**Работа с пользовательскими формами в VBA**

В настоящей заметке рассматриваются методы создания пользовательских форм и работы с ними.[[1]](#footnote-1) Пользовательские диалоговые окна создаются на основе технологии пользовательских форм, к которым можно получить доступ из редактора Visual Basic (VBE; подробнее см. [Настройка среды Visual Basic Editor](http://baguzin.ru/wp/?p=16244)).

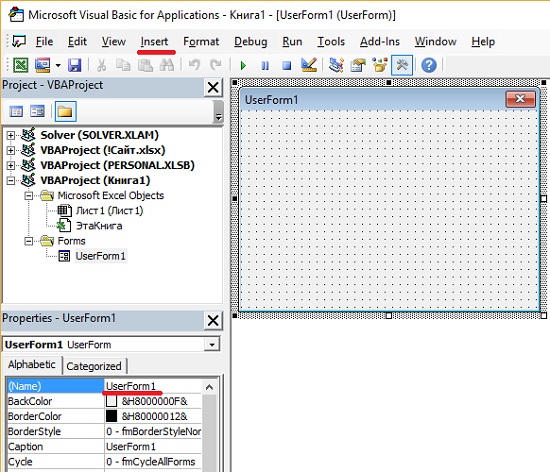


Рис. 1. Окно новой пустой формы UserForm

Стандартная последовательность шагов при этом следующая:

1. Вставьте новую форму UserForm в проект VBAProject рабочей книги.
2. Добавьте элементы управления в форму UserForm.
3. Настройте свойства добавленных элементов управления.
4. Создайте процедуры «обработчики событий» для элементов управления. Эти процедуры добавляются в модуль кода UserForm и выполняются при возникновении различных событий (например, при щелчке на кнопке).
5. Разработайте процедуру, которая отображает форму UserForm. Эта процедура находится в модуле VBA (а не в модуле кода для формы UserForm).
6. Определите простой способ вызова на выполнение процедуры, созданной в п. 5. Можно поместить кнопку на рабочий лист, команду ленты и т.д.

Рассмотрим эти шаги подробнее.

### Вставка новой формы UserForm

Чтобы добавить в проект форму UserForm, запустите VBE (например, нажав в Excel клавиши <Alt+F11>), выберите рабочую книгу в окне *Project* и выполните команду *Inserts* –> *UserForm*). Формы UserForm получают такие имена, как UserForm1, UserForm2 и т.д. Можно переименовать форму, изменив свойство *Name* в окне *Properties* (см. рис. 1). Если это окно не отображается, нажмите в VBE клавишу <F4>. В рабочей книге может быть произвольное количество форм UserForm, а каждая форма включает единственное пользовательское диалоговое окно.

### Добавление элементов управления в пользовательское диалоговое окно

Чтобы добавить элементы управления в форму UserForm, воспользуйтесь панелью *Toolbox*. Обратите внимание, что в VBE отсутствуют команды меню, предназначенные для добавления элементов управления. Если панель *Toolbox* не отображена на экране, пройдите по меню *View* –> *Toolbox* (рис. 2).

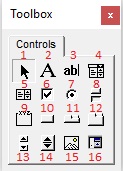


Рис. 2. Окно *Toolbox* для добавления элементов управления в пользовательскую форму

Щелкните на той кнопке в панели *Toolbox*, которая соответствует добавляемому элементу управления. После этого щелкните внутри диалогового окна для создания элемента управления (используется размер элемента по умолчанию). Также можно щелкнуть на элементе управления и, перетащив его границы в диалоговом окне, задать необходимый размер в пользовательском диалоговом окне. Добавленному элементу управления назначается имя, которое состоит из названия типа элемента управления и числового кода. Рекомендуется их переименовать, чтобы в коде VBA было понятно, с чем вы имеете дело. Согласитесь, что *РrоductListBox* звучит лучше, чем *ListBox1*.

### Элементы управления в окне Toolbox

Форма *UserForm*, которая показана на рис. 3, размещена в файле *all userform controls.xlsm*.

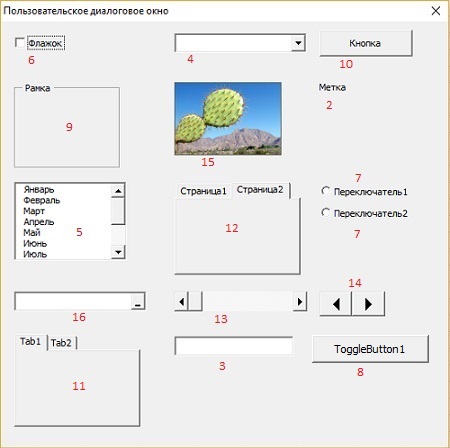


Рис. 3. Эта форма *UserForm* содержит 15 элементов управления

Элемент управления *CheckBox* (6) предоставляет пользователю возможность выбрать один из двух вариантов: включить или выключить. Если галочка установлена, то *CheckBox* имеет значение True, в противном случае – False.

*ComboBox* (4) подобен объекту *ListBox* (5). Отличие заключается в том, что *ComboBox* представляет раскрывающийся список, в котором в определенный момент времени отображается только одно значение. Кроме того, пользователю в поле списка разрешено вводить значение, которое необязательно представляет одну из опций объекта *ComboBox*.

Каждое создаваемое диалоговое окно будет иметь как минимум один элемент управления CommandButton (10). Обычно используются кнопки *ОК* и *Отмена*.

Элемент управления Frame (9) применяется в качестве оболочки для других элементов управления. Он добавляется в диалоговое окно либо в целях эстетики, либо из соображений логического группирования однотипных элементов управления. Элемент управления Frame требуется в случае, если в диалоговом окне содержится более одного набора элементов управления OptionButton.

Элемент управления Image (15) используется для представления графического изображения, которое сохранено в отдельном файле или вставляется из буфера обмена. Графическое изображение сохраняется вместе с рабочей книгой. Таким образом, вместе с рабочей книгой передавать другому пользователю копию графического файла необязательно. Некоторые графические файлы занимают много места, поэтому их включение в рабочую книгу приведет к радикальному увеличению ее размера.

Элемент управления Label (2) отображает текст в диалоговом окне.

Элемент управления ListBox (5) предоставляет список опций, из которого пользователь может выбрать один вариант (или несколько). Вы вправе указать диапазон на листе, который содержит элементы списка. Этот диапазон может состоять из нескольких столбцов. Кроме того, элемент управления ListBox может заполняться с помощью кода VBA.

Элемент управления MultiPage (12) позволяет создавать диалоговые окна с несколькими вкладками, которые подобны появляющимся после выбора команды *Формат ячеек*. По умолчанию элемент управления MultiPage состоит из двух вкладок.

Элемент управления OptionButton (7) применяется при выборе пользователем одного варианта из нескольких. Эти элементы управления всегда группируются в диалоговом окне в наборы, содержащие не менее двух опций. Когда один элемент управления OptionButton выбран, все остальные элементы управления OptionButton текущей группы автоматически становятся неактивными. Если в пользовательском диалоговом окне содержится более одного набора элементов управления OptionButton, то каждый из таких наборов должен иметь собственное значение свойства GroupName. В противном случае все элементы управления OptionButton в диалоговом окне рассматриваются как члены одной группы. Также можно вставить элементы управления OptionButton в объект Frame, что приведет к их автоматическому группированию.

