

## Лабораторная работа 1

**Тема: Объекты (элементы) управления.**

**Цель работы:** ознакомиться с базовым набором управляющих элементов, их свойствами и возможностями при построении интерфейса приложения.

Элементы управления – это объекты, содержащиеся внутри объекта - Форма. Каждый тип элемента управления имеет свой собственный набор свойств (табл. 1), методов и событий, что делает его пригодным для определенной цели.

Таблица 1

### Основные свойства объектов управления

Свойство	Назначение
Name	Имя объекта
Caption	Заголовок
Visible	Видимость
BorderStyle	Стиль границ
FontBold	Полужирный шрифт
FontItalic	Курсив
FontName	Тип шрифта
FontSize	Размер шрифта
FontUnderline	Шрифт подчеркнутый
Enabled	Доступ
Left	Координата по горизонтали
Top	Координата по вертикали
Height	Высота объекта
Width	Ширина объекта
BackColor	Цвет фона
ForeColor	Цвет шрифта
BorderColor	Цвет границ
FillStyle	Стиль заполнения
MousePointer	Вид курсора при наведении на объект

Для размещения элемента на форме нужно дважды щелкнуть на выбранном значке на панели элементов или же «нарисовать» элемент на форме - для этого необходимо щелкнуть на нем мышью и на форме растянуть

соответствующий контур. После размещения элемента последовательно задаются его свойства в окне свойств.

При выборе имени (Name) объектов Visual Basic рекомендуется использовать следующую простую схему:

- начинать название с трехбуквенного префикса;
- использовать только буквы, цифры и знак подчеркивания ( \_ );
- использовать не более 40 символов.

### **Командная кнопка (Command Button)**



Кнопка используется для управления процессом: начало, окончание, пре-рывание и т. д.

Основными свойствами являются имя, название, положение, размеры, цвет, доступность, видимость.

Дополнительно можно указать следующие свойства:

**Default** – определяет, является данная кнопка активной по умолчанию или нет. Свойство имеет два значения: True и False. По умолчанию – False. Если установлено значение True, то фокус установлен на данной кнопке, то есть кнопка активна. Нажатие клавиши Enter перехватывается и передается этой кнопке.

**Cancel** - используется подобно свойству Default. Оно обеспечивает перехват нажатия клавиши Esc и вызов события Click для соответствующей кнопки.

**Appearance** – позволяет придать кнопке трехмерный вид.

**ToolTipText** – позволяет ввести текст, который отображается в подсказке, появляющейся при зависании указателя мыши на элементе управления.

### **Надпись (Label)**

**A** Надпись предназначена для отображения текста, который пользователь не может изменить с клавиатуры. Она обладает всеми перечисленными выше общими свойствами (табл. 1). Дополнительно можно указать еще ряд свойств:

**BorderStyle** - позволяет отображать текст с рамкой или без рамки.

**Font** - это свойство позволяет оформлять шрифты, используя все возможности Windows.

**AutoSize** - автоматическое приведение ширины объекта в соответствие с длиной текста. Если свойство AutoSize равно False и длина вводимого текста больше ширины надписи, то текст усекается. Если свойство AutoSize равно True, то размер объекта приводится в соответствие с длиной текста.

### **Текстовое поле (TextBox)**



Текстовое поле является основным элементом управления, предназначенным для ввода и вывода данных.

Основные свойства текстового поля совпадают с перечисленными выше (табл. 1), но есть и особенные свойства:

**Text** – аналог свойства **Caption**. Через это свойство осуществляется, как правило, ввод данных в программу и вывод данных на экран;

**Alignment** – выравнивание текста. Имеет три значения: 0- выравнивание по левому краю, 1 - по правому краю, 2 - по центру;

**ScrollBars** – вывод линеек прокрутки. Свойство **Scrollbars** позволяет устанавливать горизонтальную, вертикальную линейки прокрутки или обе.

**MultiLine** – определяет может ли поле содержать более одной строки текста. Обычно совмещается с установкой свойства **ScrollBars**.

В текстовом поле можно выделять и заменять текст. Это осуществляется программным путем с помощью свойств **SelStart**, **SelLength**, **SelText**.

**SelStart** – начало выделения.

**SelLength** – длина выделяемого текста.

**SelText** – замена текста.

Например:

`Text1.SelStart=2` ‘начать выделение со второго символа

`Text1.SelLength=6` ‘выделить шесть символов

`Text1.SelText = “Привет”` ‘заменить выделение на слово “Привет”

**MaxLength** – определяет максимальное число символов – По умолчанию – 0, что означает максимальное значение – 32 тысячи символов.

