

Практическая работа № 28

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.

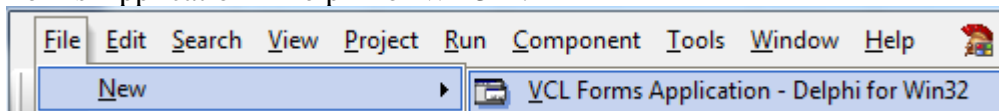
Литература:

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.

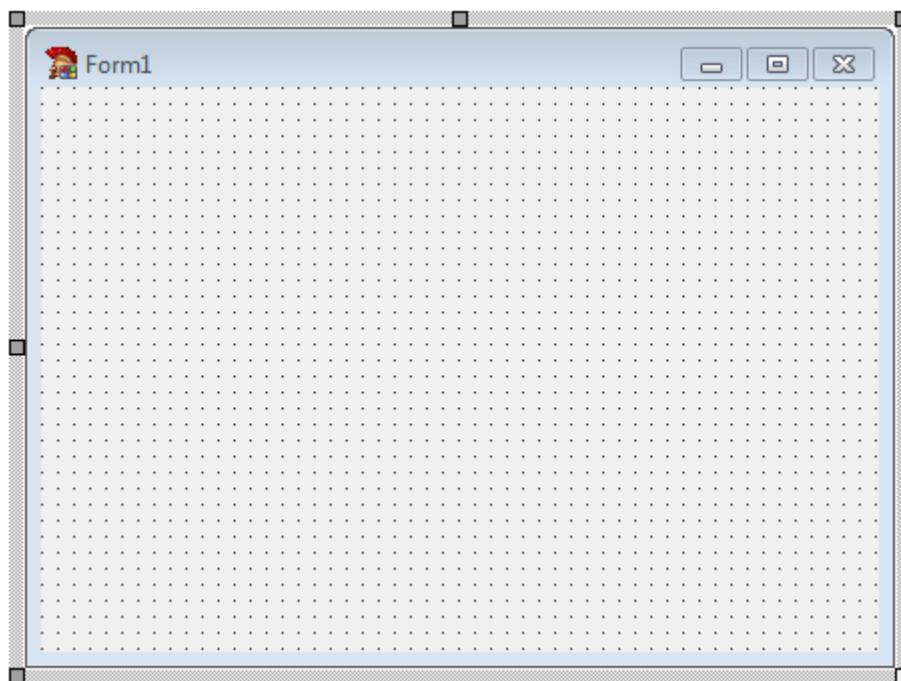
Содержание отчёта:

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

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

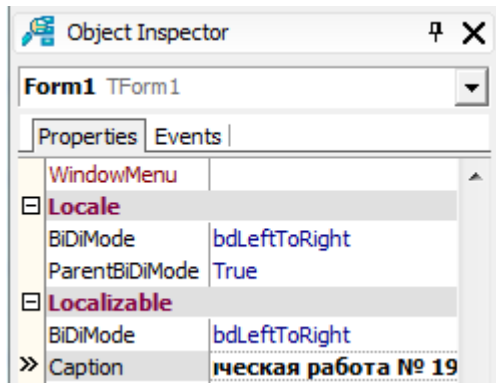


Появится окно формы, на которой размещаются все необходимые компоненты.



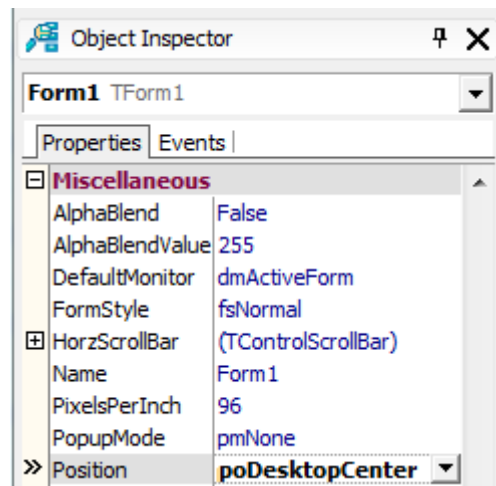
Окно формы будущей программы

В заголовок главной формы поместите текст: «<Имя Фамилия>. Практическая работа № 28». Для этого при выделенной главной форме в Инспекторе объектов на странице «Свойства» (Properties) во вкладке Action свойству Caption (Заголовок) присвойте значение «Практическая работа № 20»:



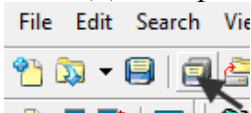
Измените при необходимости размер формы.

Свойство «Position» («Позиция») измените так, чтобы ваша программа располагалась в центре экрана:



Проверьте, что на главной панели Delphi активен зелёный треугольник . Сохраните проект под именем **Project28**.

Для сохранения нажмите кнопку «Save all» или **Shift+Ctrl+S**:



, сохраните модуль (файл **Unit1.pas**) и проект (файл **Project19.bdsproj**), создав папку **Практическая работа 28** в вашей папке (например, **Фамилия Имя**) в папке **Мои документы**.

