

Министерство образования и науки РФ
ФГБПОУ ВПО Тульский государственный университет
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

**ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПЕРЕКЛЮЧАТЕЛЕЙ.
ОТЛАДКА ПРОГРАММЫ.**

Лабораторная работа № 2
по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601

_____ Белым А.А.
(подпись)

Проверил: к. ф.-м. н., доцент

_____ Сулимова В.В.
(подпись)

Тула 2011

Цель работы

Цель работы заключается в том, чтобы научиться пользоваться простейшими компонентами организации переключений (TCheckBox, TRadioGroup), а также написать и отладить программу разветвляющегося алгоритма.

Задание на работу

Рассчитать значение функции:

$$d = \begin{cases} (f(x) - y)^3 + \operatorname{arctg}(f(x)), & x > y \\ (y - f(x))^3 + \operatorname{arctg}(f(x)), & x < y \\ (y + f(x))^3 + 0.5, & x = y \end{cases}$$

В качестве $f(x)$ использовать по выбору: $\sinh(x)$, x^2 , e^x .

This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

Схема алгоритма

На рисунке 1 представлена схема общего алгоритма расчета значения сложной функции

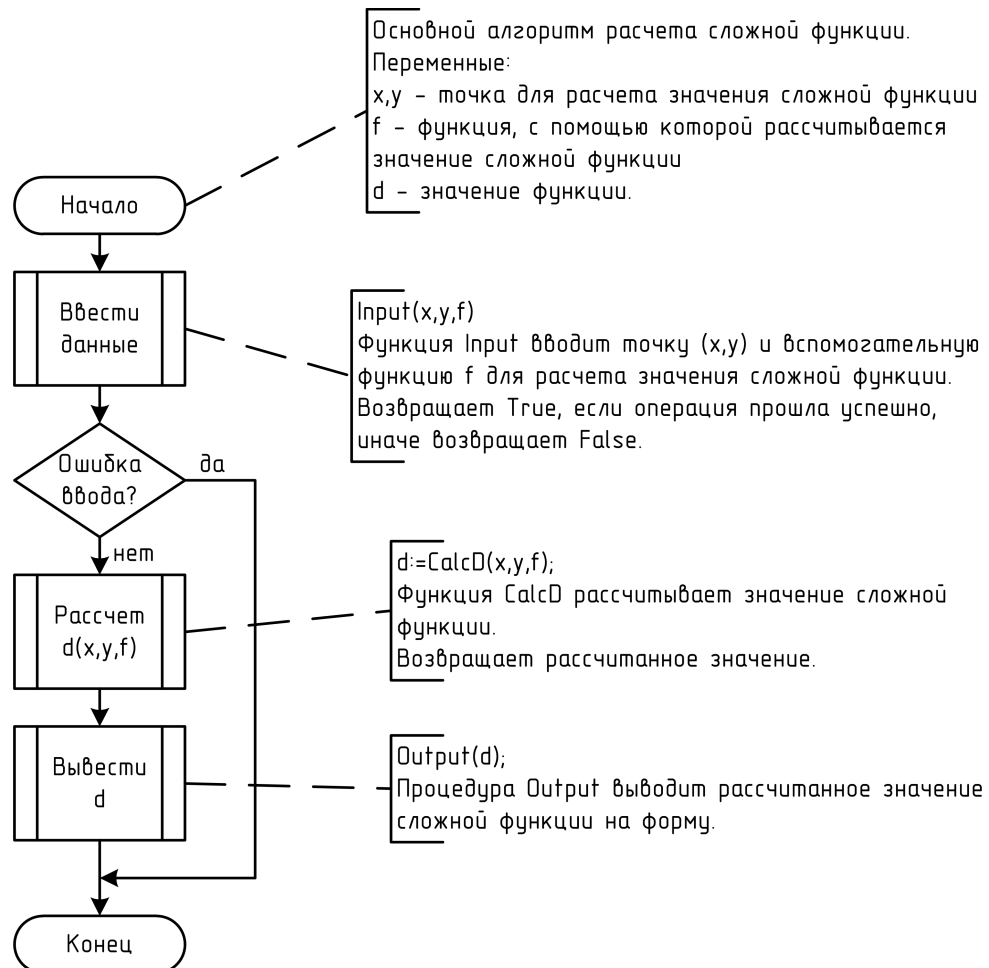


Рисунок 1 - Схема общего алгоритма расчета значения сложной функции

На рисунке 2 представлена схема алгоритма ввода данных для расчета значения сложной функции.

