ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: Объекты управления.

Объекты управления – это скелет программы в *VB*. Одни из них служат для создания *интерфейса пользователя*, например, *текстовые поля*, *командные кнопки*, другие выполняют вспомогательные функции, например, *таймер*.

Интерфейс – это внешняя оболочка приложения вместе с программами управления, доступом и другими скрытыми от пользователя механизмами управления, дающая возможность работать с документами, данными и другой информацией, хранящейся в компьютере или за его пределами. Хорошо продуманный интерфейс облегчает работу с программой и препятствует в совершении ошибок.

Рассмотрим основные (базовые) объекты управления, к которым можно отнести: *Командные кнопки*; *Надписи* (*текстовые метки*); *Кнопки выбора*; *Переключатели* (*флажки*); *Рамки*; *Текстовые поля* (*текстовые окна*); *Списки*.

Командные кнопки (***CommandButton***) **–** самый распространенный управляющий элемент в приложениях для *Windows*. Элемент управления *CommandButton* используется, чтобы: начать, прервать или закончить процесс. Когда кнопка нажата, вызывается программа, написанная в процедуре события *Click*.

Свойство *Name* (*Имя*) определяет имя командной кнопки.

Свойство *Caption* определяет текст, который должен находиться на кнопке.

Свойства *Enable* и *Visible* регулируют доступ к кнопке. Если какое-либо из этих свойств имеет значение *False* – кнопка станет недоступной. Свойство *Enable* = False означает, что кнопка блокирована. Это может пригодиться в тех случаях, когда до нажатия этой кнопки пользователь должен выполнить определенные действия (например, заполнить текстовое поле) и лишь затем нажать кнопку и перейти к следующей стадии процесса. Свойство *Visible* = *False* означает, что кнопка с таким значением свойства становится невидимой.

Надписи (***Label***) могут быть использованы (как и текстовое окно) для отображения небольшого текста или результатов простых вычислений.

Свойство *Caption* определяет текст, который выводится в элементе *Label*. Надписи часто содержат справочную информацию. Они могут использоваться как самостоятельно, так и в виде подсказок для текстового поля, списка или другого элемента.

Переключатели (***OptionButton***) **–** позволяют выбрать один (и только один) вариант из группы. Обычно они группируются в рамках, однако их можно группировать прямо на форме (рис. 5). Переключатели обладают многими полезными свойствами, но наиболее используемыми являются: *Name*; *Caption*; *Value*.

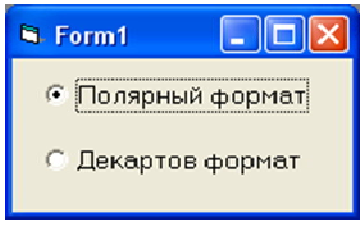

Рис. 5. *Переключатели*

Свойство *Name* (*Имя*) определяет имя элемента.

Свойство *Caption* помогает пользователю определить для чего предназначен переключатель.

Свойство *Value* указывает, выбран элемент или отключен. Когда он выбран, свойство *Value* = True, иначе *Value* = False.

Флажки (CheckBox) – применяются, когда пользователь должен дать ответ в виде: *да/нет* или *истина/ложь*. В случае положительного ответа пользователь устанавливает флажок, и он приобретает вид квадрата, в котором размещена галочка. При не установленном флажке он имеет вид пустого квадрата, обозначая отрицательный ответ. Флажки могут использоваться в форме по одному или группами.

Свойство *Value* элемента управления *CheckBox* указывает, выбран элемент, отключен или недоступен: *Value* = 1 (элемент выбран); *Value* = 0 (элемент отключен); *Value* = 2 (элемент недоступен).

Для разделения объектов по независимым группам используется элемент *Frame* (*Рамка*) (рис. 6). Рамки обычно используются не сами по себе, а в сочетании с другими элементами. Внутри рамок размещаются такие элементы как *переключатели* и *флажки*. Они группируются и в случае перемещения рамки перемещаются вместе с ней. Самым важным свойством (после *Name*) следует считать *Caption*, которое позволяет снабдить рамку содержательным названием, чтобы проще было понять, по какому признаку объединены элементы.

