

Практическая работа № 27

Тема: *Составление программ линейной структуры с использованием стандартных функций. Ввод-вывод данных в Delphi.*

Цель: Научиться составлять простейшие программы в среде Delphi, организовывать ввод-вывод данных, использовать меню и вкладки.

Время: 60 мин.

Задание: Решить задачу согласно варианта, организовав ввод данных и вывод результата.

Литература:

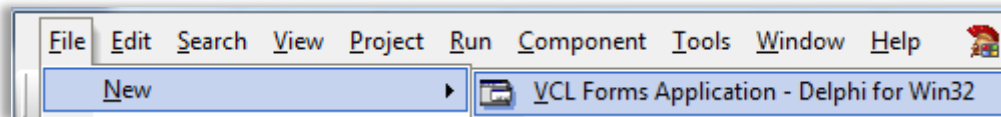
1. Фаронов В.В. Программирование на языке высокого уровня: Учебник для вузов. – СПб.: Питер, 2003.
2. Мансуров К.Т. Основы программирования в среде Lazarus.
3. Н.Б. Культин. Основы программирования в Delphi XE.
4. Бобровский С.И. Delphi 7. Учебный курс. – СПб.: Питер, 2005.

Содержание отчёта:

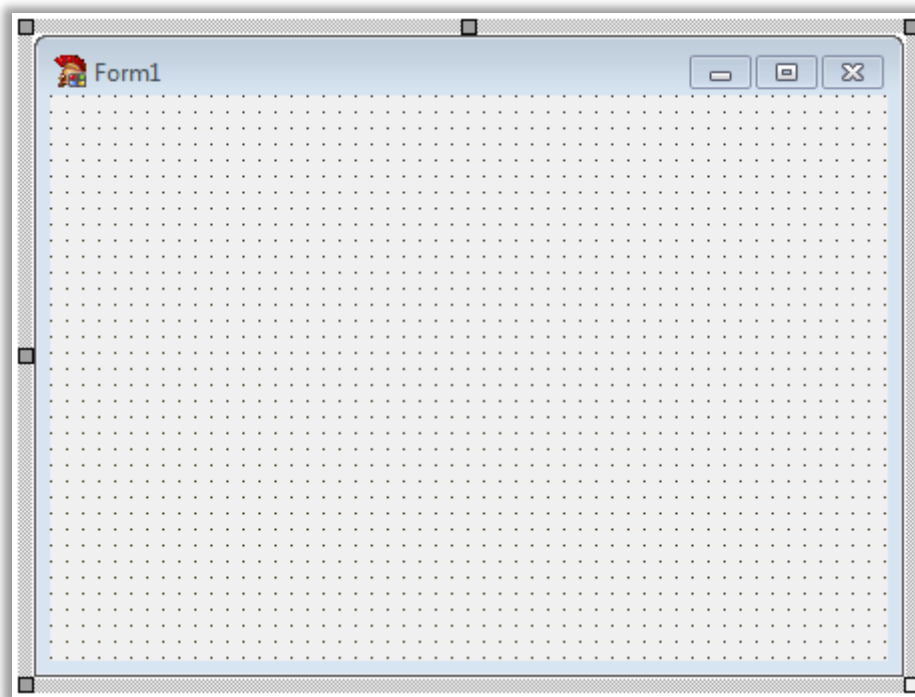
- Ответы на вопросы, поставленные в пунктах описания последовательности выполнения работы.
- Блок-схема алгоритма и текст программы.
- Выводы по работе (что изучили, чему научились).

Последовательность выполнения работы:

1. Запустите Delphi и создайте новое приложение.
 - Для этого выберите “File” – “New” – “VCL Forms Application – Delphi for Win32”:

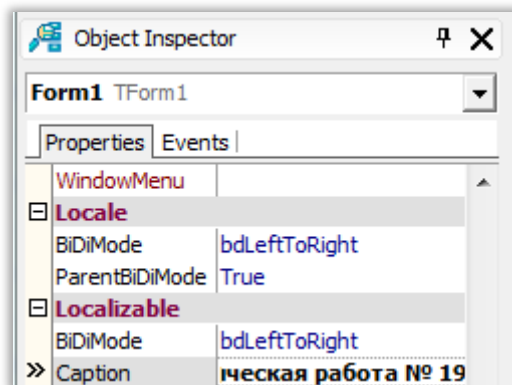


- Появится окно формы, на которой размещаются все необходимые компоненты.

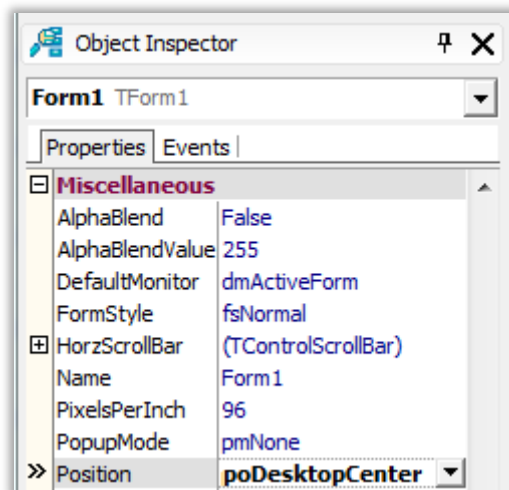



Окно формы будущей программы

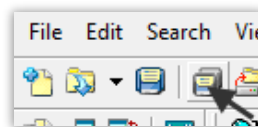
2. В заголовок главной формы поместите текст: «<Имя Фамилия>. Практическая работа № 27».
 - Для этого при выделенной главной форме в Инспекторе объектов на странице «Свойства» (Properties) во вкладке Action свойству Caption (Заголовок) присвойте значение «Практическая работа № 27»:



- Измените при необходимости размер формы.
- Свойство «Position» («Позиция») измените так, чтобы ваша программа располагалась в центре экрана:



- Проверьте, что на главной панели Delphi активен зелёный треугольник  - запуск программы.
3. Сохраните проект под именем «Pr27».
 - Для сохранения нажмите кнопку «Save all» или **Shift+Ctrl+S**:



сохраните модуль (файл **Unit1.pas**) и проект (файл **Pr27.bdsproj**), создав папку «Практическая работа 27, Фамилия Имя» в папке Мои документы.

4. Ознакомьтесь с заданием. Номер задания соответствует номеру студента по списку в журнале.