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

Варианты задания:

1. Найти натяжение T каната, перекинутого через блок, если на концах каната подвешены грузы с массами m_1 и m_2 . Зависимость T от m_1 и m_2 представлена в виде:

$$T = 2 m_1 m_2 g / (m_1 + m_2)$$
, где g – ускорение силы тяжести.
 Задачу решить при $m_1 = 100$ кг, $m_2 = 55,5$ кг, $g = 9,81$ м/с². Вывести на экран результат вычисления с пояснительной надписью «Натяжение каната».
2. Фактический объём выполненных строительно-монтажных работ составил 5432,8 тыс. грн., при этом плановое задание оказалось перевыполненным на 1,5 %. Вычислите объём планового задания.
3. Цена товара - 850 грн. В неё входит НДС – 20 % от стоимости товара. Определите стоимость товара без НДС.

4. Определить момент инерции J однородного цилиндра с радиусом R и массой m относительно оси вращения, если $m = 5,23$ кг, $R = 0,2$ м.
Момент инерции однородного цилиндра относительно оси вращения выражается зависимостью: $J = 5 m R^2$. Исходные данные и результат вычисления вывести на печать.
5. Определить радиус однородного шара, если его масса $m = 5,23$ кг, а момент инерции $J = 1,4 \cdot 10^{-2}$ кг·м².
Момент инерции J однородного шара радиусом R относительно оси вращения выражается зависимостью: $J = 2/5 m R^2$. Исходные данные и результат вычисления вывести на печать.
6. Определить внутренний и внешний радиусы полого цилиндра, если его масса $m = 5,23$ кг, а момент инерции $J = 1,4 \cdot 10^{-2}$ кг·м².
Момент инерции J для полого цилиндра с радиусами R_1 и R_2 относительно оси вращения выражается зависимостью: $J = 1/2 m (R_1^2 + R_2^2)$, считать $R_2 = 2R_1$.
Исходные данные и результат вычисления вывести на печать.
7. Найти $x = lg y$, если $y = at^2 + bt + c$; $a = 1,04$; $b = 3,785$; $c = 12,76$; $t = 100,21$. Исходные данные и результат вычисления вывести на печать.
8. Пористость грунта определяют по формуле:

$$n = \left[1 - \frac{\gamma_{об}}{\gamma_{вд}(1 + 0,1W)} \right] \cdot 100,$$

где $\gamma_{об}$ – объёмная масса грунта, $\gamma_{вд}$ – плотность грунта, W – влажность грунта в процентах.

Найти пористость грунта при следующих данных лабораторного анализа: $\gamma_{об} = 1,95$ кг/м³; $\gamma_{вд} = 2,64$ кг/м³; $W = 22$ %.

9. Высота фундамента определяется по формуле $h = (b - b_0)/2 \cdot tg \alpha$, где b, b_0 – параметры ширины фундамента, α - угол, зависящий от соотношения h, b, b_0 . Решить задачу при $b = 180$ см, $b_0 = 64$ см, $\alpha = 30^\circ$. Результат вычисления вывести на печать, указав размерность.
10. При настилке пола из керамических квадратных плиток, укладываемых по диагонали, число треугольных плиток, укладываемых у фризowego ряда, определяется по формуле $n = L / (1,41a + 2b)$, где L – длина стороны фриза; a – длина стороны квадратной плитки; b – ширина шва между плитками. Решить задачу при $L = 5230$ мм, $a = 50$ мм, $b = 1,5$ мм. Учесть, что n – целое число.
11. Используя теорему синусов (стороны треугольника пропорциональны синусам противоположных углов), найти стороны треугольника a, b, c , если два его угла соответственно равны 30° и 40° , а радиус окружности, описанной около этого треугольника, равен 5 см. Учесть, что в теореме синусов каждое из отношений равно диаметру описанной около треугольника окружности :



$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} = D$$

12. Найти радиус окружности, описанной около треугольника, если его стороны равны $5, 6$ и 7 см.

Примечание: использовать формулу $S = \frac{abc}{4R}$, формулу Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$,

где a, b, c - стороны треугольника, p – полупериметр: $p = \frac{1}{2}(a + b + c)$

13. Найти площадь треугольника, стороны которого равны $3, 4$ и 5 см.
Примечание: использовать формулу Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$, где a, b, c - стороны треугольника, p – полупериметр: $p = \frac{1}{2}(a + b + c)$

14. Найти гипотенузу прямоугольного треугольника, если его катеты равны 3 и 5 см. (Использовать теорему Пифагора).
15. В прямоугольном треугольнике катет равен другому катету, умноженному на тангенс угла, противолежащего первому катету: $a = b \operatorname{tg} \alpha$. Найти стороны прямоугольного треугольника, если один из катетов равен 6 см, а угол, ему противолежащий, равен 40° .
16. На плоскости находятся две точки А и В с координатами $A(x_1; y_1)$ и $B(x_2; y_2)$. Найти расстояние d между ними, если $d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$. Задачу решить для $x_1 = 5$ см; $y_1 = 3$ см; $x_2 = 9$ см; $y_2 = 6$ см.
17. Уравнение окружности с центром в точке $A_0(a, b)$ и радиусом R имеет вид $(x - a)^2 + (y - b)^2 = R^2$. Найти координату y точки В, если $a = 3$; $b = 2,5$; $R = 2,8$; $x = 5$ и $y < b$.
18. Объём усеченной пирамиды вычисляется по формуле: $V = \frac{1}{3} h(Q_1 + \sqrt{Q_1 Q_2} + Q_2)$ где h - высота усечённой пирамиды, Q_1 и Q_2 – площади оснований. Найти объём усечённой пирамиды, у которой $h = 5$ см; $Q_1 = 18$ см²; $Q_2 = 7$ см².

