

Отображение индикатора текущего состояния с помощью VBA

Ранее я рассмотрел [методы создания пользовательских форм и основы работы с ними](#) (если вы никогда не работали с пользовательскими формами, рекомендую для начала прочитать указанную заметку). В настоящей заметке показано использование индикатора текущего состояния – графического «измерителя», который отображает текущее состояние выполняемой задачи, например, долго работающего макроса.¹

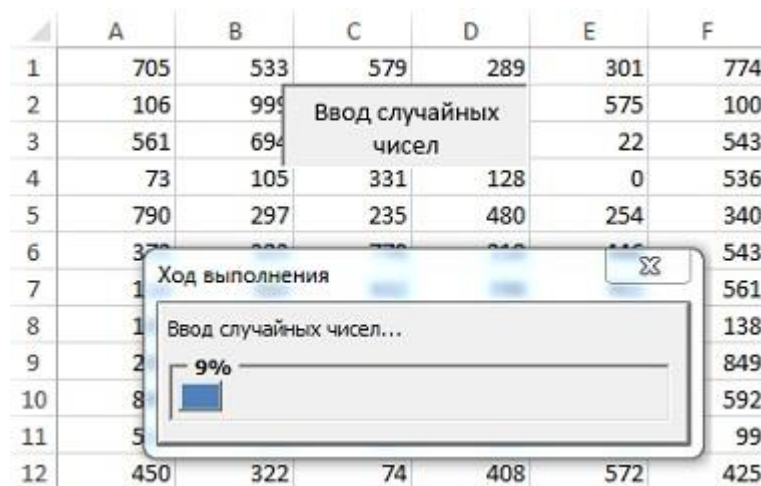


Рис. 1. В окне UserForm отображается ход выполнения макроса

Мы рассмотрим три метода создания индикаторов текущего состояния:

- Макрос, который запускается за пределами диалогового окна UserForm (отдельный индикатор текущего состояния).
- Макрос, который запускается из диалогового окна UserForm. При этом в диалоговом окне UserForm используется элемент управления MultiPage для отображения индикатора текущего состояния, пока выполняется другой макрос.
- Макрос, который запускается из диалогового окна UserForm. При этом высота диалогового окна UserForm увеличивается, а индикатор текущего состояния отображается в нижней части окна.

При использовании индикатора текущего состояния необходимо знать, насколько завершено текущее задание. Способы получения этой информации различаются в зависимости от типа выполняемого макроса. Например, если макрос записывает данные в ячейки (и количество таких ячеек известно), то остается создать код, который будет подсчитывать процентное отношение количества ячеек, содержащих данные. Даже если невозможно точно оценить, насколько далеко «зашел» макрос, пользователю небезынтересно будет узнать, что макрос еще выполняется и Excel не завис.

Отображение индикатора текущего состояния в строке состояния окна

Простой способ отображения хода выполнения макроса — использование строки состояния Excel. Его преимущество — простота реализации. Недостатком же является то, что большинство пользователей не привыкли отслеживать информацию, которая отображается в строке состояния окна, поскольку предпочитают просматривать ее в отдельном окне.

Для отображения сообщения в строке состояния используется следующий оператор:

```
Application.StatusBar = "Пожалуйста, подождите..."
```

Можно обновлять строку состояния в процессе выполнения макроса. Например, если в макросе используется переменная Pet, которая представляет состояние задачи, можно создать код, который будет периодически выполнять следующий оператор:

```
Application.StatusBar = "Выполнение... "&Pet&"% завершено"
```

¹ По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](#). – М: Диалектика, 2013. – С. 477–484.

После завершения макроса нужно вернуть строку состояния к прежнему виду. Для этого используется следующий оператор:

```
Application.StatusBar = False
```

Если строка состояния не возвращена к прежнему виду, продолжает отображаться завершающее сообщение. Помните о том, что индикатор текущего состояния замедляет выполнение макроса, поскольку обновление индикатора требует дополнительного использования системных ресурсов. Если быстроедействие макроса превыше всего, от использования индикатора текущего состояния лучше отказаться.

