Петропавловский гуманитарно-технический колледж

Контрольная работа

По предмету: «Основы объектно-ориентируемого программирования»

Вариант №7

Выполнил:

Студент з/о группа 59-ПР

Семыкин А.С.

Проверил:

Петропавловск 2008

**Содержание**

1. События клавиатуры

2. Форма. Свойства формы

3. Приложение, вычисляющее площадь треугольника

3.1 Программный код

3.2 Внешний вид приложения

**1.** **События клавиатуры**

Событие (Event) — это то, что происходит во время работы программы. В Delphi каждому событию присвоено имя. Например, щелчок кнопкой мыши - это событие OnClick, двойной щелчок мышью событие OnDblClick.

В табл. 1 приведены события клавиатуры.

Таблица 1 - События

|  |  |
| --- | --- |
| Событие | Происходит |
| OnKeyPress | При нажатии клавиши клавиатуры |
| OnKeyDown | При нажатии клавиши клавиатуры. События OnKeyDown и OnKeyPress — это чередующиеся, повторяющиеся события, которые происходят до тех пор, пока не будет отпущена удерживаемая клавиша (в этот момент происходит событие OnKeyUp) |
| OnKeyUp | При отпускании нажатой клавиши клавиатуры |

Реакцией на событие должно быть какое-либо действие. В Delphi реакция на событие реализуется как *процедура обработки события.* Таким образом, для того чтобы программа выполняла некоторую работу в ответ на действия пользователя, программист должен написать процедуру обработки соответствующего события. Следует обратить внимание на то, что значительную часть обработки событий берет на себя компонент. Поэтому программист должен разрабатывать процедуру обработки события только в том случае, если реакция на событие отличается от стандартной или не определена. Например, если по условию задачи ограничений на символы, вводимые в поле Edit, нет, то процедуру обработки события OnKeyPress писать не надо, т. к. во время работы программы будет использована стандартная (скрытая от программиста) процедура обработки этого события.

**2. Форма. Свойства формы**

Работа над новым проектом, так в Delphi называется разрабатываемое приложение, начинается с создания стартовой формы. Так на этапе разработки программы называют диалоговые окна.

Стартовая форма создается путем изменения значений свойств формы Form1 и добавления к форме необходимых компонентов (полей ввода и вывода текста, командных кнопок).

Свойства формы (табл. 2) определяют ее внешний вид: размер, положение на экране, текст заголовка, вид рамки.

Для просмотра и изменения значений свойств формы и ее компонентов используется окно Object Inspector. В верхней части окна Object Inspector указано имя объекта, значения свойств которого отображается в данный момент. В левой колонке вкладки Properties (Свойства) перечислены свойства объекта, а в правой — указаны их значения.

Таблица 2 - Свойства формы (объекта TForm)

|  |  |
| --- | --- |
| Свойство | Описание |
| Name | Имя формы. В программе имя формы используется для управления формой и доступа к компонентам формы. |
| Caption | Текст заголовка |
| Width | Ширина формы |
| Height | Высота формы |
| Top | Расстояние от верхней границы формы до верхней границы экрана |
| Left | Расстояние от левой границы формы до левой границы экрана |
| BorderStyle | Вид границы. Граница может быть обычной (bsSizeable), тонкой (bs Single) или отсутствовать (bsNone). Если у окна обычная граница, то во время работы программы пользователь может при помощи мыши изменить размер окна. Изменить размер окна с тонкой границей нельзя. Если граница отсутствует, то на экран во время работы программы будет выведено окно без заголовка. Положение и размер такого окна-во время работы программы изменить нельзя |
| BorderIcons | Кнопки управления окном. Значение свойства определяет, какие кнопки управления окном будут доступны пользователю во время работы программы. Значение свойства задается путем присвоения значений уточняющим свойствам biSystemMenu, biMinimaze, biMaximaze И biHelp. Свойство biSystemMenu определяет доступность кнопки Свернуть и кнопки системного меню, biMinimaze— кнопки Свернуть, biMaximaze— кнопки Развернуть, biHelp — кнопки вывода справочной информации |
| Icon | Значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню |
| Color | Цвет фона. Цвет можно задать, указав название цвета или привязку к текущей цветовой схеме операционной системы. Во втором случае цвет определяется текущей цветовой схемой, выбранным компонентом привязки и меняется при изменении цветовой схемы операционной системы. |
| Font | Шрифт. Шрифт, используемый "по умолчанию" компонентами, находящимися на поверхности формы. Изменение свойства Font формы приводит к автоматическому изменению свойства Font компонента, располагающегося на поверхности формы. То есть компоненты наследуют свойство Font от формы (имеется возможность запретить наследование). |