Задание:

Даны **x, y, z** . Вычислить **a, b**, если:

$$1. \quad a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}} \quad b = x(\operatorname{arctg}(z) + e)$$

$$2. \quad a = \frac{3 + e^2}{1 + x^2 |y - \operatorname{tg}(z)|} \quad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{(x-y)^2}{3}$$

$$3. \quad a = (1+y) \frac{x + \frac{y}{(x^2+4)}}{e^2 + \frac{1}{(x^2+4)}} \quad b = \frac{1 + \cos^3(y-x)}{\frac{x^2}{2} + \sin^2(z)}$$

$$4. \quad a = y + \frac{x}{y^2 + \left| \frac{x^2}{y+x^2} \right|} \quad b = (1 + \operatorname{tg}^2 \frac{z}{2})^2$$

$$5. \quad a = \frac{2 \cos^4(x - \frac{\pi}{6})}{\frac{1}{2} + \sin^2 y} \quad b = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$$

$$6. \quad a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}} \quad b = x^3(\operatorname{arctg}^3 z + e)$$

$$7. \quad a = \frac{1 + \sin^2(x+y)}{2 + \left| x - \frac{2x}{(1+x^2y^2)} \right|} + x \quad b = \cos^2(\operatorname{arctg} \frac{1}{z})$$

$$8. \quad a = \ln \left| (y - \sqrt{|x|}) \left(x - \frac{y}{z + \frac{x^2}{4}} \right) \right| \quad b = x - \frac{x^2}{3} + \frac{x+y}{x}$$

$$9. \quad a = \frac{x^2}{8 + \frac{x^2}{3} + \frac{y^2}{6}} \quad b = x(\cos^3(x+z) + 1)$$

$$10. \quad a = \frac{|5 - 2e|}{1 + x^2(y - \operatorname{tg}(z))} \quad b = |y - 4| + \frac{(y-x)^2}{6} + \frac{(x-y)^2}{7}$$

11.

$$a = (2+x) \frac{1 + \frac{y}{(x^2+3)}}{y^2 + \frac{1}{(z^2+4)}} \quad b = (1 + \operatorname{tg}^2 \frac{x}{2})^2$$

12.

$$a = \frac{z+10}{x^2 + \left| \frac{y^2}{y+x^2} \right|} + y \quad b = (1 + \cos^2 \frac{z}{2})^2$$

13.

$$a = \frac{4 \sin(x - \frac{\pi}{3})}{\frac{1}{3} + \sin^2 x} \quad b = \frac{z^2}{3 + \frac{z^2}{5}}$$

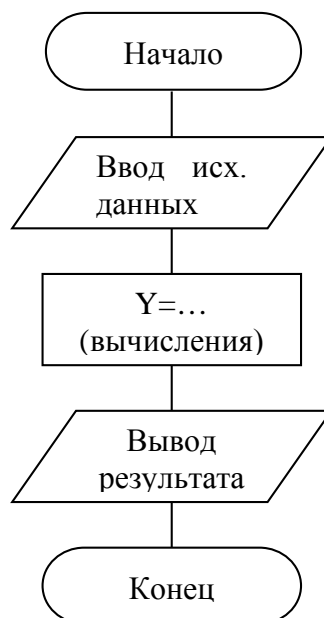
14.

$$a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + 2x^2 + \cos(x)} \quad b = x^2 (\operatorname{arctg}(z) + y)$$

15.

$$a = \frac{1 + \cos^2(x+y)}{2 + \left| x - \frac{2x}{(1+x^2z^2)} \right|} + x \quad b = \sin^2(\operatorname{arctg} \frac{1}{x})$$

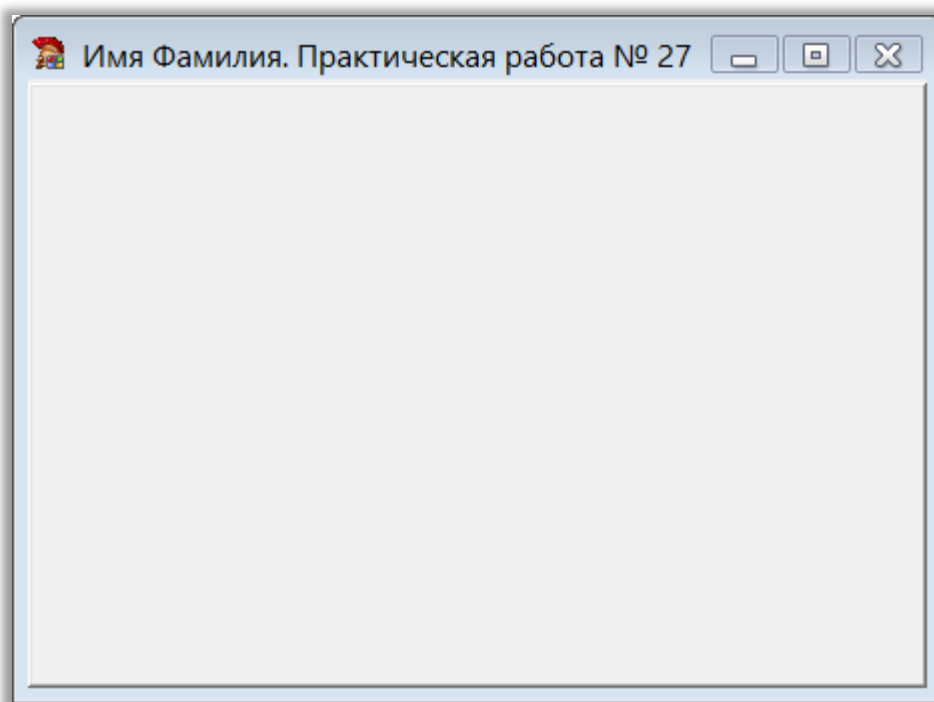
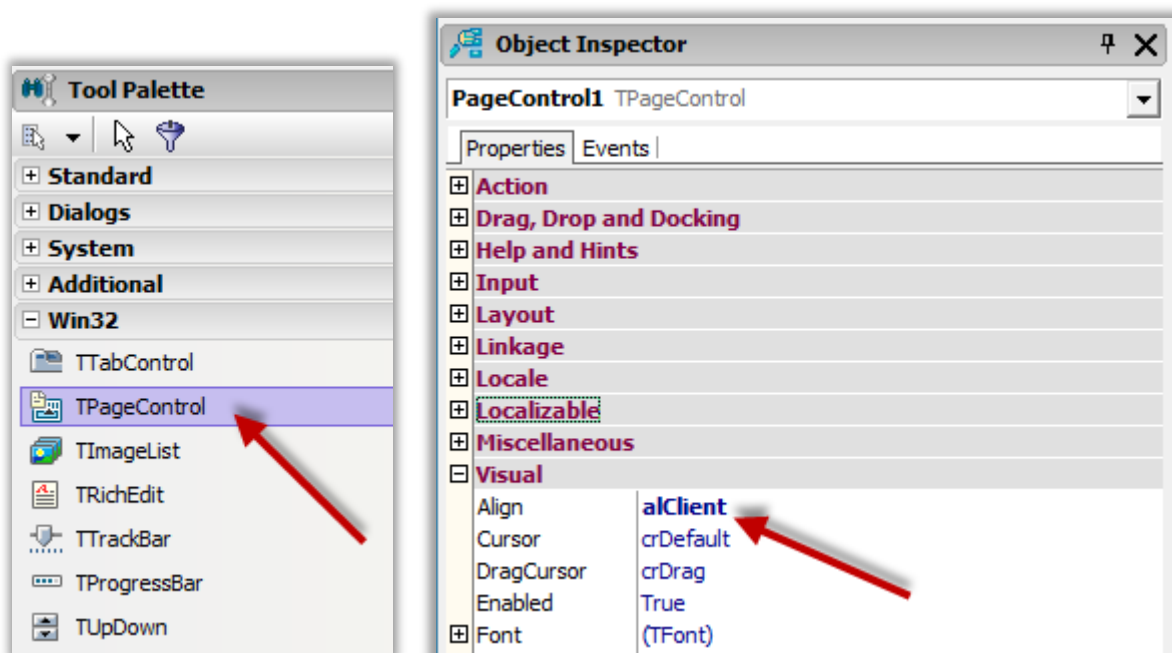
5. Составьте (по предлагаемому образцу) структуру алгоритма линейной структуры, в соответствии с которым решается поставленная задача.



