МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

УТВЕРЖДАЮ:

Зам. проректора-директора ИПР по химико-технологическому направлению

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Короткова Е.И.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2011

**Работа в среде PascalABC.NET**

Методические указания к выполнению лабораторных работ по дисциплине «Информатика» для студентов 1–го курса, обучающихся по направлениям: 240100 «Химическая технология», 240700 «Биотехнология» и 241000 «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии»

Составители Е.А.Кузьменко, Н.И. Погадаева

Томск 2011

УДК 681.3.01(075.8)

ББК 32.81я73

Работа в среде PascalABC.NET. Методические указания к выполнению лабораторных работ по дисциплине Информатика для студентов 1-го курса, обучающихся по направлениям: 240100 «Химическая технология», 240700 «Биотехнология» и 241000 «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии»/ сост. Е.А. Кузьменко, Н.И. Погадаева; Национальный исследовательский Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2011. – 44 с.

Методические указания рассмотрены и рекомендованы к изданию методическим семинаром кафедры химической технологии топлива и химической кибернетики

«\_\_\_»\_\_\_\_\_\_\_2011 года, протокол № \_\_\_.

Зав. кафедрой ХТТ,

профессор, доктор технических наук \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В. Кравцов

**Аннотация**

Методические указания к выполнению лабораторных работ по дисциплине Информатика предназначены для студентов 1-го курса, обучающихся по направлениям: 240100 «Химическая технология», 240700 «Биотехнология» и 241000 «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии». Данная дисциплина изучается в первом семестре.

Приведено описание основ работы в среде PascalABC.NET, рассмотрены примеры выполнения заданий в данной среде по основным темам раздела «Программирование на языке Паскаль».

# введение

При изучении теоретического материала по курсу «Информатика» студенты должны выполнить лабораторные работы, которые в основном связаны с программированием на языке Паскаль.

До сих пор Паскаль заслуженно считается одним из лучших языков для начального обучения программированию. Наиболее популярным решением для персональных компьютеров в 80-е - начале 90 годов стал компилятор и интегрированная среда разработки [Turbo Pascal](http://ru.wikipedia.org/wiki/Turbo_Pascal) фирмы Borland, являющийся стандартом Паскаля, но редко используемый в последнее время в связи с тем, что соответствующая операционная система MS DOS устарела. Паскаль АВС – это современная версия языка программирования Паскаль и интегрированная среда, реализующая огромные возможности платформы .NET в операционной системе Windows.

В данных методических указаниях работа в среде **PascalABC.NET рассмотрена на примерах выполнения заданий по основным темам раздела «Программирование на языке Паскаль»:**

1. Составление простейших программ на языке Паскаль. Линейные алгоритмы.
2. Программирование разветвляющихся алгоритмов, операторы IF, GOTO.
3. Выполнение циклических операций. Массивы, операторы цикла, действия с матрицами.
4. Использование стандартных алгоритмов. Составление программ с использованием подпрограмм процедур и функций.

В методических указаниях приведены пояснения по наиболее часто встречающимся ошибкам, возникающим при отладке программы в среде **PascalABC.NET.**

## РАБОТА В СРЕДЕ PascalABC.NET

PascalABC.NET — это:

* **современный язык программирования**, основанный на Delphi (Object Pascal) и сочетающий простоту языка [Паскаль](http://pascalabc.net/o-yazike-paskal) и огромные возможности платформы .NET
* бесплатная, **простая и мощная среда разработки**, ориентированная на обучение программированию.
* уникальная [Web-среда](http://pascalabc.net/WDE/), позволяющая разрабатывать и запускать программы на языке Паскаль из окна браузера, а также иметь личный каталог программ на сервере.

**Скачать бесплатно последнюю версию PascalABC.NET и ознакомиться с возможностями интегрированной среды можно на сайте** [**http://pascalabc.net**](http://pascalabc.net)**.** При завершении записи версии на рабочем столе появится иконка, с помощью которой загружается интегрированная среда **PascalABC.NET и вызывается окно браузера (рис. 1).**

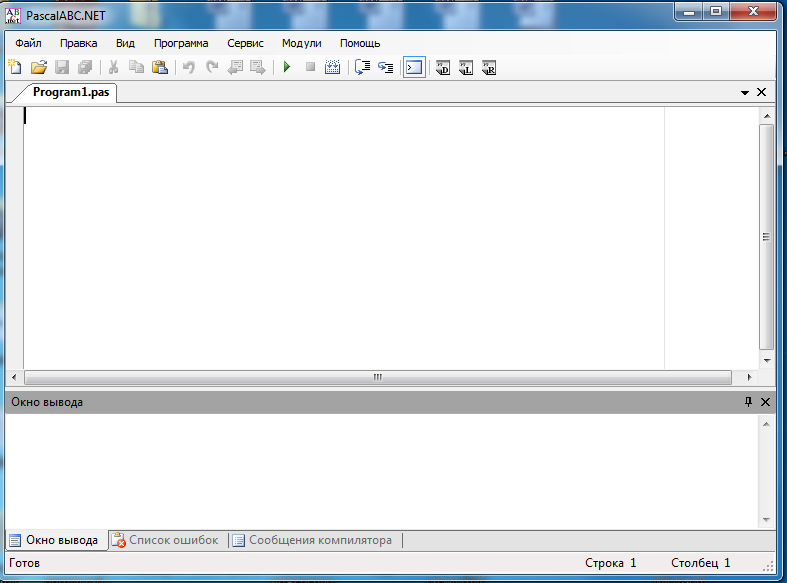
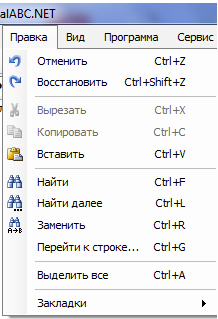
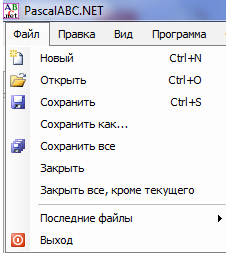
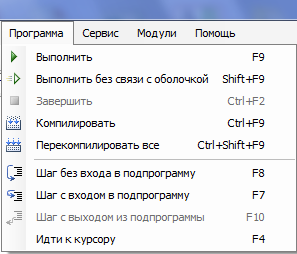
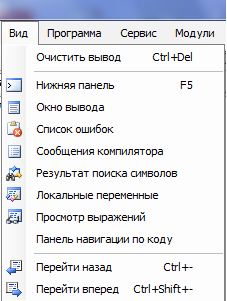
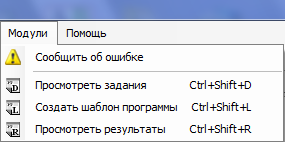
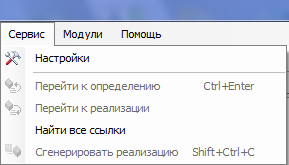


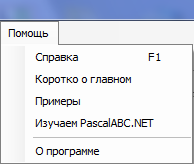
Рисунок 1

**При активизации элементов горизонтального меню в верхней части браузера, открываются горизонтальные подменю, обеспечивающие выполнение ряда нижеперечисленных функций:**









**На панель управления вынесены иконки, с помощью которых выполняются наиболее часто используемые операции:**

|  |  |
| --- | --- |
| – новый | – перейти вперед |
| – открыть | – выполнить |
| – сохранить | – завершить |
| – сохранить все | – компилировать |
| – вырезать | – шаг без входа в подпрограмму |
| – копировать | – шаг с входом в подпрограмму |
| – вставить | – окно вывода |
| – отменить | – просмотреть задания |
| – восстановить | – создать шаблон программы |
| – перейти назад | – просмотреть результаты |

В верхнем окне браузера набирается текст программы на языке программирования Паскаль, в нижнем Окне вывода появляется результат выполнения программы. Последовательность работы в среде **PascalABC.NET рассмотрим на примерах выполнения заданий по основным темам раздела «Программирование на языке Паскаль».**

# Составление простейших программ на языке Паскаль. Линейные алгоритмы

**Цель задания:** Получить представление о структуре программы на языке Паскаль, научиться использовать стандартные функции языка Паскаль для записи математических выражений.

**Пример №1.**

Составить программу на языке Паскаль и вычислить значение y:

*а=*0,3; *b*=-3.7; *с=*0,84; *х=*-5,4

**Последовательность работы**

1. Прежде чем приступить к программированию, выражение правой части уравнения нужно представить в виде последовательности вычислительных операций, ограниченных скобками, с использованием формул преобразований и стандартных функций языка Паскаль[1, с.64-66], [2, с.60-62], [3, с.64-66],:
2. Составляем программу на языке Паскаль, задавая исходные данные в разделе описания констант (const):

**Program** lab\_1;

**Const** a=0.3; b=-3.7; c=0.84; x=-5.4;

**Var** y:real;

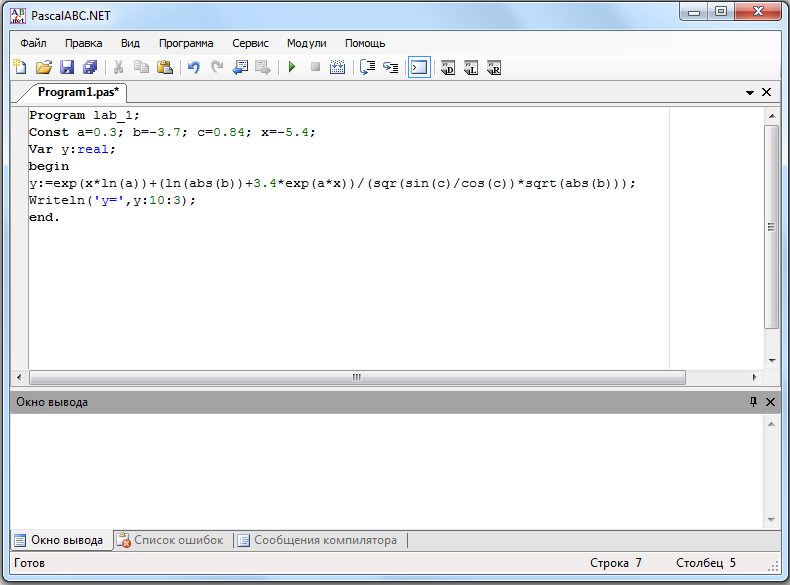
**begin**

y:=exp(x\*ln(a))+(ln(abs(b))+3.4\*exp(a\*x))/(sqr(sin(c)/cos(c))\*sqrt(abs(b)));

Writeln('y=',y:10:3);

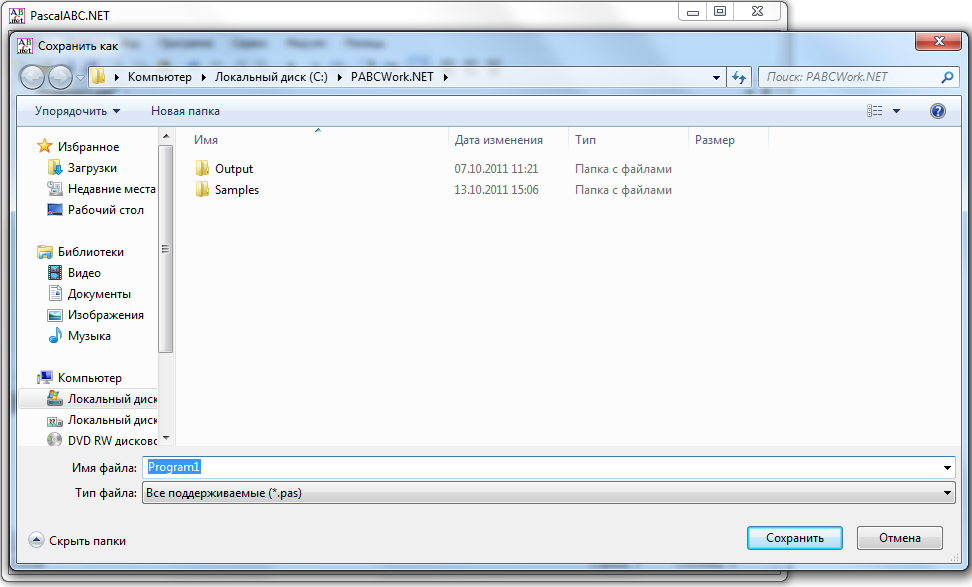
**end**.