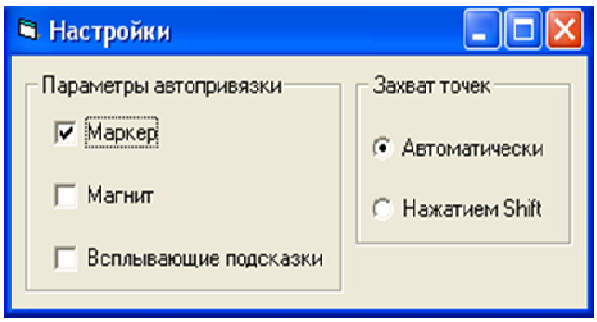

Рис. 6. *Переключатели* и *Флажки* в рамке

Текстовое поле (***TextBox***) – используется для отображения информации и ввода данных. Информация, выводимая в текстовом поле, определяется свойством *Text* и может задаваться следующими способами:

- ✓ в диалоговом окне *Properties* (*Свойства*);
- ✓ в процессе выполнения программного кода;
- ✓ при вводе данных в поле пользователем.

Чтобы поле было пустым, достаточно содержание свойства *Text* просто удалить.

Свойство *MaxLength* ограничивает длину вводимого текста заданным количеством символов и в сочетании со свойством *Password* в окне задается символ, который отображается в окне поля при вводе (***). Эти два свойства часто используются в формах для регистрации пользователя.

Свойство *Multiline* позволяет ввести текст, состоящий из нескольких строк. Для перехода на новую строку нужно нажать комбинацию клавиш *<Ctrl> <Enter>*. Если свойство *Multiline* используется в сочетании со свойством *ScrollBar*, то поле практически без всякого кодирования превращается в простейший текстовый редактор.

Свойство *Locked = True* определяет возможность пользователя только выводить информацию и данные, отображаемые в поле, можно только просматривать, но нельзя редактировать.

Событие *Change* происходит каждый раз, когда пользователь вставляет, заменяет или удаляет символы текстового поля.

Свойства, определяющие шрифт. Свойство *Font* (*Шрифт*) позволяет задать шрифт текста объекта, размер, начертание.

Упражнение 1.

1. Создайте проект.
2. Откройте окно свойств формы и установите следующие свойства:
   - *Caption* присвойте значение: Текстовое Окно;
   - *Height* (высота): 3645;
   - *Width* (ширина): 7110.
3. Создайте на форме элемент *TextBox* и установите для него следующие свойства:
   - *Имя*: Text1;
   - *Alignment* (выравнивание): 2-центровка;
   - *Font*: *шрифт*: Times New Roman, *начертание*: обычный, *Размер*: 12;

- *Height* (высота): 1455;
- *Width* (ширина): 5235;
- *Multiline* (многострочный): True;
- *Top*: 240
- *Text* : Кузнецов
  Сергей Иванович
  Механический факультет
  1 курс

4. Под элементом *TextBox* создайте три командные кнопки с именами *Command*1, *Command*2 и *Command*3 соответственно.

5. Свойствам *Caption* кнопок присвойте соответственно значения:

Command1: Очистка

Command2: Вставить время

Command3: Сообщение

6. Открывая последовательно окна программных кодов кнопок, введите в них соответствующие коды:

```
Private Sub Command1_Click()
   Text1.Text = ""
End Sub
Private Sub Command2_Click()
    Text1.Text = Time
End Sub
Private Sub Command3_Click()
   Text1.Text = "Так работает Текстовое Окно"
End Sub
```

7. Запустите приложение. Окно приложения должно иметь вид, как на рис. 7.

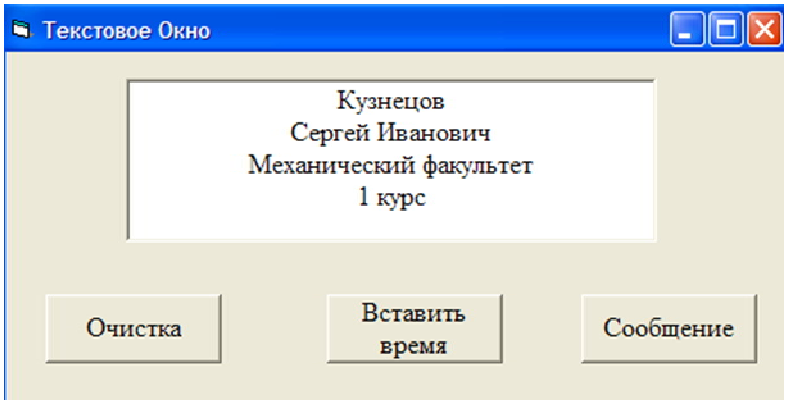

Рис. 7. Вид *Текстового Окна* из упражнения 1

8. Нажимая последовательно кнопки, отметьте изменения содержания текстового окна.