Элемент управления RefEdit (16) используется тогда, когда пользователь должен выделить диапазон ячеек на листе.

Элемент управления ScrollBar (13) в некотором смысле подобен элементу управления SpinButton. Разница заключается в том, что пользователь может перетаскивать ползунок объекта ScrollBar для изменения значения с большим приращением. Элемент управления ScrollBar рекомендуется использовать при выборе значения из большого диапазона.

Элемент управления SpinButton (14) позволяет выбрать значение после щелчка на одной из двух кнопок со стрелками. Одна из них применяется для увеличения значения, а вторая — для уменьшения. Элемент управления SpinButton часто используется совместно с элементами управления TextBox и Label, которые содержат текущее значение элемента управления SpinButton.

Элемент управления TabStrip (11) подобен элементу управления MultiPage, однако использовать его сложнее. Элемент управления TabStrip, в отличие от MultiPage, не выступает контейнером для других объектов. Как правило, элемент управления MultiPage обладает более широкими возможностями.

Элемент управления TextBox (3) позволяет пользователям вводить текст в диалоговом окне.

Элемент управления ToggleButton (8) имеет два состояния: включен и выключен. Щелчок на кнопке приводит к изменению состояния на противоположное и к изменению внешнего вида кнопки. Этот элемент управления может иметь значение True (активен) или False (неактивен). Он не относится к «стандартным» элементам управления, поэтому использование двух элементов управления OptionButton или одного CheckBox может оказаться более удачным вариантом.

### Использование элементов управления на рабочем листе

Элементы управления пользовательскими диалоговыми окнами могут встраиваться в рабочий лист (без использования *UserForm*). Доступ к элементам управления можно получить, пройдя в Excel *Разработчик* –> *Элементы управления* –> *Вставить*. Для использования подобных элементов в составе рабочего листа требуется гораздо меньше усилий, чем для создания пользовательского диалогового окна. Кроме того, в данном случае можно не создавать макросы, поскольку элемент управления можно связать с ячейкой рабочего листа.

Например, если на рабочий лист вставить элемент управления *CheckBox*, его можно связать с нужной ячейкой, задав свойство *LinkedCell*. Если флажок установлен, в связанной ячейке отображается значение ИСТИНА. Если же флажок сброшен, то в связанной ячейке отображается значение ЛОЖЬ. Например, на рис. 4 переключатель «Фикс. Ставка 20%» связан с ячейкой Н15. Рисунок отображает рабочий лист, содержащий некоторые элементы управления ActiveX (см. файл *activex worksheet controls.xlsx*). Книга включает связанные ячейки и не содержит макросов.



Рис. 4. Элементы управления ActiveX без макросов

Элементы управления на рабочем листе могут происходить из двух источников:

* *Элементы управления формами*. Эти элементы управления являются внедряемыми объектами.
* *Элементы управления ActiveX*. Эти элементы управления являются подмножеством элементов, доступных в пользовательских диалоговых окнах.

Эти элементы управления работают не одинаково. После добавления элемента управления ActiveX в рабочий лист Excel переходит в режим конструктора. В этом режиме можно настраивать свойства любого элемента управления рабочего листа, добавлять или изменять процедуры обработки событий для элемента управления, а также изменять его размер или положение. Для отображения окна свойств элемента управления ActiveX воспользуйтесь командой *Разработчик* –> *Элементы управления* –> *Свойства*.

Для создания простых кнопок можно использовать элемент управления *Button (Кнопка)*, который находится на панели инструментов *Формы (Form)*. В этом случае обеспечивается возможность запуска макроса. Если же воспользоваться элементом управления *CommandButton*, который относится к группе элементов управления *ActiveX*, то после щелчка на нем вызывается связанная процедура обработки событий (например, *CommandButton1\_Click*), которая находится в модуле кода объекта *Лист* (*Sheet*). Связать макрос с этой процедурой нельзя.

Если Excel находится в режиме конструктора, тестирование элементов управления невозможно. В этом случае нужно выйти из режима конструктора, щелкнув на кнопке *Разработчик* –> *Элементы управления* –> *Режим конструктора*. Эта кнопка работает, как переключатель.

### Настройка элементов управления пользовательского диалогового окна

Продолжим описание использования элементов управления в *UserForm.* После того, как элементов управления помещен в диалоговое окно, его можно переместить и изменить размер. Можно выделить несколько элементов управления. Для этого следует удерживать нажатой клавишу <Shift> и щелкать на объектах либо обвести указателем мыши необходимые элементы управления.

В форме UserForm содержатся вертикальные и горизонтальные направляющие, которые помогают выровнять добавленные в диалоговое окно элементы управления. При добавлении или перемещении элемент управления привязывается к направляющим, что облегчает упорядочение таких элементов в окне. Если вы не используете направляющие, можете их отключить, выбрав в VBE команду *Tools* –> *Options*. В диалоговом окне *Options* перейдите на вкладку *General* и выберите соответствующие параметры в разделе *Form Grid Settings*.

Меню *Format* окна VBE предоставляет несколько команд, которые позволяют точно разместить и выровнять элементы управления в диалоговом окне. Перед использованием этих команд необходимо указать элементы управления, к которым они будут применяться (рис. 5).

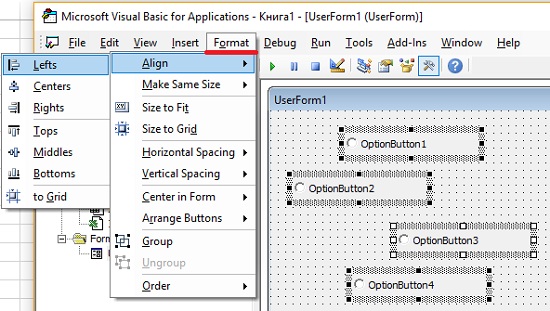


Рис. 5. Выравнивание элементов в форме UserForm

Если выделено несколько элементов управления, на выделяющей рамке одного из них (обычно последнего) вместо обычных черных маркеров появляются белые. Это означает, что этот элемент управления играет роль основы, на базе которой определяются размеры и расположение других элементов управления.

### Изменение свойств элементов управления

Каждый элемент управления характеризуется набором параметров, которые определяют внешний вид и поведение элемента управления. Свойства элемента управления можно изменять в следующих случаях:

* В момент проектирования при разработке пользовательского диалогового окна. Для этого используется окно *Properties*.
* В процессе выполнения, когда пользовательское диалоговое окно отображается на экране. Для этого воспользуйтесь инструкциями VBA.

**Работа с окном *Properties*.** В VBE окно *Properties* позволяет изменять свойства выделенного элемента управления (это может быть обычный элемент управления или сама форма *UserForm*, рис. 6).