1. Входим в среду **PascalABC.NET, используя иконку на рабочем столе компьютера**
2. В верхнем окне браузера набирается текст программы на языке Паскаль:

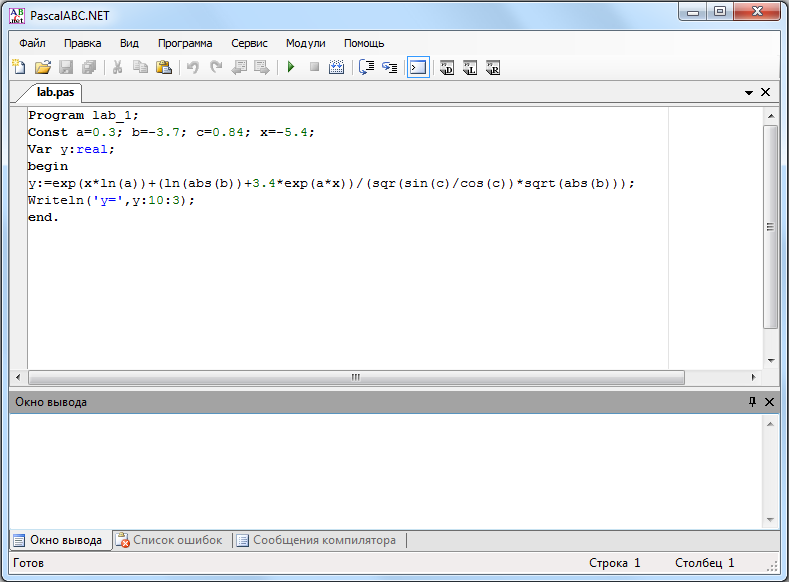


1. Необходимо сохранить текст программы, для этого воспользуемся иконкой 

Появится окно, в поле Имя файла которого вместо  следует написать имя, под которым программа будет храниться в каталоге, например lab (имя составляется из букв латинского алфавита, цифр и некоторых допустимых символов, не должно превышать 8 позиций):

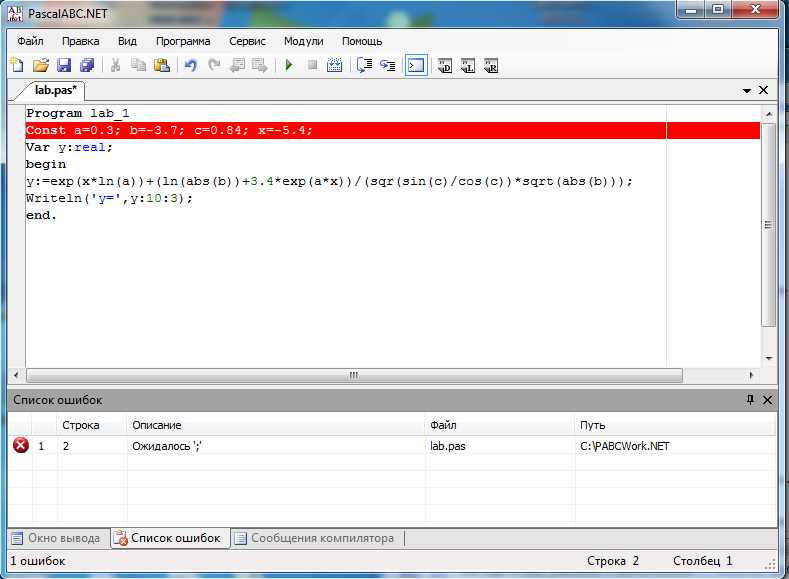
****

После этого указанное имя появится на вкладке рабочего окна.

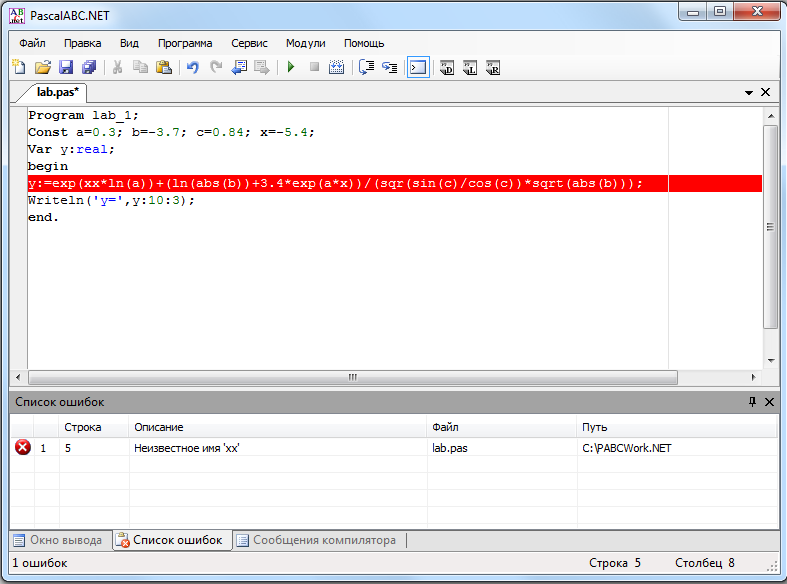


1. Воспользовавшись иконкой , выполняем программу.
2. При возникновении ошибок, в Окне вывода будет появляться подсказка. Следует отлаживать программу (исправлять ошибки), пока в Окне вывода не появится результат. Каждый раз после исправления ошибки следует сохранять последнюю версию, воспользовавшись иконкой  и выполнять программу, активизируя иконку 

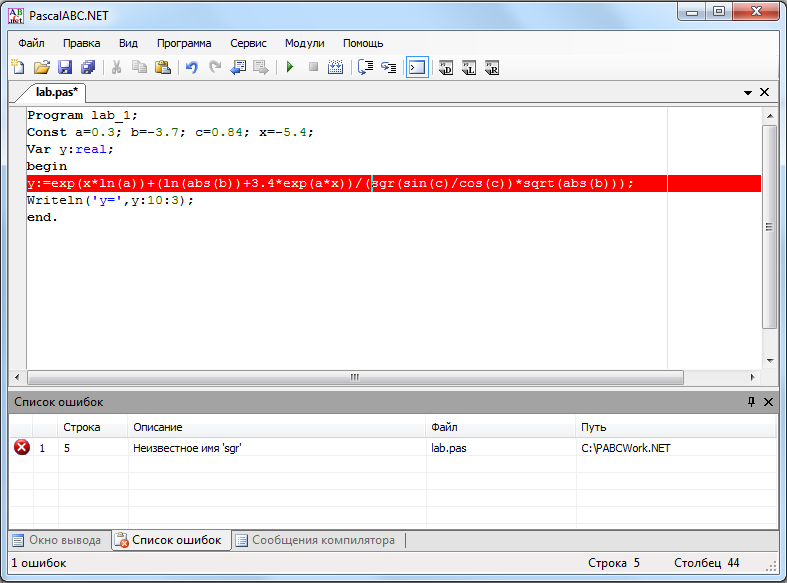
## Разъяснения по наиболее часто встречающимся ошибкам



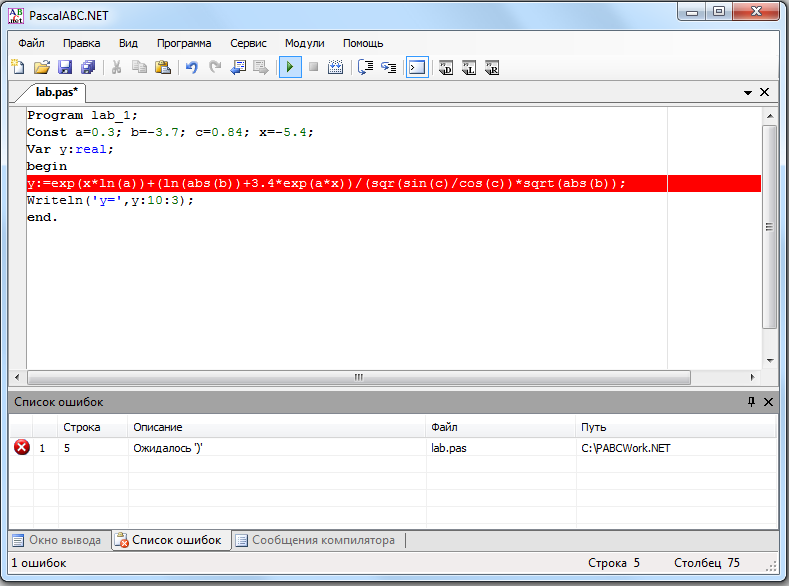
После заголовка программы **Program** lab\_1 должен стоять символ ‘;’. Аналогичная ошибка может возникать при отсутствии символа ‘;’ после любого другого оператора программы. Красной строкой выделен оператор, перед которым отсутствует разделитель ‘;’, в конце предыдущего оператора следует этот символ поставить.



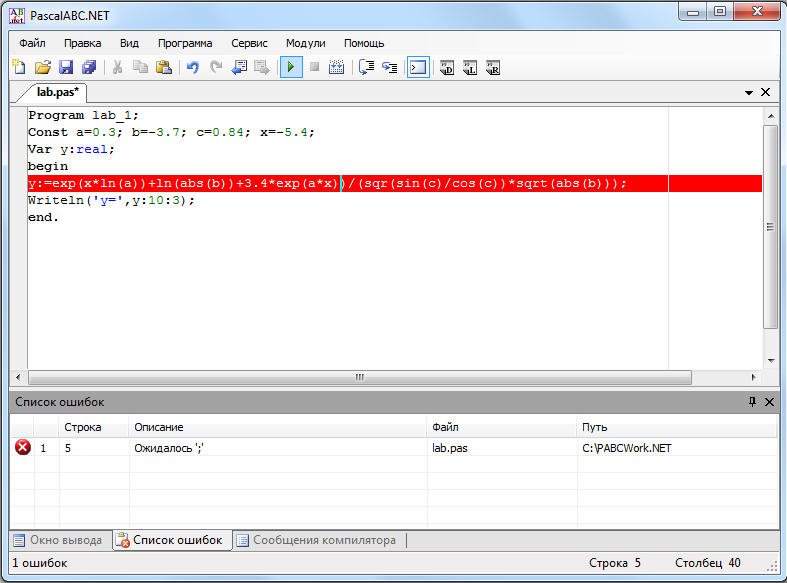
Имя ‘xx’ не описано ни в разделе описания констант (**Const**), ни в разделе описания переменных (**Var**). В данном случае, в разделе **Const** описана переменная ‘x’ (x=-5.4;), а в операторе вычисления ‘y’ используется нигде не описанная величина ‘xx’. Следует исправить ‘xx’ на ‘x’. Данная ошибка может возникать, как в случае неверного написания имени в теле программы, так и в случае отсутствия описания этого имени в разделе описаний.



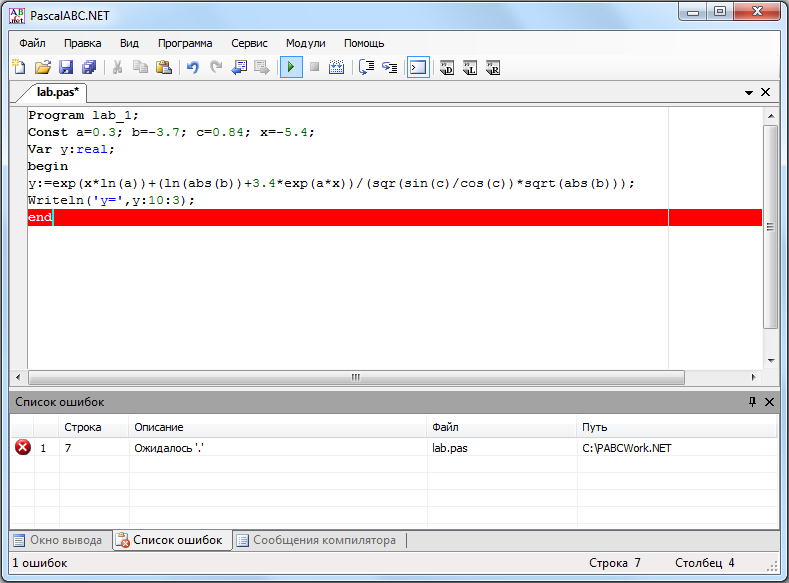
Аналогичная ошибка может возникнуть при неверном написании имени стандартной процедуры. Красной строкой выделяется строка, в которой указано неверное имя.



В данном случае, не хватает закрывающей скобки ‘)’. Это указано в нижнем окне браузера. Оператор, в который нужно внести исправления, выделен в верхнем окне красной строкой. Необходимо внимательно проверить написание оператора и указать скобку в соответствии с последовательностью выполнения вычислительных операций.