9. Завершите работу приложения кнопкой закрытия окна.

Завершая работу с текстовыми окнами, отметим, что элемент *TextBox* допускает ввод текста, объем которого не превышает 64 Кбайт. При вводе данных в одну строку, длина ее не должна превышать 255 символов. Эти параметры соответствуют текстовому файлу среднего объема.

Список (***ListBox***, ***ComboBox***) – представляет собой список значений, из которого пользователь может выбрать одно из предложенных значений. Значения в списке могут размещаться

в одну или несколько колонок. Количество колонок задается свойством *Columns* (*Колонки*). Для создания списков используются две кнопки на панели элементов управления *ListBox* и *ComboBox*:

- ✓ *ListBox* (*Список*) создаёт в форме список, в котором элементы расположены в одну или несколько колонок.
- ✓ *ComboBox* (*Поле со списком*) создает в форме раскрывающийся список.

На рис. 8 показаны списки, представляющие собой элементы управления *ListBox* (*Список*) и *ComboBox* (*Поле со списком*). Оба эти элемента используют список. Список в терминах *Visual Basic* – это массив строк, на который можно формально сослаться с помощью свойства *List.*

Списки и поля со списком – динамические элементы по своей природе. Можно изменять значение свойства *List* этих элементов управления во время проектирования (в окне свойства элемента управления), что конечно удобно при создании программы, но самое главное все-таки добавлять и удалять строки свойства *List* во время выполнения программы. Для этой цели используются методы *AddItem* (добавить элемент), *RemoveItem* (удалить элемент) и *Clear* (очистить).

Для указания позиции элемента в списке свойства *List* используется свойство *ListIndex*. Когда пользователь щелкает на какой-нибудь строке в окне списка, *Visual Basic* присваивает номер этой строки в качестве значения свойству *ListIndex* данного элемента управления. Нумерация элементов списка начинается с 0.

Упражнение 2.

1. Создайте на форме следующие объекты: три элемента *Label*; *ListBox* и *ComboBox*. Расположите их, как показано на рис. 8.

2. Свойствам *Caption* элементов *Label* присвойте текст в соответствии с рис. 8.

3. Откройте окно свойств объекта *ListBox* и в свойстве *Имя* введите List1. Выделите свойство *List* (*Список*). В правом столбце свойства появится кнопка, содержащая направленную вниз стрелку. Нажмите эту кнопку. Откроется список, позволяющий вводить значения. Введите первое значение списка. Для перехода на новую строку списка нажмите комбинацию клавиш <*Ctrl*> <*Enter*>. Введите следующее значение и сформируйте весь список, как на рис. 8.

4. Откройте окно свойств объекта *ComboBox* и в свойстве *Имя* введите Combo1. Выделите свойство *List* (Список) и, аналогично пункту 3, создайте список значений.

1. Запустите приложение на выполнение.

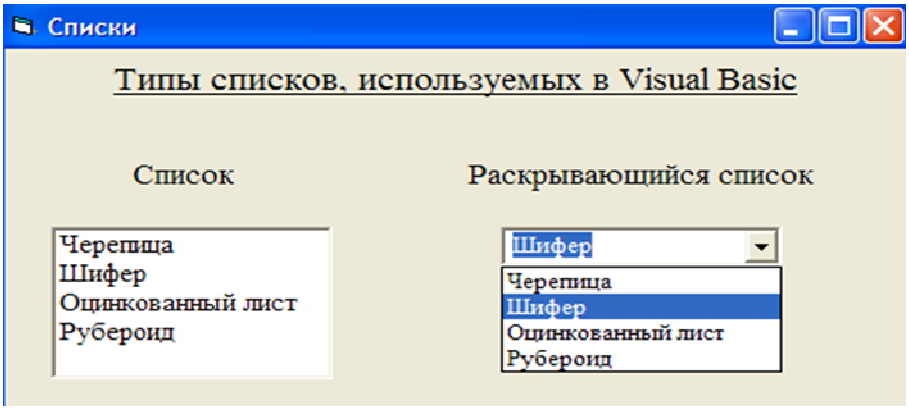

Рис. 8. Два типа списков из упражнения 2

Иногда для удобства пользователей требуется, чтобы при появлении формы на экране в списке по умолчанию было выделено наиболее часто выбираемое из него значение. Для установки значения, выбираемого по умолчанию, используется свойство *Listindex*. Например, вы хотите, чтобы при загрузке формы в списке объекта *ComboBox*, имеющем имя Combo1, выделялся второй элемент. Для

этого откройте для формы окно программного кода, и в процедуре Form_Load() (обработка события при загрузке формы) введите следующий программный код:

```
Private Sub Form_Load()
Combo1.Listindex = 1
End Sub
```

В программном коде вместо цифры 2, указывающей номер выделяемого элемента, содержится цифра 1, т. к. нумерация элементов списка начинается с 0.