$$V = \frac{1}{3} \pi \cdot R^2 H$$

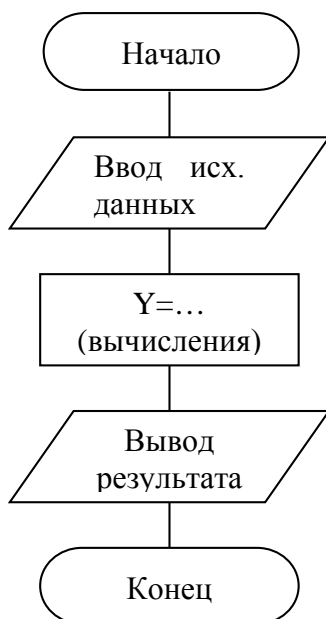
19. Объём конуса вычисляется по формуле: ,
где R - радиус основания конуса, H - высота.
Найти длину окружности, лежащей в основании конуса, если его объём $V = 25$ см³, а высота $H = 7$ см.
20. Объём шара равен $V = \frac{4}{3} \pi R^3$, где R - радиус шара. Найти радиус шара, если его объём равен $V = 30$ см³.
21. Боковое ребро правильной треугольной пирамиды равно a и составляет с плоскостью основания угол α . В эту пирамиду вписан цилиндр с квадратным осевым сечением. (Основание цилиндра лежит в плоскости основания пирамиды). Объём этого цилиндра можно найти по формуле:

$$V = \frac{\pi \cdot a^3 \sqrt{2} \sin^3 2\alpha}{128 \sin^3 \left(\frac{\pi}{4} + \alpha\right)}$$

Определить боковое ребро a пирамиды, если угол $\alpha = 1,2$ радиана, $V = 40$ см³.

22. Даны две точки $A(-1; 2; 5)$ и $B(7; -4; 8)$. Найти расстояние между А и В по формуле:
 $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$
Результат округлить до 0,1.

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



Рассмотрим построение программы для решения следующей задачи:

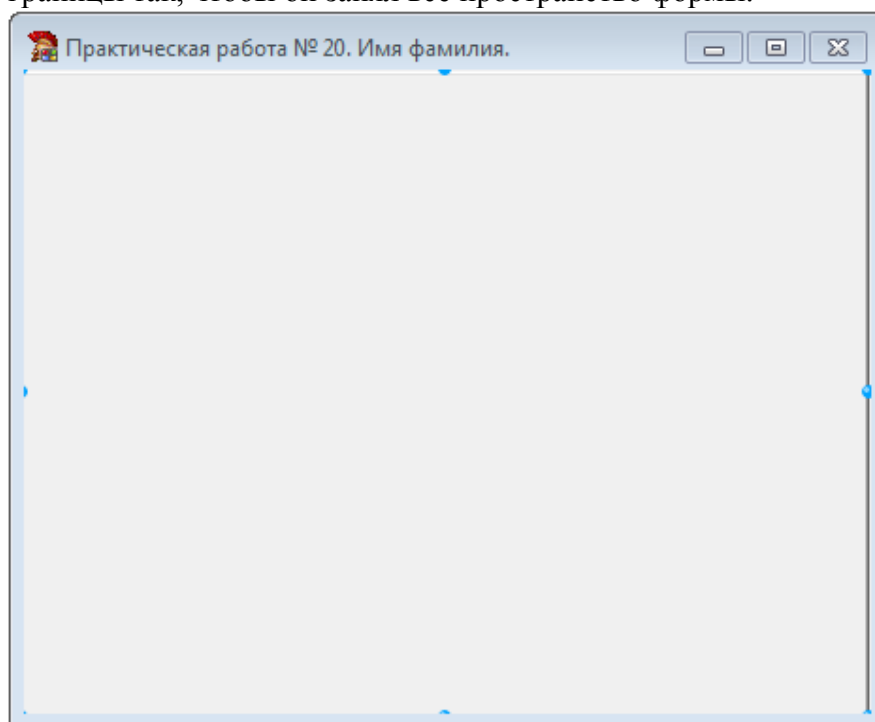
Зависимость пройденного телом пути S от времени t описывается формулой:

$$S = at + bt^2 + ct^3 .$$

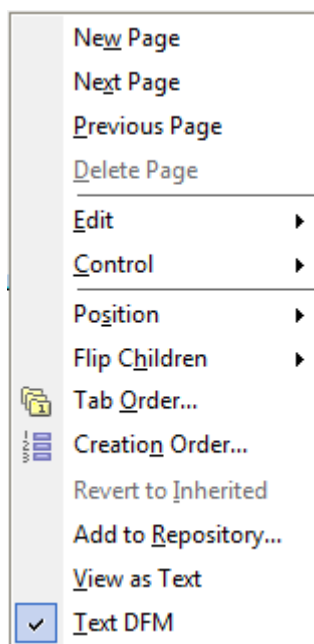
Найти пройденный телом путь при $a = 10,6$ м/с, $b = 3,04$ м/с², $c = 0,047$ м/с³, $t = 1,28$ с.