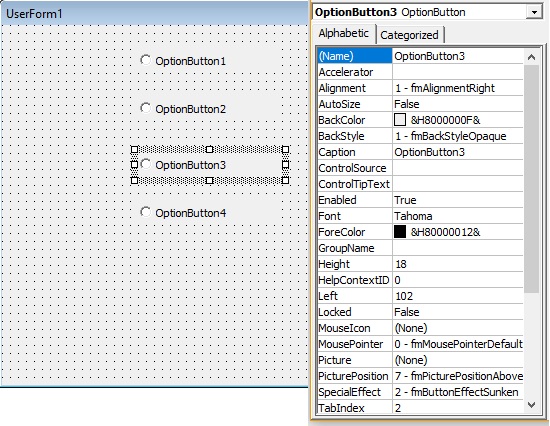


Рис. 6. Окно *Properties* для выделенного элемента управления *OptionButton*

В окне *Properties* есть две вкладки. На вкладке *Alphabetic* свойства выбранного объекта отображаются в алфавитном порядке. На вкладке *Categorized* эти свойства сгруппированы по категориям. Обе вкладки отображают одни и те же свойства.

Для того чтобы изменить свойство, необходимо щелкнуть на нем и ввести новое значение. Некоторые свойства могут принимать только ограниченный набор допустимых значений, выбираемых из соответствующего списка. После щелчка на таком свойстве в окне *Properties* будет отображена кнопка со стрелкой, указывающей вниз. Щелкните на этой кнопке, чтобы выбрать значение из предложенного списка. Например, свойство *TextAlign* может принимать одно из следующих значений: 1 — *fmTextAlignLeft*, 2 — *fmTextAlignCenter* и 3 — *fmTextAlignRight*.

После выделения отдельных свойств (например, *Font* и *Picture*) рядом с ними отображается небольшая кнопка с троеточием. Щелчок на этой кнопке приводит к вызову диалогового окна настройки свойства.

Для свойства *Picture* элемента управления *Image* необходимо указать графический файл или вставить изображение из буфера обмена. В последнем случае следует сначала скопировать его в буфер обмена, а затем выбрать свойство *Picture* элемента управления *Image* и нажать комбинацию клавиш <Ctrl+V> для вставки содержимого буфера обмена. Если выделить два или более элементов управления одновременно, в окне *Properties* отобразятся только те свойства, которые являются общими для этих объектов.

Объекту *UserForm* присущ ряд свойств, значения которых можно изменять. Эти свойства применяются в качестве значений, заданных по умолчанию, для элементов управления, которые добавляются в пользовательские диалоговые окна. Например, если изменить свойство *Font* пользовательского диалогового окна, все добавленные в окно элементы управления будут применять этот шрифт.

**Общие свойства.** Каждый элемент управления имеет как собственный набор уникальных свойств, так и ряд общих свойств, присущих другим элементам управления. Например, все элементы управления имеют свойство *Name* и свойства, определяющие его размер и расположение на форме (*Height, Width, Left и Right*). Если вы собираетесь работать с элементом управления с помощью кода VBA, присвойте ему значащее имя. Например, первый элемент управления *OptionButton*, который добавлен в пользовательское диалоговое окно, по умолчанию получит имя *ОрtionButton1*. В коде ссылка на этот объект будет выглядеть следующим образом: OptionButton1.Value = True. Но если элементу управления *OptionButton* присвоить описательное имя (например, *obLandscape*), то можно использовать такой оператор: obLandscape.Value = True.

Многие пользователи предпочитают имена, которые указывают на тип объекта. В предыдущем примере был использован префикс *ob*, который указывает на то, что объект является элементом управления *OptionButton*.

Можно изменять значения свойств нескольких элементов управления одновременно. Например, вы вправе создать на форме несколько элементов управления *OptionButton* и выровнять их по левому краю. Для этого достаточно выделить все элементы управления *OptionButton* и изменить значение свойства *Left* в окне *Properties*. Все выделенные элементы управления примут новое значение свойства *Left*.

Чтобы получить доступ к справочной информации о свойствах различных элементов управления, щелкните на свойстве в окне *Properties* и нажмите клавишу <F1>.

**Советы по использованию клавиатуры.** Многие пользователи предпочитают перемещаться по диалоговым окнам с помощью клавиатуры. Комбинации клавиш <Таb> и <Shift+Tab> позволяют циклически переключаться между элементами управления. Чтобы удостовериться, что диалоговое окно корректно реагирует на команды с клавиатуры, обратите внимание на такие моменты: порядок просмотра элементов управления и комбинации клавиш.

Порядок просмотра определяет последовательность, в которой активизируются элементы управления после нажатия пользователем клавиши <Таb> или комбинации клавиш <Shift+Tab>. Кроме того, порядок активизации указывает, какой элемент управления по умолчанию выделяется на форме первым. Если пользователь вводит текст в элемент управления *TextBox*, то этот элемент считается активным. Если после этого щелкнуть на элементе управления *OptionButton*, то именно он станет активным. Элемент управления, назначенный первым для просмотра, будет активным в момент открытия диалогового окна.

Для того чтобы указать порядок активизации, выберите команду *View* –> *Tab Order*. Кроме того, можно щелкнуть правой кнопкой мыши на *UserForm* и выбрать пункт *Тab Order* из появившегося контекстного меню. Excel отобразит диалоговое окно *Tab Order* (*Порядок просмотра*, рис. 7).

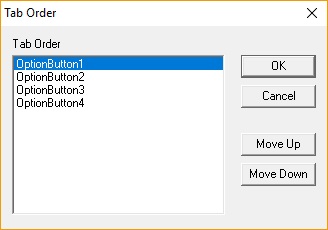


Рис. 7. В диалоговом окне *Tab Order* измените порядок просмотра элементов управления

Также можно указать порядок активизации элемента управления с помощью окна *Properties* (см. рис. 6, самое последнее из отраженных свойств). Первый активизируемый элемент управления будет иметь свойство *Tablndex* = 0. Изменение значения свойства *Tablndex* текущero объекта приведет к изменению значений свойств *Tablndex* других элементов правления. Изменения вносятся автоматически. Вы можете удостовериться в том, что значения свойства *Tablndex* всех элементов управления не больше количества элементов управления в диалоговом окне. Если нужно удалить элемент управления из списка активизируемых объектов, то присвойте его свойству *TabStop* значение *False*.

Одни элементы управления, такие как *Frame* и *MultiPage*, служат контейнерами для других элементов управления. Элементы управления в таком контейнере имеют собственный порядок просмотра (активизации). Для установки порядка просмотра группы элементов управления *OptionButtons*, находящихся внутри элемента управления *Frame*, выделите элемент управления *Frame* до того, как будет выполнена команда *View* –> *Tab Order*.

Большинству элементов управления диалогового окна можно назначить комбинацию клавиш. Таким образом, пользователь получит доступ к элементу управления, нажав <Alt> и указанную клавишу. Применив свойство *Accelerator* в окне *Properties*, можно определить клавишу для активизации элемента управления.

Некоторые элементы управления, например, *TextBox*, лишены свойства *Accelerator*, поскольку не отображают значение свойства *Caption*. Но к таким элементам можно получить доступ с помощью клавиатуры, воспользовавшись свойством *Label*. Присвойте клавишу элементу управления *Label* и расположите его в порядке просмотра перед элементом *TextBox*.

