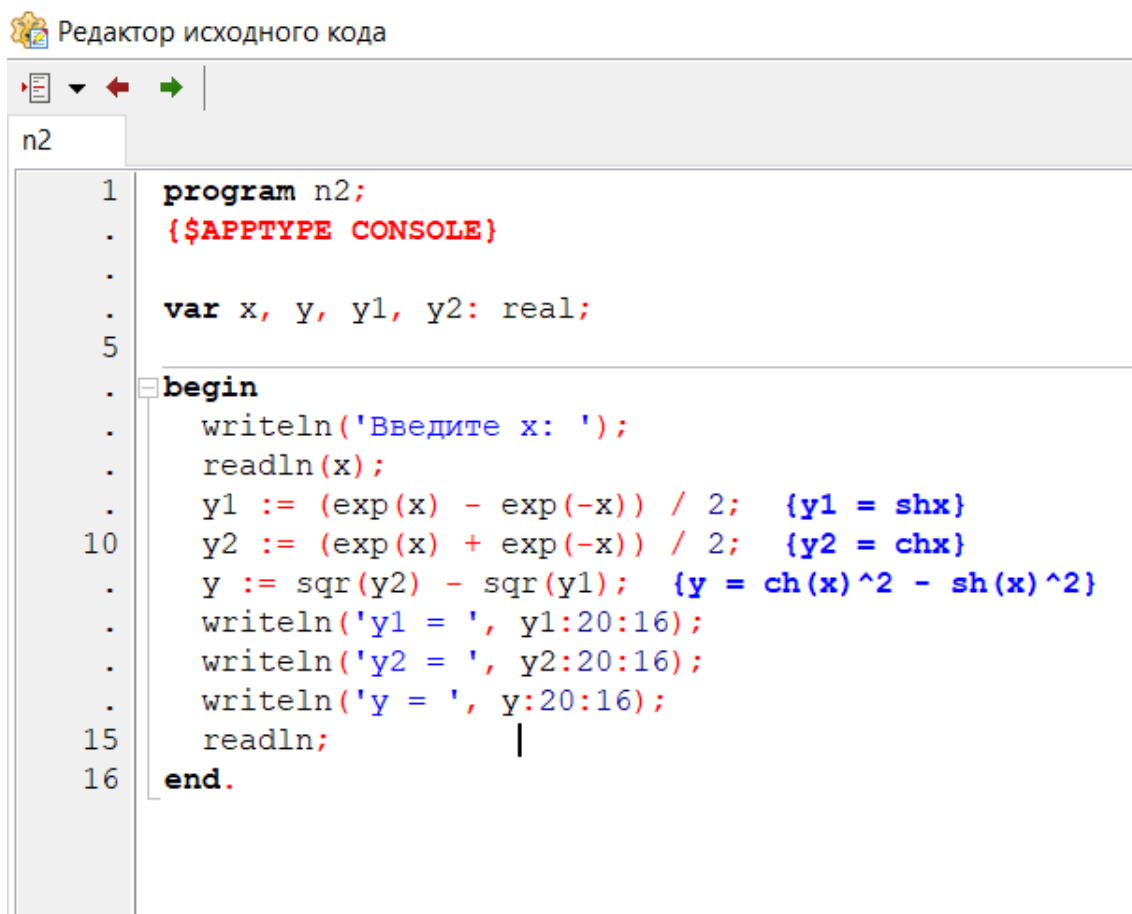
Рис. 3. -//- 3

Текст программы



```
1  program n1;
.
.  {$APPTYPE CONSOLE}
.
5  uses
.  SysUtils;
.  Var
.  y:real;
.
.  begin
10  WriteLn('До преобразований y=', y:20:16);
.  y := 1; { y=1 }
.  y := y / 6000; { y= y/6000 }
.  y:= exp(y); {y = ex}
.  y:= sqrt(y); {Квадратный корень}
15  y := y / 14; { y = y / 14 }
.  y := 14 * y; { y = 14y }
.  Y:= sqr(y); {Y = y2}
.  y := ln(y); { y = ln y }
.  y := 6000 * y; { y = 6000y }
20  WriteLn('После преобразований =', y:20:16);
.  readln;
22  end.
```