Вывести на экран: «Путь $S =$ <результат> м»

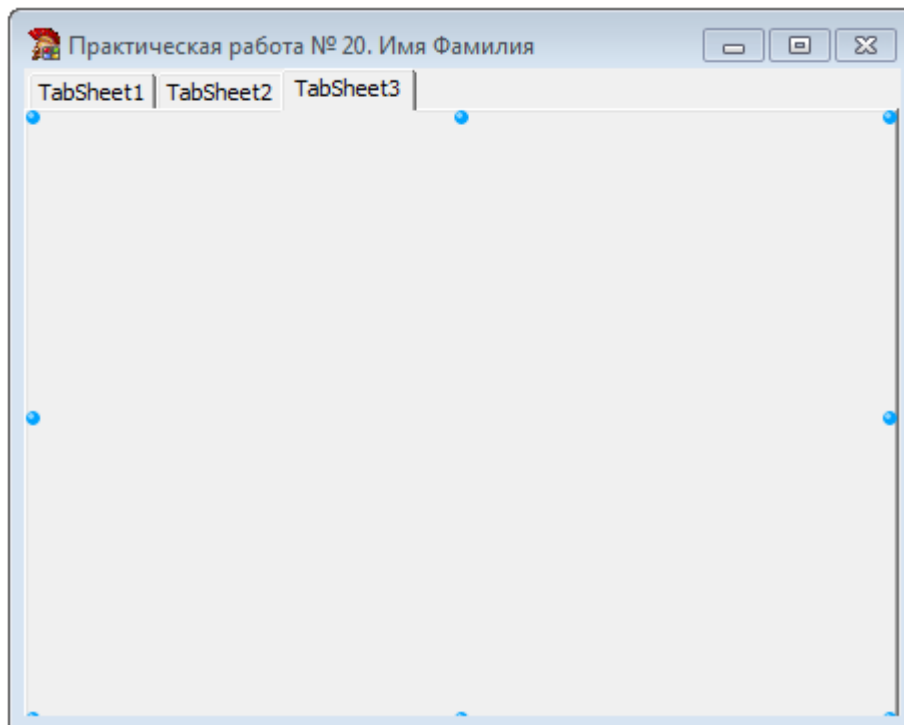
4. С целью экономии пространства окна приложения и размещения на одном и том же месте страниц разного содержания воспользуемся многостраничной панелью – компонентом **PageControl**. Перенесите компонент **PageControl** на форму (из библиотеки компонентов **Win32**) и растяните его границы так, чтобы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



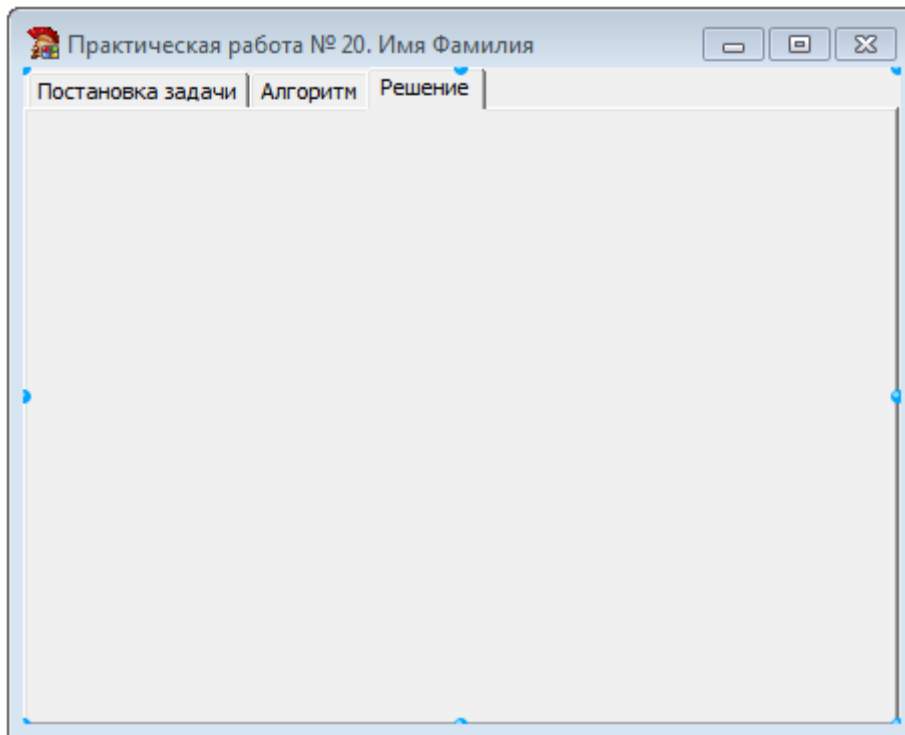
Щёлкните правой кнопкой мыши по панели. Во всплывшем контекстном меню вы можете видеть команды: **New Page** – создать новую страницу, **Next Page** – переключиться на следующую страницу, **Previous Page** – переключиться на предыдущую страницу. Каждая создаваемая страница является объектом типа **TTabSheet**.



Выберем пункт **New Page** – создать главную страницу. Появляется первая страница **TabSheet1**. Аналогично создаём вторую страницу **TabSheet2** и третью – **TabSheet3**.