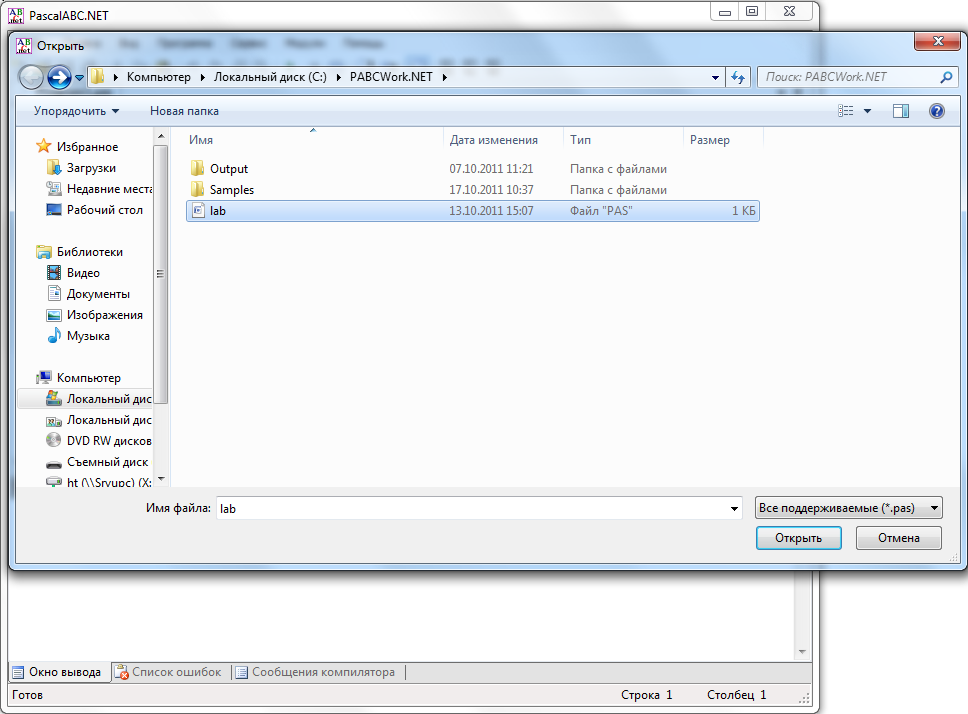


Возникновение данной ошибки связано с недостаточным количеством открывающих скобок ‘)’. Следует внимательно проверить написание оператора, выделенного красной строкой, и указать скобку в соответствии с последовательностью выполнения вычислительных операций.



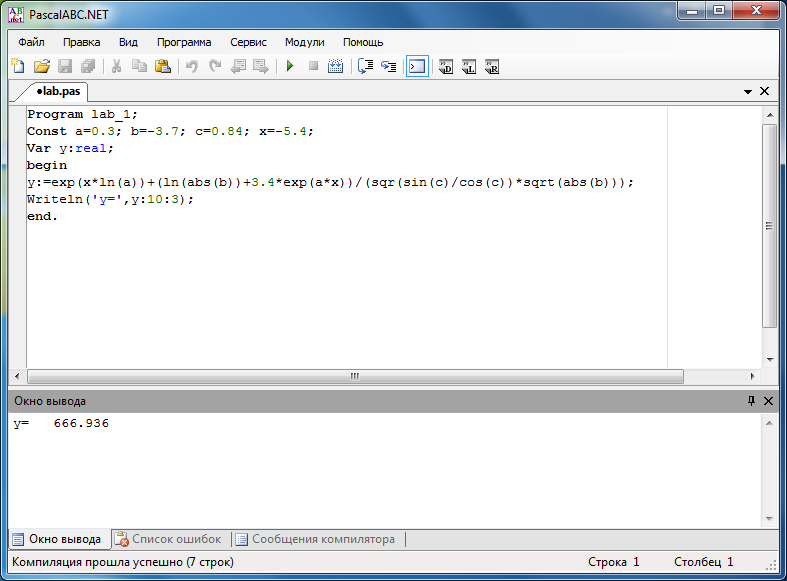
Отсутствует символ ‘.’ В конце программы.

1. Если не удалось завершить отладку программы за один прием, следует записать последнюю версию на диск, используя иконку , и в следующий раз продолжить отладку, предварительно войдя в среду **PascalABC.NET** и воспользовавшись иконкой . При этом появится окно:



Следует выделить в каталоге имя отлаживаемого файла (lab) и активизировать клавишу **Открыть**, а затем продолжить отладку.

Когда программа будет отлажена и в окне вывода появится результат:



После получения результата можно приступать к оформлению отчета.

# Программирование разветвляющихся алгоритмов

**Цель задания:** Научиться составлять программы на языке Паскаль на основе разветвляющихся алгоритмов с использованием условного оператора *if…then*.

**Пример №2**

Вычислить значение:

**

Значение *х* задать с клавиатуры.

**Последовательность работы**

1. Составляем блок-схему алгоритма, которая будет отражать последовательность расчета и написания программы. В данном случае используем разветвляющийся тип алгоритма [1,с.56-57], [2,с.52-53],[3,с.55-56]:

Начало

Ввод *b,c,x*

*x* < 1

*y = a + b∙x*



Конец

*x* = 1

*x* < 1

Ввод *y*



2. Составляем программу расчета на языке Паскаль, используя, для передачи управления, условный оператор перехода *if…then…else*.

Текст программы:

**Program** Lab2;

**Const** b=2.4; c=1.7; (описание постоянных параметров)

**Var** x, a, y: real; (описание переменных параметров)

**Begin**

**write** (‘ Введите x=’); (Ввод параметра х с клавиатуры)

**readln** (x);

a:=3.5\*b/(b+exp(3\*ln(c)));

**if** x < 1 **then** y:=a+b\*x; (расчет параметра у)

**if** x = 1 **then** y:=sqrt(abc(a+b/x)) **else**

y:=sqr(a\*x+b\*x);

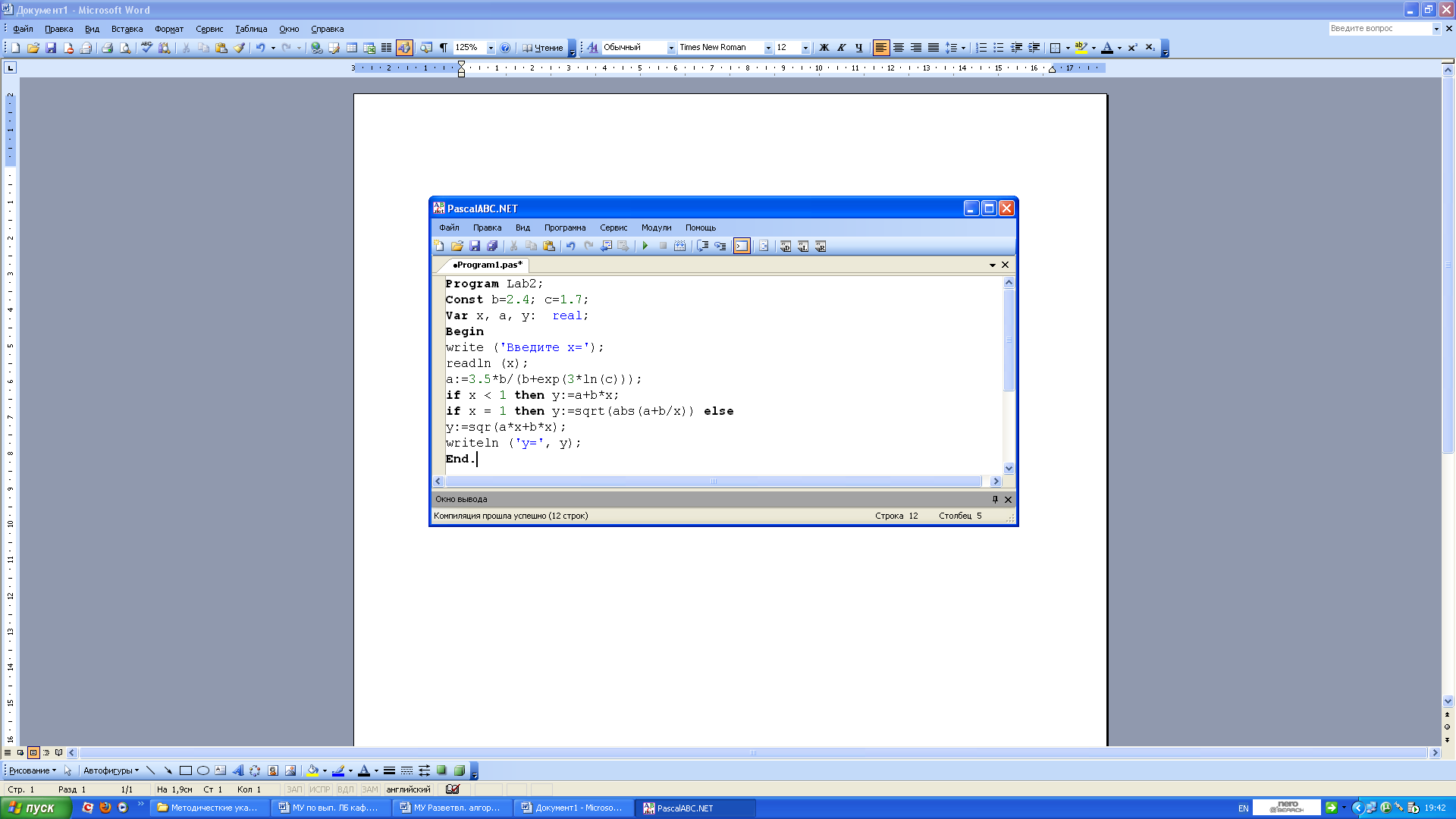
**writeln** (‘y=’, y); (вывод параметра у на экран)

**End.**

3. Входим в среду **PascalABC.NET, используя**

**иконку на рабочем столе компьютера**.

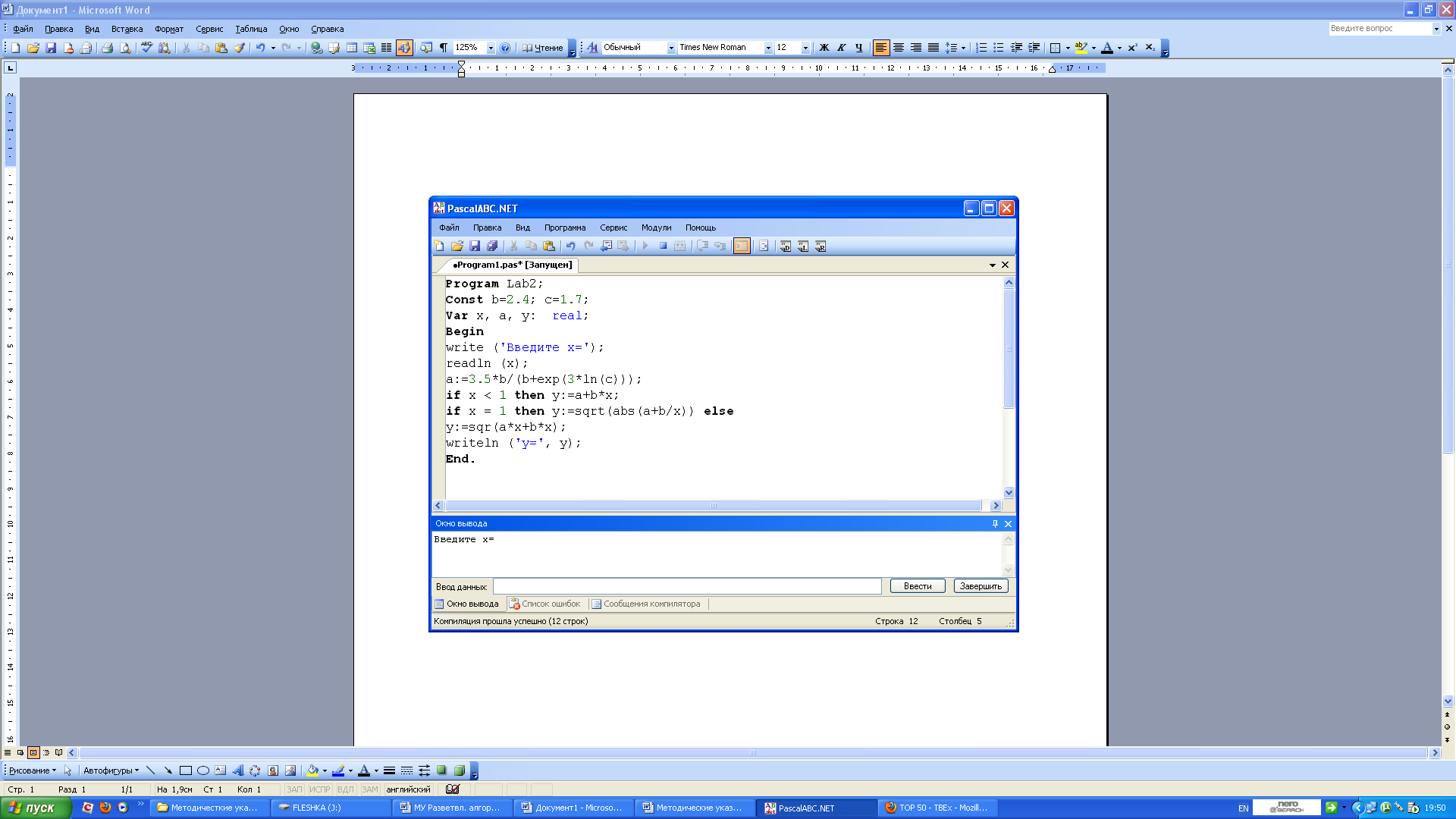
1. В верхнем окне браузера набирается текст программы на языке Паскаль:



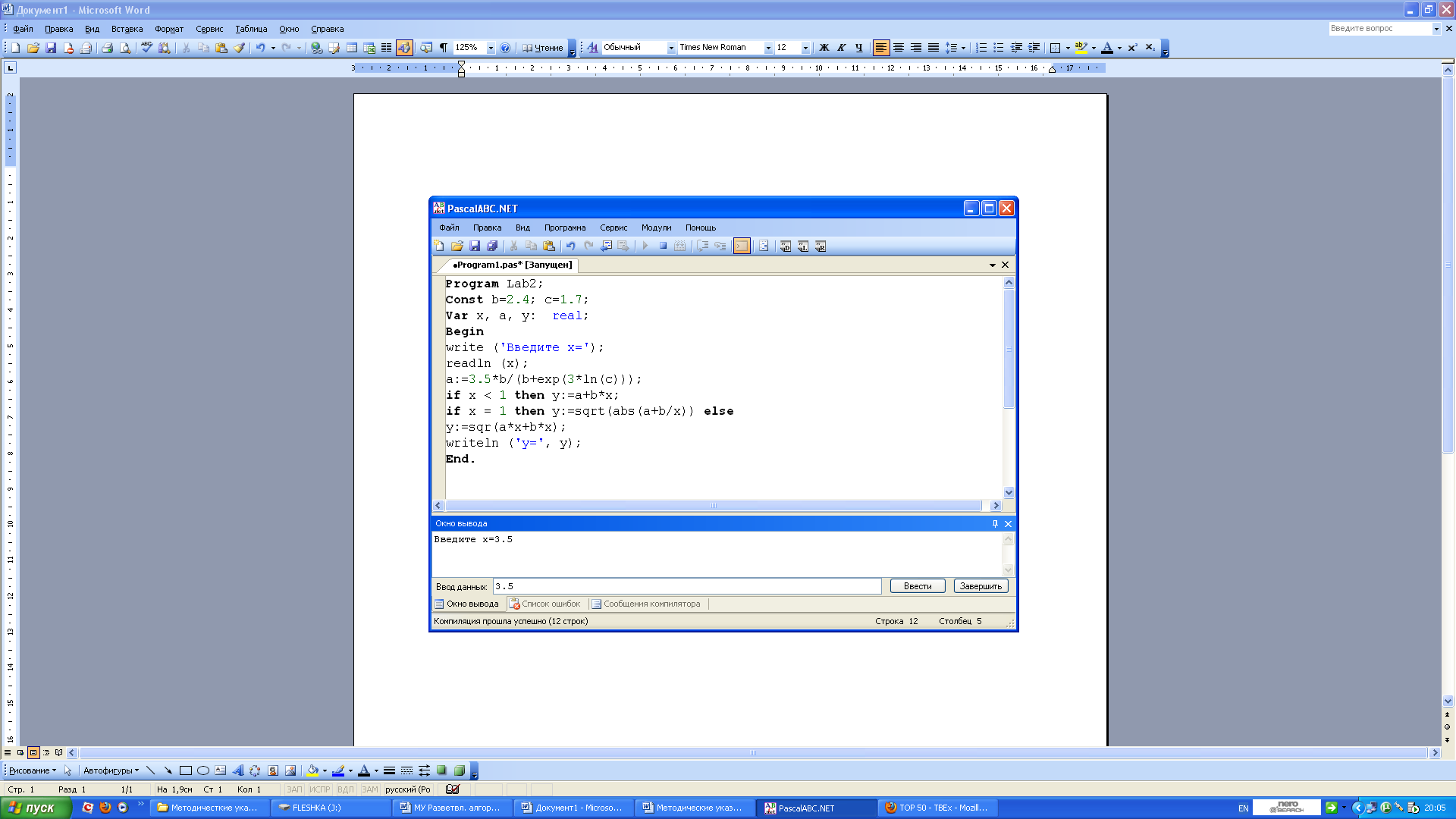
5. Сохраняем текст программы (подробно последовательность действий описана в предыдущем примере).

6. Воспользовавшись иконкой , выполняем программу.

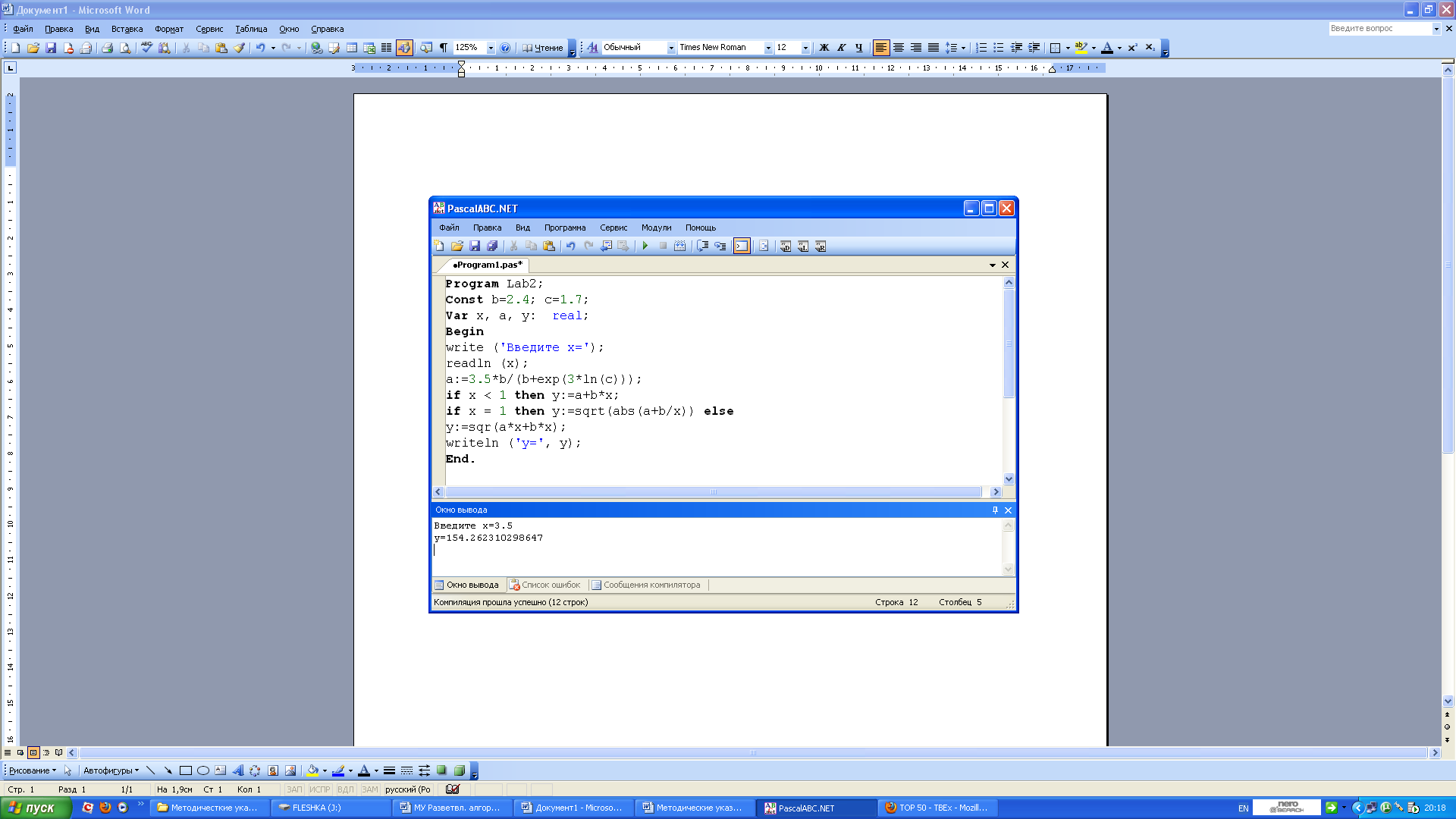
7. При возникновении ошибок, в Окне вывода будет появляться подсказка. Следует отлаживать программу (исправлять ошибки), пока в Окне вывода не появится дополнительная строка для ввода значения *x*:



Произвольно задайте значение *x* (например, 3,5) и нажмите «Ввести»:



После нажатия клавиши «Ввести» в окне вывода появится результат вычисления величины *у*:

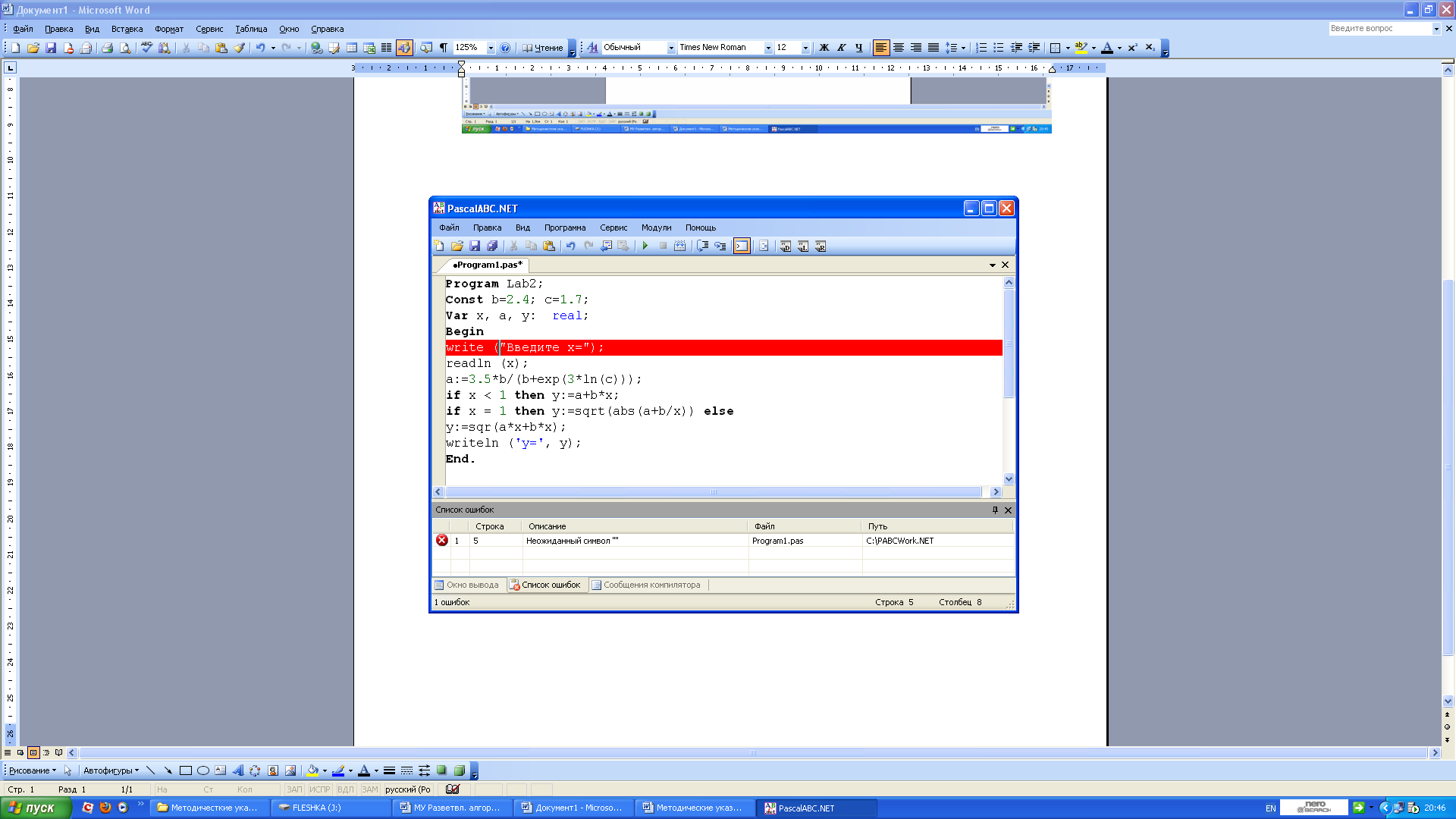


Повторите несколько раз вычисления, изменяя значение *х* в строке ввода.