**PasswordChar** – позволяет заменять вводимые символы звездочками. Это свойство используется для ввода пароля.

**Locked** – запрещает пользователям изменять содержимое поля. Поле можно просматривать, но нельзя редактировать или удалять. Однако, значение поля можно изменить программным путем.

Объект **TextBox** может обрабатывать 23 события. Основные события текстового поля связаны с вводом данных: **Click**, **DbClick**, **KeyDown**, **KeyUp**, **GotFocus**, **LostFocus**, которые рассмотрены были ранее;

**Change** - изменение значения текстового поля.

При вводе данных с клавиатуры в активное текстовое поле программа не делает различий между буквами и цифрами: все вводится как текст. Поэтому для перевода текста в числа и обратно чисел в текст, используются функции преобразования символьных переменных:

**Val(C)** – преобразование текста в число ;

**Str(N)** – преобразование числа в текст или **Str\$(N)** - для преобразования в символьную переменную переменных типа Variant.

Например:

A = Val(Text1.Text) ‘перевод значения текстового поля в число.

Text2.Text = Str\$(a) ‘перевод числа в текст.

### Окно с рисунком (PictureBox)



Элемент управления PictureBox предназначен для отображения рисунков и других графических объектов.

К основным и наиболее часто используемым свойствам относятся:

**Align** - определяет положение PictureBox в форме: 0 - None, 1 - Top, 2 - Bottom, 3 - Left, 4 - Right. В зависимости от значения этого свойства объект будет закрепляться у одного из краев формы или будет сохранять положение, заданное разработчиком (None). Если элемент управления закрепляется у одного из краев формы, то его размер (ширина и высота) всегда будут устанавливаться в соответствии с размерами формы.

**Picture** - позволяет загружать графические объекты и сохранять их, содержит отображаемый графический объект.

Для загрузки графического объекта щелкните мышью по кнопке троеточие в поле свойства Picture – появится диалоговое окно Load Picture, которое позволяет осуществить поиск и загрузку нужного файла с диска. Например: C:\Windows\облака.bmp.

В процессе работы программы загрузку изображений можно осуществлять с помощью функции **LoadPicture**:

Объект.Picture = LoadPicture("спецификация\_файла")

Выгрузка рисунка в процессе работы осуществляется этой же функцией без указания имени файла:

Объект.Picture = LoadPicture()

### Флажки и переключатели



**Флажки (CheckBox)** – это элементы управления, которые можно отмечать «галочками», выбирая из ряда опций одну или несколько.



**Переключатели (OptionButton)** – это элементы управления, которые можно отмечать точкой, выбирая один элемент из группы.

Важнейшим событием для этих элементов является событие **Click**, а основным свойством – его значение **Value**. С помощью этого свойства можно

определить состояние объекта. Свойство Value может иметь три значения: 0 – не отмечен; 1- отмечен; 2 – отмечен, но недоступен. Последнее значение может быть установлено только программно.

### Списки и поля со списками

Имеется несколько типов полей со списками: простые списки  (ListBox), раскрывающиеся списки  (ComboBox) или поле со списком, а также элемент ImageCombo.

Списки позволяют выбирать значения из списка, вносить записи в процессе работы программы и на этапе разработки. Если данные не умещаются в окне, то автоматически появляется линейка прокрутки.

К основным и наиболее часто используемым свойствам относятся: Text, List, ListIndex, ListCount, Columns, Sorted, ItemData, MultiSelect.

**Text** – хранит значение выбранного элемента списка.

**List** – это свойство хранит все значения списка. Все записи в списке имеют индекс (как массивы), нумерация элементов списка начинается с нуля. Зная индекс элемента, можно выбрать его из списка. Синтаксис команды:

`<переменная>=lstBox.List(i)`

**ListIndex** - возвращает индекс элемента списка:

`< номер_элемента >= lstBox.ListIndex`

Можно комбинировать свойства List и ListIndex:

`<Элемент_списка>=lstBox.List(lstBox. ListIndex)`

Если в списке не выбран ни один элемент, то значение свойства ListIndex равно -1.

**ListCount** – сохраняет текущее значение числа элементов списка.

**Columns** – это свойство позволяет в процессе разработки отображать данные в несколько столбцов. Заполнение столбцов в этом случае осуществляется последовательно – сначала заполняется первый столбец, затем второй и т.д.

**Sorted** – определяет способ расположения элементов в списке. Если это свойство установлено в True, то все элементы будут сортироваться по алфавиту, даже если они добавлены с указанием индекса. Индекс последнего добавленного элемента имеет свойство **NewIndex**. Значение свойства сортировка устанавливается только на этапе разработки.