Рассмотрим построение программы для случая, когда

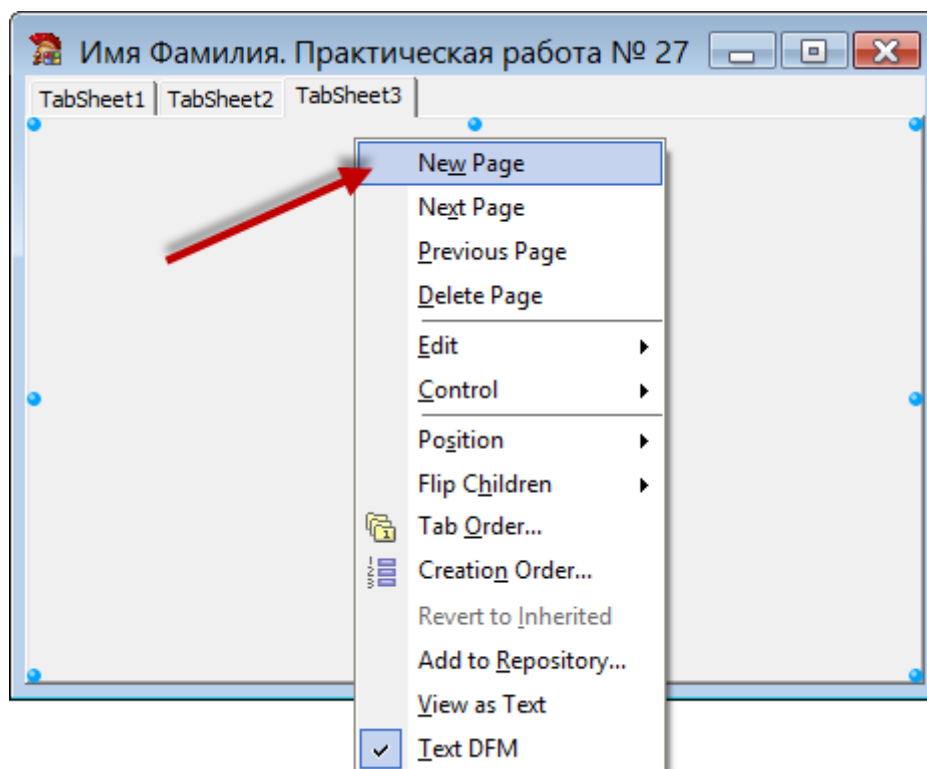
$$a = (1+y) \frac{x + \frac{y}{(x^2+4)}}{e^2 + \frac{1}{(x^2+4)}} \quad b = \cos^2\left(\operatorname{arctg} \frac{1}{z}\right)$$

С целью экономии пространства окна приложения и размещения на одном и том же месте страниц разного содержания воспользуемся многостраничной панелью – компонентом **PageControl**. Перенесите компонент **PageControl** на форму (из библиотеки компонентов **Win32**) и задайте его свойству **Align** значение **alClient** (растяните его границы так, чтобы он занял всё пространство формы).