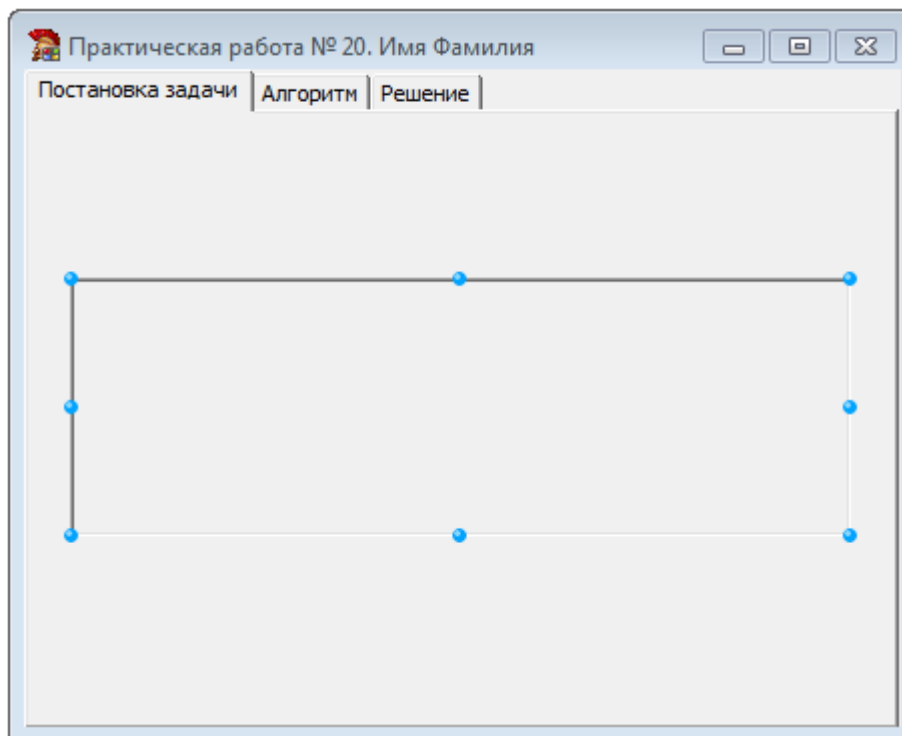


В Инспекторе объектов присвоим свойству **Caption** объекта **TabSheet1** значение «Постановка задачи», **TabSheet2** – «Алгоритм», а **TabSheet3** – «Решение». Для изменения свойств страницы необходимо её выделить, для этого - сначала щёлкнуть ЛКМ на закладке страницы, а потом – посередине её.

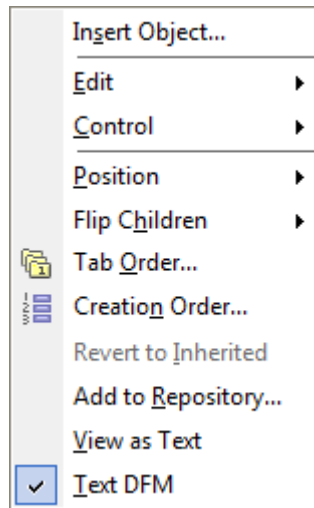


5. Формулировка постановки задачи может содержать специальные математические символы, которые не могут быть введены с клавиатуры, например, в виде метки **Label**. Поэтому воспользуемся контейнером **OLE – OLEContainer** (библиотека **System**) и создадим область клиента для объекта **OLE** (OLE (англ. *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 технологию в *ActiveX*).

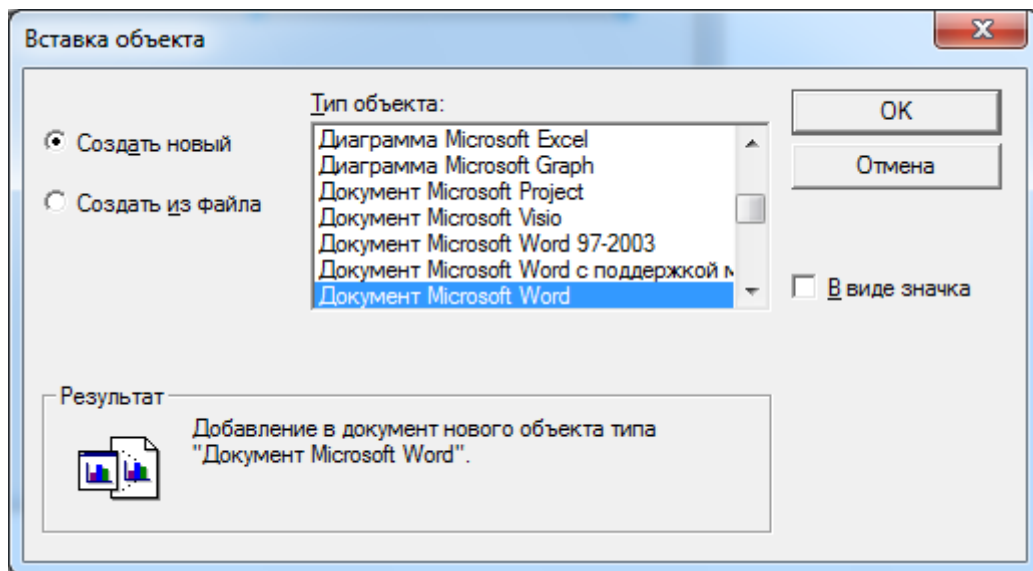
Разместим контейнер OLE на странице TabSheet1 «Постановка задачи».