### Отображение пользовательского диалогового окна

Для того чтобы отобразить пользовательское диалоговое окно с помощью VBA, необходимо создать процедуру, которая вызывает метод *Show* объекта *UserForm*. Форму *UserForm* невозможно отобразить, не выполнив как минимум одну строку кода VBA:

Sub ShowForm()

UserForm1.Show

End Sub

Данная процедура должна располагаться в стандартном модуле VBA, а не в модуле формы *UserForm*. При отображении пользовательская форма остается на экране до тех пор, пока ее не скроют. Обычно в нее добавляют элемент управления CommandButton, который запускает процедуру закрытия формы. Эта процедура либо выгружает пользовательскую форму с помощью метода Unload, либо удаляет ее с экрана с помощью метода Hide объекта UserForm.

**Отображение немодальной формы.** По умолчанию отображается модальная форма. Это означает, что форма должна исчезнуть с экрана прежде, чем пользователь выполнит какие-либо действия на рабочем листе (т.е. редактирование данных невозможно). Немодальную форму также можно отобразить. В этом случае вы вправе продолжать работу в Excel, не скрывая саму форму. Для отображения немодальной формы используется следующий синтаксис:

UserForm1.Show vbModeless

**Тестирование пользовательского диалогового окна.** Обычно в процессе разработки возникает необходимость в тестировании формы UserForm. Для этого можно воспользоваться одним из способов:

* выполните команду *Run* –> *Run Sub/UserForm*;
* нажмите <F5>;
* щелкните на кнопке *Run Sub/UserForm*, которая находится на стандартной панели инструментов.

Выбор одного из этих методов приводит к запуску события инициализации диалогового окна. Как только диалоговое окно будет отображено в тестовом режиме, можно проверить порядок активизации объектов, а также поддержку комбинаций клавиш.

**Отображение пользовательского диалогового окна на основе значения переменной.** В некоторых случаях приходится выбирать, какое окно UserForm будет отображено. Если название пользовательского диалогового окна хранится в виде строковой переменной, можно воспользоваться методом Add для добавления объекта UserForm в коллекцию UserForms с последующим обращением к методу Show из коллекции UserForms. В приведенном ниже примере название объекта UserForm присваивается переменной MyForm, после чего отображается пользовательское диалоговое окно.

MyForm = "UserForm1"

UserForms.Add(MyForm).Show

**Загрузка пользовательского диалогового окна.** В VBA поддерживается оператор Load. Загрузка пользовательского диалогового окна приводит к сохранению объекта формы в памяти. Однако до тех пор пока не будет выполнен метод Show, форма останется невидимой для остальной части программы. Для загрузки окна UserForm1 воспользуйтесь оператором:

Load UserForm1

Если вы применяете сложное диалоговое окно, то вам может понадобиться предварительно загрузить его в память, чтобы в случае необходимости быстро отобразить его с помощью метода Show. Как правило, использовать метод Load в подобном случае не имеет смысла.

**О процедурах обработки событий.** Как только диалоговое окно появляется на экране, пользователь начинает с ним взаимодействовать, т.е. генерирует события. Например, щелчок на элементе управления CommandButton приводит к возникновению события Click объекта CommandButton. Вам необходимо создать процедуры (обработчики событий), которые будут выполняться при возникновении соответствующих событий.

Процедуры обработки событий находятся в модуле кода объекта UserForm. Наряду с этим процедура обработки события может вызывать другие процедуры, которые находятся в стандартном модуле VBA.

В коде VBA можно изменять свойства элементов управления, пока пользовательское диалоговое окно отображается на экране (т.е. на этапе выполнения). Например, можно назначить элементу управления ListBox процедуру, которая изменяет текст элемента управления Label при выборе элемента списка.

### Закрытие пользовательского диалогового окна

Для закрытия формы UserForm1 воспользуйтесь командой: Unload UserForm1. Если же код находится в модуле кода формы UserForm, воспользуйтесь оператором: Unload Me. В этом случае ключевое слово Me применяется для идентификации пользовательской формы.

Обычно в коде VBA команда Unload выполняется только после того, как форма UserForm выполнит все свои функции. Например, форма UserForm может содержать элемент управления CommandButton, который используется в качестве кнопки ОК. Щелчок на этой кнопке приводит к выполнению заранее определенного макроса. Одна из функций макроса заключается в выгрузке формы UserForm из памяти. В результате пользовательское диалоговое окно отображается на экране до тех пор, пока макрос, содержащий оператор Unload, не завершает свою работу.

Когда форма UserForm выгружается из памяти, элементы управления, содержавшиеся на ней, возвращаются в первоначальное состояние. Другими словами, в коде нельзя обращаться к значениям, указываемым пользователем, после того как форма будет выгружена из памяти. Если значения, введенные пользователем, будут применяться позже (после выгрузки диалогового окна UserForm), то необходимо присвоить их переменной с областью действия Public, которая определена в стандартном модуле VBA. Кроме того, значение всегда можно сохранить в ячейке листа.

Окно формы UserForm автоматически выгружается из памяти после того, как пользователь щелкает на кнопке *Закрыть* (обозначается символом х в заголовке окна). Это действие также приводит к возникновению события QueryClose объекта UserForm, за которым следует событие Terminate пользовательского диалогового окна. Объект UserForm может использовать метод Hide. После его вызова диалоговое окно исчезает, но остается в памяти, поэтому в коде можно получить доступ к различным свойствам элементов управления:

UserForml.Hide

Если ваш код находится в модуле кода объекта UserForm, можно воспользоваться оператором:

Me.Hide

Если по какой-либо причине пользовательское диалоговое окно должно быть немедленно скрыто в процессе выполнения макроса, воспользуйтесь методом Hide в самом начале процедуры, а затем укажите команду DoEvents. Например, в следующей процедуре форма UserForm немедленно исчезнет после того, как пользователь щелкнет на кнопке CommandButton1. Последний оператор процедуры выгружает пользовательское диалоговое окно из памяти.

Private Sub CommandButton1\_Click()

Me.Hide

Application.ScreenUpdating = True

For r = 1 To 10000

Cells(r, 1) = r

Next r

Unload Me

End Sub

В рассматриваемом примере переменной ScreenUpdating присвоено значение True, в результате чего Excel полностью скрывает окно UserForm. Если этот оператор не использовать, окно UserForm остается видимым.

### Пример создания пользовательского диалогового окна

В примере представлено диалоговое окно, предназначенное для получения имени и пола пользователя. В диалоговом окне вы найдете элемент управления TextBox, используемый для ввода имени, и два элемента управления OptionButton для указания пола. Информация, полученная в диалоговом окне, заносится в пустую строку рабочего листа.

**Создание пользовательской формы.** Создайте рабочую книгу, содержащую только один рабочий лист. Нажмите комбинацию клавиш <Alt+F11> для активизации VBE. В окне *Project* выберите проект рабочей книги и выполните команду *Inserts* –> *UserForm*. Воспользуйтесь окном *Properties* для изменения значения свойства *Caption* формы *UserForm* на *Укажите имя и пол* (если окно *Properties* не отображается, нажмите <F4>).