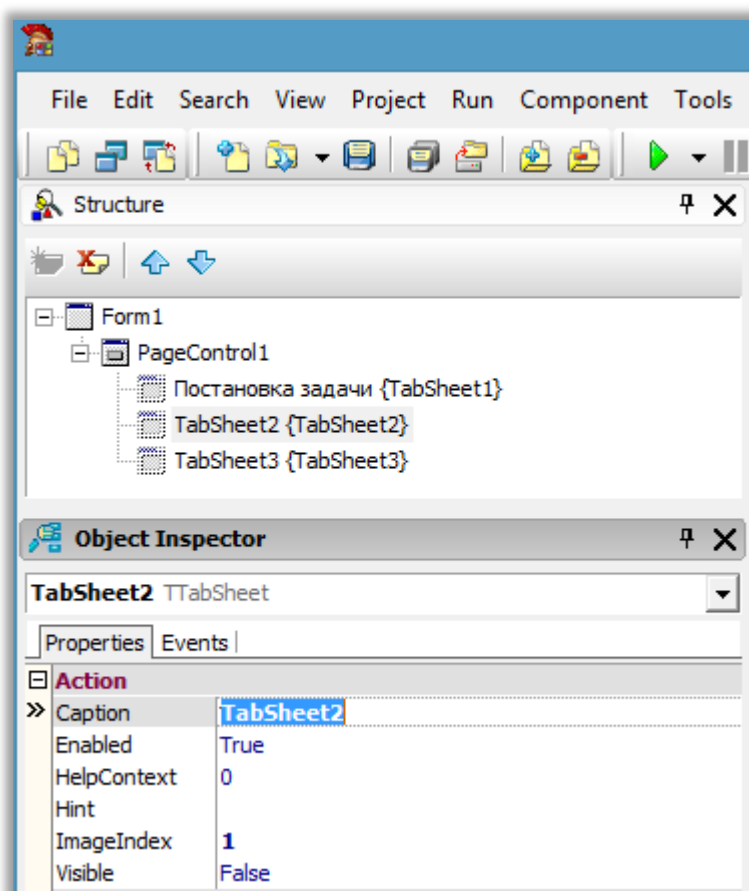


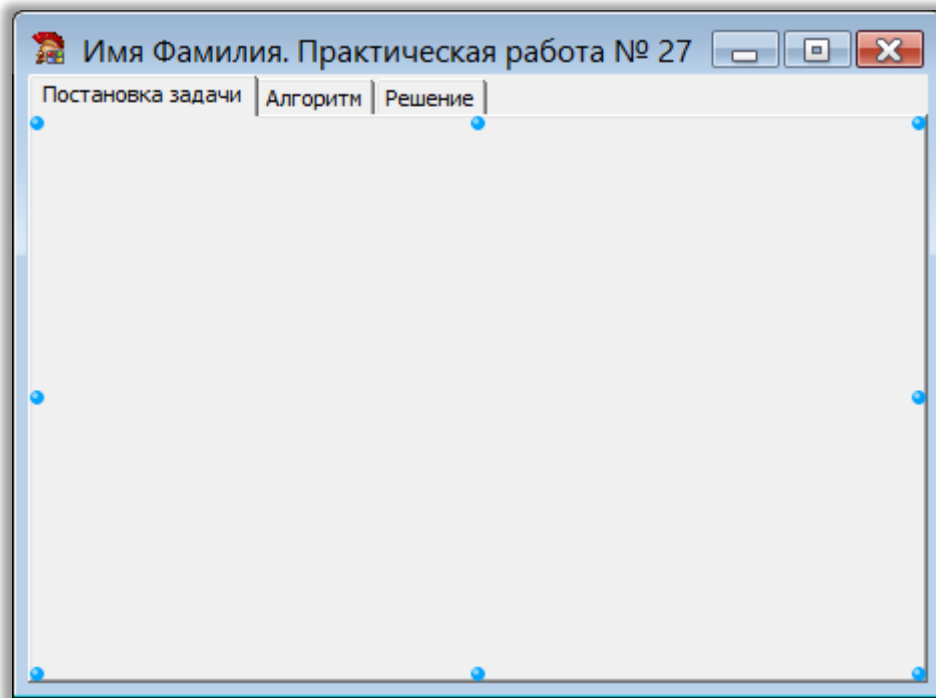
Щёлкните правой кнопкой мыши по панели. Во всплывшем контекстном меню вы можете видеть команды: **New Page** – создать новую страницу, **Next Page** – переключиться на следующую страницу, **Previous Page** – переключиться на предыдущую страницу. Каждая создаваемая страница является объектом типа **TTabSheet**. Выберем пункт **New Page** – создать главную

страницу. Появляется первая страница **TabSheet1**. Аналогично создаём вторую страницу **TabSheet2** и третью – **TabSheet3**.



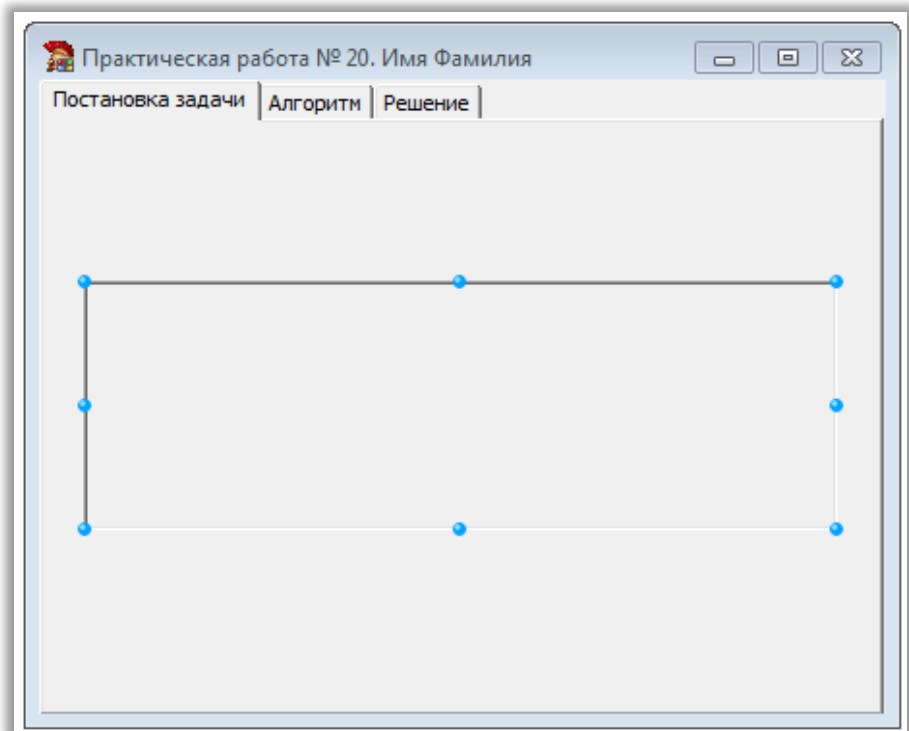
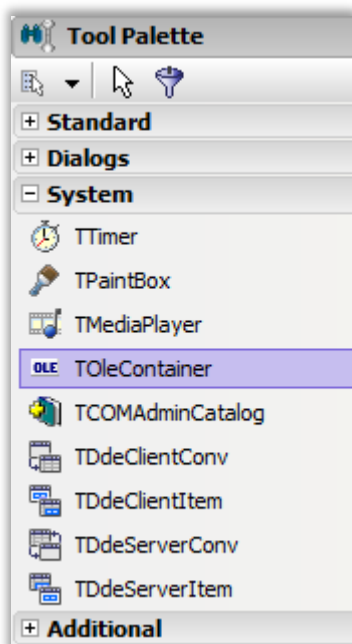
В Инспекторе объектов присвоим свойству **Caption** объекта **TabSheet1** значение «Постановка задачи», **TabSheet2** – «Алгоритм», а **TabSheet3** – «Решение».