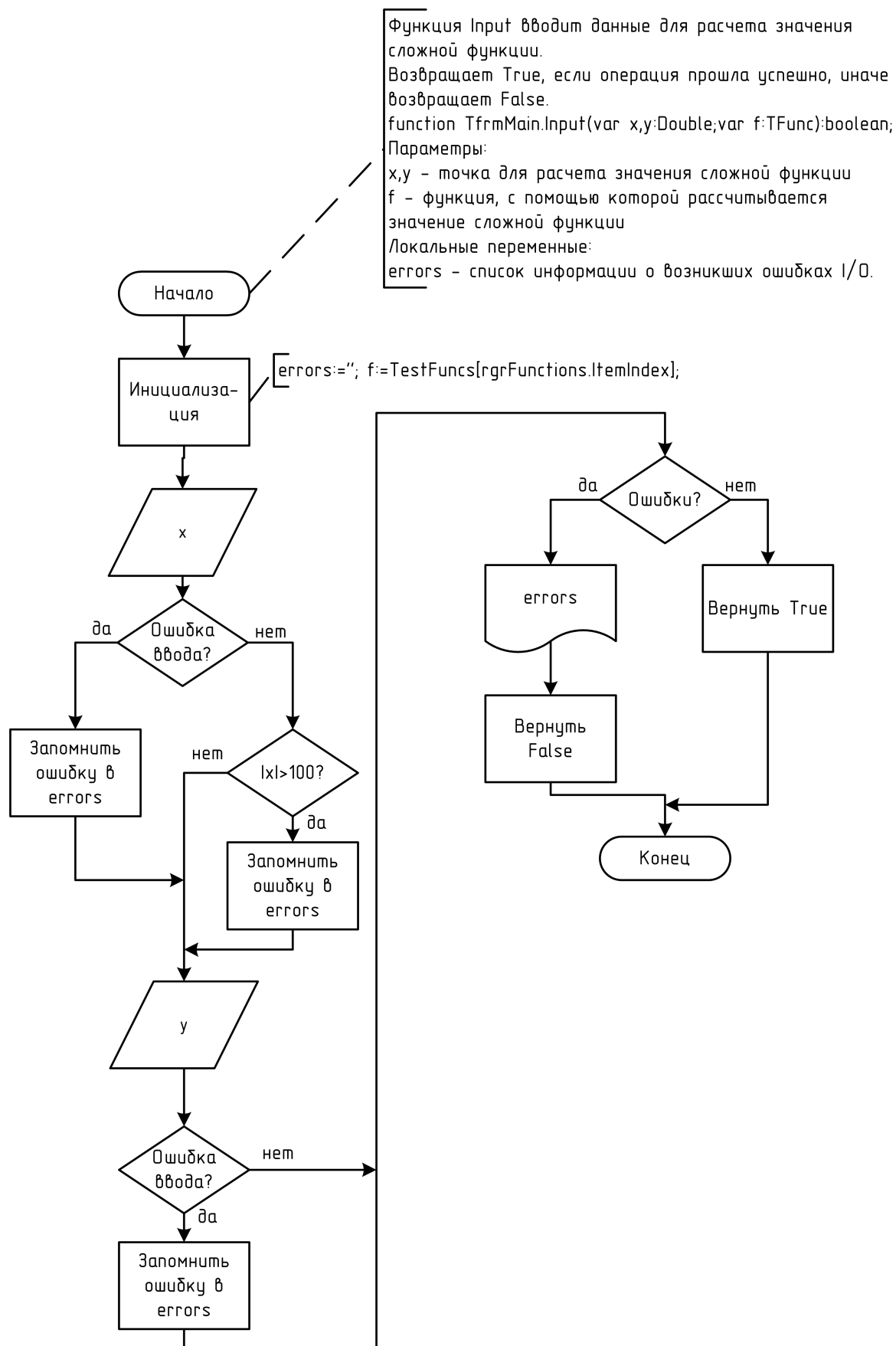


Рисунок 2 - Схема алгоритма ввода данных

На рисунке 3 представлена схема алгоритма расчета значения сложной функции.