**MultiSelect** – это свойство позволяет выбирать одновременно несколько элементов. Данное свойство имеет три значения:

0 – множественный выбор невозможен;

1 – простой множественный вы-бор; 2 – расширенный выбор.

## Методы списков

**AddItem** - добавление элементов в список:

```
List1.AddItem <Элемент [ , Индекс ]>
```

```
List1.AddItem “Персональная выставка”,5
```

**RemoveItem** - удаление элементов из списка:

```
List1.RemoveItem.ИндексЭлемента
```

**ListBox Clear** - удаление всех элементов списка.

## Создание проекта

Создать приложение для ознакомления с возможностями элементов управления.

### Порядок выполнения

1. Создайте новый проект. Для этого в главном меню выполните команду **Файл→Новый проект (Ctrl+N)**.
2. Сохраните проект под именем «Объекты\_управления».
3. Установите значения свойств элементов согласно данным таблицы 2. Приведите свою форму в соответствие с образцом (рис. 1).

Таблица 2

Объект	Свойство	Значение
1	2	3
Форма	Name	FrmLab3_2
	Caption	Свойства списка и комбинированного списка
Рамка	Caption	Выбранная запись и ее номер
Метки	Font	Размер 10
	Name	Lbl1
	Caption	(отсутствует)
	Name	Lbl2
	Caption	(отсутствует)

Список	Name	LstFruct
	List	Груша Слива Персик Ананас Вишня Апельсин Лимон Мандарин (переход к следующему элементу списка Ctrl/ Enter)
	Sorted	True
	Height	1035
	Style	0-Standard

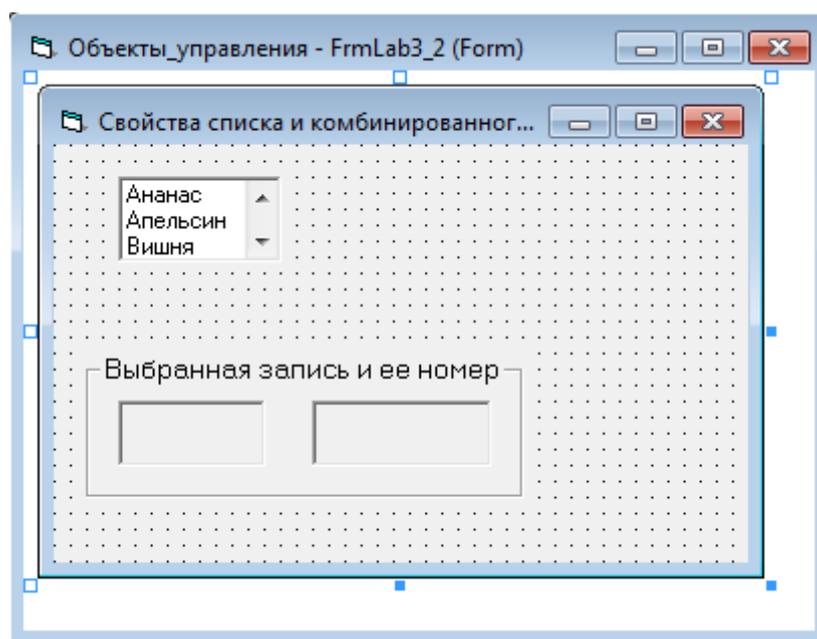


Рис. 1. Интерфейс приложения

4. В окне кода процедуры обработки события **Click** для объекта **LstFruct** добавьте строки:

**Lbl2.Caption = LstFruct.ListIndex**

**Lbl1.Caption = LstFruct.Text**

5. Проверьте работу созданной программы (обратите внимание, что список отсортирован и нумерация элементов списка начинается с нуля).

6. Остановите программу, установите для объекта **LstFruct** значение **1-CheckBox** для свойства **Style** и проверьте работу программы.

7. Добавьте на форму объекты и измените их свойства в соответствии с табл. 3. Приведите свою форму в соответствие с образцом рис. 2.

Таблица 3

Объект	Свойство	Значение
Рамка	Caption	Количество записей в списке
	Font	Размер 10
Метка	Name	Lbl3
	Caption	(отсутствует)
Комбинированный список	Name	CboProduct
	List	Хлеб Молоко Сыр Сметана Масло Творог
	Sorted	False
	Width	1000
	Style	0-DropDown Combo
	Text	(отсутствует)
Командные кнопки	Name	CmdAdd
	Caption	Добавить
	Name	CmdDel
	Caption	Удалить

8. В окне кода процедуры обработки события **Click** для объекта **CmdAdd** добавьте строки:

```
CboProduct.AddItem CboProduct.Text  
Lbl3.Caption = CboProduct.ListCount
```