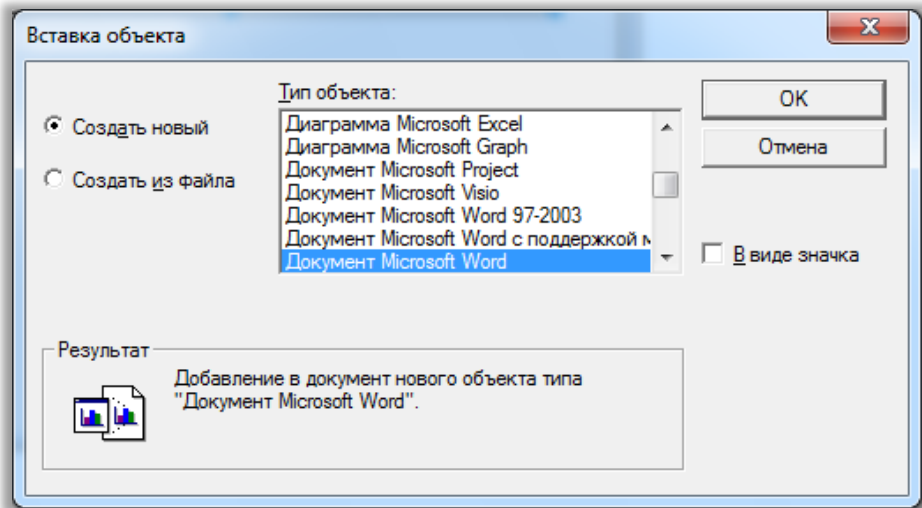
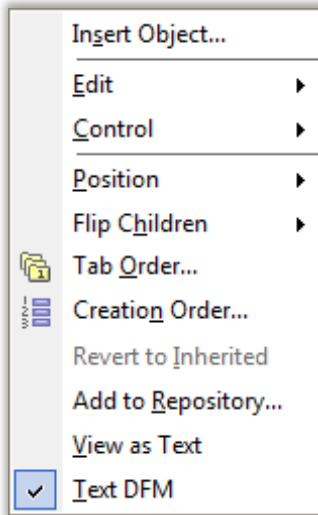


Формулировка постановки задачи может содержать специальные математические символы, которые не могут быть введены с клавиатуры, например, в виде метки **Label**. Поэтому воспользуемся контейнером **OLE – OLEContainer** (библиотека **System**) и создадим область клиента для объекта **OLE** (OLE (англ. *Object Linking and Embedding*, произносится как oh-lay [олэй]) — технология связывания и внедрения объектов в другие документы и объекты, разработанная корпорацией Майкрософт. В 1996 году Microsoft переименовала эту технологию в *ActiveX*).

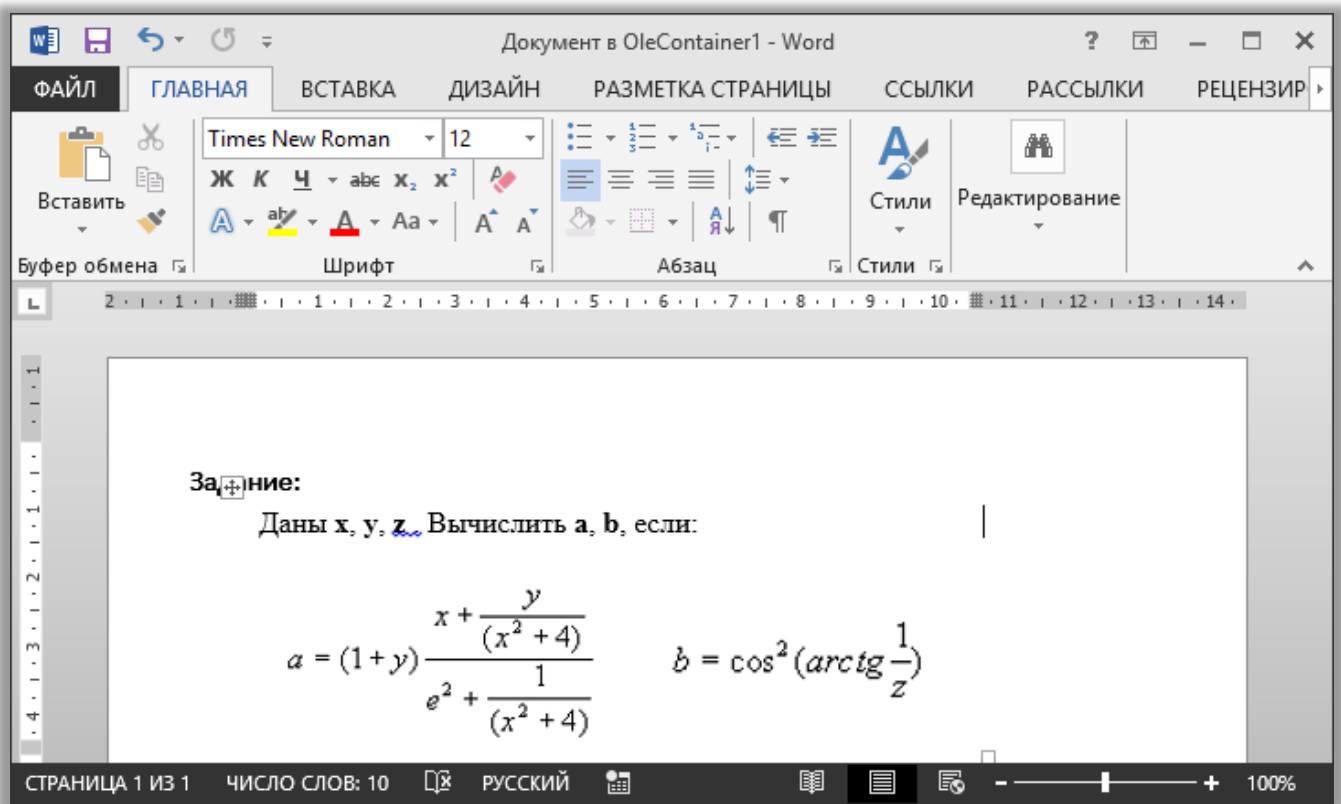
Разместим контейнер OLE на странице TabSheet1 «Постановка задачи».



Щёлкнув ПКМ, вызовем контекстное меню и выберем первый пункт **Insert Object...**, в котором выберем вставку объекта – «Документ Microsoft Word».