Создание отдельного индикатора текущего состояния

Такой индикатор не инициализируется путем отображения формы `UserForm`. Следующий макрос очищает рабочий лист, и записывает 20 тысяч случайных чисел в диапазон ячеек (см. также файл *progress indicator1.xlsm*).

```
Sub GenerateRandomNumbers()  
' Вставка случайных чисел на активный лист  
Const RowMax As Long = 500  
Const ColMax As Long = 40  
Dim r As Long, c As Long  
If TypeName(ActiveSheet) <> "Worksheet" Then Exit Sub  
Cells.Clear  
For r = 1 To RowMax  
    For c = 1 To ColMax  
        Cells(r, c) = Int(Rnd * 1000)  
    Next c  
Next r  
End Sub
```

После небольшого изменения макроса (описанного в следующем разделе) диалоговое окно `UserForm` отображает индикатор процесса выполнения макроса (рис. 1).

Создание диалогового окна `UserForm`, включающего индикатор текущего состояния

Выполните следующие шаги:

1. Вставьте новое диалоговое окно `UserForm` и измените значение свойства `Caption` на *Ход выполнения процесса*.
2. Добавьте элемент управления `Frame` и присвойте ему имя `FrameProgress`.
3. Добавьте элемент управления `Label` в состав элемента управления `Frame` и назначьте ему имя `LabelProgress`. Удалите заголовок этого элемента управления, а также сделайте его фон красным (посредством свойства `BackColor`). На данный момент размеры и расположение этого элемента управления не важны.
4. Добавьте еще один элемент управления `Label` над элементом управления `Frame`, с помощью которого вы будете описывать происходящее (необязательно). В нашем примере с помощью этого элемента управления добавляется надпись *Ход выполнения процесса*.
5. Настройте диалоговое окно `UserForm` и элементы управления таким образом, чтобы они выглядели, как на рис. 2.

