

Узбекское Агентство Информатизации и Связи  
Ташкентский Университет Информационных Технологий

Кафедра: «ПОИТ»

Курсовая работа

По предмету:  
«Объектно-ориентированные языки программирования»  
На тему:  
«Классы (тема № 1). Номер варианта 1.17».

Выполнила: Елибаева Э.С

Ст. гр. 252-10

Приняла:

Ташкент 2013

«Утверждаю»  
Заведующий кафедрой

«\_\_» \_\_\_\_\_ 2013г.

Задание  
на курсовую работу

**По дисциплине:** Объектно-ориентированные языки программирования.

**Студентка:**

**Руководитель**

**Защита курсовой работы:** \_\_\_\_\_

**Тема:** Классы (тема № 1). Номер варианта 1.17

**Задание:** Создать класс для трёхмерной фигуры (в соответствии с вариантом)

Реализовать в классе методы вычисления поверхности и объёма, а также рисование и вращение. Применить указанные методы в программе.

Вариант 1.17 усеченная пирамида, основание равнобедренный треугольник

Подпись преподавателя:\_\_\_\_\_

Подпись студента:\_\_\_\_\_

Дата выдачи задания:

## **Аннотация**

Данная курсовая работа по предмету: «Объектно-ориентированные языки программирования» на тему: «Классы (тема № 1). Номер варианта 1.17». В данной курсовой работе рассматривается реализация классов для графики, а также исследование вызовов конструкторов и методов класса. Рассчитываются поверхность и объём пирамиды, а также рисование и вращение пирамиды.

Объём работы: 18 листов формата А4.

## **Содержание**

1. Цель.....	5
2. Постановка задачи. ....	5
3. Вариант задания 1.17 .....	5
4. Реализация задания.....	5
4.1. Теоретическая часть.....	5
4.1.1. Общие сведения о теле «пирамида» и равнобедренном треугольнике.....	5
4.1.2. Преобразования на плоскости и в пространстве.....	7
4.1.3. Проецирование пространственных фигур и тел на плоскость.....	8
4.2. Практическая (программная) реализация.....	10
4.2.1. Описание созданных классов .....	10
4.2.2. Описание использованных в работе классов языка Delphi.....	13
4.3. Внешний вид ПС.....	14
4.3.1. Описание управляющих элементов.....	14
5.	
Заключение.....	15
6.	
Приложения.....	16
6.1. Полный листинг программы.....	16
6.2. Список использованной литературы.....	23

## **1. Цель**

Реализация классов для графики. Выполнить исследование вызовов конструкторов и методов класса.

## **2. Постановка задачи**

Создать класс для трёхмерной фигуры (в соответствии с вариантом)

Реализовать в классе методы вычисления поверхности и объёма, а также рисование и вращение. Применить указанные методы в программе.

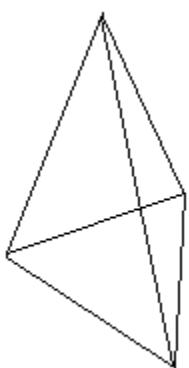
## **3. Вариант задания 1.17**

Тело: усеченная пирамида, основание равнобедренный треугольник

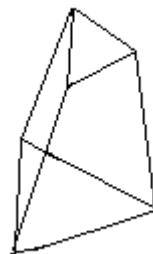
## **4. Реализация задания**

### **4.1. Теоретическая часть**

#### **4.1.1. Общие сведения о теле «пирамида» и равнобедренном треугольнике**



Пирамида



Усеченная

Пирамида – многогранник, основание которого – многоугольник, а остальные грани – треугольники, имеющие общую вершину.

По числу углов основания различают пирамиды треугольные, четырехугольные и т.д.