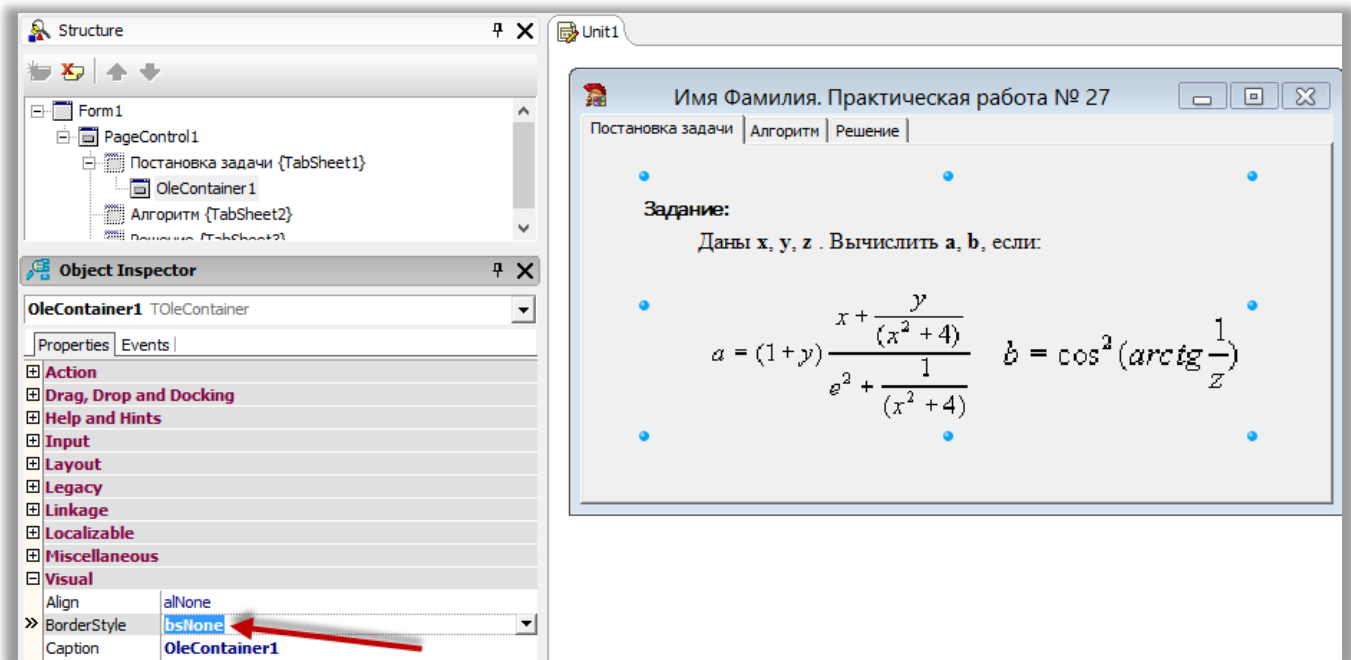


В открывшемся новом документе вводим с клавиатуры или копируем готовый текст формулировки постановки задачи, затем закрываем документ.

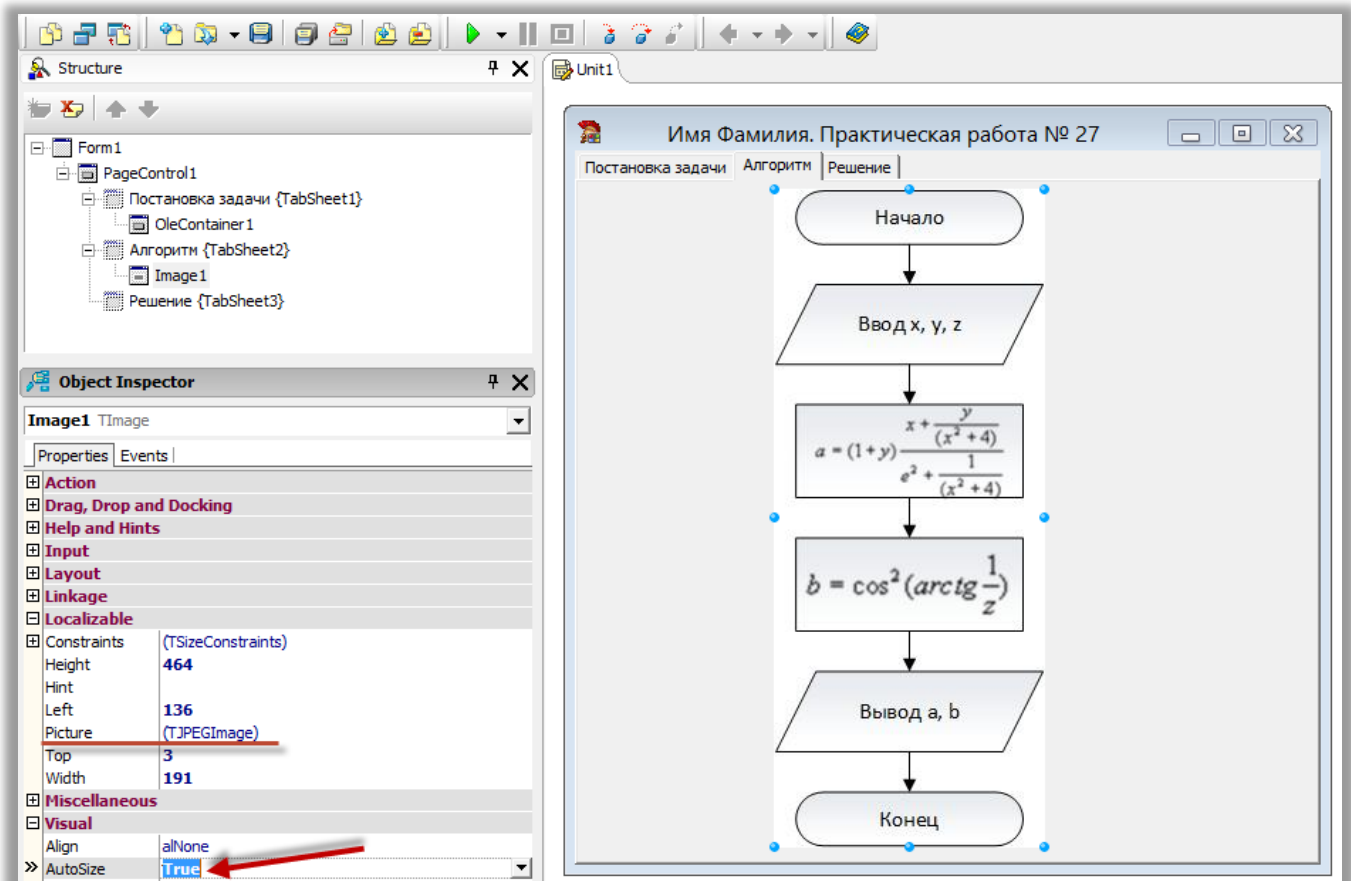


Текст размещается в контейнере OLE.

Для того, чтобы убрать рамку вокруг текста, можно свойству **BorderStyle** присвоить значение **bsNone**:



На странице **TabSheet2** «Алгоритм» располагаем блок-схему алгоритма в виде файла-картинки (растрового изображения), воспользовавшись холстом **TImage** из библиотеки **Additional**.