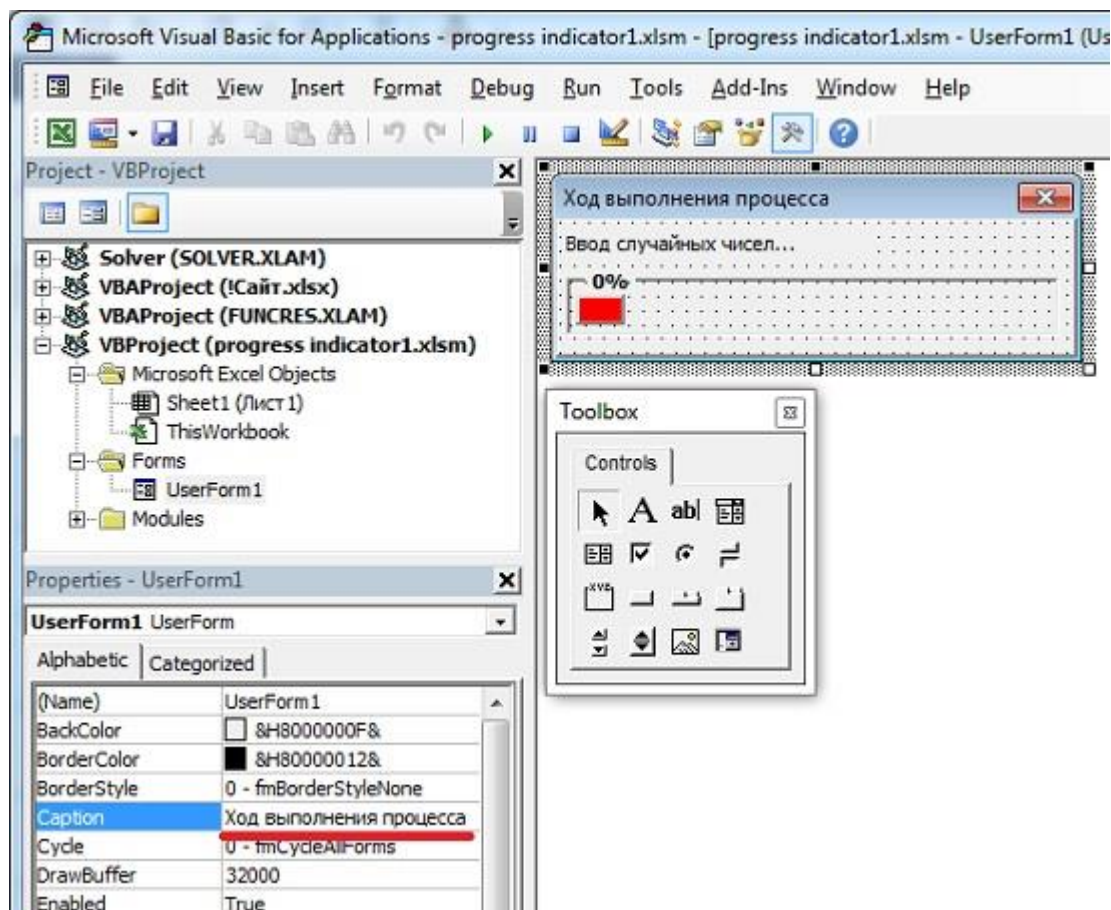


Рис. 2. Окно формы UserForm может играть роль индикатора хода выполнения процесса

Можно изменить тип форматирования элементов управления. Например, свойство `SpecialEffect` элемента `Frame` таким образом, чтобы последний стал «вдавленным».

Создание процедур обработки событий. Важно, чтобы процедура автоматически запускалась при отображении диалогового окна `UserForm`. Один из вариантов подразумевает использование события `Initialize`. Но это событие возникает еще до того, как диалоговое окно отображается на экране, поэтому такой вариант не подходит. С другой стороны, событие `Activate` возникает в тот момент, когда диалоговое окно `UserForm` отображается на экране, поэтому в данном случае можно остановиться на его использовании.

Вставьте следующую процедуру в модуль кода диалогового окна `UserForm`. Эта процедура всего лишь вызывает процедуру `GenerateRandomNumbers`, когда диалоговое окно `UserForm` отображается на экране. Процедура `GenerateRandomNumbers`, которая хранится в модуле кода VBA, является фактическим макросом, который будет работать, пока на экране отображается индикатор текущего состояния.

```
Private Sub UserForm_activate()
    Call GenerateRandomNumbers
End Sub
```

В процедуре `GenerateRandomNumber` имеется дополнительный модуль, который отслеживает текущее состояние, сохраняя соответствующие сведения в переменной `PctDone`.

```
Sub GenerateRandomNumbers ()
' Вставка случайных чисел на активный лист
Dim Counter As Long
Const RowMax As Long = 500
Const ColMax As Long = 40
Dim r As Long, c As Long
Dim PctDone As Double
If TypeName(ActiveSheet) <> "Worksheet" Then Exit Sub
Cells.Clear
```

```

Counter = 1
For r = 1 To RowMax
    For c = 1 To ColMax
        Cells(r, c) = Int(Rnd * 1000)
        Counter = Counter + 1
    Next c
    PctDone = Counter / (RowMax * ColMax)
    Call UpdateProgress(PctDone)
Next r
Unload UserForm1
End Sub

```

Процедура `GenerateRandomNumbers` включает два цикла. Во внутреннем цикле вызывается процедура `UpdateProgress`, которая принимает только один аргумент (переменная `PctDone`, которая «ответственна» за отображение процесса выполнения макроса). Эта переменная может принимать значения от 0 до 100.

```

Sub UpdateProgress(Pct)
    With UserForm1
        .FrameProgress.Caption = Format(Pct, "0%")
        .LabelProgress.Width = Pct * (.FrameProgress.Width - 10)
        .Repaint
    End With
End Sub

```

Создание процедуры запуска. Для отображения диалогового окна `UserForm` введите следующий код в модуль `VBA`.

```

Sub ShowUserForm()
    With UserForm1
        'Использование цвета темы рабочей книги
        .LabelProgress.BackColor = ActiveWorkbook.Theme. _
            ThemeColorScheme.Colors(msoThemeAccent1)
        .LabelProgress.Width = 0
        .Show
    End With
End Sub

```

Можно сделать так, чтобы строка состояния соответствовала текущей теме рабочей книги. Для этого в процедуру `ShowUserForm` добавьте следующий оператор.

```

.LabelProgress.BackColor = ActiveWorkbook.Theme. _
    ThemeColorScheme.Colors(msoThemeAccent1)

