

Microsoft® Small Basic

Объект Controls

Предполагаемое время работы с этим уроком: 1 час



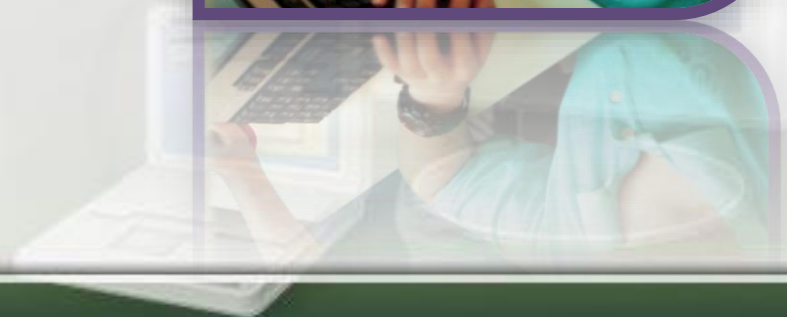
Объект Controls

В этом уроке вы изучите следующее.

Использование различных свойств объекта **Controls**.

Использование различных операций объекта **Controls**.


Использование событий элементов управления для кнопок и текстовых полей в программе.



Общие сведения об объекте Controls

















Пока вы изучили использование различных объектов в Small Basic, таких как **GraphicsWindow**, **Shapes**, **File** и **Math**.

В этом уроке будет представлен объект **Controls** Small Basic. С помощью этого объекта можно отображать простые элементы управления, такие как кнопки и текстовые поля, в графическом окне.

Объект

Controls

Объект Controls позволяет добавлять и перемещать элементы управления, а также взаимодействовать с ними.

-  LastClickedButton
-  LastTypedTextBox
-  AddButton
-  AddMultiLineTextBox
-  AddTextBox
-  GetButtonCaption
-  GetTextBoxText
-  HideControl
-  Move
-  Remove
-  SetButtonCaption
-  SetSize
-  SetTextBoxText
-  ShowControl
-  ButtonClicked
-  TextTyped

Операции объекта Controls

Перед созданием программы с использованием объекта **Controls** рассмотрим некоторые операции объекта **Controls** и их параметры.

AddTextBox — с помощью этой операции можно определить текстовое поле, которое будет отображаться в графическом окне. В качестве параметров необходимо указать координаты x и y текстового поля.

AddButton — с помощью этой операции можно определить кнопку, которая будет отображаться в графическом окне. В качестве параметров необходимо указать название кнопки и координаты x и y .



```
textbox = Controls.AddTextBox(200, 150)  
button = Controls.AddButton("Кнопка", 150, 200)
```

Операции объекта Controls

```
Controls.GetButtonCaption(button)  
Controls.SetButtonCaption(button, "Щелчок")
```

GetTextBoxText — можно получить текст, отображающийся в текстовом поле, указав его имя в качестве параметра для этой операции.

SetTextBoxText — можно определить текст для отображения в текстовом поле, указав его имя и необходимый текст в качестве параметров для этой операции.

GetButtonCaption — эта операция позволяет получить название кнопки. Необходимо указать имя кнопки в качестве параметра.

SetButtonCaption — эта операция позволяет установить или изменить название кнопки. В качестве параметров необходимо указать имя кнопки и новое название.

```
Controls.GetTextBoxText(textbox)  
Controls.SetTextBoxText(textbox, "Hello World!")
```

Операции объекта Controls

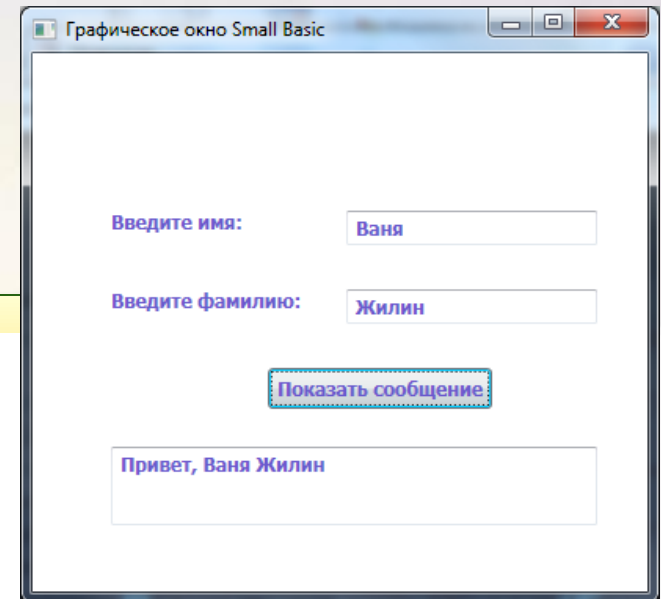
Помимо добавления к программе полезных элементов управления можно выполнить определенные операции и определить параметры для добавляемых элементов управления.

Рассмотрим объект **Controls** на примере.

```
GraphicsWindow.DrawText(50, 100, "Введите имя:")
firstname = Controls.AddTextBox(200, 100)
GraphicsWindow.DrawText(50, 150, "Введите фамилию:")
lastname = Controls.AddTextBox(200, 150)
button = Controls.AddButton("Показать сообщение", 150, 200)
message = Controls.AddMultiLineTextBox(50, 250)
Controls.SetSize(message, 310, 50)

Controls.ButtonClicked = show_message

Sub show_message
  If Controls.GetButtonCaption(button) = "Показать сообщение" Then
    Controls.SetTextBoxText(message, "Привет, " + Controls.GetTextBoxText(firstname) + " " + Controls.GetTextBoxText(lastname))
  EndIf
EndSub
```



Нажмите кнопку  на панели  управления.