Элементы пирамиды:

- Апофема – высота боковой грани правильной пирамиды
- Боковые грани – треугольники, сходящиеся в вершине пирамиды
- Боковые ребра – общие стороны боковых граней
- Вершина пирамиды – точка, не лежащая в плоскости основания
- Высота- отрезок перпендикуляра, проведенного через вершину пирамиды к плоскости ее основания
- Диагональное сечение пирамиды – сечение пирамиды, проходящее через вершину и диагональ основания
- Основание – многоугольник, которому не принадлежит вершина пирамиды

Если все боковые ребра равны, то:

- Около основания пирамиды можно описать окружность, причем высота пирамиды проецируется в центр окружности
- Боковые ребра образуют с плоскостью основания равные углы

Если боковые грани наклонены к плоскости основания под одним углом, то:

- Высоты боковых граней равны
- Площадь боковой поверхности равна половине произведения периметра основания на высоту боковой грани

Формулы:

Объем пирамиды  $V = \frac{1}{3} S \cdot h$ , где  $S$  – площадь основания и  $h$  - высота

Боковая поверхность – это сумма площадей боковых граней  $S_b = \sum_i S_i$

Полная поверхность – это сумма боковой поверхности и площади основания

$$S_p = S_b + S_o$$

Усеченной пирамидой называют многогранник, заключенный между пирамидой и секущей плоскостью, параллельный ее основанию

Равнобедренный треугольник – треугольник, в котором две стороны равны между собой.

Свойства равнобедренного треугольника:

- Углы, противолежащие равным сторонам равны между собой. Также равны биссектрисы, медианы и высоты, проведенные из этих углов
- биссектриса, медиана и высота, проведенные к основанию, совпадают между собой. Центры описанной и вписанной окружностей лежат на этой линии

- углы, противолежащие равным сторонам, всегда острые

Периметр равнобедренного треугольника может быть вычислен по формулам:

$$P = 2a + b$$

$$P = 2R(2 \sin \alpha + \sin \beta)$$

Площадь равнобедренного треугольника может быть вычислена по формулам:

$$S = \frac{1}{2}a^2 \sin \beta = \frac{1}{2}ab \sin \alpha$$

$$S = \frac{1}{2}b \sqrt{\left(a + \frac{1}{2}b\right) \left(a - \frac{1}{2}b\right)}$$

#### **4.1.2. Преобразования на плоскости и в пространстве**

Каждое преобразование более низкого уровня является одновременно и преобразованием более высокого.

1) преобразования на плоскости

1.1) перенос на вектор  $n$  ( $a, b$ ):

$$M(n) = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

$P' = M(n)$ ;  $P$  и  $P'$  – однородные координаты.

1.2) поворот на угол  $\varphi$  против часовой стрелки вокруг начала координат:

$$M(\varphi) = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1.3) масштабирование относительно начала координат:

$$M(e) = \begin{bmatrix} e1 & 0 & 0 \\ 0 & -e2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2) преобразования в пространстве

2.1) перенос на вектор  $n$  ( $a, b, c$ ):

$$M(n) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2) вращение относительно оси OZ:

$$M(z, \varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3) вращение относительно оси OX:

$$M(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4) вращение относительно оси OY:

$$M(y, \psi) = \begin{bmatrix} 0 & \cos \psi & -\sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5) масштабирование относительно начала координат

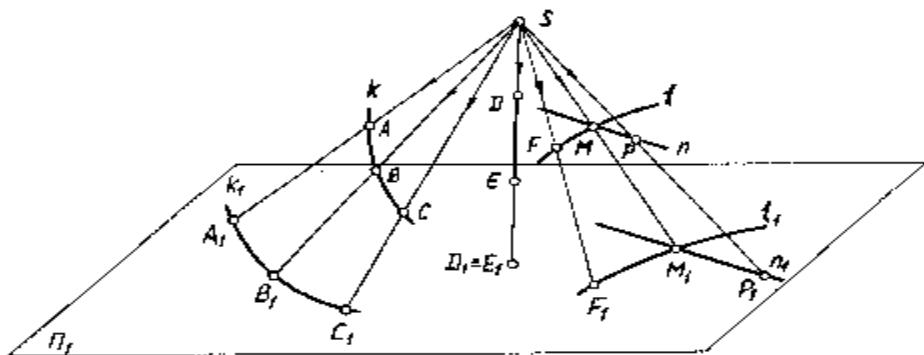
$$M(n) = \begin{bmatrix} e1 & 0 & 0 & 0 \\ 0 & e2 & 0 & 0 \\ 0 & 0 & e3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4.1.3. Проецирование пространственных фигур и тел на плоскость