При создании формы в первую очередь следует изменить значение свойства caption (Заголовок). В нашем примере надо заменить текст Form1 на "скорость бега". Чтобы это сделать, нужно в окне Object Inspector щелкнуть мышью на строке Caption, в результате чего будет выделено текущее значение свойства, в строке появится курсор, и можно будет ввести текст "скорость бега" (рис. 1).

Аналогичным образом можно установить значения свойств Height и width, которые определяют высоту и ширину формы. Размер формы и ее положение на экране, а также размер других элементов управления и их положение на поверхности формы задают в пикселах, т. е. точках экрана. Свойствам Height и width надо присвоить значения 250 и 330 соответственно.

Форма — это обычное окно. Поэтому его размер можно изменить точно так же, как размер любого другого окна, т. е. захватом и перемещением (с помощью мыши) границы. По окончании перемещения границ автоматически изменятся значения свойств Height и width. Они будут соответствовать установленному размеру формы.



Рис. 1 - Установка значения свойства путем ввода значения

Положение диалогового окна на экране после запуска программы соответствует положению формы во время ее разработки, которое определяется значением свойств тор (отступ от верхней границы экрана) и Left (отступ от левой границы экрана). Значения этих свойств также можно задать путем перемещения окна формы при помощи мыши.

При выборе некоторых свойств, например, Borderstyle, справа от текущего значения свойства появляется значок раскрывающегося списка. Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. 2).

Некоторые свойства являются сложными, т. е. их значение определяется совокупностью значений других (уточняющих) свойств. Перед именами сложных свойств стоит значок "+", при щелчке на котором раскрывается список уточняющих свойств (рис. 3). Например, свойство BorderIcons определяет, какие кнопки управления окном будут доступны во время работы программы. Так, если свойству biMaximize присвоить значение False, то во время работы программы кнопки Развернуть в заголовке окна не будет.

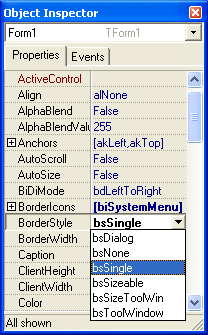


Рис. 2 - Установка значения свойства путем выбора из списка

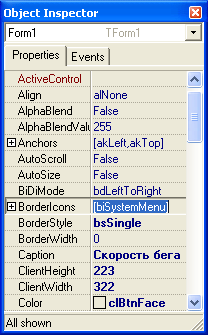


Рис. 3 - Раскрытый список вложенных свойств сложного свойства BorderIcons

Рядом со значениями некоторых свойств отображается командная кнопка с тремя точками. Это значит, что для задания значения свойства можно воспользоваться дополнительным диалоговым окном. Например, значение сложного свойства Font можно задать путем непосредственного ввода значений уточняющих свойств, а можно воспользоваться стандартным диалоговым окном выбора шрифта.

**3. Приложение, вычисляющее площадь треугольника**

Задание: Создать приложение вычисляющее площадь треугольника. Стороны треугольника вводятся с помощью компонента Edit.

**3.1 Программный код**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var

a,b,c,p,s:real;

begin

label4.Caption:=('');

if (((edit1.Text='')or (edit2.Text='')) or (edit3.Text='')) then

label4.Caption:=('Введите все 3 величины сторон треугольника')

else begin

a:=strtofloat(edit1.Text);

b:=strtofloat(edit2.Text);

c:=strtofloat(edit3.Text);

if (((a=0) or (b=0)) or (c=0)) then

label4.Caption:=('Треугольника с такими величинами сторон не существует')

else

begin

p:=(a+b+c)/2;

s:=sqrt(p\*(p-a)\*(p-b)\*(p-c));

if s=0 then

label4.Caption:=('Треугольника с такими величинами сторон не существует')

else

label4.Caption:=('Площадь треугольника = '+floattostr(s));

end;

end;

end;

end.

* 1. **Внешний вид приложения**