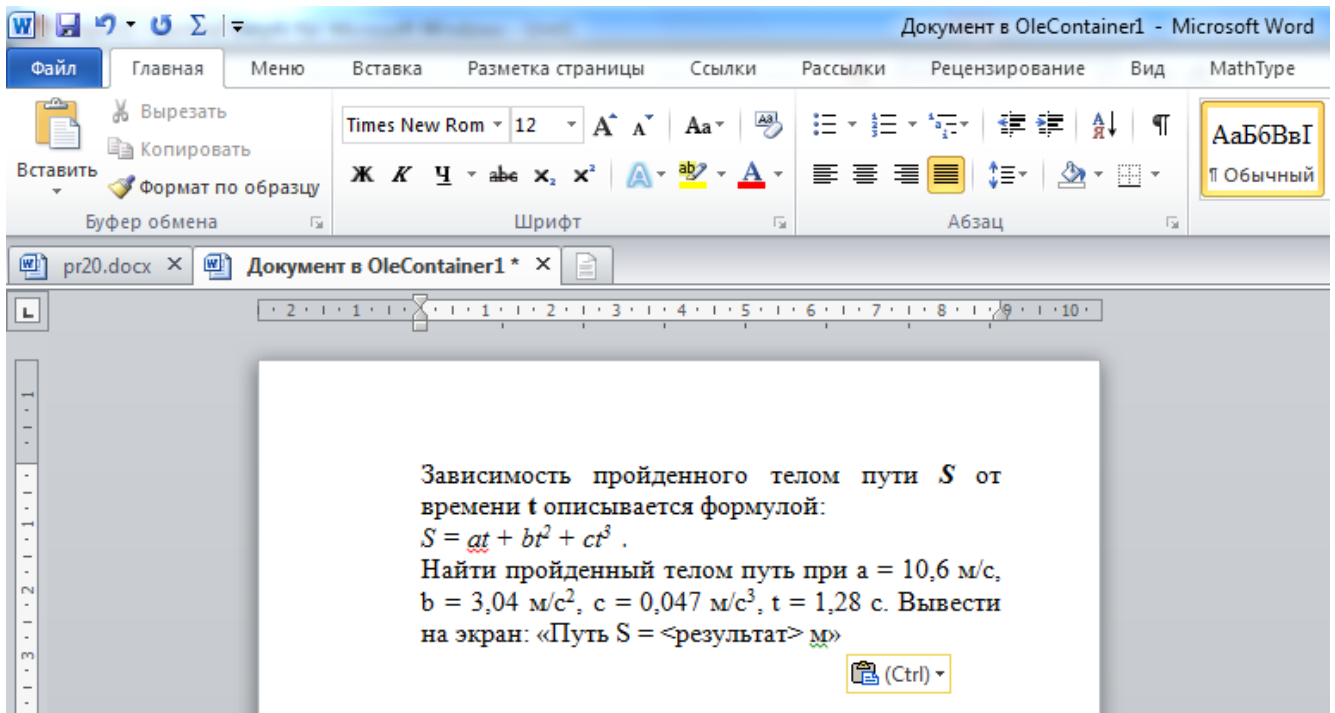
Щёлкнув ПКМ, вызовем контекстное меню



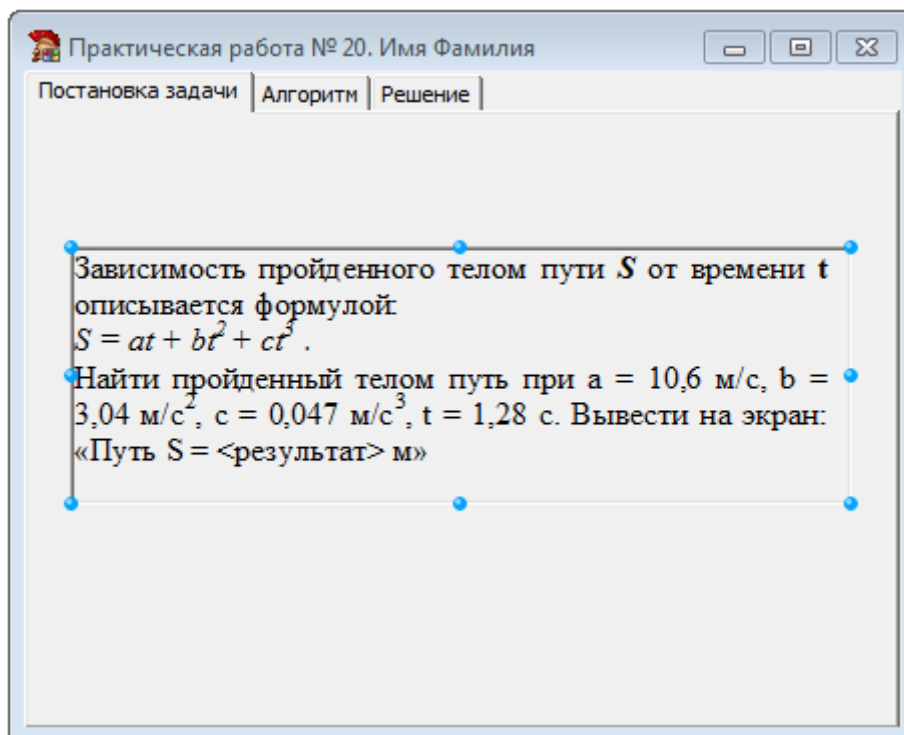
и выберем первый пункт **Insert Object...**, в котором выберем вставку объекта – «Документ Microsoft Word».