Рис. 4. Текст программы задания 1.



```
1  program n2;
.  {$APPTYPE CONSOLE}
.
.  var x, y, y1, y2: real;
5
.  begin
.  writeln('Введите x: ');
.  readln(x);
.  y1 := (exp(x) - exp(-x)) / 2; {y1 = shx}
10  y2 := (exp(x) + exp(-x)) / 2; {y2 = chx}
.  y := sqr(y2) - sqr(y1); {y = ch(x)^2 - sh(x)^2}
.  writeln('y1 = ', y1:20:16);
.  writeln('y2 = ', y2:20:16);
.  writeln('y = ', y:20:16);
15  readln;
16  end.
```

Рис. 5. Текст программы задания 2.

```

1  program n3;
.
.  {$APPTYPE CONSOLE}
.
5  var x, y1, y2, y: real;
.  begin
.      writeln('Enter x: ');
.      readln(x);
.      y1 := sin(x);
10     y2 := cos(x);
.      y := sqr(y1) + sqr(y2);
.      writeln('y1 = ', y1:20:16);
.      writeln('y2 = ', y2:20:16);
14     writeln('y = ', y:20:16);
15     readln;
16 end.

```

Рис. 6. Текст программы задания 3.

Тестовые данные и результаты тестирования

Задание 1

Абсолютная погрешность составила $1.0000000000012632 - 1 = 0.0000000000012632$

Относительная погрешность = 0.0000000000012632

Данные погрешности относятся к начальным погрешностям, погрешностям при округлении и погрешностям операций.

Задание 2

Чем больше числа, с которыми производятся операции в условиях ограниченности разрядов для их представления, тем больше погрешность (см. таблицу 1).

Таблица 2. Результаты тестирования программы задания 2. Тип real.

x	y1	y2	y	Δ	δ
5	74.2032	74.20994	1.0000000	0.000000	0.000000
	1057778	8524787	00001819	0000018	0000018

	87520	8480	0	190	190
10	11013.2 3287470 3393000 0	11013.23 2920103 3240000	1.0000000 29802322 4	0.000000 0298023 224	0.000000 0298023 224
15	1634508 .686235 9024000 000	1634508. 6862362 0830000 00	1.0000000 00000000 0	0	0
20	2425825 97.7048 9514000 00000	2425825 97.70489 5140000 0000	0.0000000 00000000 0	1	1
25	3600244 9668.69 2940000 0000000	3600244 9668.692 9400000 000000	0.0000000 00000000 0	1	1

Таблица 3. Результаты тестирования программы задания 2. Тип double.

x	y1	y2	y	Δ	δ
5	74.2032 1057778 87520	74.20994 8524787 8480	1.0000000 00001819 0	0.000000 0000018 190	0.000000 0000018 190
10	11013.2 3287470 3393000 0	11013.23 2920103 3240000	1.0000000 29802322 4	0.000000 0298023 224	0.000000 0298023 224
15	1634508 .686235 9024000 000	1634508. 6862362 0830000 00	1.0000000 00000000 0	0	0

20	2425825 97.7048 9514000 00000	2425825 97.70489 5140000 0000	0.0000000 00000000 0	1	1
25	3600244 9668.69 2940000 0000000	3600244 9668.692 9400000 000000	0.0000000 00000000 0	1	1

Таблица 4. Результаты тестирования программы задания 2. Тип single.

x	y1	y2	y	Δ	δ
5	74.2032 0892000 00000	74.20994 5680000 0000	1.0000000 00000000 0	0	1
10	11013.2 3242000 0000000 0	11013.23 3400000 0000000	24.000000 00000000 00	23	23
15	1634508 .625000 0000000 000	1634508. 6250000 0000000 00	0.0000000 00000000 0	1	1
20	2425825 92.0000 0000000 00000	2425825 92.00000 0000000 0000	0.0000000 00000000 0	1	1
25	3600245 1460.00 0000000 0000000	3600245 1460.000 0000000 000000	0.0000000 00000000 0	1	1