```

При выполнении процедуры `ShowUserForm` ширина объекта `Label` устанавливается равной 0. После этого вызывается метод `Show` объекта `UserForm1`, что приводит к отображению диалогового окна `UserForm` (которое играет роль индикатора текущего состояния). Когда диалоговое окно `UserForm` отображается на экране, вызывается событие `Activate`, которое приводит к выполнению процедуры `GenerateRandomNumbers`. Процедура `GenerateRandomNumbers` включает код, который вызывает процедуру `UpdateProgress` при каждом изменении переменной счетчика цикла `r`. Обратите внимание, что процедура `UpdateProgress` использует метод `Repaint` объекта `UserForm`. Если бы этого оператора не было, изображение на экране не обновлялось бы. Перед завершением процедуры `GenerateRandomNumbers` ее последний оператор выгружает диалоговое окно `UserForm` из памяти.

Для того чтобы модифицировать эту методику, необходимо разобраться, как определяется процент завершения выполняемой задачи. После этого значение состояния задачи можно будет присваивать переменной `PctDone`. Определение этой величины осуществляется различными способами в зависимости от приложения. Если код выполняется в цикле (как в данном примере), вычисление процента выполнения сравнительно несложное. Если же код выполняется не в цикле, определение состояния выполнения происходит в различных точках кода.

Отображение сведений о текущем состоянии с помощью элемента управления MultiPage

В предыдущем примере макрос запускался не из диалогового окна UserForm. Во многих случаях долго выполняющийся макрос можно вынудить «уйти со сцены», щелкнув мышью на кнопке ОК в диалоговом окне UserForm. В этом же разделе будет описан лучший способ, использование которого возможно при следующих допущениях:

- проект завершен и отлажен;
- в проекте используется диалоговое окно UserForm (без элемента управления MultiPage) для запуска долго выполняющегося макроса;
- существует метод оценки степени завершения выполняемой задачи.

Как и в предыдущем примере, в рабочий лист вводятся случайные числа. Отличие заключается в том, что в приложении содержится диалоговое окно UserForm, в котором пользователем определяется количество строк и столбцов для ввода случайных чисел (рис. 3; см. также файл *progress indicator2.xlsm*).

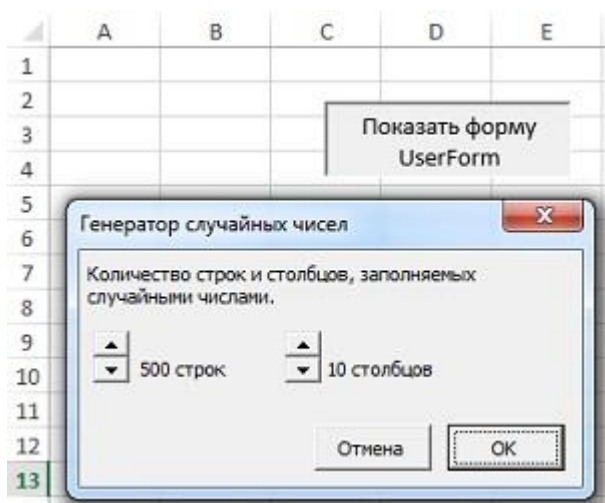


Рис. 3. Пользователь определяет количество строк и столбцов, в которые вводятся случайные числа

Изменение диалогового окна. На данном этапе предполагается, что диалоговое окно UserForm уже настроено. Нам осталось добавить элемент управления MultiPage. Первая страница элемента управления MultiPage будет содержать все первоначальные элементы управления. На второй странице располагаются элементы управления, которые используются для отображения индикатора текущего состояния. Когда макрос начнет выполняться, в коде VBA значение свойства Value элемента управления MultiPage изменится. В результате будут скрыты исходные элементы управления и отображены элементы управления, которые используются для создания индикатора текущего состояния.

Первым шагом будет добавление элемента управления MultiPage в диалоговое окно UserForm. После этого необходимо переместить все существующие в диалоговом окне UserForm элементы управления на первую страницу элемента управления MultiPage. Затем следует активизировать вторую страницу элемента управления MultiPage и настроить ее так, чтобы она выглядела, как показано на рис. 4. В данном случае используется та же комбинация элементов управления, что и в предыдущем примере.