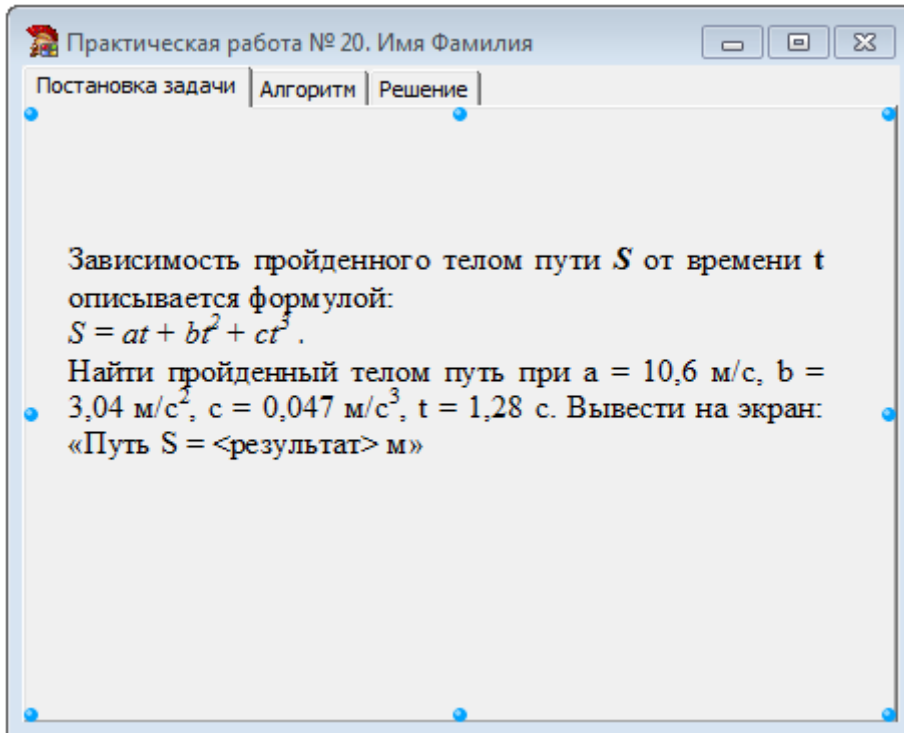
В открывшемся новом документе вводим с клавиатуры или копируем готовый текст формулировки постановки задачи, затем закрываем документ.



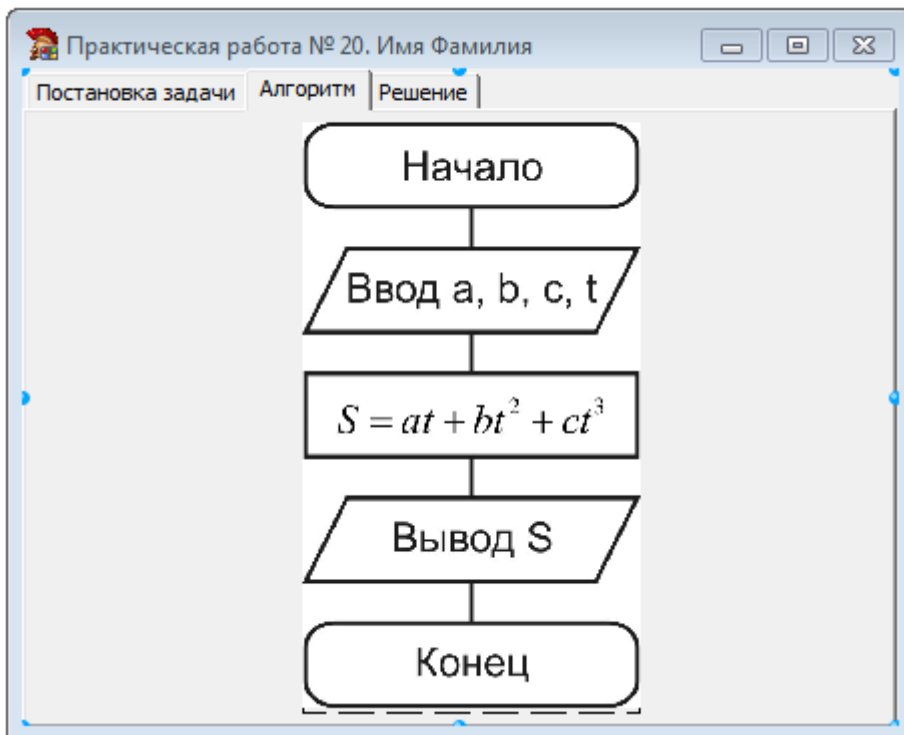
Текст размещается в контейнере OLE.