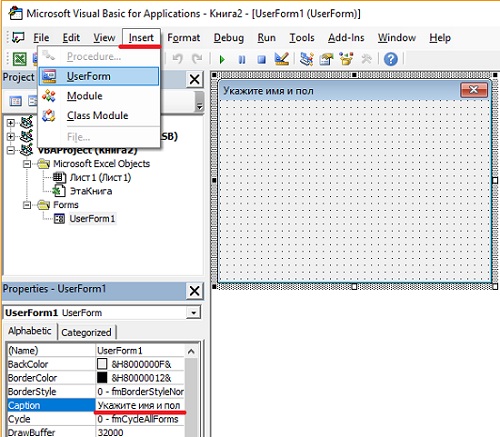


Рис. 8. Создание пользовательской формы

Добавьте элемент управления Label и настройте его свойства (рис. 9):

*Свойство Значение*

Accelerator И

Caption Имя:

Tablndex 0

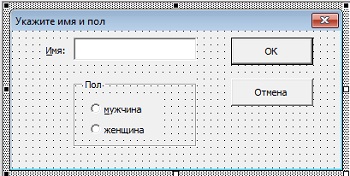


Рис. 9. Пользовательская форма с элементами управления

Добавьте элемент управления TextBox и измените его свойства:

Name TextName

Tablndex 1

Добавьте элемент управления Frame и измените его свойства:

Caption Пол

Tablndex 2

Добавьте элемент управления OptionButton в состав элемента Frame и измените его свойства:

Accelerator М

Caption Мужчина

Name OptionMale

Tablndex 0

Добавьте еще один элемент управления OptionButton в состав элемента Frame и измените его свойства:

Accelerator Ж

Caption Женщина

Name OptionFemale

Tablndex 1

Добавьте элемент управления CommandButton за пределами элемента Frame и измените его свойства:

Caption OK

Default True

Name OKButton

Tablndex 3

Добавьте еще один элемент управления CommandButton и настройте его свойства:

Caption Отмена

Default False

Name CloseKButton

Tablndex 4

При создании нескольких похожих элементов управления может оказаться, что быстрее копировать существующий элемент управления, чем создавать новый. Для копирования элемента управления удерживайте нажатой клавишу <Ctrl> при перетаскивании элемента управления, что приведет к созданию копии. После этого останется изменить свойства скопированного элемента управления.

**Создание кода для отображения диалогового окна.** После создания элементов управления на лист необходимо добавить элемент управления *ActiveX*, называемый *Кнопка* (*CommandButton*). Эта кнопка будет запускать процедуру, которая предназначена для отображения формы *UserForm*. Для этого выполните следующие действия:

1. Перейдите в окно Excel (например, воспользуйтесь комбинацией клавиш <Alt+F11>).
2. Выберите команду *Разработчика* –> *Элементы управления* –> *Вставить* и щелкните на значке *Кнопка* (*CommandButton*), который находится в разделе *Элементы ActiveX*.
3. Разместите кнопку на рабочем листе. Чтобы измените ее подпись, щелкните на кнопке правой кнопкой мыши и выберите в контекстном меню команду *Объект CommandButton* –> *Edit*. Для изменения других свойств объекта щелкните на нем правой кнопкой мыши и выберите команду *Properties*.
4. Дважды щелкните на объекте *CommandButton*. Это приведет к активизации VBE. При этом отобразится модуль кода для листа с открытой пустой процедурой обработки событий объекта *CommandButton*, который расположен на рабочем листе.
5. Введите единственный оператор в процедуру CommandButton1\_Click (рис. 10). В процедуре используется метод Show объекта UserForml для отображения пользовательского диалогового окна.

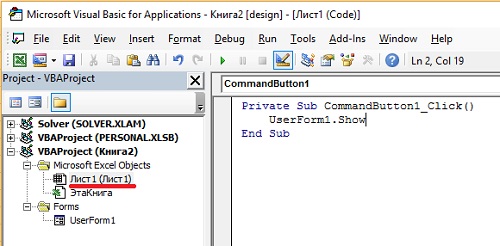


Рис. 10. Процедура *CommandButton1\_Click* вызывается после щелчка на кнопке рабочего листа

**Тестирование диалогового окна.** После щелчка на кнопке, находящейся на рабочем листе, ничего не произойдет. Точнее, кнопка будет выделена, но это не приведет к инициализации каких-либо действий. Причина этого заключается в том, что программа Excel по-прежнему остается в режиме конструктора, в который она переходит автоматически после добавления элемента управления ActiveX. Для выхода из режима конструктора щелкните на кнопке *Разработчик* –> *Элементы* *управления* –> *Режим конструктора*. Если же требуется изменить кнопку, снова перейдите в режим конструктора. После выхода из режима конструктора щелчок на кнопке приведет к отображению пользовательского диалогового окна.

Когда диалоговое окно будет отображено, введите произвольный текст в текстовом поле и щелкните на кнопке ОК. В результате ничего не произойдет, что совершенно естественно, так как для объекта UserForm не создано ни одной процедуры обработки событий. Для закрытия диалогового окна щелкните на крестике в его заголовке.

**Добавление процедур обработки событий.** Перейдите в VBE. Удостоверьтесь в том, что пользовательское окно отображено на экране, и дважды щелкните на кнопке *Отмена*. Активизируется окно кода для формы *UserForm*, а также добавляется пустая процедура CloseButton\_Click. Обратите внимание, что название процедуры состоит из имени объекта, символа подчеркивания и названия обрабатываемого события. Добавьте в процедуру единственный оператор:

Private Sub CloseButton\_Click()

Unload UserForm1

End Sub

Эта процедура, которая вызывается после щелчка на кнопке Отмена, выгружает из памяти форму UserForm1.

Щелкните на значке *View Object* в верхней части окна *Project Explorer* или дважды кликните на строке *UserForm1*, чтобы отобразить форму UserForm1. Дважды щелкните на кнопке *ОК* и введите код процедуры – обработчика событий для события Click объекта OKButton:

Private Sub OKButton\_Click()

Dim NextRow As Long

' Проверка активности листа Лист1

Sheets("Лист1").Activate

' Проверка ввода имени

If TextName.Text = "" Then

MsgBox "Введите имя."

TextName.SetFocus

Exit Sub

End If

' Определение следующей пустой строки

NextRow = \_

Application.WorksheetFunction.CountA(Range("A:A")) + 1

' Передача имени

Cells(NextRow, 1) = TextName.Text

' Передача пола

If OptionMale Then Cells(NextRow, 2) = "Мужчина"

If OptionFemale Then Cells(NextRow, 2) = "Женщина"

' Очистка элементов управления для ввода следующей записи

TextName.Text = ""

TextName.SetFocus

OptionMale = False

OptionFemale = False

End Sub

Перейдите в Excel и щелкните на кнопке еще раз, чтобы отобразить пользовательское диалоговое окно. С его помощью можно добавлять новые имена в рабочий лист.

Процедура OKButton\_Click работает следующим образом. Сначала она проверяет, активен ли лист Лист1, введено ли имя. После этого запускается функция Excel СЧЁТЗ (CountA) для определения следующей пустой ячейки в столбце А. Затем текст из текстового поля TextBox передается в определенную ячейку столбца А. С помощью операторов If определяется выделенный элемент управления OptionButton, что обеспечивает запись соответствующего текста в столбец В (пол). Далее элементы диалогового окна очищаются и окно перезапускается, чтобы обеспечить возможность введения следующей записи. Заметим, что щелчок на кнопке *ОК* не приведет к закрытию диалогового окна. Для завершения ввода данных (и выгрузки пользовательского диалогового окна) щелкните на кнопке *Отмена*.