##### ЦЕНТРАЛЬНОЕ ПРОЕЦИРОВАНИЕ

Основными видами проецирования являются центральное и параллельное. Центральное проецирование представляет собой общий случай проецирования геометрических образов из некоторого центра на плоскость.

Пусть задана плоскость  $\Pi_1$  и кривая линия  $k$  с точками  $A, B, C$  (рис. 1)



Возьмем некоторую точку  $S$ , не лежащую в плоскости  $\Pi_1$ . Через точку  $S$  и точки  $A, B, C$  кривой  $k$  проведем прямые до пересечения с плоскостью  $\Pi_1$  в точках  $A_1, B_1, C_1$ . Проведя таким образом через  $S$  и каждую точку кривой  $k$  прямые, получим в плоскости  $\Pi_1$  изображение  $k_1$  кривой  $k$ .

В соответствии с описанным построением введем следующие понятия:

$S$  - центр проекций;  $\Pi_1$  - плоскость проекций; кривая  $k$  с точками  $A, B, C$  - объект проецирования;  $SA, SB, SC$  - проецирующие лучи;  $A_1, B_1, C_1$  - центральные проекции точек  $A, B, C$ ;  $k_1$  - центральная проекция кривой  $k$ .

Рассматривая каждую пространственную фигуру как совокупность точек, можно сказать, что проекция фигуры представляет собой множество проекций ее точек.

### **Свойства центрального проецирования:**

1. Любая точка (кроме S) проецируется на плоскость проекций в единственную точку (рис.1).
2. Каждой точке (A, B, C, D,...), принадлежащей какой-либо линии (кривой или прямой), соответствует проекция (A1, B1, C1, D1, ...) этой точки на проекции данной линии (рис.1).
3. Кривая в общем случае проецируется в кривую, а прямая - в прямую. Если прямая совпадает с проецирующим лучом, например DE (рис.1), то она проецируется в точку D1 ° E1. Плоскость, проходящая через центр проекций, проецируется в прямую и называется проецирующей. Кривая, все точки которой принадлежат проецирующей плоскости, проецируется в прямую.
4. Точка пересечения линий проецируется в точку пересечения проекций этих линий (рис.1).

Центральное проецирование обладает большой наглядностью и применяется в строительном черчении, в архитектуре, в живописи и т.п. Недостатком центрального проецирования является сложность построения изображения предмета и определения истинных размеров. Поэтому оно имеет ограниченное применение в техническом черчении.

## **4.2. Практическая (программная) реализация**

Для реализации поставленной задачи выбран объектно-ориентированный язык программирования Delphi 7. Использованная среда разработки – Borland Delphi 7.

### **4.2.1. Описание созданных классов**

В ходе работы был создан класс, с помощью объектов которого определяется пространственное тело, и выполняются операции над ним.

класс TPiram (пирамида) – описывает пирамиду, заданную несколькими объектами класса Point и определяет методы для ее рисования, вращения и математических операций.

```
procedure TPiram.proekt(alfa:real);  
begin  
    form1.image1.Canvas.Rectangle(0,0,form1.image1.Width,form1.image1.Height );  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*x*cos(alfa+pi*2/3)),round(form1.image1.Height div  
2+x*sin(alfa+pi*2/3))+100);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*x*cos(alfa+pi*4/3)),round(form1.image1.Height div  
2+x*sin(alfa+pi*4/3))+100);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);  
  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa+pi*2/3)),round(form1.image1.Height div  
2+y*sin(alfa+pi*2/3))+100-h*2);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa+pi*4/3)),round(form1.image1.Height div  
2+y*sin(alfa+pi*4/3))+100-h*2);  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);  
  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);
```

```
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);  
  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*x*cos(alfa+pi*2/3)),round(form1.image1.Height div  
2+x*sin(alfa+pi*2/3))+100);  
  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa+pi*2/3)),round(form1.image1.Height div  
2+y*sin(alfa+pi*2/3))+100-h*2);  
  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*x*cos(alfa+pi*4/3)),round(form1.image1.Height div  
2+x*sin(alfa+pi*4/3))+100);  
  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa+pi*4/3)),round(form1.image1.Height div  
2+y*sin(alfa+pi*4/3))+100-h*2);  
  
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div  
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);  
  
    form1.image1.Canvas.LineTo(round(form1.image1.Width div  
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);  
  
end;
```