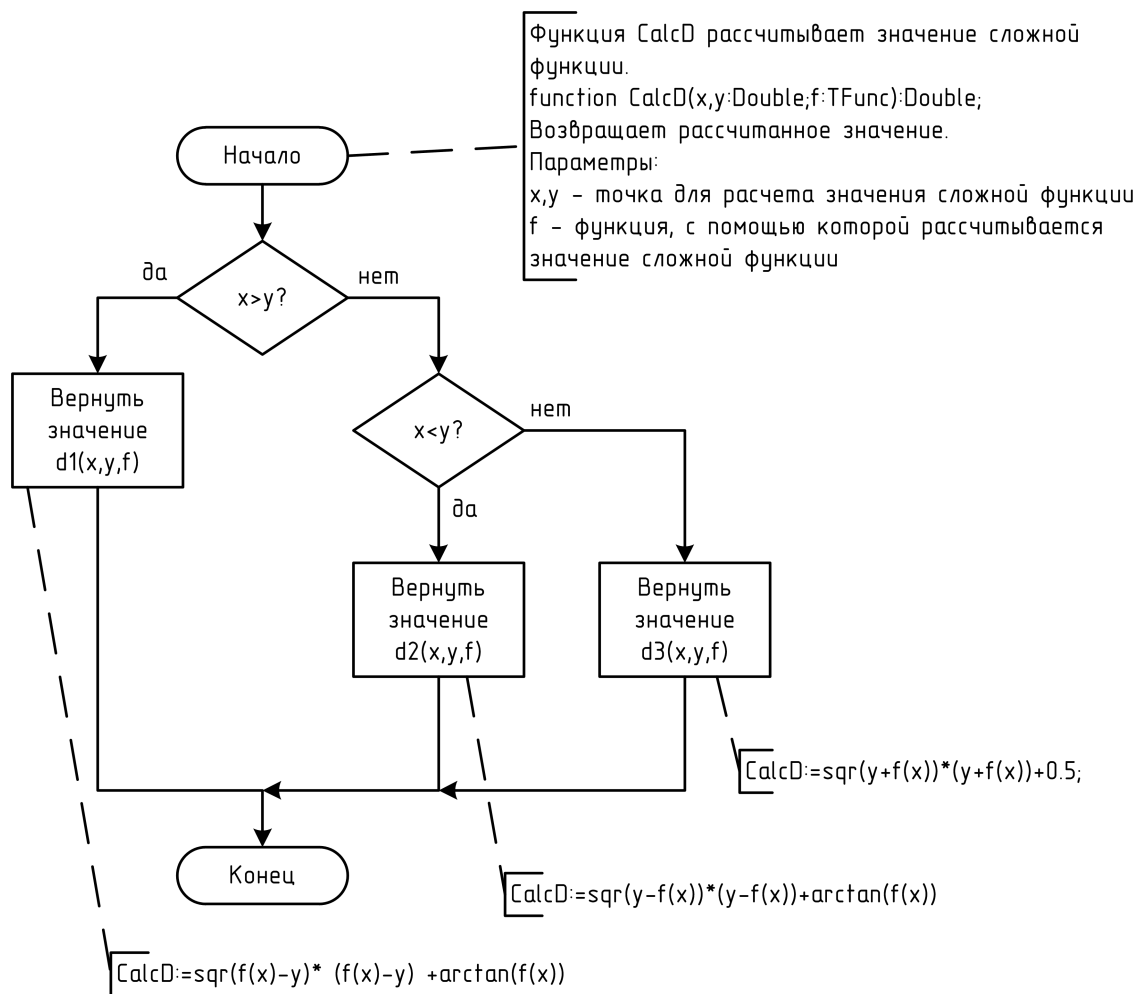


Рисунок 3 - Схема алгоритма расчета значения сложной функции

На рисунке 4 представлена схема алгоритма вывода рассчитанного значения сложной функции.

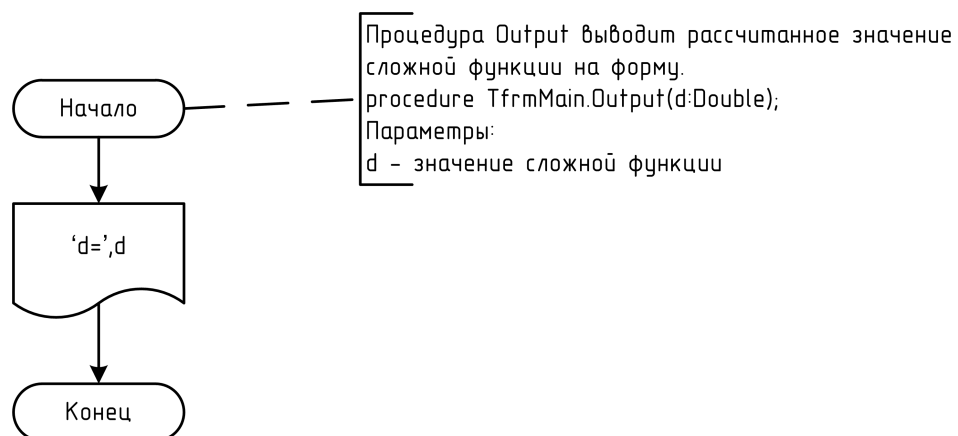


Рисунок 4 - Схема алгоритма вывода результата

Инструкция пользователю

Программа позволяет рассчитать значение функции:

$$d = \begin{cases} (f(x) - y)^3 + \arctg(f(x)), x > y \\ (y - f(x))^3 + \arctg(f(x)), x < y \\ (y + f(x))^3 + 0.5, x = y \end{cases}$$

, где $f(x)$ по выбору может быть: $\sinh(x)$, x^2 , e^x .

