# Визуальное программирование

## Глава 6. Форма и элементы управления

##### Редактор Visual Basic

Редактор Visual Basic является отдельной программой, которая была разработана с единственной целью – помочь нам в создании и редактировании макросов VBA.

Для просмотра, отладки, редактирования и явного написания макросов перейдите в редактор VBA (Сервис🡪 Макрос🡪 Редактор).

В левой части редактора содержится окно Project (Проект). Если окно закрыто, то его можно открыть из меню View🡪 Project Explorer.

В окне Project отображается содержимое шаблона Normal, т.е. общего шаблона, а также проекты всех открытых в данный момент документов. Если говорить просто, проектом называется файл любого приложения Office и все связанные с ним элементы VBA, включая макросы и пользовательские формы.

Если в документе уже есть макросы, то в редакторе Visual Basic сначала, как правило, нужно открыть модуль – элемент VBA, который содержит макросы.

На рисунке показано окно редактора, в котором видим:

1. окно Project;
2. окно кода для документа Словарь с Макросом 2;
3. пользовательскую форму UserForm1;
4. Toolbox – панель элементов управления;
5. Properties – окно свойств.

Чтобы открыть модуль, выполните следующие действия:

1. В окне Project выберите проект нужного документа, раскройте его. Для этого щелкните на знаке «плюс (+)», который расположен слева от этой папки. Затем аналогичным образом и раскройте папку Modules (Модули).
2. Дважды щелкните на NewMacros. Откроется окно кодов (Code) справа. Оно содержит записанные с макрорекодера макросы.

Замечание. В главе 1 предлагалось открывать макросы другим путём.

##### Форма и элементы управления

Любое окно, которое мы видим, работая в Windows, является с точки зрения программиста VB и VBA формой. Форма предназначена для организации удобного диалога с помощью элементов управления. Форму можно считать главным элементом приложения, поскольку все остальные элементы помещаются в неё как в контейнер.

Кнопки, поля ввода и другие элементы расположены на Панели элементов (Toolbox). После щелчка на пиктограмме требуемого элемента управления (Control) курсор принимает форму крестика. Теперь нужно отметить в форме начальный угол элемента управления, нажав левую кнопку мыши, растянуть элемент до нужного размера и отпустить кнопку.

Название элемента вы узнаете по всплывающей подсказке (для этого задержите курсор на элементе). Имена элементов по умолчанию формируются из названия элемента и порядкового номера, например: Label1, Label2 и т. д.

В таблице1 приведены названия основных элементов на русском и на английском языках.

Таблица 2

|  |  |
| --- | --- |
| Название | Название |
| Label | Надпись | TextBox | Текстовое окно |
| Frame  | Рамка | CommandButton | Командная кнопка |
| CheckBox | Флажок | OptionButton | Переключатели |
| ListBox | Список | ComboBox  | Раскрывающийся список  |
| ScroolBar | Полоса прокрутки | Image | Изображение |

##### Свойства элементов управления

Каждый элемент управления имеет свойства. Большинство из них отвечают за внешний вид и расположение элемента на форме. При конструировании свойства выделенного в данный момент элемента задаются в окне Properties. Самый простой способ выбрать элемент – щелчок на нём в окне формы.

 Свойства элемента могут быть расположены в алфавитном порядке, либо по категориям. Набор свойств зависит от типа элемента управления. Например, у формы есть свойство Picture (рисунок), у элемента ComboBox (раскрывающийся список) есть свойство, определяющее число колонок в нём. Список свойств состоит из двух столбцов: в левом перечислены названия свойств, а в правом – их значения. Значения одних свойств могут быть произвольными и вводятся с клавиатуры (например, свойство Caption), значения других фиксированы и выбираются из списка значений (например, True или False для свойства Visible), третьи устанавливаются из диалогового окна (например, Font). Как обычно в приложениях Office, многоточие в строке со свойством открывает диалоговое окно.

У большинства элементов свойств много. Рассмотрим и обсудим свойства, которые нам обязательно понадобятся (табл. 3 – 4).

Как видно из рисунка, свойства Top и Left задают координаты верхнего левого угла элемента управления, свойства Height и Width – его высоту и ширину. Отсчет в системе координат ведется сверху вниз (Y) и слева направо (X). Эти свойства определяются автоматически, когда мы устанавливаем или изменяем размеры и положение элемента с помощью мыши. Но можно задать их более точно в окне свойств.

Top

Left

Height

Width

Свойство Name служит для обращения к элементу.

Часто при работе приложения требуется сделать недоступными для пользователя некоторые элементы управления. Для этого используют два свойства – Enabled и Visible. Свойство Enabled определяет, будет элемент управления реагировать на событие или нет. Если значение свойства равно False, элемент управления будет недоступен, и пользователь не cможет его использовать. Обычно при этом элемент подсвечивается серым цветом. Свойство Visible позволяет сделать элемент управления невидимым. Если его значение равно False, то он не виден.

Свойство Text элемента TextBox аналогично свойству Caption для других элементов. Если предполагается ввод текста или чисел пользователем во время работы проекта, то используется элемент TextBox. Введённые данные сохраняются в свойстве Text.