Упражнение 3.

1. Создайте на форме следующие объекты: *ListBox*, два элемента *TextBox*, четыре элемента *Label* и командную кнопку *CommandButon*. Расположите их, как показано на рис. 9.

2. Свойствам *Caption* элементов *Label* присвойте соответствующий текст, как на рис. 9.

3. Откройте окно свойств объекта *ListBox*. Задайте объекту имя List1. В свойстве *List* (*Список*) введите список дисциплин. Свойству *Sorted* присвойте значение True (чтобы вставляемые в список дисциплины сортировались в алфавитном порядке).

4. Откройте окно свойств объекта *CommandButon* и создайте на командной кнопке название: Добавить в список.

5. Откройте окно программного кода объекта *CommandButon* и введите в процедуру Command1_Click() (обработка события объекта при нажатии) программный код для добавления дисциплины в список:

```
Private Sub Command1_Click()
List1.AddItem Text1.Text  'Добавление записи в объект List1 из Text1
End Sub
```

6. Откройте окно программного кода объекта *ListBox* и введите в процедуру List1_Click() (обработка события объекта при выборе записи) программный код для переноса выбранной дисциплины из списка в текстовое поле *Text*2:

```
Private Sub List1_Click()
Text2.Text = List1.Text     'Добавление записи в объект Text2 из List1
 List1.RemoveItem(List1.ListIndex)  'Удаление выбранной записи из List1
 End Sub
```

7. Запустите программу. Форма должна иметь вид, как показано на рис. 9.

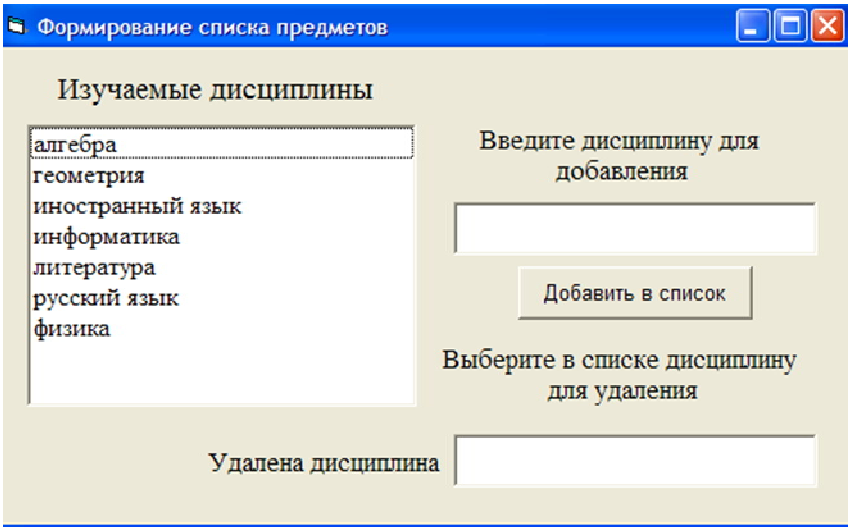

Рис. 9. Работа со списками из упражнения 3

Упражнение 4. Создайте новый проект, в котором на форме расположите два окна списков. В один из списков введите названия городов. Составьте программу, которая щелчком мыши переносит запись из одного списка в другой. Задайте в обоих списках сортировку строк по алфавиту.

Контрольные вопросы:

1. Что вы понимаете под интерфейсом пользователя?
2. Перечислите свойства и назначение *Командной кнопки* (*CommandButton*).
3. Перечислите свойства и назначение *Надписи* (*Label*).
4. Перечислите свойства и назначение *Переключателей*(*OptionButton*), *Флажков*(*CheckBox*).
5. Как изменить параметры шрифта для отображаемого текста?
6. Перечислите свойства и назначение *TextBox* (*Текстовое поле*).
7. Перечислите свойства и назначение *ListBox* и *ComboBox* (*Списки*). Какие методы позволяют добавлять и удалять из списка значения?