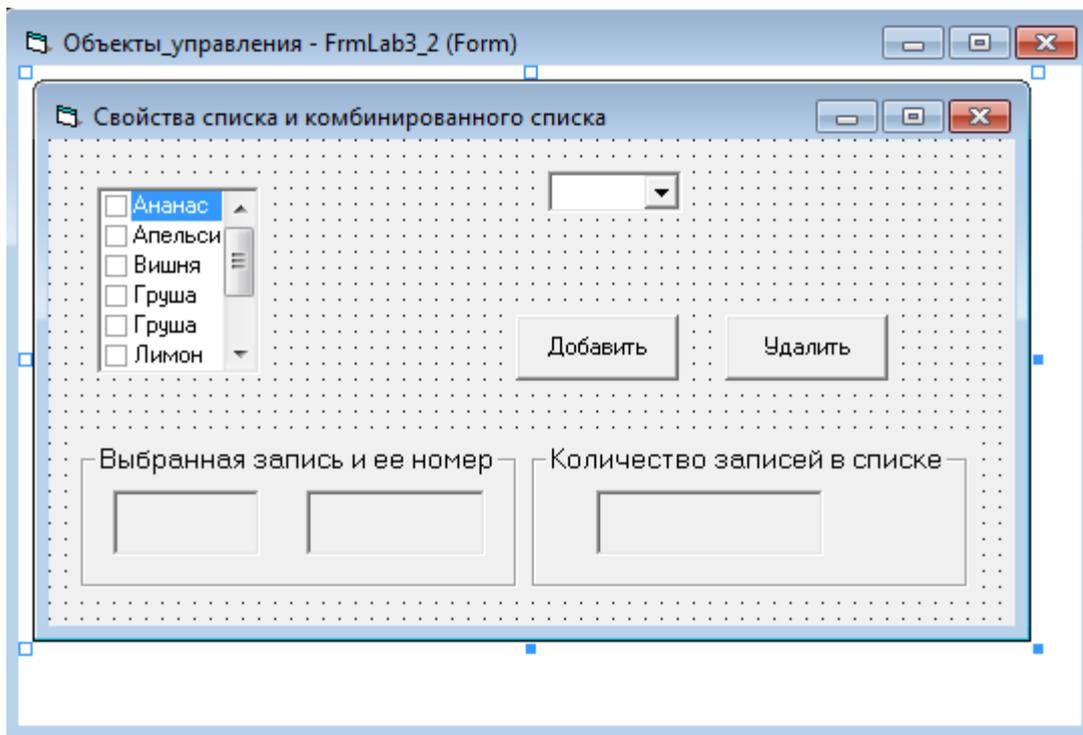


Рис. 2. Интерфейс приложения

9. В окне кода процедуры обработки события **Click** для объекта **CmdDel** добавьте строки:

**CboProduct.ListIndex = 0**

**CboProduct.RemoveItem CboProduct.ListIndex**

**lbl3.Caption = CboProduct.ListCount**

10. Проверьте работу программы (рис. 3). Сохраните проект в рабочей папке.

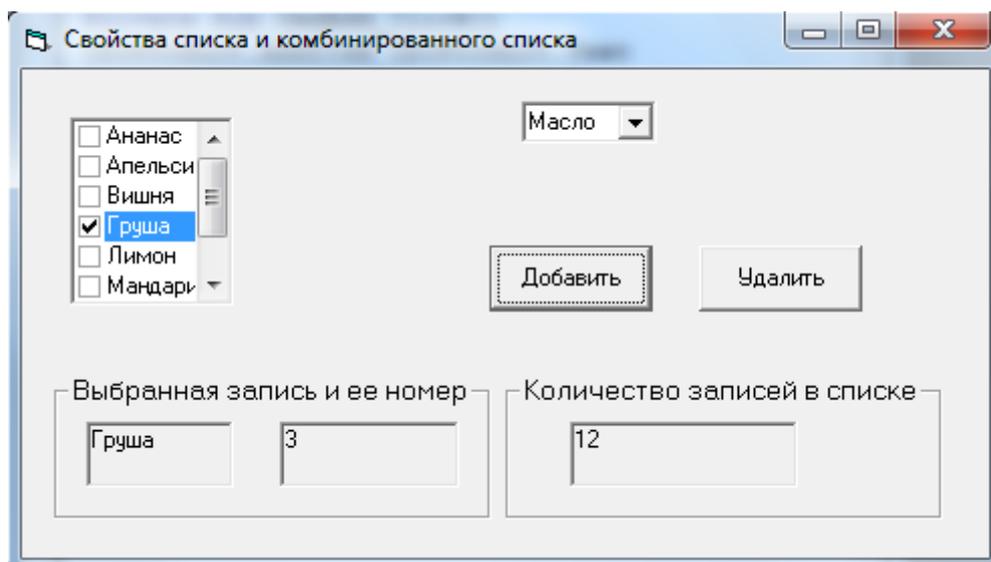


Рис. 3. Результат работы программы

11. Добавьте в проект «Объекты\_управления» новую форму.