Таблица 5. Результаты тестирования программы задания 2. Тип extended.

x	y1	y2	y	Δ	δ
5	74.2032	74.20994	1.0000000	0.000000	0.000000
	1057778	8524787	00001819	0000018	0000018
	87520	8480	0	190	190
10	11013.2	11013.23	1.0000000	0.000000	0.000000
	3287470	2920103	29802322	0298023	0298023
	3393000	3240000	4	224	224
	0				
15	1634508	1634508.	1.0000000	0	0
	.686235	6862362	00000000		
	9024000	0830000	0		
	000	00			
20	2425825	2425825	0.0000000	1	1
	97.7048	97.70489	00000000		
	9514000	5140000	0		
	00000	0000			
25	3600244	3600244	0.0000000	1	1
	9668.69	9668.692	00000000		
	2940000	9400000	0		
	0000000	000000			

Чем больше двоичных разрядов отведено под число, тем более точно его хранимое значение, то есть меньше погрешность округления, что особенно заметно при сравнении типов single и extended (таблицы 4 и 5 соотв.)

На точность реально влияет изменение переменных y1, y2 – x в случаях выше является целым, погрешность округления у x стремится к 0, точность у напрямую зависит от точности y1 и y2, и только от них, потому что математически $y = 1$.

Задание 3

Таблица 6. Результаты тестирования программы задания 3.

x	y	Δ	δ
5	1.000000000 0000000	0	0
10	1.000000000 0000000	0	0
15	1.000000000 0000000	0	0
20	1.000000000 0000000	0	0
25	1.000000000 0000000	0	0

То, что $\text{abs}(\sin(x)) < 1$ и $\text{abs}(\cos(x)) < 1$, позволяет с высокой точностью вычислять и хранить их значения, в отличие от больших чисел вроде $\cosh(25)^2$. За счет этого обеспечивается малая (практически нулевая) погрешность округления и операций.

Вывод

Погрешности зависят от многих факторов, в частности, от разрядов, доступных для хранения значений, т. е. типа переменной, от величины вычисляемого значения и др. Было подтверждено наличие погрешностей, связанных с означенными факторами, измерены их абсолютные и относительные значения.

Часть 2. Программирование разветвляющегося вычислительного процесса

Вариант 19

Цель работы

Изучить технологию программирования разветвляющихся вычислительных процессов.

Задание

Даны действительные числа x, y . Вычислить $f(x, y)$:

$$f(x, y) = \begin{cases} x - 2y + xy, & \text{при } x < y; \\ (\sin x)x, & \text{при } x = y; \\ y - 2x + 1 & \text{при } x > y. \end{cases}$$

Протестировать все ветви алгоритма.