1. Добавьте элемент управления Frame, присвоив ему имя FrameProgress.
2. Добавьте элемент управления Label в состав элемента управления Frame, присвоив ему имя LabelProgress. Удалите заголовок этого элемента управления, а также сделайте его фоновый цвет красным.
3. Добавьте еще один элемент управления Label, описывающий суть происходящего (необязательно).
4. Активизируйте элемент управления MultiPage в целом (а не отдельную вкладку), а свойству Style присвойте значение 2 – fmTabStyleNone. (Это приведет к сокрытию

всех вкладок.) Возможно, придется изменить размер элемента управления MultiPage, чтобы учесть скрытые вкладки.

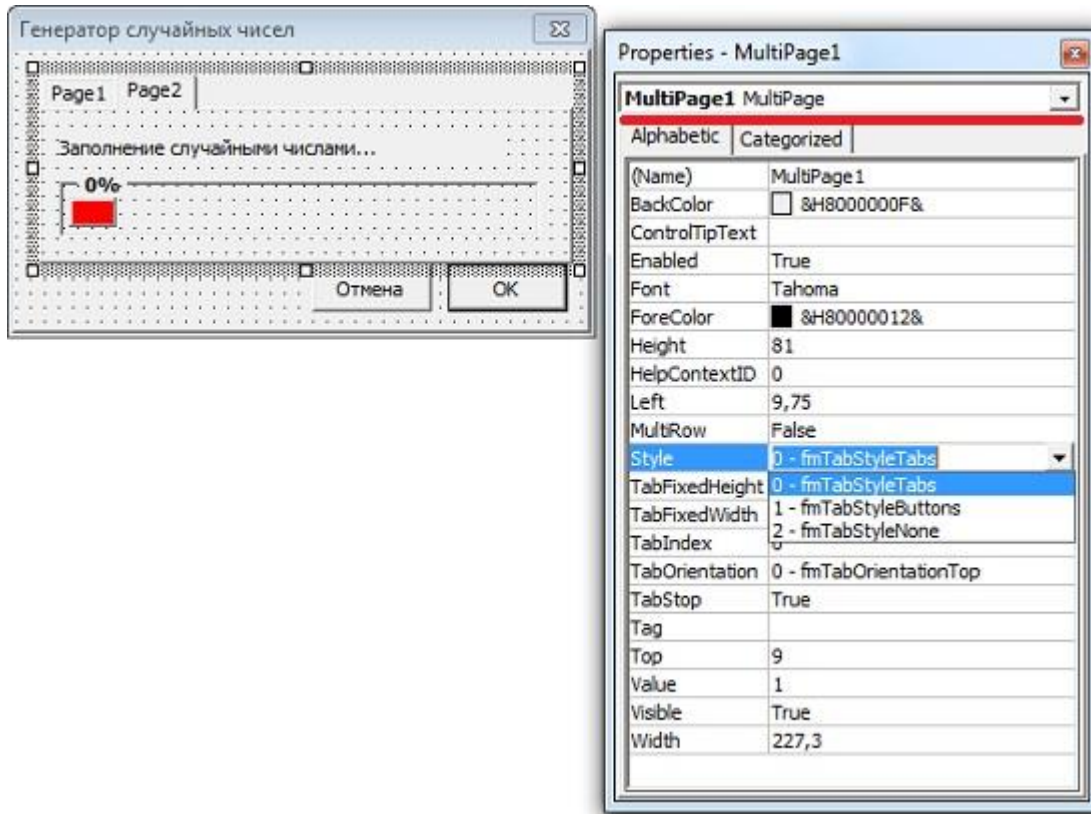


Рис. 4. Вторая вкладка элемента управления MultiPage, которая применяется для отображения индикатора текущего состояния

Простейший способ выделения элемента управления MultiPage, когда вкладки скрыты, — выбор его в раскрывающемся списке, который находится в окне Properties (подчеркнут карсной линией на рис. 4). Для выбора определенной страницы укажите величину свойства Value для элемента MultiPage: 0 — для Page1, 1 — для Page2 и т.д.