12. Установите значения свойств элементов согласно данным таблицы 4. Приведите свою форму в соответствии с образцом (рис. 4).

Таблица 4

<b>Объект</b>	<b>Свойство</b>	<b>Значение</b>
1	2	3
Frame1 (рамка)	Caption	Выбери фигуру
Frame2	Caption	Ширина границы фигуры
Frame3	Caption	Стиль заполнения фигуры
Frame4	Caption	Стиль границ
Frame5	Caption	Цвет границ
Option1 (Переключатели)	Name Caption	Opt1 Прямоугольник
Option2	Name Caption	Opt2 Квадрат
Option3	Name Caption	Opt3 Овал
Option4	Name Caption	Opt4 Окружность
Option5	Name Caption	Opt5 Прямоугольник с закругленными углами
HScroll1 (Горизонтальные прокрутки)	Name Max Min	HsbWidth 20 1
HScroll2	Name Max Min	HsbFillSt 7 0
HScroll3	Name Max Min	HsbBorderSt 6 0
HScroll4	Name Max Min	HsbRed 255 0

1	2	3
HScroll5	Name	HsbBlue
	Max	255
	Min	0
HScroll6	Name	HsbGreen
	Max	255
	Min	0
Shape1	Name	Shape1

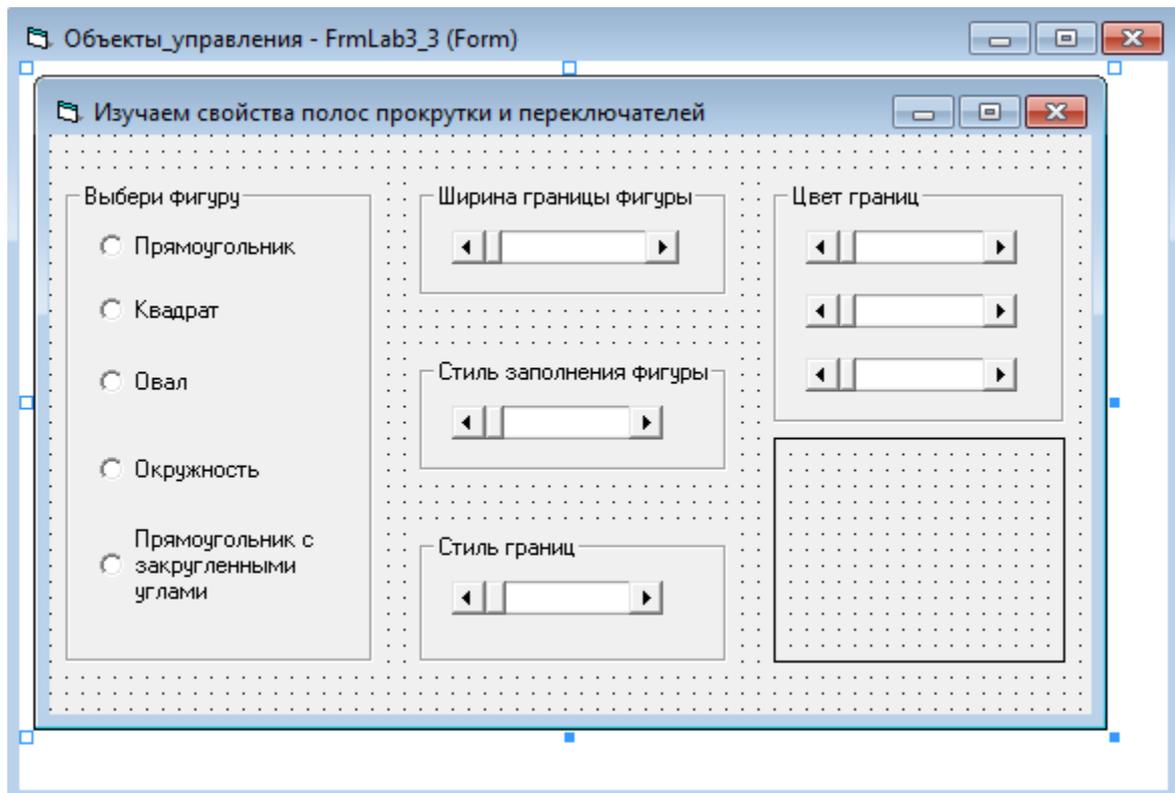


Рис. 4. Интерфейс приложения

Чтобы **поместить** переключатель **внутри рамки**, необходимо **один раз щелкнуть на нем в панели элементов и перетащить его внутрь рамки**.