Разъяснения по наиболее часто встречающимся ошибкам

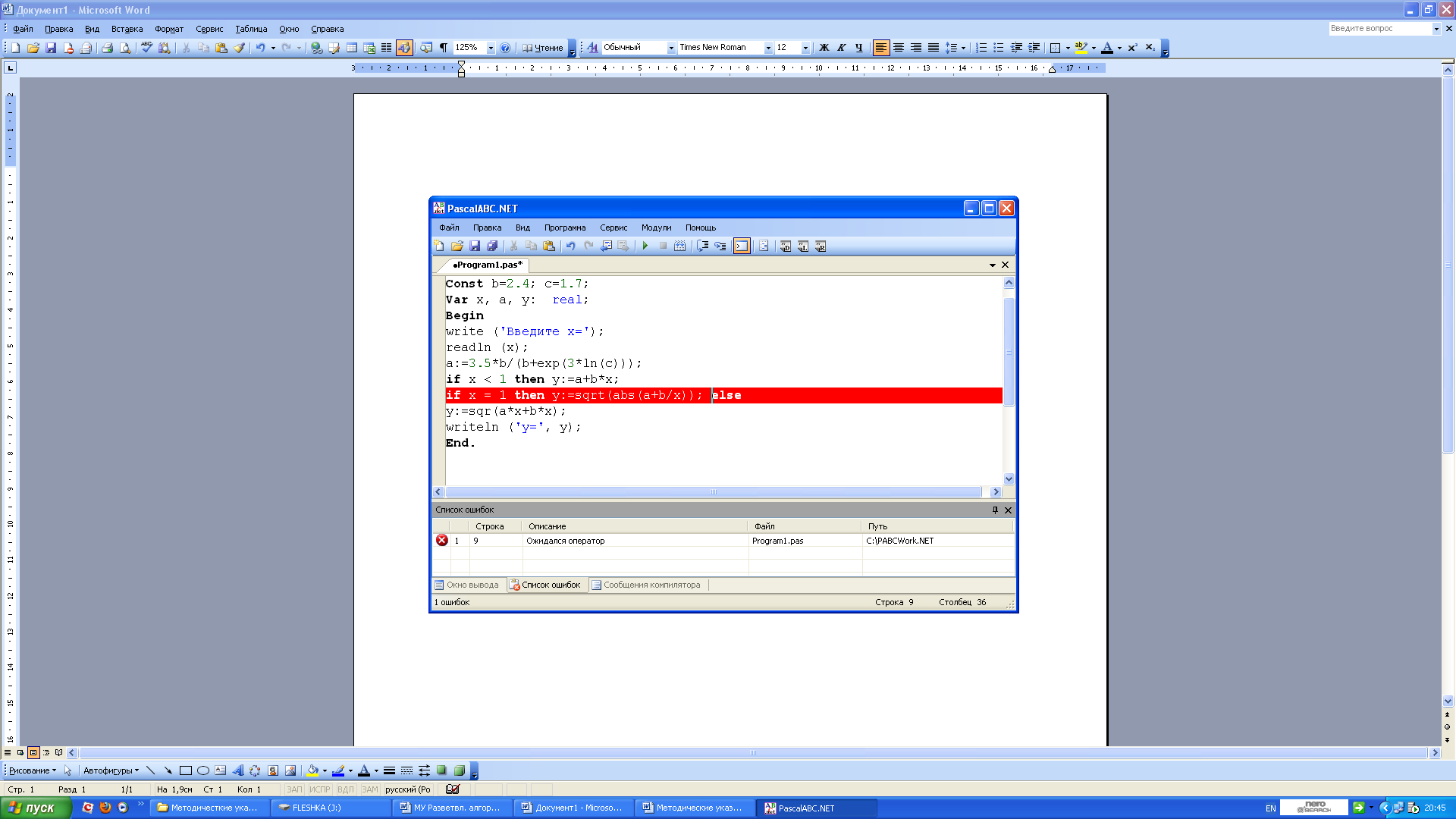
Основные ошибки, встречающиеся в программах, описаны в предыдущем примере. В данном разделе приведены те ошибки, которые могут дополнительно возникнуть при выполнении текущей лабораторной работы.

1. Неправильное введение апострофов



Апострофы в программе необходимо вводить символом **‘**. Двойной апостроф **“** не допускается.

2. Неправильное введение условного оператора



Все составные части условного оператора написаны правильно, однако перед зарезервированным словом *else* стоит точка с запятой, что не допустимо. Точка с запятой ставится лишь после закрытия условного оператора или его началом.

Последовательность работы при незавершенной отладке приведена в предыдущем примере.

# Выполнение циклических операций. Массивы, операторы цикла, действия с матрицами

**Цель задания:** Научиться составлять программы на языке Паскаль на основе циклических алгоритмов с использованием операторов цикла *while..do* и *repeat…until.* Научится работать с матрицами.

**Пример задания №3**

Вычислить значение:

|  |
| --- |
|  |

*b*=0,5; *a0*=12; *ak*=20; *h*=2; *c(4)*=6,7,8,9.

Для организации цикла при вычислении суммы использовать оператор цикла *while..do*

## Последовательность работы

1. Для упрощения расчетов разобьем выражение на две составляющие:

, 

2. Составляем программу расчета, используя операторы цикла *while..do и for.*

Текст программы:

**Program** Lab3;

**Const** b=0.5; {описание постоянных параметров}

a0=12; ak=20; h=2;

c:array[1..4]of real=(6,7,8,9); {описание массива}

**Var** a,k,f,y: real; j:integer; {описание переменных}

**Begin**

a:=a0;

k:=0; f:=1;

**while** a<=ak **do begin** k:=k+5\*a\*b; {расчет суммы}

a:=a+h; **end;**

**for** j:=1 **to** 4 **do** {обращение к массиву}

f:=f\*sqrt(c[j]); {расчет произведения}

y:=k+4.3\*f; {расчет параметра у}

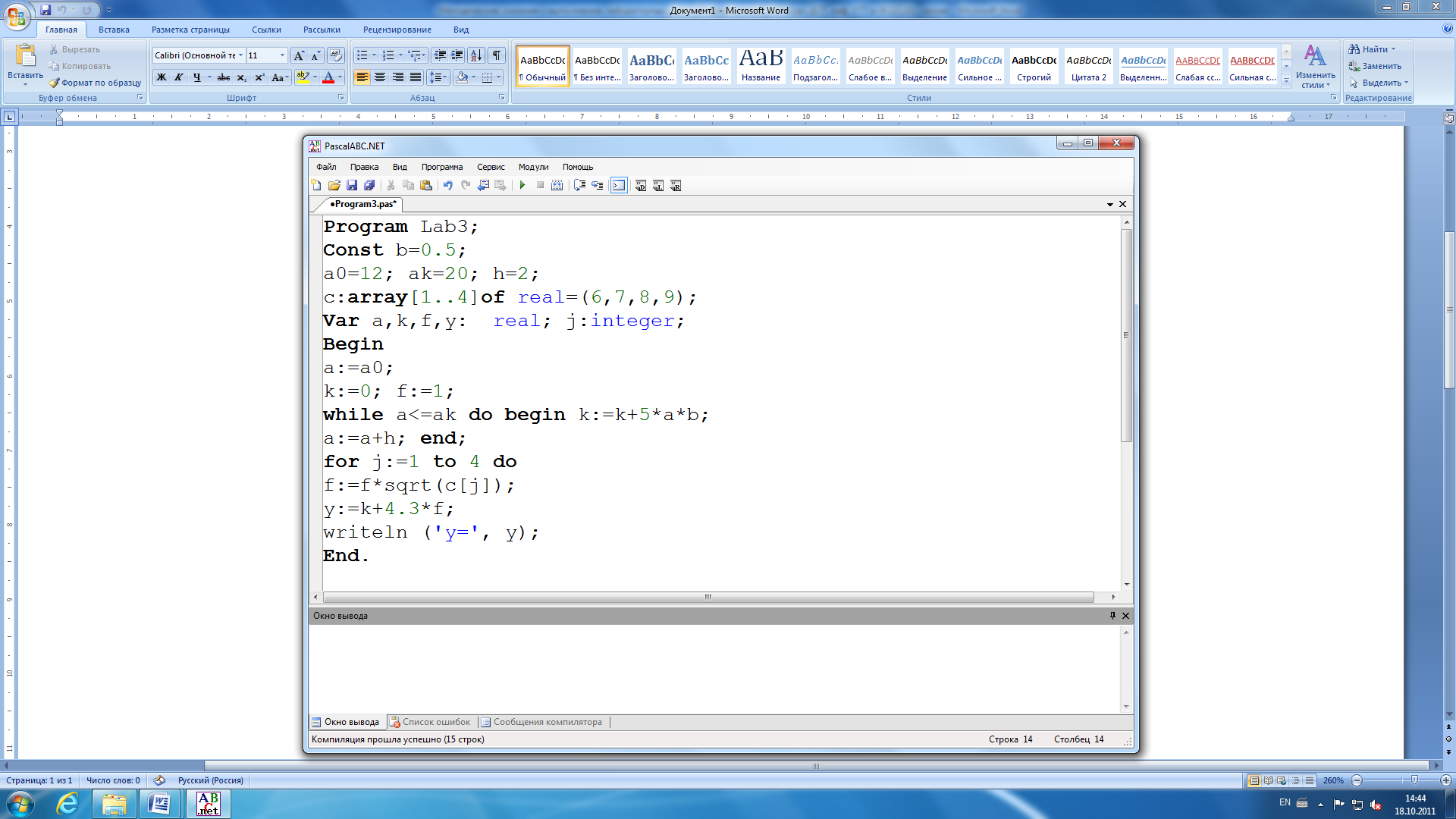
**writeln** (‘y=’, y); {вывод параметра у на экран}

**End.**



3. Входим в среду PascalABC.NET, используя

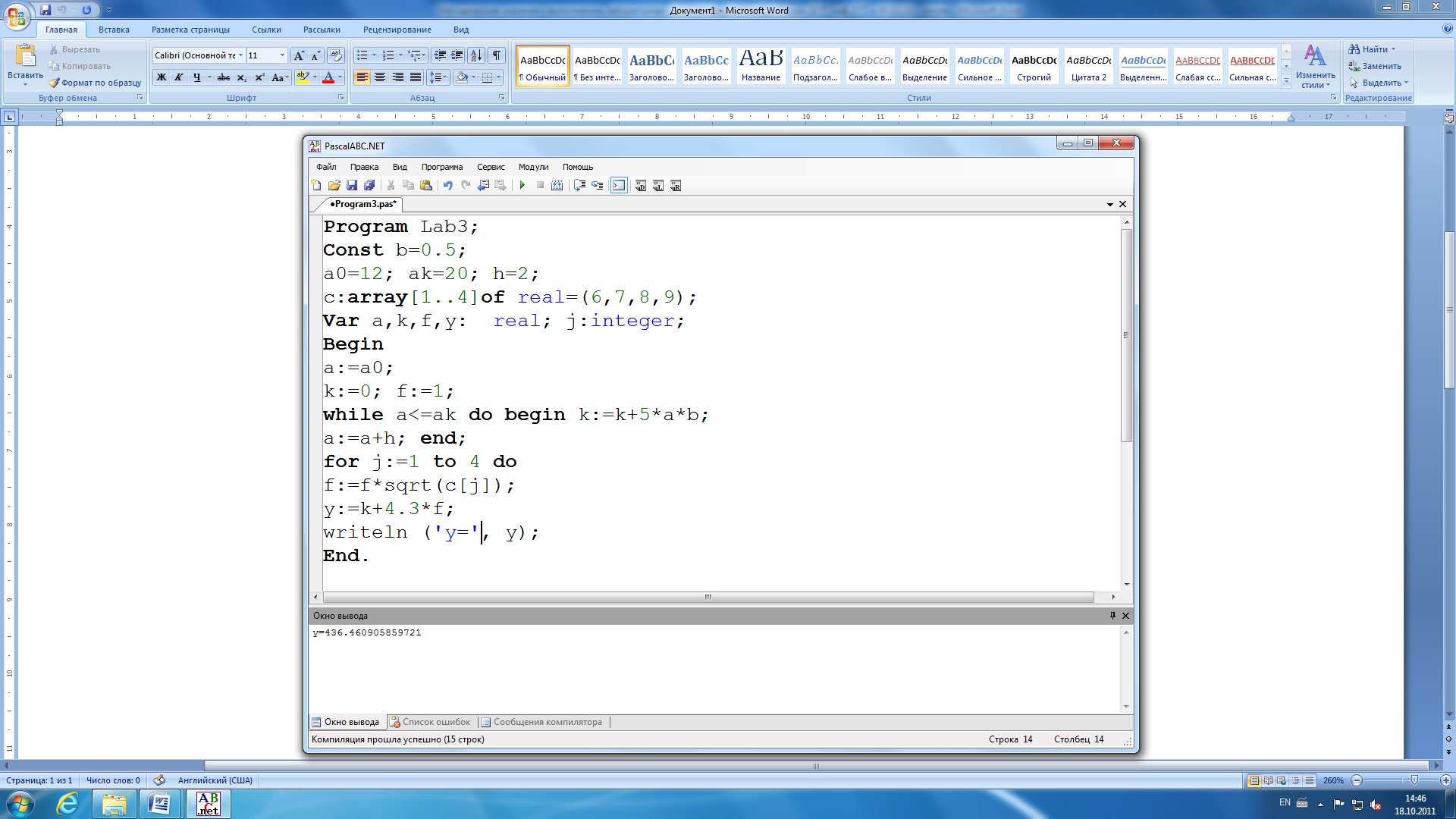
иконку на рабочем столе компьютера.

4. В верхнем окне браузера набирается текст программы на языке Паскаль: 

5. Сохраняем текст программы (подробно последовательность действий описана в примере №1).