Для расчета значения функции сначала выберите вспомогательную функцию, установив переключатель в позицию напротив выбранной функции. Далее введите числа x и y , от которых рассчитывается значение. После ввода данных нажмите кнопку “Посчитать”. Если в данных были допущены ошибки, программа сообщит об этом во всплывающем окне.

Если данные введены правильно, то программа выведет результат в нижнем текстовом поле.

Инструкция программисту

При разработке программы был определен тип `TFunc=function(x:Double):Double` - тип вспомогательных функций. При разработке программы построение интерполяционного многочлена Ньютона с разделенными разностями были написаны следующие процедуры и функции:

1. Процедура `btnRunClick` - основная процедура программы расчета сложной функции.

Параметры процедуры представлены в таблице 1 :

Таблица 1 - Параметры основной процедуры программы

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

Локальные переменные процедуры представлены в таблице 2 :

Таблица 2 - Локальные переменные основной процедуры программы

имя	тип	предназначение
x,y	Double	точка для расчета значения сложной функции
f	TFunc	функция, с помощью которой рассчитывается значение сложной функции
d	Double	значение функции.

2. Функция Input вводит данные для расчета значения сложной функции.

Возвращает True, если операция прошла успешно, иначе возвращает False.

В функции объявляются константы $n=3$ - количество вспомогательных функций, и `TestFuncs:Array[0..n-1] of TFunc = (f1,f2,f3)` - массив вспомогательных функций. `function TfrmMain.Input(var x,y:Double;var f:TFunc):boolean;`

Параметры функции представлены в таблице 3 :

Таблица 3 - Параметры функции ввода данных

имя	тип	предназначение
x,y	Double	точка для расчета значения сложной функции
f	TFunc	функция, с помощью которой рассчитывается значение сложной функции

Локальные переменные функции представлены в таблице 4 :

Таблица 4 - Локальные переменные функции ввода данных

имя	тип	предназначение
errors	string	список информации о возникших ошибках I/O.

3. Функция CalcD рассчитывает значение сложной функции.

Возвращает рассчитанное значение.

`function CalcD(x,y:Double;f:TFunc):Double;`

Параметры функции представлены в таблице 5 :

Таблица 5 - Параметры функции расчета значения сложной функции

имя	тип	предназначение
x,y	double	точка для расчета значения сложной функции
f	TFunc	функция, с помощью которой рассчитывается значение сложной функции

4. Процедура Output выводит рассчитанное значение сложной функции на форму.

```
TfrmMain.Output(d:Double);
```

Параметры процедуры представлены в таблице 6 :