Вставка процедуры UpdateProgress. Вставьте следующую процедуру в модуль кода диалогового окна UserForm.

```
Sub UpdateProgress(Pct)
    With UserForm1
        .FrameProgress.Caption = Format(Pct, "0%")
        .LabelProgress.Width = Pct * (.FrameProgress.Width - 10)
    End With
    DoEvents
End Sub
```

Процедура UpdateProgress вызывается из макроса после щелчка пользователем на кнопке ОК и выполняет обновление индикатора текущего состояния.

Далее необходимо модифицировать процедуру, которая выполняется после щелчка на кнопке ОК. Данная процедура выступает обработчиком события Click и называется OKButton_Click. Для начала необходимо вставить оператор в начало процедуры:

```
MultiPage1.Value = 1
```

Этот оператор приводит к активизации второй вкладки элемента управления MultiPage (страницы, на которой отображается индикатор текущего состояния). На следующем шаге необходимо самостоятельно создать код, который будет вычислять степень выполнения задачи. Полученное значение следует присвоить переменной PctDone. Скорее всего, расчеты будут производиться внутри цикла. После этого нужно добавить приведенный ниже оператор, который будет обновлять индикатор текущего состояния:

Call UpdateProgress (PctDone)

Отображение индикатора текущего состояния без применения элемента управления MultiPage

Методика, рассмотренная в этом разделе, немного проще предыдущей, так как не требует применения элемента управления MultiPage. Вместо этого индикатор текущего состояния содержится в нижней части диалогового окна UserForm. Для того чтобы элементы управления, составляющие индикатор текущего состояния, изначально не были видны, высота диалогового окна UserForm была уменьшена до соответствующего размера. Как только нужно будет отобразить индикатор текущего состояния, высота диалогового окна UserForm увеличится, что сделает индикатор видимым на экране (см. файл *progress indicator3.xlsm*).

На рис. 5 показано диалоговое окно UserForm в редакторе VBE. Свойство Height диалогового окна UserForm имеет значение 172. Но перед тем как отобразить диалоговое окно UserForm, значение свойства Height устанавливается равным 124 (это приводит к тому, что элементы управления, представляющие индикатор текущего состояния, становятся невидимыми). Как только пользователь щелкнет на кнопке ОК, код VBA изменит значение свойства Height на 172. Для этого используется следующий оператор:

```
Me.Height = 172
```