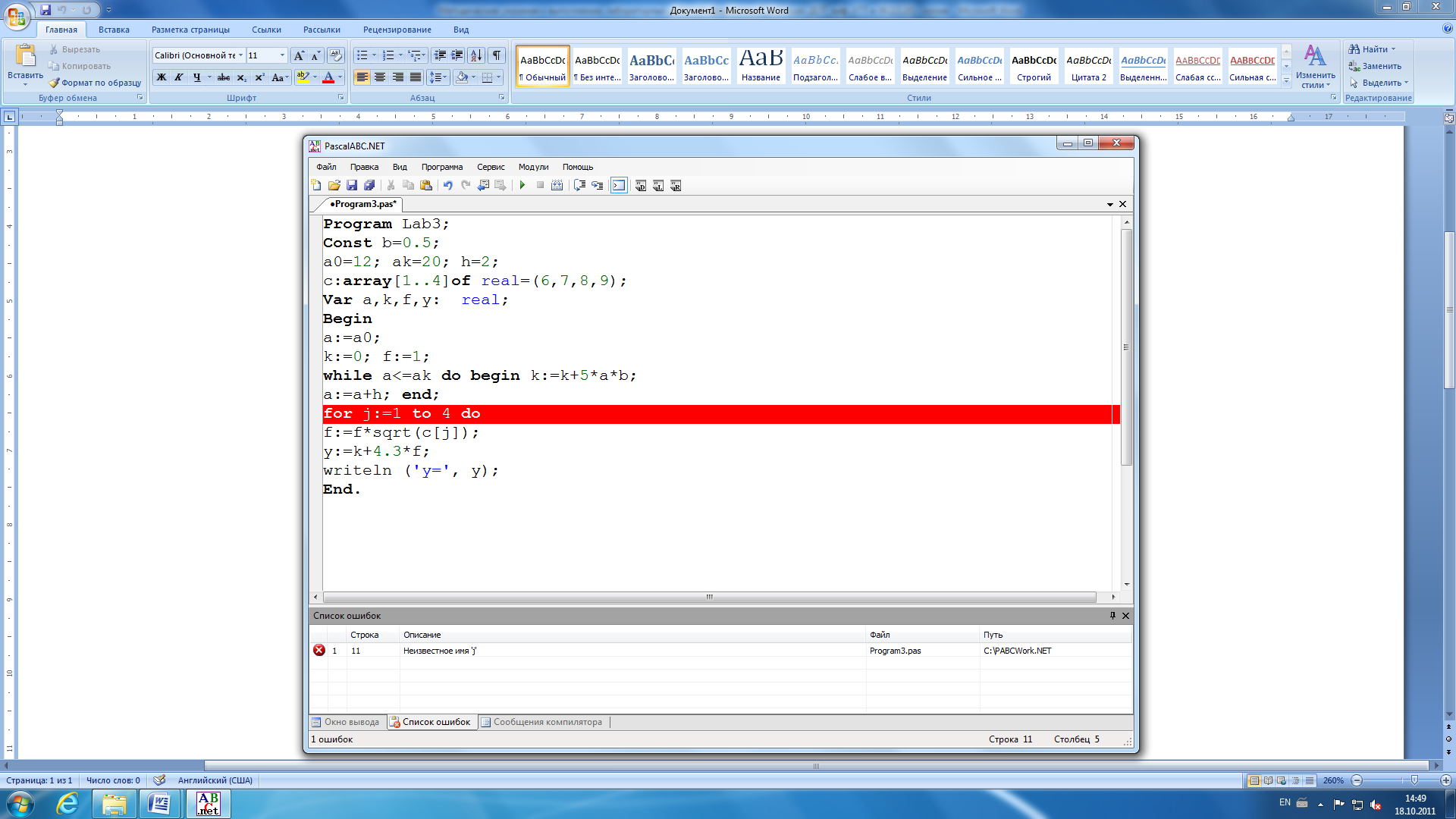
6. Воспользовавшись иконкой , выполняем программу.

7. При возникновении ошибок, в Окне вывода будет появляться подсказка. Следует отлаживать программу (исправлять ошибки), пока в Окне вывода не появится результат:

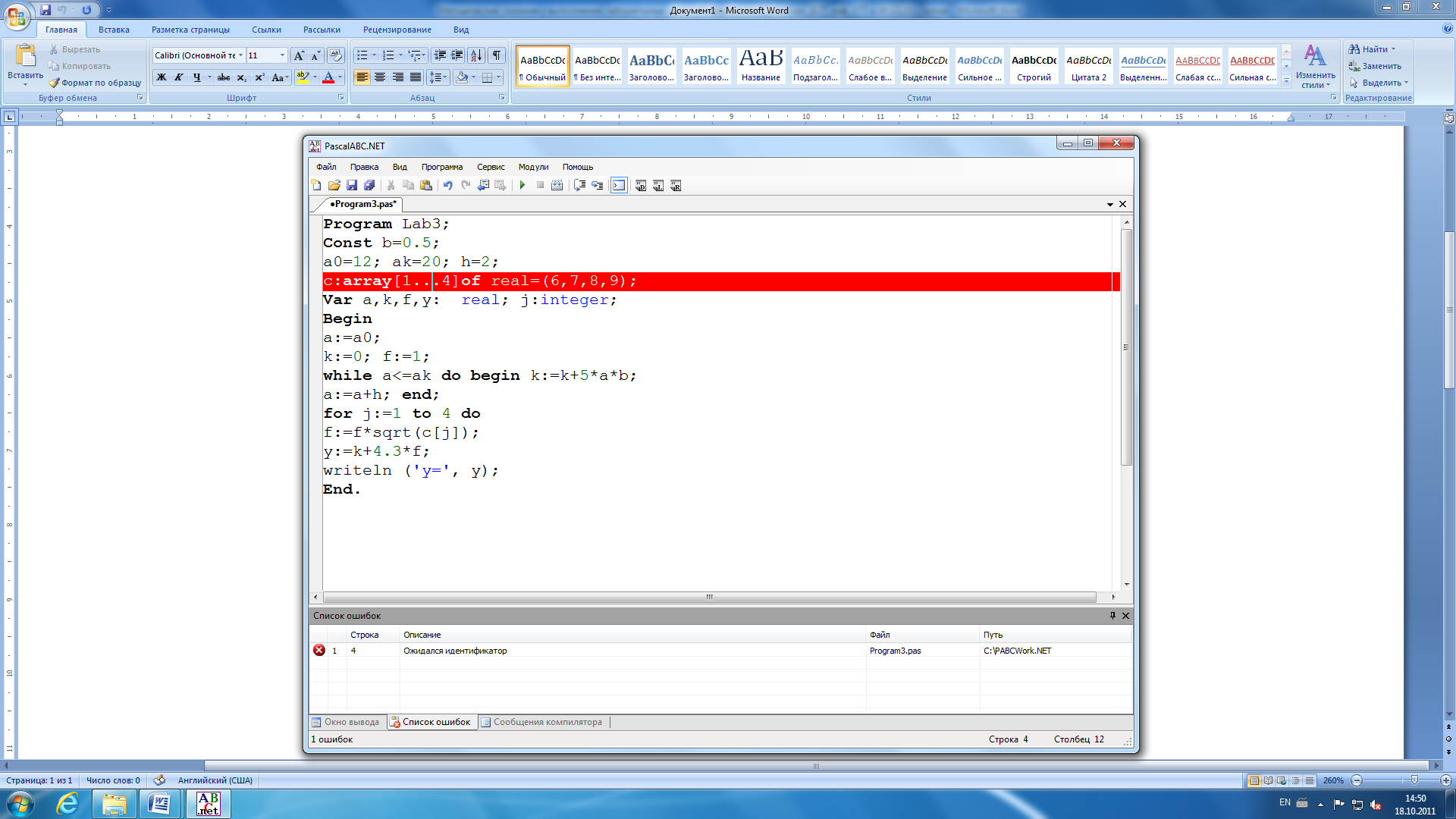


**Разъяснения по наиболее часто встречающимся ошибкам**

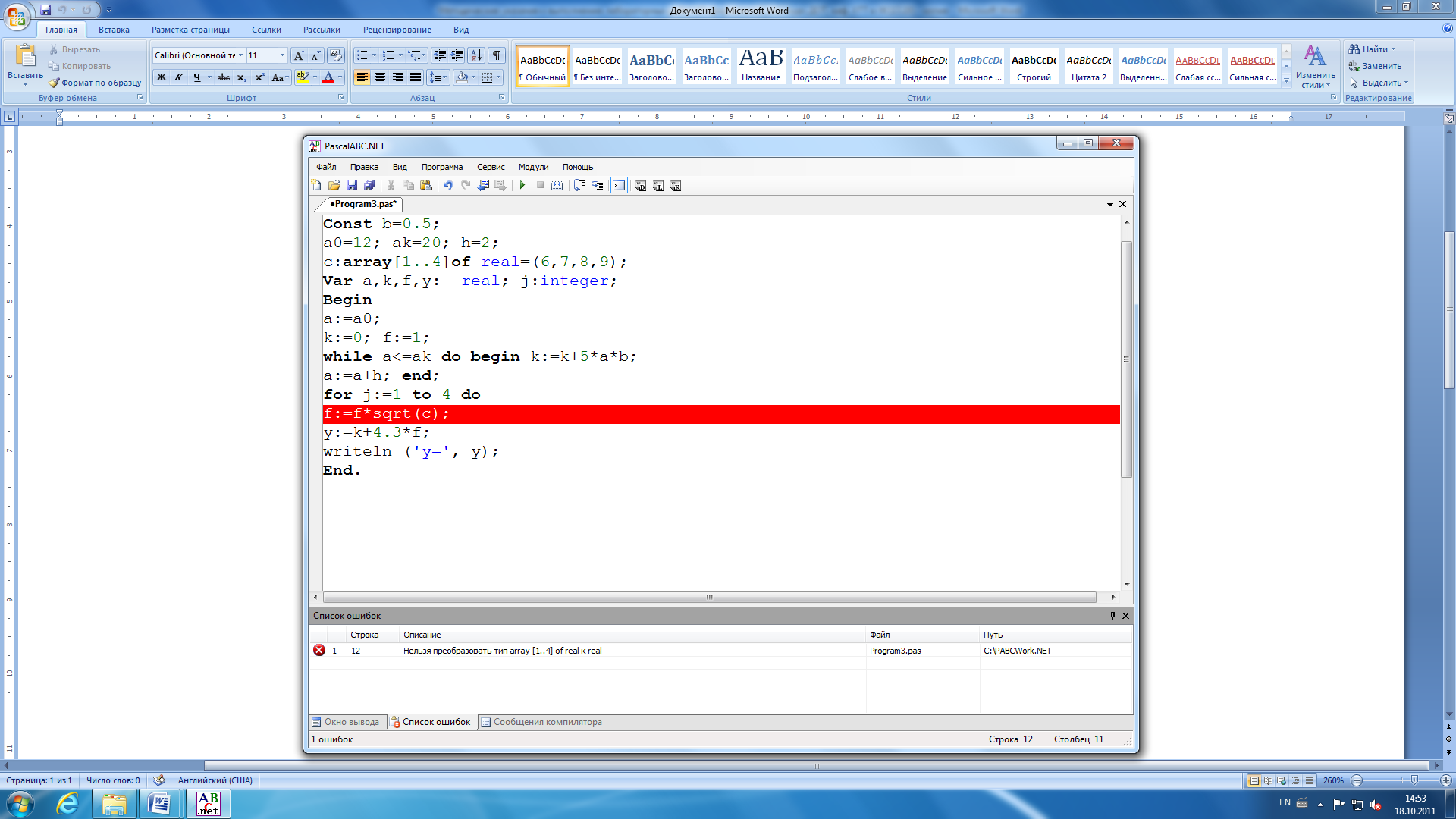
Основные ошибки, встречающиеся в программах, описаны в предыдущих примерах. В данном разделе приведены те ошибки, которые могут дополнительно возникнуть при выполнении текущей лабораторной работы.



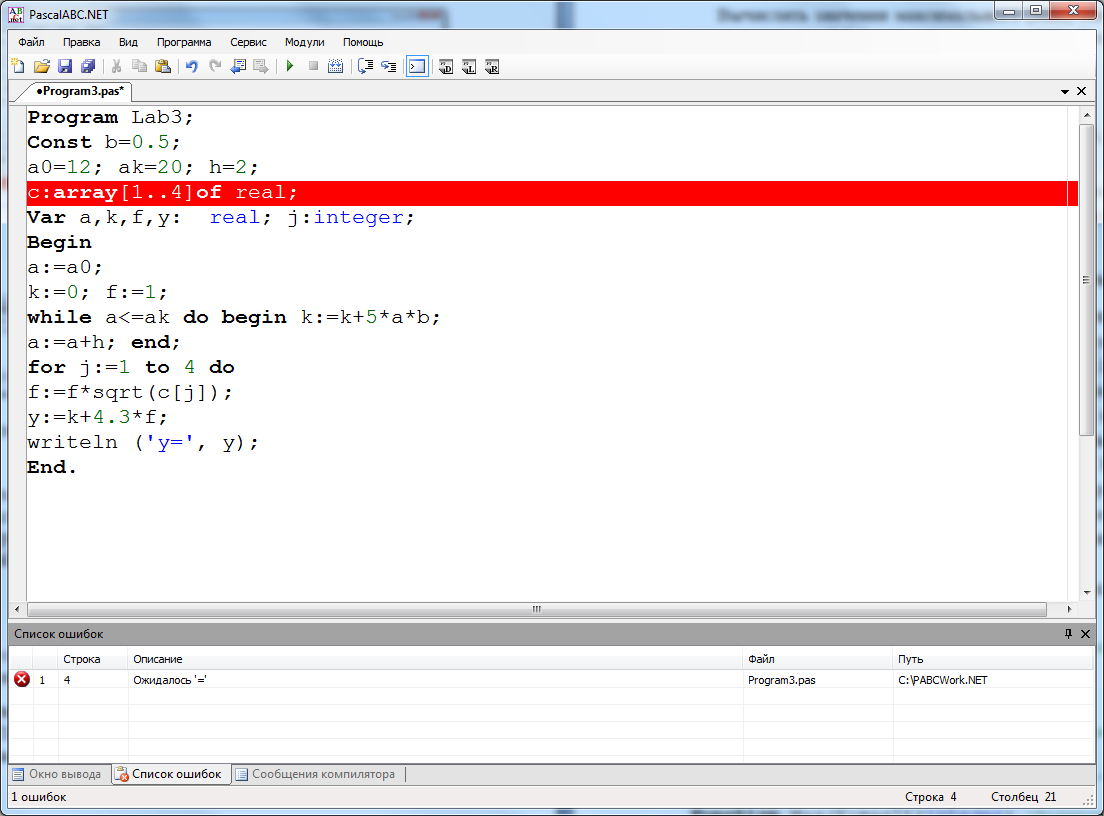
Номер элемента матрицы, обозначенный за *j*, не описан в программе. Следует описать его в разделе ***Var***, следующим образом *j:integer*.