Проверьте работоспособность комбинаций клавиш: Alt+М – активизирует мужской пол, Alt+Ж – женский. Рабочая книга с рассмотренным примером находится в файле *get name and sex.xlsm*.

### События объекта UserForm

Каждый элемент управления в форме UserForm (а также сам объект UserForm) разрабатывается для того, чтобы реагировать на определенные события. Эти события возникают в результате действий пользователя или генерируются программой Excel. Можно создать код, который будет выполняться при возникновении определенного события. Некоторые действия приводят к возникновению сразу нескольких событий. Например, щелчок на кнопке со стрелкой, направленной вверх, в элементе управления *SpinButton* приведет к возникновению события *SpinUp* и события *Change*. После того как пользовательское диалоговое окно будет отображено с помощью метода *Show*, Excel сгенерирует события *Initialize* и *Activate* объекта *UserForm*. В Excel также поддерживаются события, связанные с объектами Sheet (Лист), Chart (Диаграмма) и ThisWorkbook (ЭтаКнига).

**Получение дополнительных сведений о событиях.** Для того чтобы получить информацию о событиях, которые генерируются конкретным элементом управления, выполните следующее:

* Добавьте элемент управления в пользовательское диалоговое окно.
* Дважды щелкните на элементе управления, чтобы открыть модуль кода для объекта *UserForm*. VBE вставит пустую процедуру обработки события, принятого по умолчанию.
* Щелкните на раскрывающемся списке в правом верхнем углу окна модуля и просмотрите полный список событий, которые поддерживаются текущим элементом управления (на рис. 11 показан список событий для элемента управления *CheckBox*).
* Выберите событие из списка, и VBE создаст пустой обработчик события.

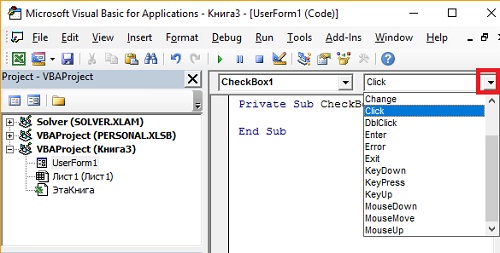


Рис. 11. Список событий для элемента управления *CheckBox*

Имя процедуры обработки событий включает имя объекта, который сгенерировал событие. Таким образом, если изменить имя элемента управления, придется внести соответствующие изменения и в имя процедуры обработки события. Имя процедуры не изменяется автоматически! Чтобы облегчить работу, присвойте описательные имена элементам управления до того, как приступите к созданию процедуры обработки соответствующих событий.

**События объекта UserForm.** Несколько событий непосредственно связано с отображением и выгрузкой объекта UserForm:

* Initialize. Происходит перед загрузкой и отображением формы UserForm. Не происходит, если объект UserForm до этого был скрыт.
* Activate. Происходит в момент активизации объекта UserForm.
* Deactivate. Происходит в момент деактивизации объекта UserForm. Не происходит при сокрытии формы UserForm.
* QueryClose. Происходит перед выгрузкой объекта UserForm.
* Terminate. Происходит после выгрузки объекта UserForm.

Важно правильно выбрать подходящее событие для процедуры обработки событий, а также проанализировать порядок выполнения событий. Использование метода *Show* приводит к возникновению событий *Initialize* и *Activate* (в указанном порядке). Применение команды *Load* приводит к вызову события *Initialize*. Команда *Unload* вызывает события *QueryClose* и *Terminate* (в указанном порядке). Метод *Hide* не приводит к вызову каких-либо событий.

В файле *userform events.xlsm* описанные события перехватываются и в момент возникновения события возникает окно сообщения. Если изучение событий объекта *UserForm* вызывает у вас затруднения, то, проанализировав код этого примера, вы получите ответы на многие вопросы.

**События элемента управления SpinButton.** Для того чтобы разобраться в концепции событий, мы подробно рассмотрим события, связанные с элементом управления *SpinButton* (рис. 12). Файл *spinbutton events.xlsm* демонстрирует применение событий, генерируемых объектами *SpinButton* и *UserForm* (первый содержится во втором). Рабочая книга включает несколько процедур обработки событий — по одной для каждого события элемента управления *SpinButton* и объекта *UserForm*. Каждая из этих процедур добавляет текст в столбце А с описанием события.

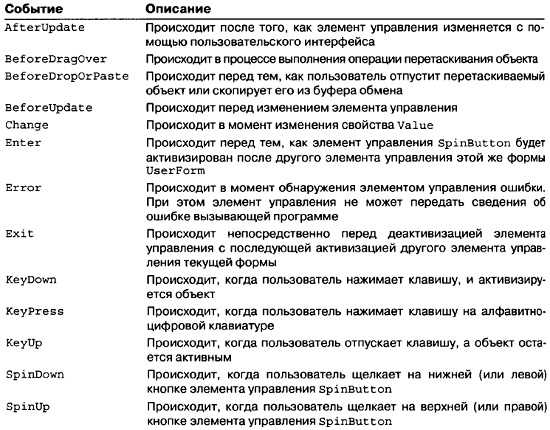


Рис. 12. События элемента управления *SpinButton*

Пользователь может управлять объектом *SpinButton* с помощью мыши или (если элемент управления активен) клавиш управления курсором. Когда пользователь щелкает мышью на верхней кнопке элемента управления *SpinButton*, происходят следующие события: *Enter* (генерируется только в том случае, если элемент управления неактивен); *Change*; *SpinUp*.

Пользователь может нажать клавишу <Таb> для того, чтобы сделать активным элемент управления *SpinButton*. Только после этого можно использовать клавиши управления курсором для изменения значения элемента управления. Если все именно так и происходит, то события генерируются в следующем порядке: *Enter*; *KeyDown*; *Change*; *SpinUp* (или *SpinDown*); *KeyUp*.

Элемент управления *SpinButton* может изменяться в коде VBA, что также провоцирует возникновение соответствующих событий. Например, оператор SpinButton1.Value = 0 устанавливает свойство *Value* элемента управления *SpinButton1* равным 0, а это приводит к возникновению события *Change*. Такой результат достигается только в том случае, если исходное свойство *Value* не равно нулю.

Вы вправе предположить, что выполнить отмену генерирования событий можно, установив свойство *EnableEvents* объекта *Application* равным *False*. Но это свойство поддерживается только объектами, которые являются «истинными» в Excel: *Workbook*, *Worksheet* и *Chart*.

**Совместное использование элементов управления SpinButton и TextBox.** Элемент управления *SpinButton* имеет свойство *Value*, но не может отображать значение этого свойства. В большинстве случаев требуется, чтобы пользователь мог изменить значение элемента управления *SpinButton* непосредственно, а не многократно щелкая на элементе управления. Эффективным решением может стать объединение элемента управления *SpinButton* с элементом управления *TextBox*, что позволяет пользователю вводить значение элемента управления *SpinButton* непосредственно, используя для этого поле элемента управления *TextBox*. Кроме того, щелчок на элементе управления *SpinButton* позволит изменить значение, отображаемое в элементе управления *TextBox*.