Таблица 3. Общие свойства элементов Form, CommandButton, Label, TextBox, OptionButton, CheckBox

|  |  |
| --- | --- |
| Название свойства | Смысл (перевод) |
| Name | Имя |
| Width | Ширина  |
| Height | Высота |
| Top | Расстояние до верхнего края |
| Left | Расстояние до левого края |
| Caption | Подпись: то, что можно видеть на элементе управления (кроме TextBox) |
| Back Color | Цвет объекта (кроме CommandButton) |
| Visible | Видимость (True/False) |
| Enabled | Доступность элемента (True/False) |
| AutoSize | Автоматическое изменение размера, в зависимости от текста (подписи) на элементе или размера картинки на изображении |
| Value | 0 – (True ), 1– (False)– (непомеченный, помеченный). Для OptionButton и CheckBox |

Параметры шрифта в элементах управления устанавливаются в диалоговом окне свойства Font. Особенность этого свойства в том, что само оно является объектом и имеет свойства. Только для цвета шрифта используется отдельное свойство, т.к. для других элементов оно задаёт цвет линии. Свойства шрифтов представлены в табл.4.

Таблица 4. Параметры шрифта

|  |  |
| --- | --- |
| Свойство  | Значение  |
| Font.Name | Имя шрифта |
| Font.Size | Размер шрифта |
| Font.Bold | Полужирный  |
| Font.Italic | Курсив |
| Font.Underline | Подчеркивание |
| ForeColor | Цвет шрифта или линии |

##### Форма для установки параметров построения графика функции

Вернёмся к нашей задаче, которую мы решали в главе 2 в Excel: автоматизировать построение графика функции из определённого нами набора. Теперь пришло время задать параметры этой задачи в диалоге. Будем использовать для этой цели форму. Какие же параметры этого процесса надо задать? Обычно для определения функции задают область определения, т.е. наименьшее и наибольшее X. В зависимости от функции также важен шаг, через который будут определяться значения функции. Если это квадратичная функция, то это может быть 0,5 или единица. Для тригонометрических функций лучше взять шаг значительно меньше и являющийся долей числа пи.

Во втором задании для самостоятельной работы (после второго шага построения графика, на странице 55) предлагался подобный вариант работы.

Внимание! Вместо количества точек можно вводить шаг прогрессии, поскольку по количеству точек однозначно определяется шаг и наоборот. Действительно, если разность между максимумом и минимумом разделить на шаг, то получим число промежутков. Количество точек на единицу больше числа промежутков. Поэтому в дальнейшем будем использовать тот или иной вариант, имея ввиду, что в своей работе вы можете использовать любой из вариантов ввода данных.

Для создания формы, подобной той, которая изображена на рисунке, понадобятся:

* элементы Label для пояснений пользователю, т.е. элементы, в которых написано «Функция», «На интервале от до» и «Количество точек»;
* элементы TextBox для ввода минимального значения (от), максимального значения (до), а также количества точек;
* элементы ComboBox для списка возможных для выбора функций;
* кнопка CommandButton, выполняющая построение графика.

После размещения элементов изменим их свойства. Начнём со свойства Caption для элементов Label и CommandButton. Выделяем каждый элемент по очереди и в окне свойств Properties изменяем свойство Caption.

Удобно и логично, чтобы надписи на всех элементах были выполнены одним шрифтом. Если нужно для всех элементов установить одинаковое значение какого-либо свойства, то нужно пометить все эти элементы. Это можно сделать щелчком мыши, удерживая клавишу CTRL. Другой способ – активизировать Select Object (стрелка) на панели элементов Toolbox и, удерживая левую клавишу мыши, «захватить» в прямоугольную рамку нужные элементы. После выделения найдите свойство Font, нажмите кнопку с многоточием – откроется диалоговое окно, в котором можно установить нужные параметры. Шрифты должны быть достаточно крупными, чтобы пользователю не приходилось напрягать зрение или слишком близко подвигаться к экрану.

Для элементов TextBox надо изменить свойство Text. Поскольку эти окна предназначены для ввода данных пользователем, сотрём значение свойства Text.

##### Использование введённых данных.

Итак, пользователь ввел минимальное и максимальное значения. Необходимо посчитать второе значение на оси X, поскольку именно его мы должны ввести, чтобы задать шаг арифметической прогрессии. Используем функцию Val, которая возвращает числовое значение текстового аргумента.

Допустим для определённости, что минимальное значение введено в элемент с именем Text1, максимальное - в элемент Text2, а шаг - в элемент Text3. Тогда соответствующие строчки в процедуре Автозаполнение надо изменить следующим образом:

Dim Min as Single, Max as Single, Step as Single

Min= Text1. Text

Max= Text1. Text

Step= Text3. Text

 Range("A3").Select

 ActiveCell.FormulaR1C1 = Val(Min)

 Range("A4").Select

 ActiveCell.FormulaR1C1 = Val(Min) + Val(Step)

Кроме того, надо подсчитать номер последней ячейки, в которую будет введено значение. Учитывая, что первое значение в третье строчке, получим:

 Number = Int((Val(Max) - Val(Min)) / Val(Step)) + 3

Это число будет использоваться во всех операторах, связанных с автозаполнением:

 Selection.AutoFill Range("A3:A" & Number)

 Range("A3:A" & Number).Select

В последних двух строках вместо номера ячейки с помощью операции конкатенация (&) добавляется номер последней ячейки.

Внимание! Если вы вводите переменные, то используйте один макрос. Если же переменные будут использоваться в нескольких макросах, то необходимо описать эти переменные в секции (General)(Declarations). Смотри приложение Переменные.