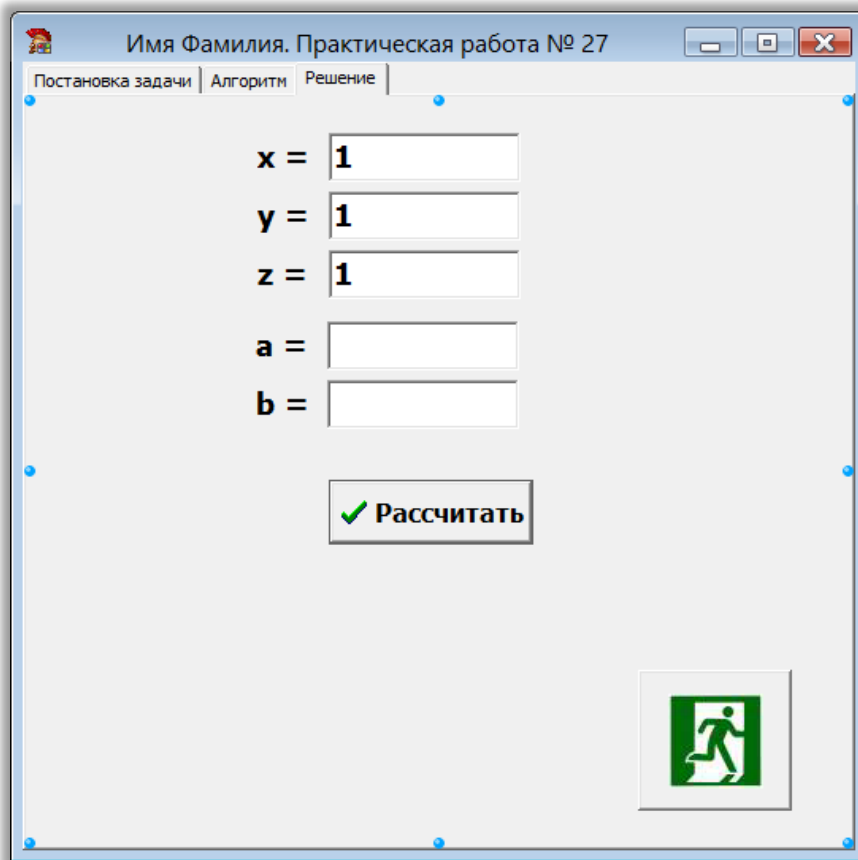


На странице **TabSheet3** «Решение» располагаем компоненты **Label** и **Edit** для ввода исходных данных и вывода результатов. Имя объектов **Edit** (свойство **Name**) изменяем на **Edit_x** (компонент Edit для ввода значения x), **Edit_y** (компонент Edit для ввода значения y), **Edit_z** (компонент Edit для ввода значения z), **Edit_a** (компонент Edit для вывода значения a), **Edit_b** (компонент Edit для вывода значения b). Свойство **Text** для элементов **Edit_x**, **Edit_y**, **Edit_z**

изменяем, записываем туда значение «1» (значение «по-умолчанию», для упрощения процесса отладки программы).

Свойству **ReadOnly** объектов **Edit_a** и **Edit_b** присваиваем значение True, чтобы исключить возможность ввода пользователем данных в окно вывода результатов.

Внизу формы располагаем две кнопки. Кнопку «Рассчитать» берём типа **TBitBtn** (свойству **Kind** присваиваем значение **bkOk**), вторую кнопку для выхода из программы – типа **TSpeedButton** (выбираем свойство **Glyph** и загружаем картинку **exit.bmp**).



Дважды нажмём на кнопку «Рассчитать» и введём программный код, который будет выполняться при нажатии пользователем этой кнопки.

Для ввода вещественных чисел («с плавающей запятой») применяется функция **StrToFloat**, преобразующая строки (текст) в числа.

Для вывода вещественных чисел («с плавающей запятой») применяется функция **FloatToStr**, преобразующая числа в строки (текст).

В Delphi обычные математические операции можно выполнить, используя соответствующие им специальные функции. Список этих функций приведен ниже в таблице 1.

Табл. 1 – Математические функции

Формула	Функция	Значение
$ x $	Abs (x)	Абсолютное значение n
$[x]$	Trunc(x)	Целая часть числа x
$\{x\}$	Frac(x)	Дробная часть X
\sqrt{x}	Sqrt (x)	Квадратный корень из x
x^2	Sqr (x)	Квадрат x
Sin x	Sin (x)	Синус x
Cos x	Cos (x)	Косинус x

arctg x	Arctan (x)	Арктангенс x
e ^x	Exp(x)	Экспонента x
ln x	Ln(x)	Натуральный логарифм x
π	Pi	Число π
x ⁿ	exp(n*ln(x))	Число x в степени n
ⁿ √x	exp(1/n*ln(x))	Корень в n-ой степени
	Random(n)	Случайное целое число в диапазоне от 0 до n- 1

Для округления до n знаков после запятой используется функция FormatFloat:

label1.caption:=FormatFloat('#.##', ваше_число). Можно использовать функцию FloatToStrF: FloatToStrF(a,ffFixed,6,2). Здесь a – ваше число, ffFixed – формат вывода вещественного числа, 2 – количество знаков после запятой, 6 – точность (количество знаков перед запятой)

Операторы деления

7 | 2

6 | 3 ← div — целочисленное деление, т.е. дробная часть отбрасывается. (7 : 2 = 3)

1 ← mod — получение остатка от целочисленного деления (7 : 2 = 1)

Язык программирование Delphi имеет стандартный модуль **Math**, который позволяет использовать в него входящие процедуры и функции для математических преобразований. **Для подключения модуля необходимо прописать Math в разделе uses!!!**

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, OleCtrls, ComCtrls, Buttons, StdCtrls, Math;
```

Ниже описаны некоторые математические функции, для работы с числовыми типами.

- **ArcSin(x)** - вычисляется арксинус аргумента,
- **ArcCos(x)** - возвращается арккосинус аргумента,
- **Tan(x)** - определяется тангенс аргумента (в радианах задается угол),
- **Cotan(x)** - возвращается котангенс угла (угол в радианах),
- **Log10(x)** - вычисление десятичного логарифма,
- **SinCos(x, sin, cos)** - вычисляется косинус и синус угла одновременно,
- **Log2(x)** - производит вычисление двоичного логарифма числа,
- **LogN(y,x)** - вычисляется логарифм x по основанию y,
- **Max(a,b)** - выбирает максимальное число из 2 чисел,
- **Min(a,b)** - возвращает минимальное число из двух чисел,
- **Power (X, N)** - возведение числа X в произвольную степень N.
- **hypot** - вычисление гипотенузы по длине катетов;
- **radtodeg, degtorad** - преобразование радианов в градусы и наоборот;
- **floor** - округление в меньшую сторону;
- **ceil** - округление в большую сторону;
- **sincos** - вычисление синуса и косинуса;
- **mean** - среднее арифметическое в комбинации чисел.
(Например: caption:=FloatToStr(mean([10,10,2])))- находит среднее арифметическое трех данных чисел, но, конечно же, искомым чисел может быть и больше, и меньше указанных в примере трех.)

Таким образом, получим следующий код:

```
var
  Form1: TForm1;

implementation
  {$R *.dfm}

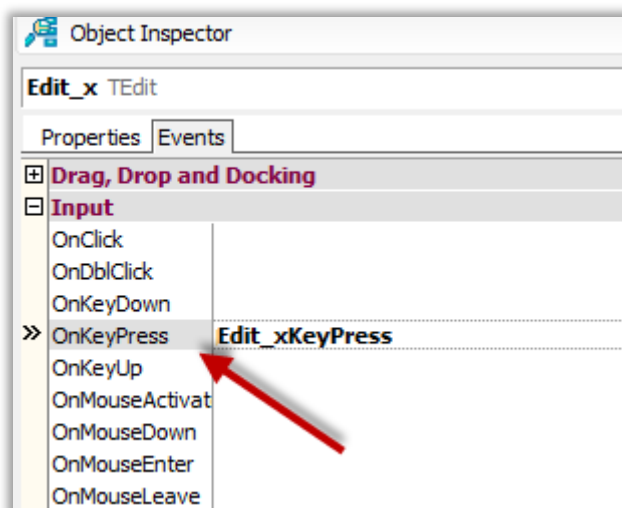
procedure TForm1.BitBtn1Click(Sender: TObject);
var
  x,y,z,a,b: real;
begin
  x:=StrToFloat(Edit_x.Text);
  y:=StrToFloat(Edit_y.Text);
  z:=StrToFloat(Edit_z.Text);
  a:=(1+y)*(x+y/(x*x+4))/(sqr(exp(x))+1/(x*x+4));
  b:=sqr(cos(arctan(1/z)));
  Edit4.Text:=FloatToStrF(a,ffFixed,6,2);
  Edit5.Text:=FormatFloat('0.##',b);
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  Close;
end;

end.
```

Здесь для вывода значения **b** используется не функция **FloatToStr**, а другая функция **FormatFloat**, которая округляет выводимое число до нескольких (в данном случае – двух) знаков после запятой; причём если слева от разделителя – значащая цифра, она и выводится, иначе выводится 0, если справа от разделителя – значащая цифра, она и выводится, иначе не выводится ничего.

Для того, чтобы пользователь мог вводить в окошках ввода только цифры и символ точки (или запятой, в зависимости от того какой десятичный разделитель используется в операционной системе), в обработчик события **OnKeyPress** объектов **Edit_x**, **Edit_y**, **Edit_z** вставляем оператор, подменяющий все символы, кроме цифр и запятой, нулевым символом, который не заносится в текст окна ввода:



```

procedure TForm1.Edit_xKeyPress(Sender: TObject; var Key: Char);
begin
  if Key='.' then Key:=',';
  if not (Key in ['0'..'9',',',',',#08]) then Key:=#0;
end;

procedure TForm1.Edit_yKeyPress(Sender: TObject; var Key: Char);
begin
  if Key='.' then Key:=',';
  if not (Key in ['0'..'9',',',',',#08]) then Key:=#0;
end;

procedure TForm1.Edit_zKeyPress(Sender: TObject; var Key: Char);
begin
  if Key='.' then Key:=',';
  if not (Key in ['0'..'9',',',',',#08]) then Key:=#0;
end;

```


И, наконец, задаём обработчик события нажатия кнопки завершения работы программы:

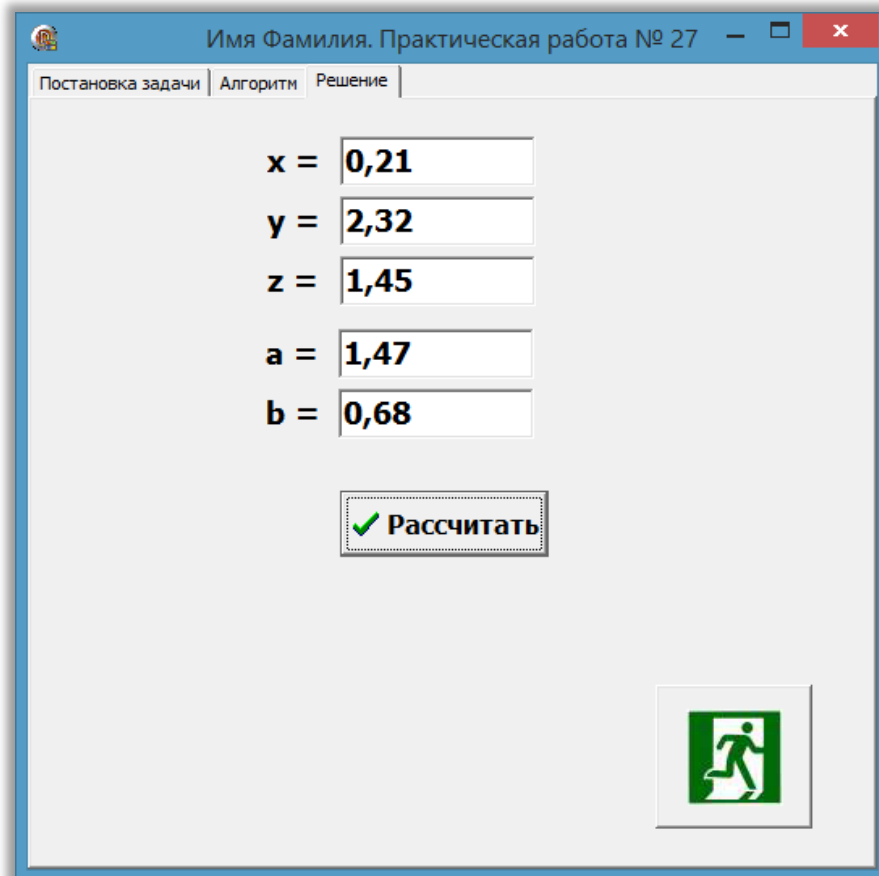
```

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  88   Close;
end;

```

6. Сохраните проект, запустите и протестируйте его.

- *Имя генерируемого компилятором исполняемого файла совпадает с именем проекта.*
 Для запуска программы на выполнение нажмите командную кнопку «Run»  или F9.



7. Покажите работу и отчёт преподавателю.