Таблица 6 - Параметры процедуры вывода значения функции

имя	тип	предназначение
d	Double	значение сложной функции

Текст программы

Ниже представлен текст программы на языке Delphi 7, реализующей расчет значения сложной функции.

```
unit UnitMain;
interface

uses
  Windows, SysUtils,
  Forms, Dialogs,
  Math, Controls, StdCtrls, ExtCtrls, jpeg, Classes;

type
  TFunc = function(x:Double):Double;
  TfrmMain = class(TForm)
    lblInfo: TLabel; //информация о программе
    grpParams: TGroupBox; //группа параметров ввода
    btnRun: TButton; //кнопка запуска
    btnExit: TButton; //кнопка выхода
    txtX: TLabelledEdit; //поле ввода x
    txtY: TLabelledEdit; //поле ввода y
    imgFormula: TImage; //формула расчета
    memResult: TMemo; //поле вывода результатов
    rgrFunctions: TRadioGroup; //переключатели функций
    procedure btnRunClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);

  private
    { Private declarations }
    function Input(var x,y:Double;var f:TFunc):boolean;
    procedure Output(d:Double);
  public
    { Public declarations }

  end;
  function CalcD(x,y:Double;f:TFunc):Double;
var
  frmMain: TfrmMain;

implementation

{$R *.dfm}
(*
Процедура Output выводит рассчитанное значение сложной функции на форму.
Параметры:
d – значение сложной функции
*)
procedure TfrmMain.Output(d:Double);
begin
  memResult.Text:=memResult.Text+'d='+FloatToStr(d)+'#13#10';
end;

function f1(x:Double):Double;
begin
  f1:=sinh(x);
end;
function f2(x:Double):Double;
begin
  f2:=sqr(x);
end;
function f3(x:Double):Double;
begin
  f3:=exp(x);
```

```

end;

(*
Функция Input вводит данные для расчета значения сложной функции.
Возвращает True, если операция прошла успешно, иначе возвращает False.
Параметры:
x,y – точка для расчета значения сложной функции
f – функция, с помощью которой рассчитывается значение сложной функции
Локальные переменные:
errors – список информации о возникших ошибках I/O.
*)
function TfrmMain.Input(var x,y:Double;var f:TFunc):boolean;
var errors:string;
const n=3;
const TestFuncs:Array[0..n-1] of TFunc = (f1,f2,f3);
begin
  errors:='';
  f:=TestFuncs[rgrFunctions.ItemIndex];
  if not TryStrToFloat(txtX.Text,x) then
    errors:=#10#13+'x – не вещественное число!';
  else if abs(x)>100 then
    errors:= #10#13+'x слишком большое!';
  if not TryStrToFloat(txtY.Text,y) then
    errors:=errors+#10#13+'y – не вещественное число!';
  if (errors<>'') then begin
    MessageDlg('Ошибки: '+errors,mtError,[mbOK],0);
    Input:=False;
  end else Input:=True;
end;

(*
Функция CalcD рассчитывает значение сложной функции.
Возвращает рассчитанное значение.
Параметры:
x,y – точка для расчета значения сложной функции
f – функция, с помощью которой рассчитывается значение сложной функции
*)
function CalcD(x,y:Double;f:TFunc):Double;
begin
  if x>y then
    CalcD:=sqr(f(x)-y)*(f(x)-y)+arctan(f(x))
  else if x<y then
    CalcD:=sqr(y-f(x))*(y-f(x))+arctan(f(x))
  else
    CalcD:=sqr(y+f(x))*(y+f(x))+0.5;
end;

(*
Процедура btnRunClick – основная процедура программы расчета сложной функции.
Локальные переменные:
x,y – точка для расчета значения сложной функции
f – функция, с помощью которой рассчитывается значение сложной функции
d – значение функции.
*)
procedure TfrmMain.btnRunClick(Sender: TObject);
var x,y,d:Double;f:TFunc;
begin
  if Input(x,y,f) then begin
    d:=CalcD(x,y,f);
    Output(d);
  end;
end;

procedure TfrmMain.btnExitClick(Sender: TObject);
begin
  Application.Terminate;

```

end;

end.

Тестовый пример

Ниже на рисунке 5 представлен пример работы программы при $x=0$, $y=-1$, и вспомогательной функции $f(x) = e^x$.

Лабораторная работа №2

Программа вычисляет значение выражения:

$$d = \begin{cases} (f(x) - y)^3 + \arctg(f(x)), & x > y \\ (y - f(x))^3 + \arctg(f(x)), & x < y \\ (y + f(x))^3 + 0.5, & x = y \end{cases}$$

f(x)=

☐ sh(x)

☐ x^2

☒ exp(x)

x: 0

y: -1

Посчитать

Выход

Белым Андрей Андреевич, гр.220601
Лабораторная работа №2
d=8,78539816339745

Рисунок 5 - Пример работы программы с правильными исходными данными

На рисунке 6 представлен пример работы программы с неправильными исходными данными.

Лабораторная работа №2

Программа вычисляет значение выражения:

$$d = \begin{cases} (f(x) - y)^3 + \arctg(f(x)), & x > y \\ (y - f(x))^3 + \arctg(f(x)), & x < y \\ (y + f(x))^3 + 0.5, & x = y \end{cases}$$

f(x)=

☐ sh(x)

☒ x^2

☐ exp(x)

x: 1e100

y: abcd

Посчитать

Выход

Белым Андрей
Лабораторная

Error

Ошибки:
x слишком большое!
y - не вещественное число!

OK

Рисунок 6 - Пример работы программы с неправильными исходными данными

Вывод

В этой лабораторной работе я изучил операторы ветвления и циклов Delphi и компонент TMaskEdit. Именно операторы ветвления и циклов позволяют программе изменять поведение при получении внешних данных, а компонент TMaskEdit помогает организовать их безопасное получение.