13. В окне кода процедуры обработки события **Change** для объектов **Hsbred**, **HsbBlue**, **HsbGreen** добавьте строки:

```
Shape1.BorderColor = RGB(Hsbred.Value, Hsbgreen.Value, HsbBlue.Value)
```

14. В окне кода процедуры обработки события **Click** для объекта **Opt1** добавить строки:

```
Shape1.Shape = 0
```

15. Самостоятельно введите код для других переключателей, меняя только число:

- Квадрат 1
- Овал 2
- Окружность 3
- Прямоугольник с закругленными углами 5

16. Проверьте работу программы (рис. 5).

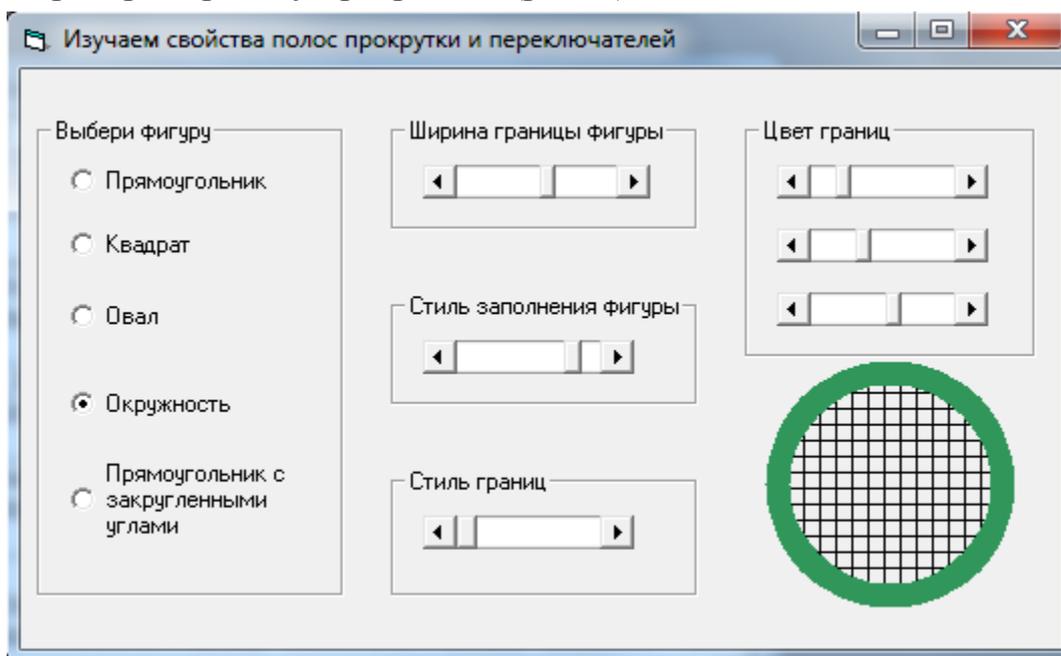


Рис. 5. Работа приложения

17. Добавьте в проект «Объекты\_управления» новую форму.

18. Установите значения свойств элементов согласно данным таблицы 5.

Приведите свою форму в соответствии с образцом (рис. 6).

Таблица 5

Объект	Свойство	Значение
1	2	3
Метки	Caption	212
	Caption	100
	Caption	По Фаренгейту
	Caption	По Цельсию
	Caption	-40
	Caption	-40

1	2	3
Вертикальная прокрутка	Name	VsbTherm
	Max	-40
	Min	100
	LargeChange	10
	SmallChange	1
Текстовые окна	Name	TxtF
	Text	(отсутствует)
	Name	txtC
	Text	(отсутствует)