Проект программы

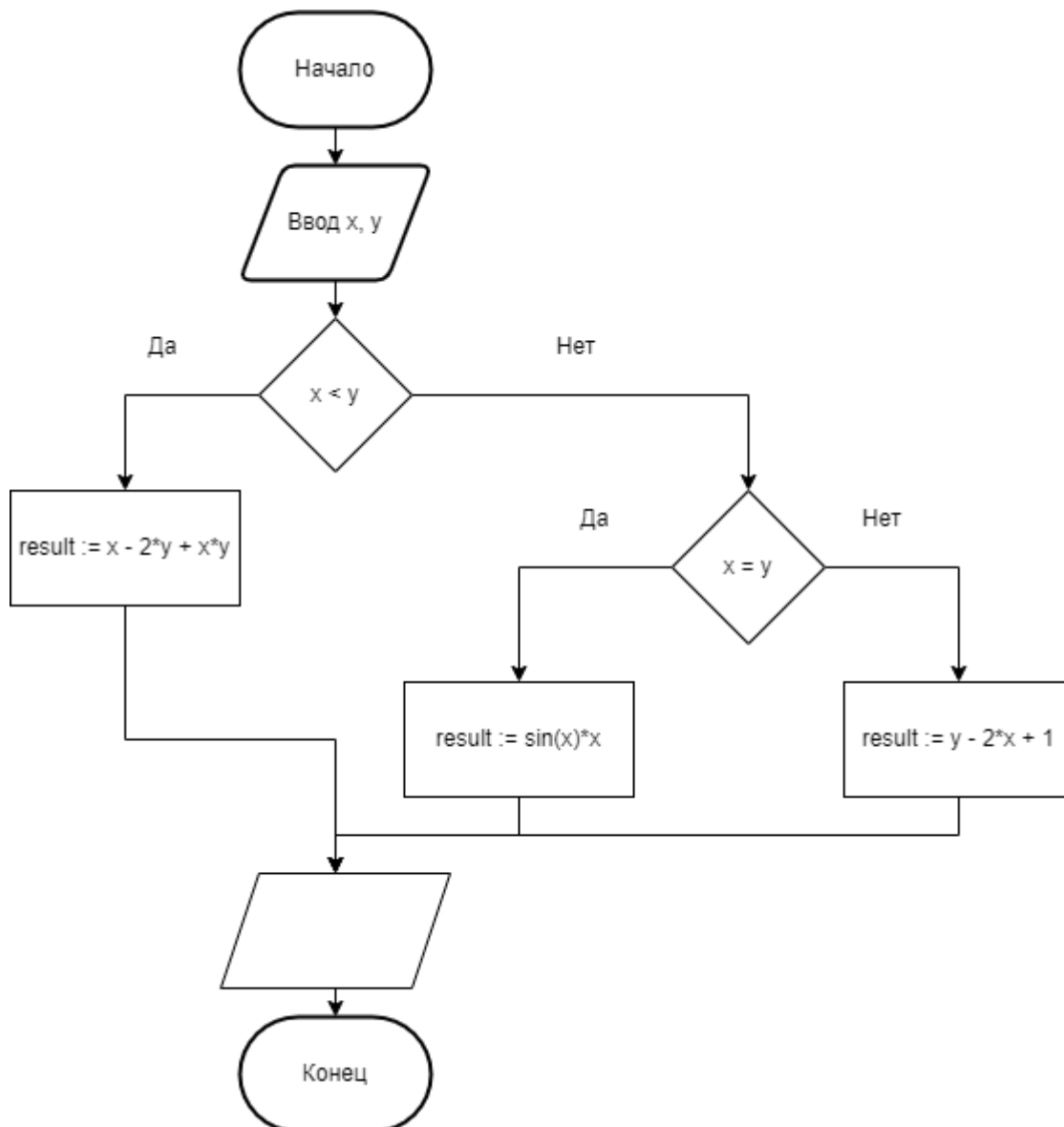


Рис. 7. Проект программы части 2

Текст программы

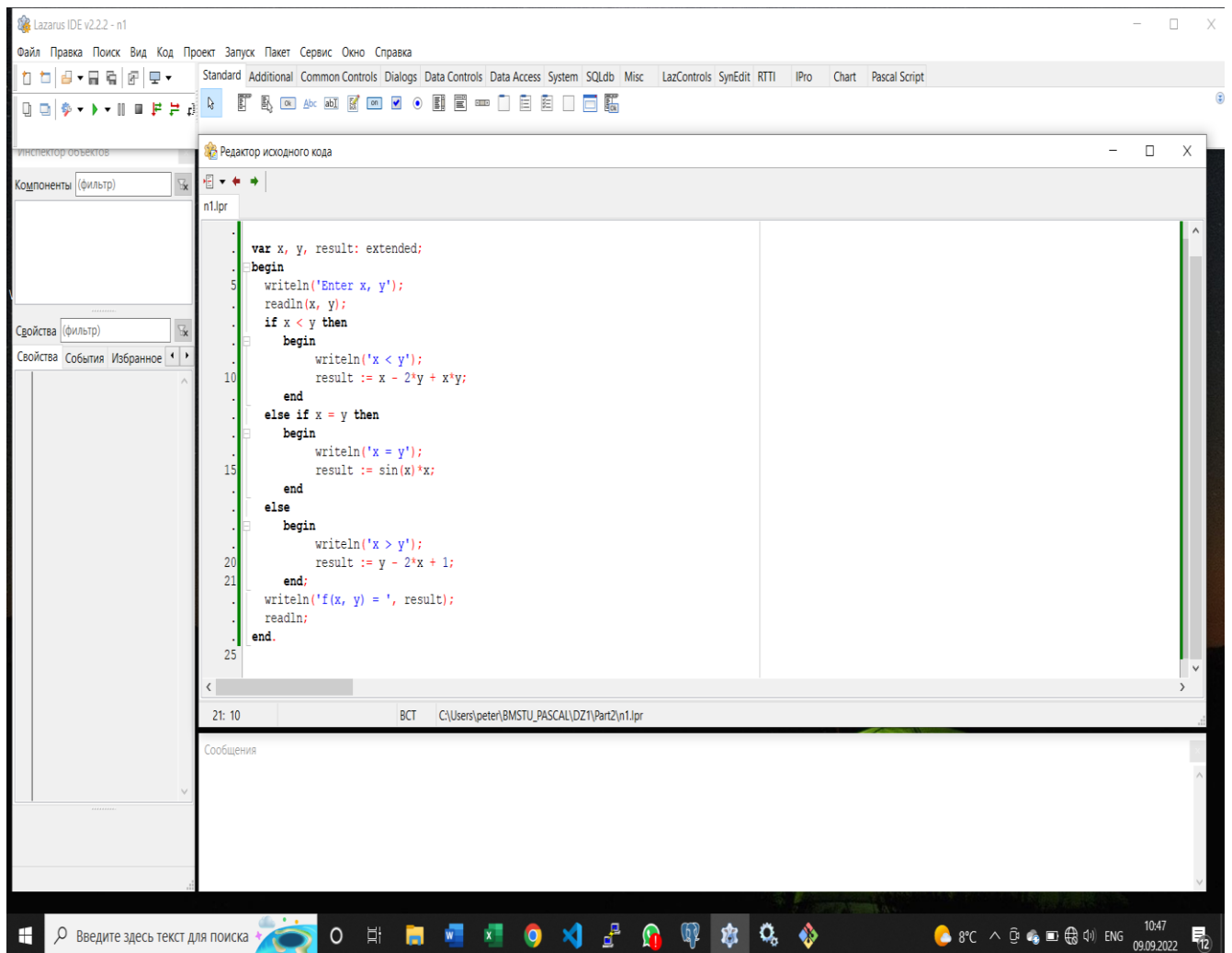


Рис. 8. Текст программы части 2

Тестовые данные и результаты тестирования

Таблица 2. Результаты тестирования программы части 2

x	y	F(x, y)
0	1	-2.0000000000000000E+000
1	2	-1.0000000000000000E+000
1	1	8.4147098480789650E-001
2	2	1.8185948536513632E+000
2	1	-2.0000000000000000E+000
1	0	-1.0000000000000000E+000

Вывод

Был написан разветвляющийся алгоритм на ЯП Pascal, а также составлен проект данного алгоритма.

Часть 3. Программирование циклического процесса.

Типы циклов

Цель работы

Изучить технологию программирования циклических процессов и типы циклов.

Задание

Решить задачу, организовав итерационный цикл.

Вычислить значение $\sin(3)$ с точностью ξ , используя разложение в ряд:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

Значение точности вводится с клавиатуры. Проверить программу при $\xi = 10^{-3}, 10^{-4}$.

Определить, как изменяется число итераций при изменении точности.