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

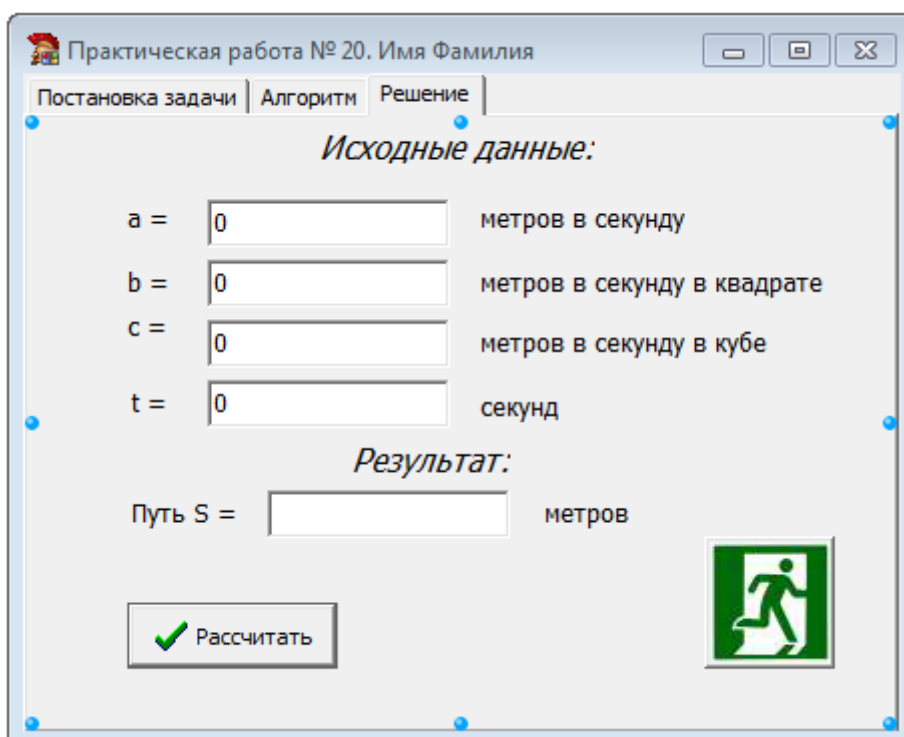


6. На странице **TabSheet2** «Алгоритм» располагаем блок-схему алгоритма в виде файла-картинки (растрового изображения), воспользовавшись холстом **TImage** из библиотеки **Additional**.



7. На странице **TabSheet2** «Решение» располагаем компоненты **Label** и **Edit** для ввода исходных данных и вывода результатов. Имя объектов **Edit** (свойство **Name**) изменяем на **Edit_a** (компонент **Edit** для ввода коэффициента a), **Edit_b** (компонент **Edit** для ввода коэффициента b), **Edit_c** (компонент **Edit** для ввода коэффициента c), **Edit_t** (компонент **Edit** для ввода времени t), **Edit_S** (компонент **Edit** для вывода пути S); в качестве текста записываем нули (свойство **Text**). Свойству **ReadOnly** объекта **Edit_S** 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**).



8. 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

$n\sqrt{x}$	$\exp(1/n*\ln(x))$	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);

Операторы деления

$7 \div 2$

$6 \div 3 \leftarrow \text{div}$ — целочисленное деление, т.е. дробная часть отбрасывается. ($7 : 2 = 3$)

$1 \leftarrow \text{mod}$ — 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;
50  a,b,c,t,S:real;

implementation

{$R *.dfm}

63  procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  a:=StrToFloat(Edit_a.Text);
  b:=StrToFloat(Edit_b.Text);
  c:=StrToFloat(Edit_c.Text);
  t:=StrToFloat(Edit_t.Text);
  S:=a*t+b*power(t,2)+c*power(t,3);
  Edit_S.Text:=FormatFloat('0.###',S);
end;

```

Здесь для вывода значения пути S используется не функция **FloatToStr**, а другая функция **FormatFloat**, которая округляет выводимое число до нескольких (в данном случае – трёх) 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_a**, **Edit_b**, **Edit_c**, **Edit_S** вставляем оператор, подменяющий все символы, кроме цифр и точки (запятой), нулевым символом, который не занесётся в текст окна ввода:

```

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

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

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

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


```

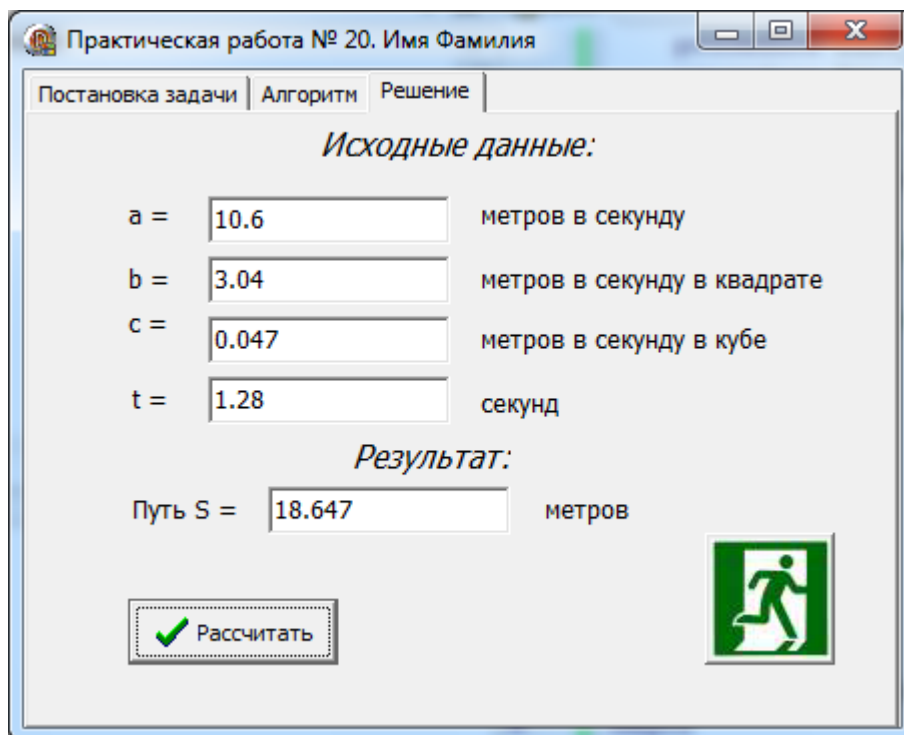
9. И, наконец, задаём обработчик события нажатия кнопки завершения работы программы:

```

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

```

10. 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**.



11. Покажите работу и отчёт преподавателю.