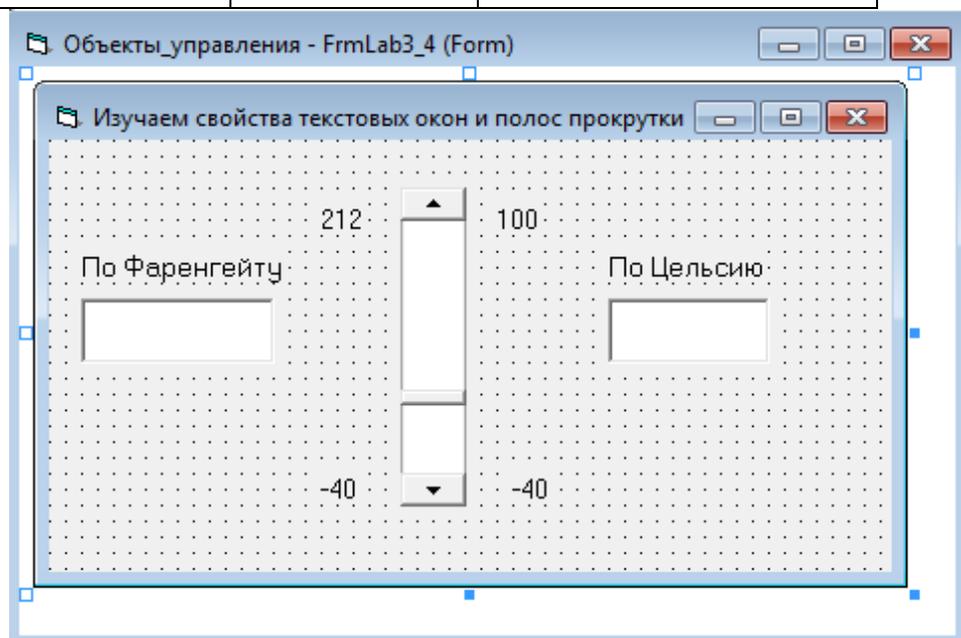


Рис. 6. Интерфейс приложения

19. В окне кода процедуры обработки события **Change** добавьте строки для объекта **VsbTherm** проверьте работу приложения:

```

Dim tempC, tempF As Integer
tempC = CInt ( vsbTherm.Value )
TxtC.Text = Str ( tempC )
tempF = CInt ( tempC * 9 / 5 + 32 )
txtF.Text = Str( tempF )

```

20. Проверьте работу приложения (рис. 7):

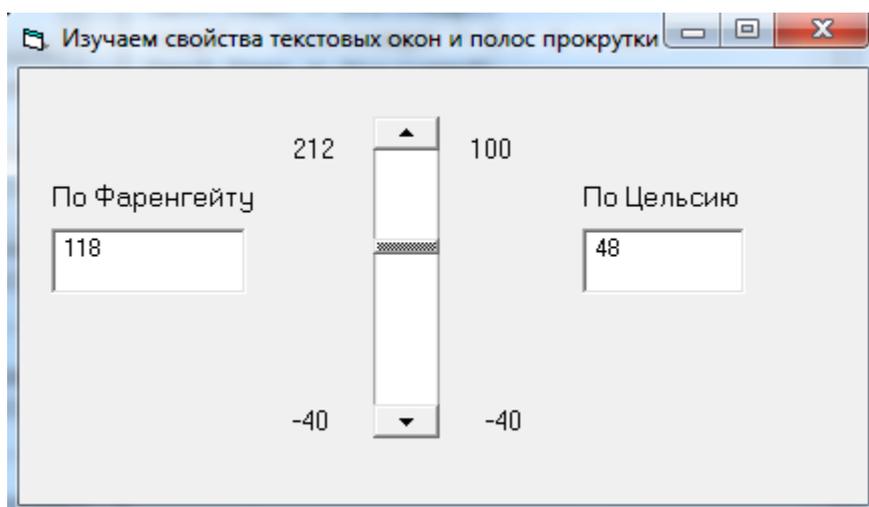


Рис. 7. Результат работы приложения

21. В графическом редакторе **Paint** создайте четыре рисунка «зима», «весна», «лето», «осень» и сохраните рисунки в отдельных файлах на диске **C:**.

22. Добавьте в проект «Объекты\_управления» новую форму. Установите значения свойств элементов согласно данным таблицы 6. Приведите свою форму в соответствии с образцом (рис. 8).

Таблица 6

Объект	Свойство	Значение
1	2	3
Форма	Name	frmPaint
	WindowState	2-Maximized
Метка	Name	LblResult
	Caption	0
Полоса прокрутки	Name	HsbSet
	Max	15
	Min	1
Командные кнопки	Name	CmdWinter
	Caption	Зима
	Name	CmdSpring
	Caption	Весна
	Name	CmdSummer
	Caption	Лето

1	2	3
Командные кнопки	Name	CmdAutumn
	Caption	Осень
	Name	cmdExit
	Caption	Выход