На рис. 13 приведен пример (см. также файл *spinbutton and textbox.xlsm*). Свойство *Min* элемента управления *SpinButton* имеет значение 1, а свойство *Мах*— значение 100. Таким образом, щелчок на одной из стрелок элемента управления *SpinButton* приведет к изменению значения в пределах от 1 до 100. Код, реализующий «связывание» элементов управления *SpinButton* и *TextBox* сводится к созданию процедур обработки событий, которые будут синхронизировать свойство *Value* элемента управления *SpinButton* и свойство *Text* элемента управления *TextBox*.

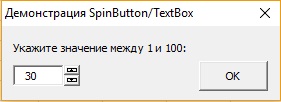


Рис. 13. Комбинирование элементов управления *SpinButton* и *TextBox*

Представленная процедура выполняется каждый раз при возникновении события *Change* элемента управления *SpinButton*. Таким образом, процедура выполняется тогда, когда пользователь щелкает на элементе управления *SpinButton* или изменяет его значение, нажав одну из клавиш управления курсором.

Private Sub SpinButton1\_Change()

TextBox1.Text = SpinButton1.Value

End Sub

Если пользователь введет значение непосредственно в элемент управления *TextBox*, то будет сгенерировано событие *Change*, после чего должен выполняться следующий код:

Private Sub TextBox1\_Change()

Dim NewVal As Integer

NewVal = Val(TextBox1.Text)

If NewVal >= SpinButton1.Min And \_

NewVal <= SpinButton1.Max Then \_

SpinButton1.Value = NewVal

End Sub

Эта процедура начинается с вызова функции VBA *Val*, которая преобразует текст элемента управления *TextBox* в числовое значение (если элемент управления *TextBox* содержит строку, то функция *Val* возвращает значение 0). Следующий оператор определяет, попадает ли значение в указанный диапазон допустимых значений. Если это так, то свойство *Value* элемента управления *SpinButton* устанавливается равным значению, которое введено в поле элемента управления *TextBox*.

Пример организован таким образом, что щелчок на кнопке *ОК* (которая называется *OKButton*) передает значение элемента управления *SpinButton* в активную ячейку. Процедура обработки события *Click* элемента управления *CommandButton* выглядит следующим образом:

Private Sub OKButton\_Click()

' Enter the value into the active cell

If CStr(SpinButton1.Value) = TextBox1.Text Then

ActiveCell = SpinButton1.Value

Unload Me

Else

MsgBox "Некорректная запись.", vbCritical

TextBox1.SetFocus

TextBox1.SelStart = 0

TextBox1.SelLength = Len(TextBox1.Text)

End If

End Sub

Данная процедура выполняет проверку: анализируются текст, введенный в поле элемента управления *TextBox*, и значения элемента управления *SpinButton*. Такая процедура обрабатывает ситуации неверного ввода данных. Например, если пользователь введет в поле элемента управления *TextBox* текст *Зt*, то значение элемента управления *SpinButton* не изменится, а результат, который помещается в активную ячейку, будет отличным от ожидаемого. Обратите внимание, что значение свойства *Value* элемента управления *SpinButton* преобразуется в строку с помощью функции *CStr*. Это позволяет предотвратить ошибку, которая возникает, когда числовое значение сравнивается с текстовым. Если значение элемента управления *SpinButton* не соответствует содержимому элемента управления *TextBox*, то на экране отображается специальное сообщение. Причем объект *TextBox* активен, а его содержимое — выделено (с помощью свойств *SelStart* и *SelLength*). Таким образом, пользователю проще исправить неправильные значения.

**О свойстве Tag.** Каждый объект *UserForm* и каждый элемент управления имеет свойство *Tag*. Оно не представляет конечные данные и по умолчанию не имеет значения. Свойство *Tag* можно использовать для хранения информации, которая будет применена в программе. Например, можно создать набор элементов управления *TextBox* в пользовательском диалоговом окне. От пользователя требуется ввести текст только в некоторые из них. В отдельные поля вводить текст необязательно. Можно применять свойство *Tag* для идентификации полей, которые нужно заполнять. В таком случае значение свойства *Tag* — это строка, например, *Required*. Поэтому при написании кода обработки введенных пользователем данных можно ссылаться на свойство *Tag*.

Приведенный ниже пример представляет собой функцию, которая проверяет все элементы управления *TextBox* объекта *UserForm1* и возвращает количество пустых текстовых полей, которые *требуют* ввода информации.

Function EmptyCount()

Dim ctl As Control

EmptyCount= 0

For Each ctl In UserFormi.Controls

If TypeName(ctl) = "TextBox" Then

If ctl.Tag = "Required" Then

If ctl.Text = "" Then

EmptyCount = EmptyCount + 1

End If

End If

End If

Next ctl

End Function

### Ссылка на элементы управления пользовательского диалогового окна

При работе с элементами управления, находящимися в форме *UserForm*, код VBA обычно содержится в модуле кода объекта *UserForm*. Кроме того, на элементы управления диалогового окна можно ссылаться из модуля кода VBA общего назначения. Для выполнения этой задачи необходимо задать правильную ссылку на элемент управления, указав имя объекта *UserForm*. В качестве примера рассмотрим процедуру, которая введена в модуле кода VBA. Эта процедура отображает пользовательское диалоговое окно, которое называется *UserForm1*.

Sub GetData ()

UserForm1.Show

End Sub

Предположим, что в диалоговом окне *UserForm1* содержится текстовое поле *TextBox1* и вам необходимо указать значение текстового поля по умолчанию:

Sub GetData()

UserForm1.TextBox1.Value = "Джон Доу"

UserForm1.Show

End Sub

Еще одним способом установки значения по умолчанию является использование события *Initialize* объекта *UserForm*. Можно написать код процедуры *UserForm\_Initialize*, который будет располагаться в модуле кода диалогового окна:

Private Sub UserForm\_Initialize()

TextBox1.Value = "Джон Доу"

End Sub

Обратите внимание, что при обращении к элементу управления из модуля кода диалогового окна необязательно вводить в ссылку имя объекта *UserForm*. Подобное определение ссылок на элементы управления имеет свое преимущество: всегда можно воспользоваться средством Auto List Member, которое позволяет выбирать имена элементов управления из раскрывающегося списка.

Вместо того чтобы использовать фактическое имя объекта *UserForm*, предпочтительнее применить имя *Me*. В противном случае, если имя объекта *UserForm* изменится, вам придется изменять все ссылки (с его участием) в коде.

**Использование коллекций элементов управления.** Элементы управления пользовательских диалоговых окон образуют отдельную коллекцию. Например, следующий оператор отображает количество элементов управления в форме UserForm1:

MsgBox UserForm1.Controls.Count

В VBA не поддерживаются коллекции для каждого типа элемента управления. Например, не существует коллекции элементов управления *CommandButton*. Но тип элемента управления можно определить с помощью функции *TypeName*. Следующая процедура использует структуру For Each для циклического просмотра элементов коллекции *Controls*. В результате отображается количество элементов управления *CommandButton*, которые входят в коллекцию элементов управления объекта *UserForm1* (этот код вы найдете в файле *all userform controls.xlsm*).