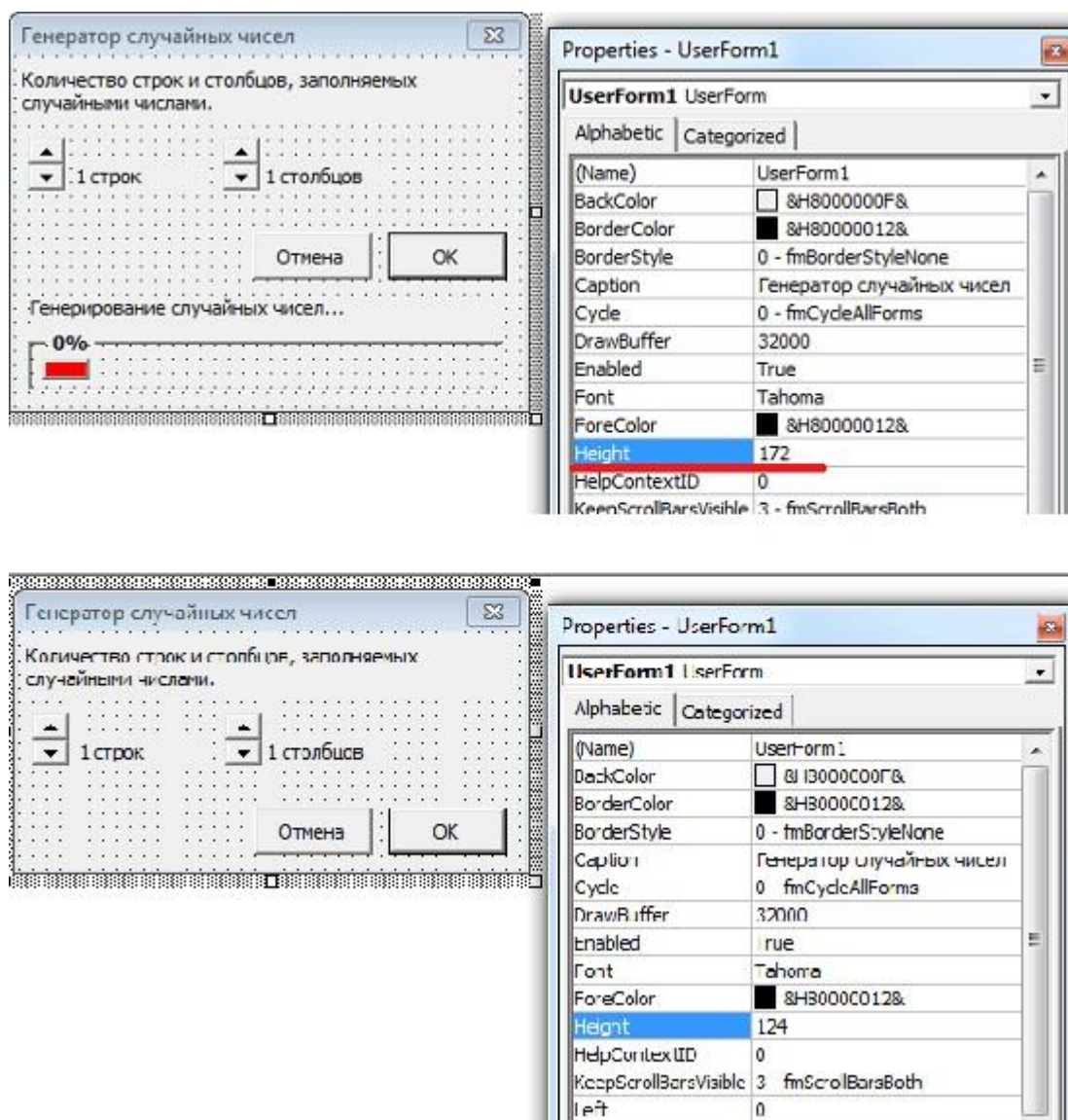


Рис. 5. Уменьшение высоты диалогового окна UserForm приводит к сокрытию элементов управления, образующих индикатор текущего состояния

На рис. 6 показано диалоговое окно UserForm с отображенным индикатором текущего состояния.

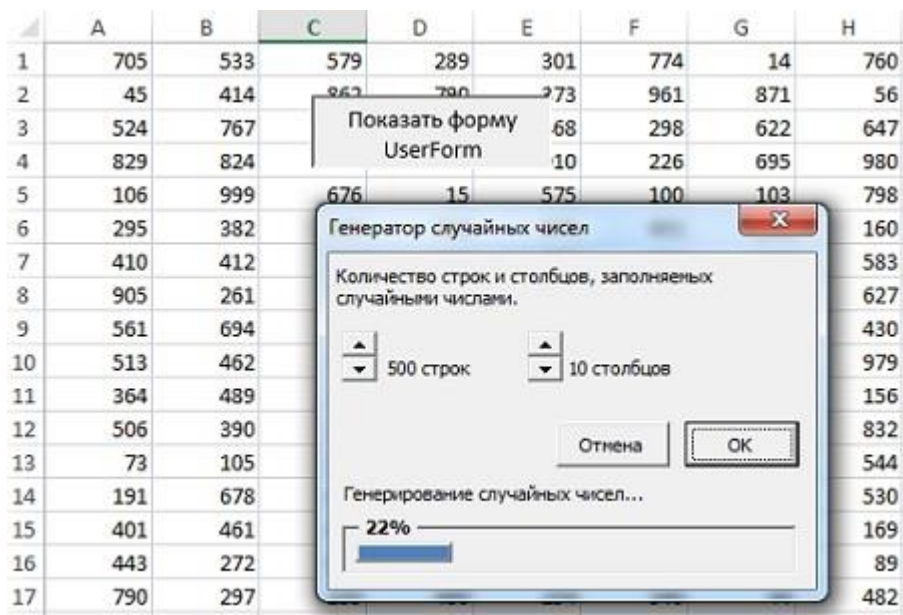


Рис. 6. Индикатор текущего состояния в действии