Проект программы

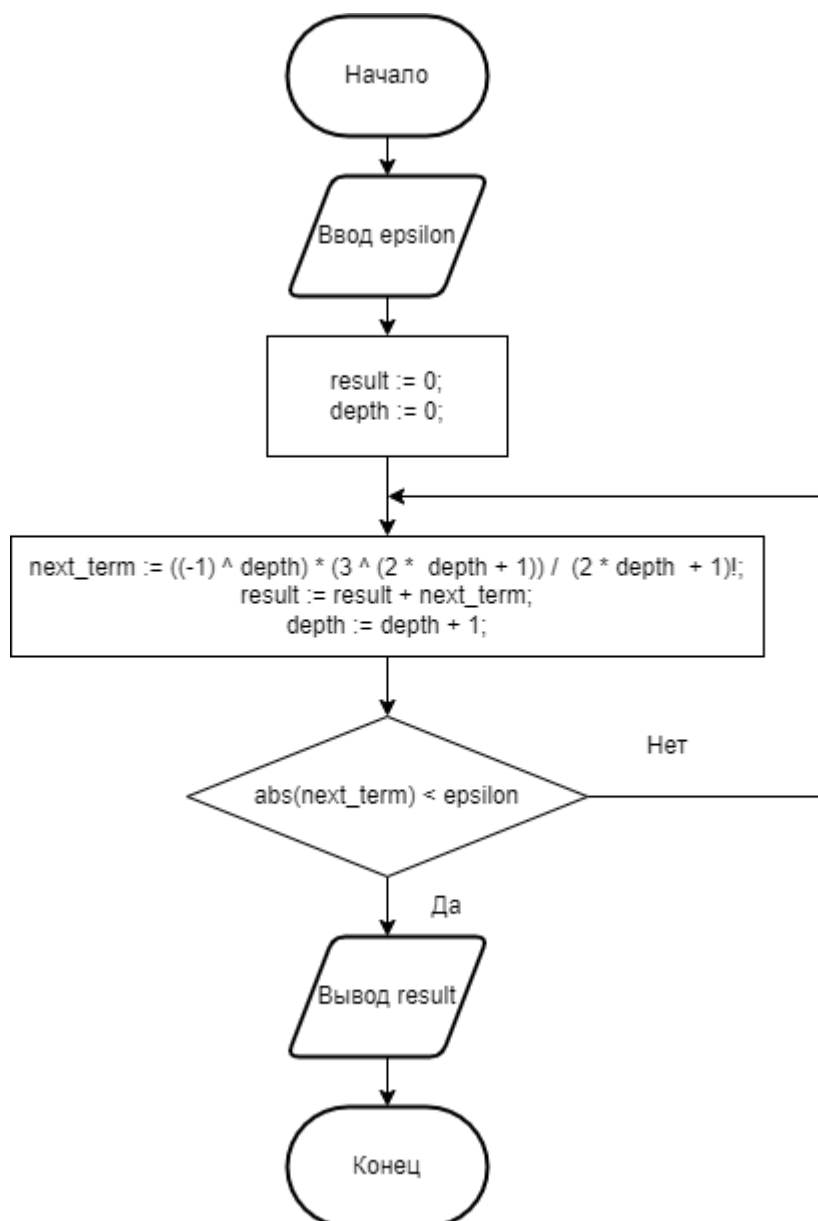


Рис. 9. Проект программы части 3.

Текст программы

Редактор исходного кода

```
n1.lpr
1  program n1;
.
.
.  function Factorial(x: integer): extended;
.      var res: extended; i: longint;
.      begin
.          res := 1;
.          for i := 1 to x do
.              res := res * i;
.          Factorial := res;
10     end;
.
.
.  function Pow(x: extended; power: integer): extended;
.      var i: longint; res: extended;
.      begin
15         res := 1;
.         for i := 1 to power do
.             res := res * x;
.         Pow := res;
.         end;
.
20
.  var eps, result, next_term: real; depth: longint;
.  begin
.      writeln('Enter epsilon');
.      readln(eps);
.      depth := 0;
25     result := 0;
.      repeat
27         next_term := Pow(-1, depth) * Pow(3, 2 * depth + 1) / Factorial(2 * depth + 1);
.         result := result + next_term;
.         depth := depth + 1;
30     until abs(next_term) <= eps;
.      writeln('Sin(3) = ', result, ' iterations = ', depth);
.      readln;
.      end.
35
```

Рис. 10. Текст программы части 3.

Тестовые данные и результаты тестирования

Таблица 3. Результаты тестирования программы части 3.

Epsilon	Sin(3) с точностью epsilon	Количество итераций
0.001	1.4113062718531458E-001	7
0.0001	1.4113062718531458E-001	8

Вывод

Была решена задача, требующая проектирования и построения программы с итерационным циклом с постусловием. Были изучены технология программирования циклических процессов и типы циклов.