Sub CountButtons()

Dim cbCount As Integer

Dim ctl As Control

cbCount = 0

For Each ctl In UserForm1.Controls

If TypeName(ctl) = "CommandButton" Then \_

cbCount = cbCount + 1

Next ctl

MsgBox cbCount

End Sub

### Настройка панели инструментов Toolbox

Если объект *UserForm* активен в редакторе VBE, на панели *Toolbox* отображаются элементы управления, которые можно добавить в пользовательское диалоговое окно. Панель *Toolbox* изначально содержит одну вкладку. Щелкните на ней правой кнопкой мыши и в контекстном меню выберите параметр *New Page* (Добавить страницу). Кроме того, можно изменить текст, который отображается на вкладке. Для этого выберите параметр *Rename* (Переименовать) из контекстного меню.

Рекомендуется предварительно настроить элементы управления и сохранить их для дальнейшего использования. Можно, например, в форме создать элемент управления *CommandButton*, который настроен на выполнение роли кнопки ОК. Можно изменять параметры кнопки: Width (Ширина), Height (Высота), Caption (Подпись), Default (По умолчанию) и Name (Имя). После этого перетащите модифицированный элемент управления *CommandButton* на панель инструментов *Toolbox*. Это приведет к созданию элемента управления. Щелкните на элементе управления правой кнопкой мыши, чтобы переименовать его или изменить значок.

Также можно создать раздел панели *Toolbox*, в котором будет содержаться несколько элементов управления. Например, вы вправе создать два элемента управления *CommandButton*, которые будут представлять кнопки *ОК* и *Отмена*. Настройте их так, как это необходимо. Затем выберите обе кнопки и переместите их на панель инструментов *Toolbox*. Впоследствии можно использовать новый элемент управления панели *Toolbox* для быстрого создания необходимых кнопок.

Этот метод также применим к элементам управления, которые используются в качестве контейнера. Например, создайте элемент управления *Frame* и добавьте в него четыре модифицированных элемента управления *OptionButton* (соответствующим образом расположив их на форме). После этого перетащите элемент управления *Frame* на панель инструментов *Toolbox*, чтобы создать модифицированный элемент управления *Frame*.

Можно разместить модифицированные элементы управления на отдельной вкладке панели *Toolbox*. Таким образом, появляется возможность экспортировать вкладку панели *Toolbox* для совместного применения другими пользователями Excel. Для экспорта вкладки панели *Toolbox* щелкните на ней правой кнопкой мыши и выберите пункт меню *Export Page*.

Среди прилагаемых к заметке файлов находится страничный файл под именем *newcontrols.pag*, который включает некоторые настроенные элементы управления. Можно импортировать этот файл в качестве новой вкладки окна *Toolbox*. Щелкните правой кнопкой мыши на вкладке и выберите команду *Import Page*. В результате панель *Toolbox* будет как на рис. 14.

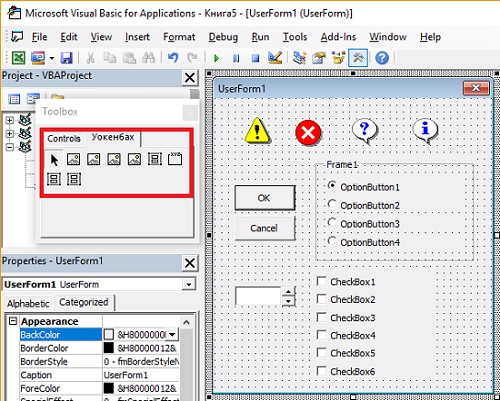


Рис. 14. В окне *Toolbox* появилась страница с новыми элементами управления

### Добавление элементов управления ActiveX

В пользовательском диалоговом окне содержатся и другие элементы управления ActiveX, разработанные компанией Microsoft и независимыми производителями. Для того чтобы добавить дополнительные элементы управления ActiveX на панель инструментов *Toolbox*, щелкните правой кнопкой мыши на ней и выберите пункт *Additional Controls* (Дополнительные элементы управления). В результате будет отображено диалоговое окно, показанное на рис. 15.

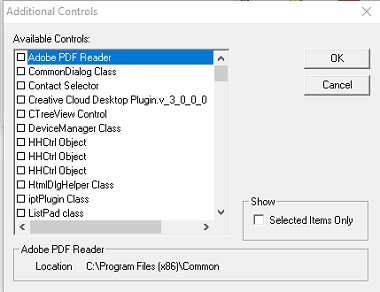


Рис. 15. В диалоговом окне *Additional Controls* можно найти дополнительные элементы управления ActiveX

В диалоговом окне *Additional Controls* содержатся все элементы управления ActiveX, установленные в системе. Выберите элементы управления, которые необходимо добавить на панель инструментов. После этого щелкните на кнопке *ОК* для добавления значков каждого из выбранных элементов управления. Не все элементы управления ActiveX, установленные в системе, поддерживаются пользовательскими диалоговыми окнами. Более того, большая их часть не поддерживается, к тому же некоторые элементы управления требуют лицензии на использование в приложениях. Если лицензия отсутствует, на экране появится сообщение об ошибке.

### Создание шаблонов диалоговых окон

Зачастую при создании пользовательского диалогового окна каждый раз на форму добавляются одни и те же элементы управления. Например, все пользовательские диалоговые окна имеют два элемента управления *CommandButton*, используемых в качестве кнопок *ОК* и *Отмена*. В предыдущем разделе рассматривались методы комбинирования элементов управления с целью получения одного элемента управления, обладающего функциями двух. Еще одной программной уловкой может служить шаблон диалогового окна, который при необходимости импортируется для последующего создания на его основе других проектов. Преимущество шаблонного подхода заключается в следующем: процедуры обработки событий сохраняются вместе с шаблоном.

Начните с создания пользовательского диалогового окна, содержащего все элементы управления и настройки, которые необходимо повторно использовать в других проектах. После этого убедитесь, что диалоговое окно выделено. Выберите команду *File* –> *Export File* (или нажмите комбинацию клавиш <Ctrl+E>). После этого на экране появится запрос на ввод имени файла. Затем для создания проекта на основе шаблона выполните команду *File* –> *lmport File*, чтобы загрузить ранее сохраненное диалоговое окно.

**Имитация диалоговых окон Excel.** Внешний вид и поведение диалоговых окон Windows изменяются от программы к программе. При разработке приложений для Excel рекомендуется придерживаться стиля диалоговых окон Excel. Наилучшим методом изучения эффективных способов создания диалоговых окон является повторное создание одного из стандартных диалоговых окон Excel. Например, удостоверьтесь, что вы правильно определили комбинации клавиш и активизировали элементы управления. Для создания копии одного диалогового окна Excel следует протестировать его в различных условиях. Один только анализ диалоговых окон Excel поможет улучшить познания в вопросах структуры окон и методов создания элементов управления. Со временем вы убедитесь, что невозможно повторить отдельные диалоговые окна Excel даже с помощью VBA.

1. По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](http://baguzin.ru/wp/?p=16255). – М: Диалектика, 2013. – С. 405–438. [↑](#footnote-ref-1)