#### **4.2.2. Описание использованных в работе процедур языка Delphi**

##### **1. Ввод пирамиды**

```
procedure TPiram.vvod(a1,b1,h1:integer);
begin
  a:=a1; b:=b1; h:=h1;
  x:=round(2/3*a*sin(pi/3));
  y:=round(2/3*b*sin(pi/3));
end;
```

##### **2. Вращение пирамиды**

```
procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
  alfa:=x*pi / 400;
  piramida.proekt(alfa);
end;
```

##### **3. Создание пирамиды**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  a2:=strToInt(edit1.text);
  b2:=strToInt(edit2.text);
  h2:=strToInt(edit4.text);
  piramida.vvod(a2,b2,h2);
  piramida.per(p);
  piramida.pl(s);
  piramida.ob(v);
  edit5.Text:=IntToStr(p);
  edit6.Text:=IntToStr(s);
  edit7.Text:=IntToStr(v);
  piramida.proekt(alfa);
end;
```

##### **4. Изменение формы**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  a2:=strToInt(edit1.text);
  b2:=strToInt(edit2.text);
  h2:=strToInt(edit4.text);
  piramida:=TPiram.creat(a2,b2,h2);
  alfa:=pi/6;
  piramida.proekt(alfa);
end;
```

## 5. Удаление формы

```
procedure TForm1.FormDestroy(Sender: TObject);
begin
  piramida.Free;
end;
```

### 4.3. Внешний вид ПС



Интерфейс ПС состоит из одного окна (формы), где отображено пространство для изображения и управляющие элементы.

#### 4.3.1. Описание управляющих элементов

a = 100 см      b = 50 см      h = 100 см

1) Задание параметров пирамиды две стороны и высота

Новая пирамида задается с помощью кнопки «Создать».

Создать

2) Математические операции (вычисление площади боковой поверхности, площади полной поверхности и объема пирамиды).

$$P = \boxed{750} \text{ см}$$

$$S = \boxed{8665} \text{ см}^2$$

$$V = \boxed{252591} \text{ см}^3$$

Вычисляется автоматически после создания пирамиды.

### 3) Вращение пирамиды

Для вращения необходимо движение мыши в нужном направлении

## 5. Заключение

В данной работе отражены основы работы с пространственными преобразованиями и программная реализация их с помощью объектно-ориентированного метода средствами визуального программирования.

## 6. Приложения

### 6.1. Полный листинг программы

```
unit Unit11;

interface

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

type
  TPiram = class(TObject)
    a,b,h,x,y: integer;
    constructor creat(a1,b1,h1:integer);
    public
    procedure vvod(a1,b1,h1:integer);
    procedure per( var p: integer);
    procedure ob( var v:integer);
    procedure pl( var s:integer);
    procedure proekt(alfa:real);
    end;
  TForm1 = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Edit2: TEdit;
    Edit4: TEdit;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Button1: TButton;
    Label13: TLabel;
    Label14: TLabel;
```

```

Label15: TLabel;
Label16: TLabel;
Label17: TLabel;

procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
Form1: TForm1; piramida:TPiram;
alfa:real;
a2,b2,h2,u2,p,s,v: integer;
implementation

{$R *.dfm}

constructor TPiram.creat(a1,b1,h1:integer);
begin
inherited create;
a:=a1; b:=b1; h:=h1;
x:=round(2/3*a*sin(pi/3));
y:=round(2/3*b*sin(pi/3));
end;
procedure TPiram.vvod(a1,b1,h1:integer);
begin
a:=a1; b:=b1; h:=h1;
x:=round(2/3*a*sin(pi/3));
y:=round(2/3*b*sin(pi/3));
end;
procedure TPiram.per( var p: integer);
begin p:=3*(a+b+h);
end;
procedure TPiram.ob( var v:integer);
begin
if a=b then v:=round(h*a*a*sin(pi/3)/2)
else v:= round(a*h*a*a*sin(pi/3)/(6*(a-b))-(a*h/(a-b)-h)*b*b*sin(pi/3)/6);