При описании размерности массива в квадратных скобках следует ставить две точки (..), вместо указанных в программе трех (…).



В программе между параметрами должно быть соответствие по типу данных. Так как в разделе описаний данный параметр (С) описан как массив, то и при вычислении необходимо указать, что это параметр содержит несколько значений С[j].



При описании массива в разделе описания ***Const*** после указания типа необходимо поставив знак равно перечислить численные значения, входящие в массив, в круглых скобках:

c:array[1..4]of real=(6,7,8,9);

8. Последовательность работы при незавершенной отладке приведена в примере №1. Следует отлаживать программу (исправлять ошибки), пока в Окне вывода не появится результат

# Использование стандартных алгоритмов. Составление программ с использованием подпрограмм процедур и функций

**Цель задания:** Получить представление об использовании подпрограмм на языке Паскаль, научиться программировать типовые алгоритмы, используемые при вычислении математических выражений.

**Пример задания №4.**

Вычислить значения максимального (Max) элемента массива *K*(5)  
и минимального (Min) элемента массива *C*(9). Полученные значения подставить в формулу:

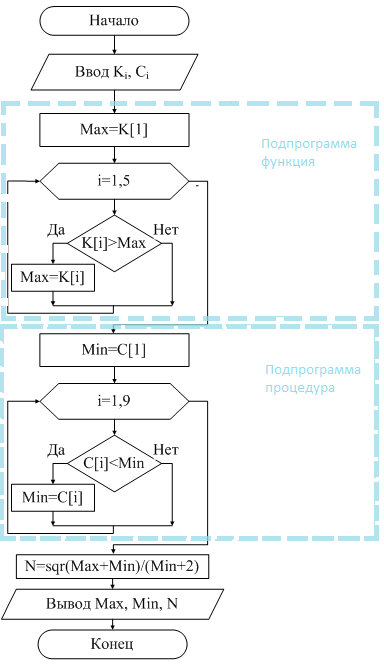


*K*(5)= {7, 80, 25, 72, 31}; *C*(9)= {6.1, 4.2, 3.3, 0.4, –1.5, –8.6, 29.7, 5.8, 17.9}.

Поиск значения максимального элемента оформить в виде подпрограммы функции, поиск значения минимального элемента – в виде подпрограммы процедуры.

**Последовательность работы**

1. Разрабатываем алгоритм решения задачи. Программирование типовых алгоритмов поиска минимального и максимального элементов описано в [1, с.115-116], [2, с.106-107], [3, с.109-110]. Программирование типовых алгоритмов вычисления суммы и произведения, вычисления факториала, нахождения количества и пр. описано в [1, с.113-132], [2, с.105-112], [3, с.107-115]. Минимальный и максимальный элементы массивов вычисляем в разных циклах. Вычисление максимального элемента будем оформлять в виде подпрограммы функции, минимального элемента - в виде подпрограммы процедуры. Программирование с использованием подпрограмм описано в [1, с.94-99], [2, с.88-92], [3, с.89-93].



**Блок-схема алгоритма**

1. Составляем программу на языке Паскаль:

**Program** lab\_43;

**Var** {Описание переменных параметров}

N,Min:real; i:integer;

**Type** mas1=**array**[1..5] **of** integer;

mas2=**array**[1..9] **of** real;

**Const** {Описание постоянных параметров}

K:mas1=(7, 80, 25, 72, 31);

C:mas2=(6.1, 4.2, 3.3, 0.4, -1.5, -8.6, 29.7, 5.8, 17.9);

**Function** Max(K:mas1):integer; {Подпрограмма функция}

**Var** K1,i:integer; {Вычисление максимального элемента}

**Begin**

K1:=K[1];

**For** i:=1 **to** 5 **do**

**if** K[i]>K1 **then** K1:=K[i];

Max:=K1;

**end**;

**Procedure** Minimum(C:mas2;**Var** Min:real); {Подпрограмма процедура}

**Var** i:integer; {Вычисление минимального элемента}

**Begin**

Min:=C[1];

**For** i:=1 **to** 9 **do**

**if** C[i]<Min **then** Min:=C[i];

**end**;

**Begin** {Основная программа}

Minimum(C,Min); {Обращение к подпрограмме процедуре}

N:=sqr(Min+Max(K))/(Min+2);

{Вывод результатов вычислений}

Writeln('Min=',Min,' Max=', Max(K), ' N=',N:5:2);

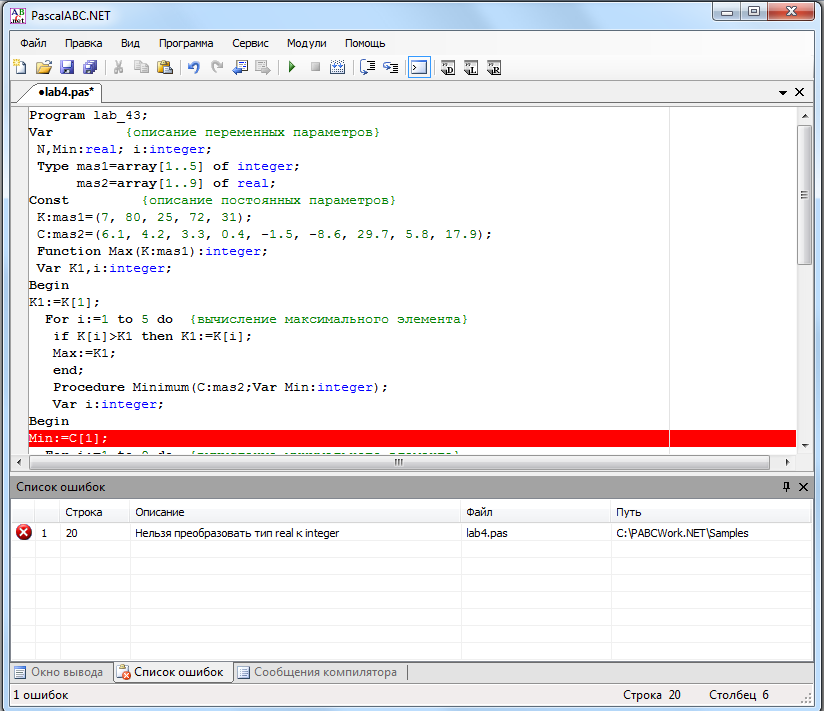
**End**.

В приведенном тексте программы, заключенные в фигурные скобки фрагменты являются комментариями (не воспринимаются компилятором). Комментарии используются для пояснения программы и в студенческих работах не обязательны.

1. Входим в среду **PascalABC.NET, используя иконку на рабочем столе компьютера**. Подробно работа в среде **PascalABC.NET описана в методических указаниях к лабораторной работе № 1.**
2. В верхнем окне браузера набираем текст программы на языке Паскаль.
3. Сохраняем текст программы в файле, присвоив ему имя, например ***lab4***.
4. Воспользовавшись иконкой , выполняем программу.
5. При возникновении ошибок, в Окне вывода будет появляться подсказка. Следует отлаживать программу (исправлять ошибки), пока в Окне вывода не появится результат. Каждый раз после исправления ошибки следует сохранять последнюю версию, воспользовавшись иконкой  и выполнять программу, активизируя иконку 

## Разъяснения по наиболее часто встречающимся ошибкам

По большинству возможных ошибок и вариантов их устранения приведены рекомендации в предыдущих примерах. Здесь приведем ошибки, возникающие, в основном, при использовании подпрограмм.

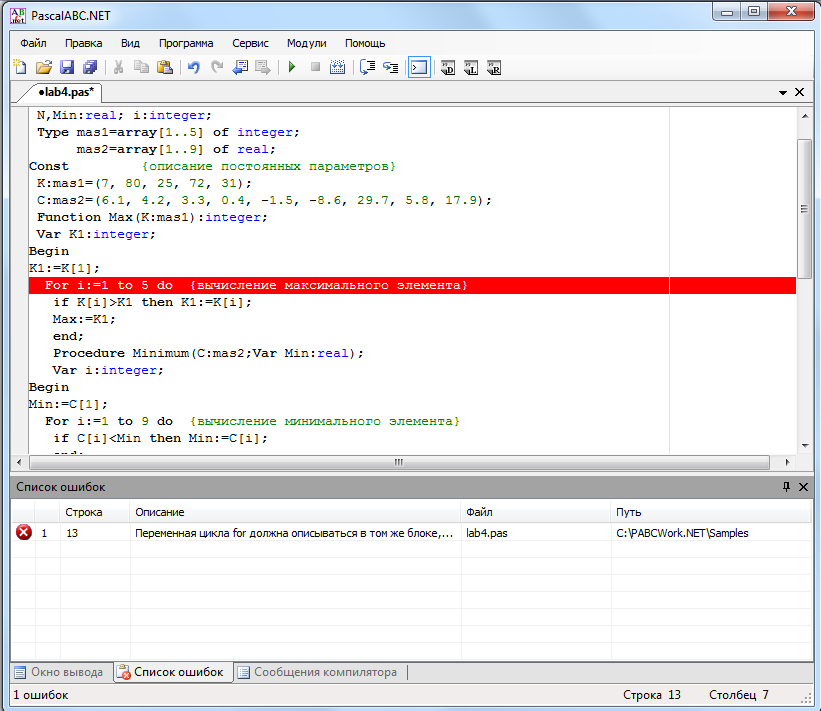


Эта ошибка связана с тем, что в описании глобальных переменных Min описан как вещественная переменная (Min:real;),а в списке формальных параметров процедуры

(**Procedure** Minimum(C:mas2;**Var** Min:integer);) - как целое.

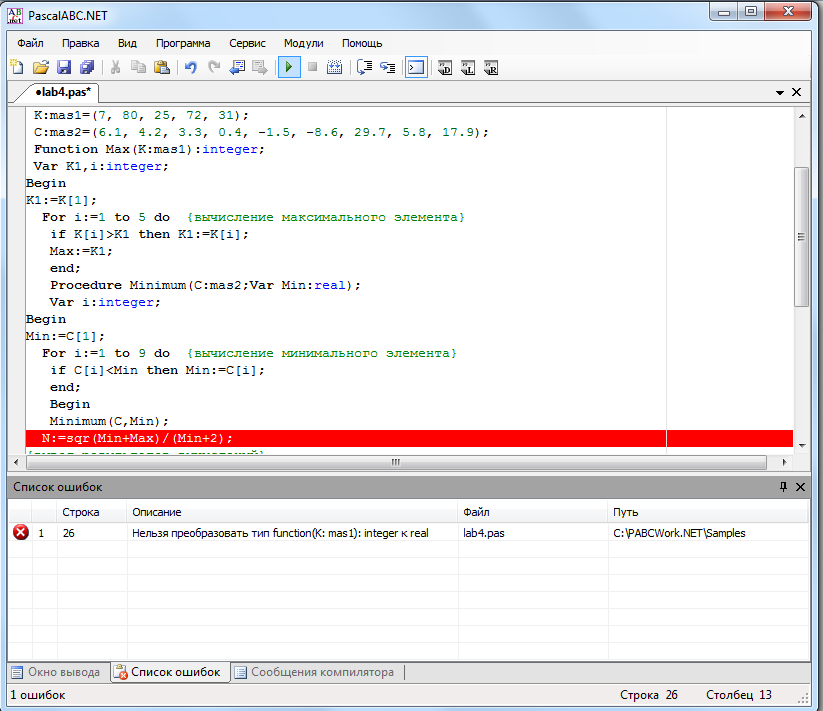
Необходимо исправить описание в списке формальных параметров:

**Procedure** Minimum(C:mas2;**Var** Min:real);.



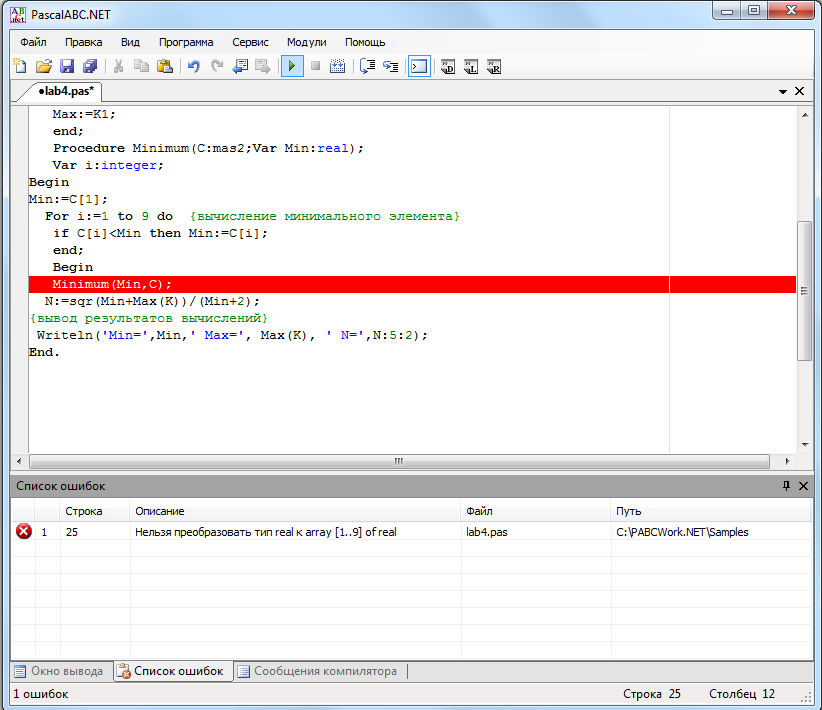
Используемые внутри подпрограммы имена должны быть описаны в списке локальных переменных, в данном случае, переменную цикла For **i** следует внести в список описания локальных переменных:

**Var** K1,i:integer;(вместо **Var** K1:integer;)

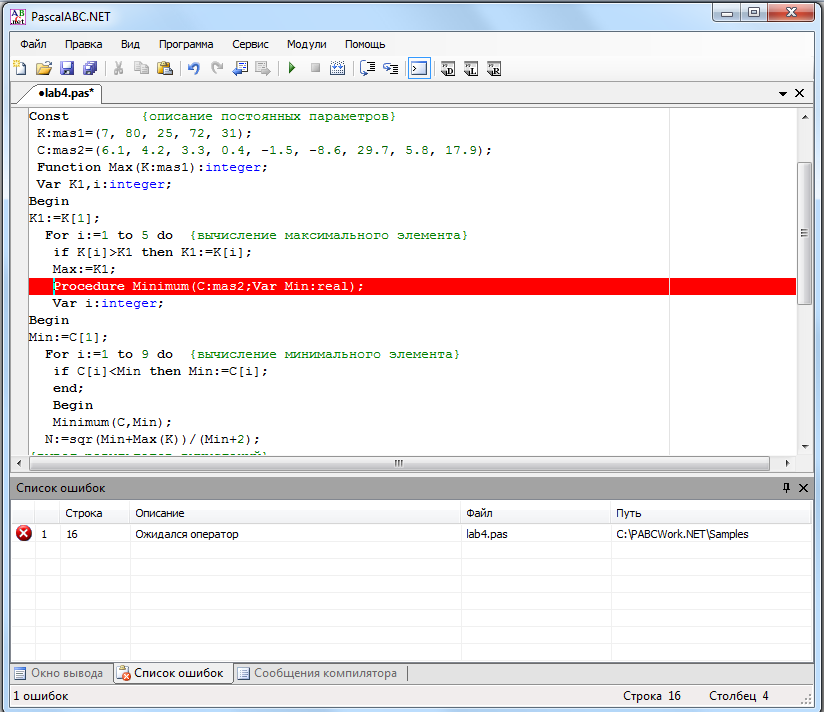


В данном случае, Max описан, как подпрограмма функция:

(**Function** Max(K:mas1):integer;), поэтому использоваться может только со списком фактических параметров. Т.е. оператор, выделенный красной строкой должен записываться следующим образом: N:=sqr(Min+Max(K))/(Min+2);

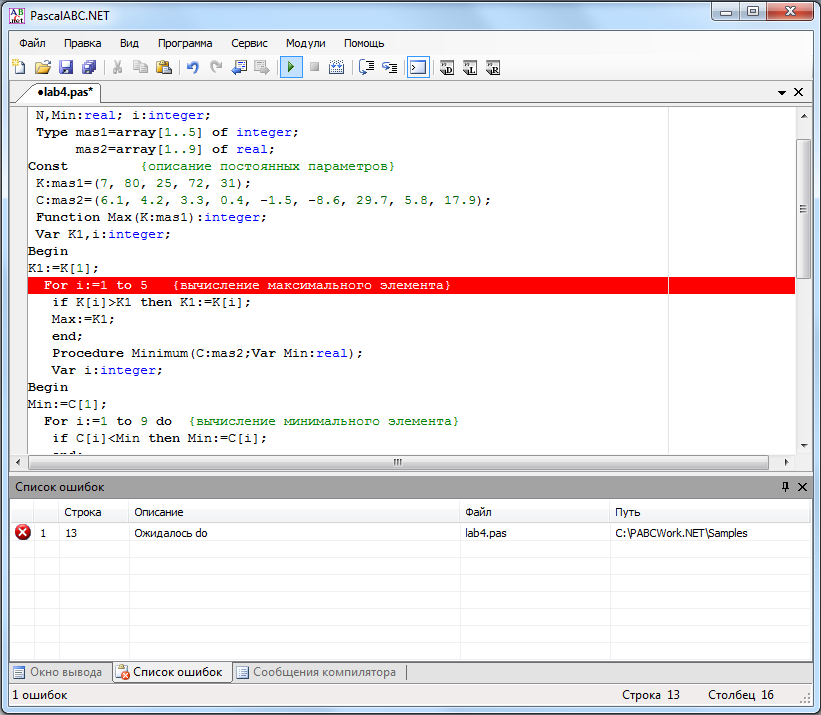


При использовании подпрограммы процедуры (это же касается и подпрограммы функции) между формальными и фактическими параметрами должно быть соответствие по количеству, порядку следования и типу данных. В данном случае, последовательность списка фактических параметров при обращении к подпрограмме процедуре в основной программе Minimum(Min,C)не совпадает с последовательностью списка формальных переменных в заголовке подпрограммы процедуры **Procedure** Minimum(C:mas2;**Var** Min:real). Следует отметить, что имена переменных списков могут не совпадать, а вот соответствие типов переменных списков обязательно диагностируется компилятором и сообщается о возникновении данной ошибки. В нашем случае, обращение к подпрограмме процедуре в основной программе должно быть следующим: Minimum(C,Min).

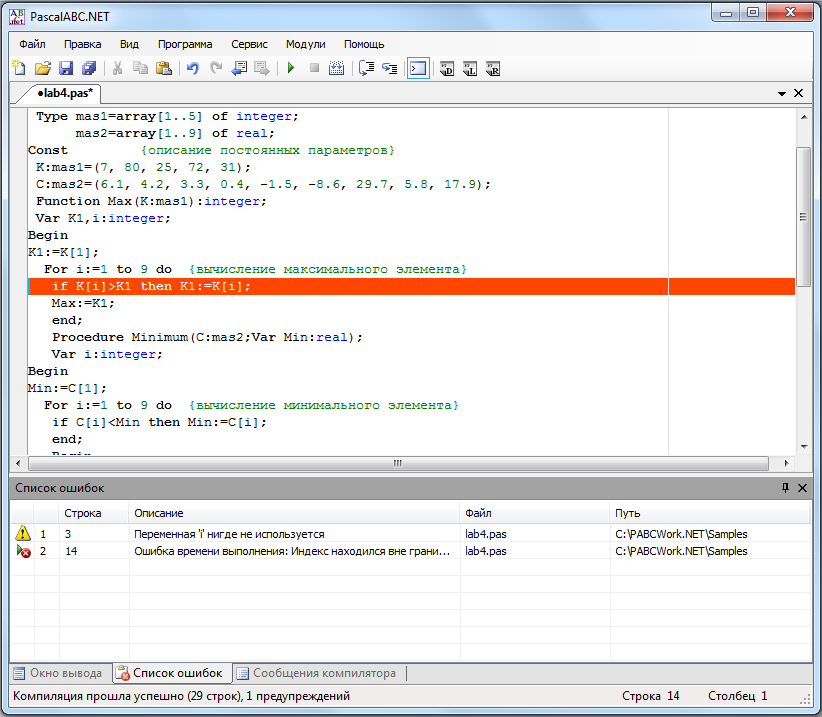


Отсутствует оператор **end**; в конце описания подпрограммы функции, перед оператором, выделенным красной строкой

**Procedure** Minimum(C:mas2;**Var** Min:real);, являющимся заголовком следующей подпрограммы. Следует перед этим оператором поставить оператор **end**;.

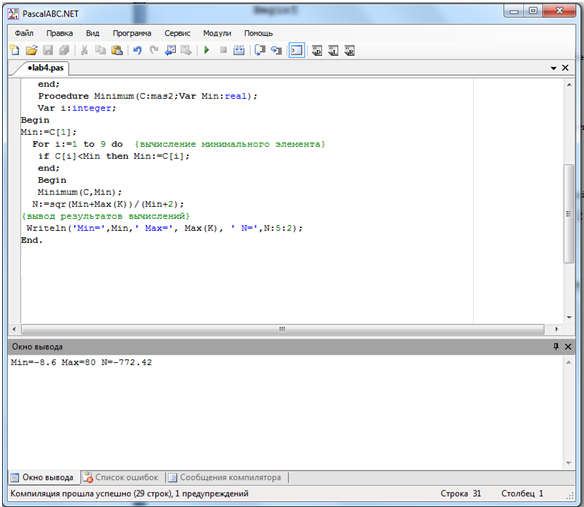


В выделенной строке пропущен оператор цикла **do**. Правильная запись: **For** i:=1 **to** 5 **do**



Верхняя граница цикла (9) выходит за верхнюю границу массива K[i] (5), поэтому и выходит сообщение «…Индекс находится вне границы». В строке, задающей границы циклической процедуры, следует изменить верхнюю границу: **For** i:=1 **to** 5 **do**

Когда программа будет отлажена, в Окне вывода появится результат:



1. Если не удалось завершить отладку программы за один прием, следует записать последнюю версию на диск, и в следующий раз, открыв эту версию продолжить работу (подробно описано в примере № 1).

# Список рекомендуемых источников

1. Мойзес О.Е., А.В. Кравцов. Информатика. Ч. 1: учеб. пособие. – Томск: Изд-во ТПУ, 2007. – 126 с.
2. Кравцов А.В., Мойзес О.Е., Кузьменко Е.А., Баженов Д.А., Коваль П.И. Информатика и вычислительная математика: учеб. пособие для студентов химических специальностей технических вузов. – Томск: Изд-во ТПУ, 2003. – 245 с.
3. Мойзес О.Е., Баженов, Д.А.Коваль П.И., Кузьменко Е.А. Информатика /Учебное пособие, Томск: Изд. ТПУ, 2000. – 119 с.
4. Зубанов Ф. Microsoft Windows 2000. – 2-е изд., испр. – М.: Русская Редакция, 2000. – 592 с.
5. Интернет ресурс: http://pascalabc.net.

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ……………………………………………………………........ 3

## РАБОТА В СРЕДЕ PascalABC.NET ……………………………………....4

# Составление простейших программ на языке Паскаль. Линейные алгоритмы………..……………………………………7

# Программирование разветвляющихся алгоритмов, операторы IF, GOTO……….……...………………..………………...25

# Использование стандартных алгоритмов. Составление программ с использованием подпрограмм процедур и функций»…….……………….…..42

# Список рекомендуемых источников……………………….64

## Учебное издание

**Работа в среде PascalABC.NET**

## Методические указания по выполнению лабораторных работ по дисциплине Информатика

*Составители*

КУЗЬМЕНКО Елена Анатольевна

ПОГАДАЕВА Надежда Игоревна

Рецензент

*кандидат технических наук,*

*доцент кафедры ХТТ и ХК*

*О.Е. Мойзес*

Редактор *С.В. Ульянова*

Компьютерная верстка *Т.И. Тарасенко*

**Отпечатано в Издательстве ТПУ в полном соответствии  
с качеством предоставленного оригинал-макета**

|  |  |  |
| --- | --- | --- |
| Подписано к печати . Формат 60×84/16. Бумага «Снегурочка».  Печать Xerox. Усл.печ.л. 2,09. Уч.-изд.л. 1,89.  Заказ . Тираж экз. | | |
| nqa_iso9001 | Национальный исследовательский Томский политехнический университет  Система менеджмента качества  Томского политехнического университета сертифицирована  NATIONAL QUALITY ASSURANCE по стандарту ISO 9001:2008 | ukas015 |
| logo_izd_TPU. 634050, г. Томск, пр. Ленина, 30.  Тел./факс: 8(3822)56-35-35, www.tpu.ru | | |