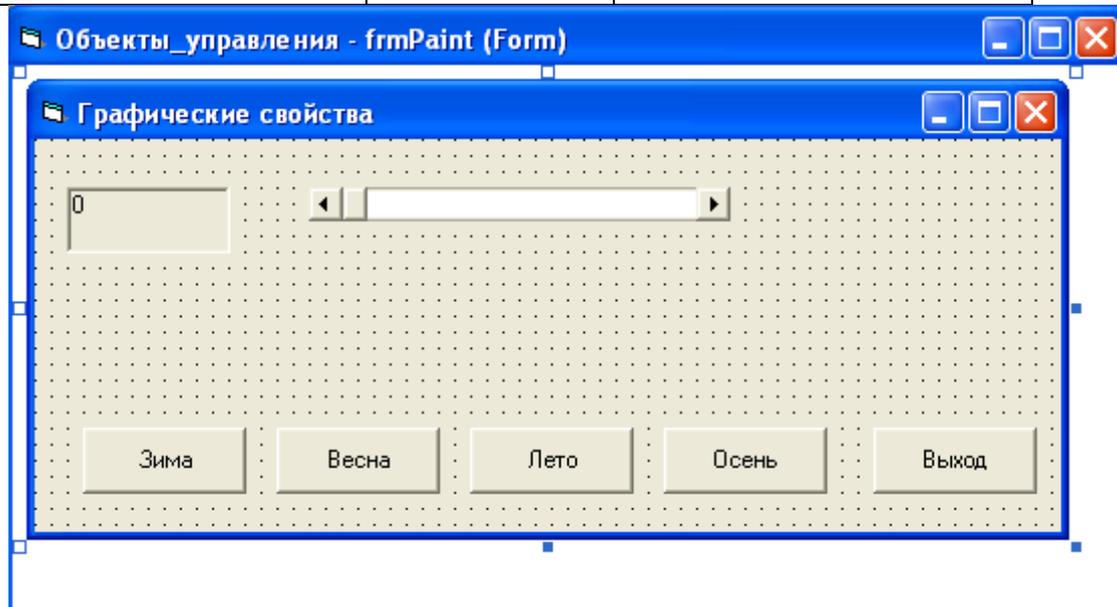


Рис. 8. Интерфейс приложения

23. В окне кода процедуры обработки события **Change** набрать для объекта **HsbSet**:

```
lblResult.Caption = Str(hsbSet.Value)  
frmPaint.BackColor = QBColor(hsbSet.Value)
```

24. Такой же код набрать для обработки события **Scroll** для объекта **HsbSet**.

25. В окне кода процедуры обработки события **Click** набрать для объекта **cmdExit**:

```
MsgBox "До свидания",0,"Выход"  
End
```

26. В окне кода процедуры обработки события **Click** набрать для объекта **cmdWinter**:

```
frmPaint.Picture = LoadPicture("c:\Мои Документы\зима.bmp")
```

27. Проверить работу приложения (рис. 9):

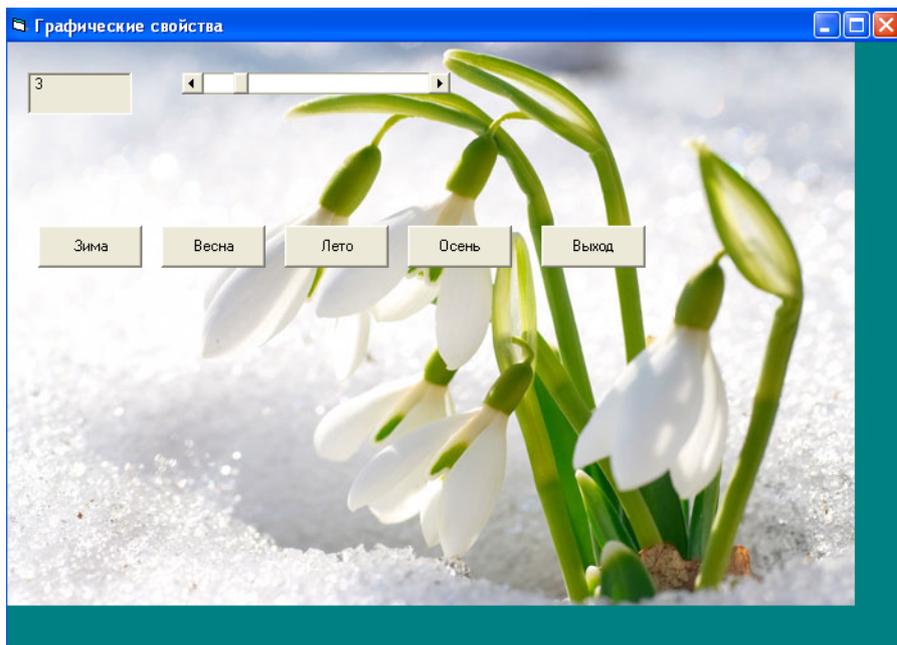


Рис. 9. Результаты работы приложения