```

```

end;
procedure TPiram.pl( var s:integer);
begin
if a=b then s:=round(2*a*a*sin(pi/3)/2+3*a*h)
else
s:=round(a*a*sin(pi/3)/2+b*b*sin(pi/3)/2+a*a*s in(pi/3)/(2*cos(arctan(a*h/(a-
b))*2/x))-b*b*sin(pi/3)/(2*cos(arctan(a*h/(a-b))*2/x)));
end;
procedure TPiram.proekt(alfa:real);
begin
form1.image1.Canvas.Rectangle(0,0,form1.image1.Width,form1.image1.Height
);
form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*x*cos(alfa+pi*2/3)),round(form1.image1.Height div
2+x*sin(alfa+pi*2/3))+100);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*x*cos(alfa+pi*4/3)),round(form1.image1.Height div
2+x*sin(alfa+pi*4/3))+100);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);

form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa+pi*2/3)),round(form1.image1.Height div
2+y*sin(alfa+pi*2/3))+100-h*2);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa+pi*4/3)),round(form1.image1.Height div
2+y*sin(alfa+pi*4/3))+100-h*2);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);

form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);
form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*x*cos(alfa+pi*2/3)),round(form1.image1.Height div
2+x*sin(alfa+pi*2/3))+100);
form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa+pi*2/3)),round(form1.image1.Height div
2+y*sin(alfa+pi*2/3))+100-h*2);

```

```

    form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*x*cos(alfa+pi*4/3)),round(form1.image1.Height div
2+x*sin(alfa+pi*4/3))+100);
    form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa+pi*4/3)),round(form1.image1.Height div
2+y*sin(alfa+pi*4/3))+100-h*2);
    form1.image1.Canvas.MoveTo(round(form1.image1.Width div
2+3*x*cos(alfa)),round(form1.image1.Height div 2+x*sin(alfa))+100);
    form1.image1.Canvas.LineTo(round(form1.image1.Width div
2+3*y*cos(alfa)),round(form1.image1.Height div 2+y*sin(alfa))+100-h*2);

end;

procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
alfa:=x*pi / 400;
piramida.proekt(alfa);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
a2:=strtoint(edit1.text);
b2:=strtoint(edit2.text);
h2:=strtoint(edit4.text);

piramida.vvod(a2,b2,h2);
piramida.per(p);
piramida.pl(s);
piramida.ob(v);
edit5.Text:=inttostr(p);
edit6.Text:=inttostr(s);
edit7.Text:=inttostr(v);
piramida.proekt(alfa);
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
a2:=strtoint(edit1.text);
b2:=strtoint(edit2.text);
h2:=strtoint(edit4.text);
piramida:=TPiram.creat(a2,b2,h2);
alfa:=pi/6;

```

```
piramida.proekt(alfa);  
end;
```

```
procedure TForm1.FormDestroy(Sender: TObject);  
begin  
  piramida.Free;  
end;  
  
end.
```

## **6.2. Список использованной литературы**

1. Delphi – «32 урока»
2. Беклемишев – «Курс аналитической геометрии»
3. Валентин Озеров «916 советов по Delphi»
4. Джон Матчо «Delphi»