Запуск (F5)

Свойства и операции объекта Controls

Рассмотрим возможности нескольких других операций и свойств объекта **Controls**.

HideControl — эту операцию можно использовать для скрытия существующего элемента управления в графическом окне.
ShowControl — эта операция используется для отображения ранее скрытого элемента управления в графическом окне.
Remove — эту операцию можно использовать для удаления элемента управления из графического окна.

```
Controls.HideControl(textbox)  
Controls.ShowControl(button)  
Controls.Remove(textbox)
```

Свойства и операции объекта Controls

```
Controls.SetSize(textbox, 300, 50)  
Controls.Move(textbox, 100, 150)
```

SetSize — с помощью этой операции можно указать фиксированный размер элемента управления. В качестве параметров необходимо указать имя, высоту и ширину элемента управления.

Move — с помощью этой операции можно переместить элемент управления в другое место в графическом окне. В качестве параметров необходимо указать имя, левую и верхнюю координаты элемента управления.

LastClickedButton — эту операцию можно использовать для поиска последней нажатой кнопки в графическом окне.

LastTypedTextBox — эту операцию можно использовать для поиска последнего текстового поля, в которое был введен текст.

```
Controls.LastClickedButton  
Controls.LastTypedTextBox
```

Объект Controls

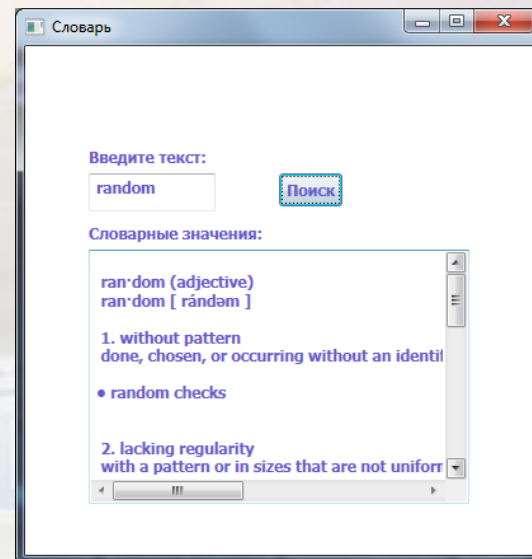
Теперь напишем простую программу, включающую в себя объект **Controls**. Эта программа отображает определения указанного слова.

```
Controls.SetTextBoxText(textbox, "")
GraphicsWindow.DrawText(50, 140, "Словарные значения: ")

multitxt = Controls.AddMultiLineTextBox(50, 160)
Controls.SetSize(multitxt, 300, 200)
Getdfn = Controls.AddButton("Поиск", 200, 100)
GraphicsWindow.DrawText(80, 80, "")
meaning = Dictionary.GetDefinition(Controls.GetTextBoxText(textbox))
Controls.SetTextBoxText(multitxt, meaning)
Controls.ButtonClicked = Showmeaning

Sub Showmeaning
  If Controls.GetButtonCaption(Getdfn) = "Поиск" Then
    meaning = Dictionary.GetDefinition(Controls.GetTextBoxText(textbox))
    Controls.SetTextBoxText(multitxt, meaning)
  EndIf
EndSub
```

ВЫВОД



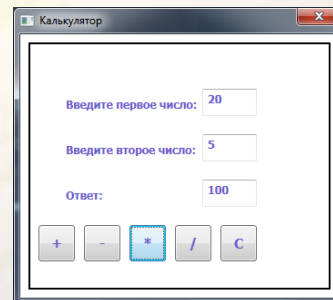
События элементов управления

Теперь, когда вы знакомы с объектом **Controls** в Small Basic, ознакомимся с событиями, которые можно использовать для этого объекта. События элементов управления создают действия в программе при нажатии пользователем кнопки или вводе текста в текстовое поле.

ButtonClicked вызывает событие при нажатии пользователем кнопки.

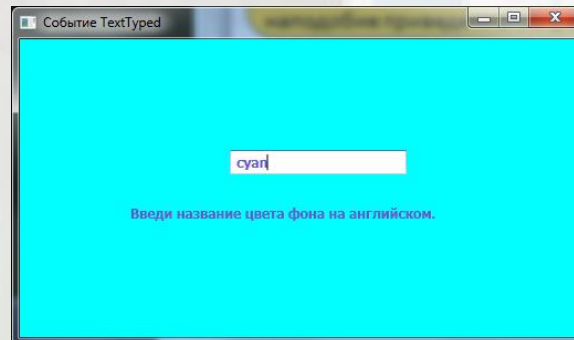
TextTyped вызывает событие при вводе пользователем текста в текстовое поле.

Событие **ButtonClicked** можно использовать для создания простого калькулятора наподобие приведенного справа:



Вот пример события **TextTyped**:

```
GraphicsWindow.Show()  
GraphicsWindow.PenWidth = 20  
GraphicsWindow.DrawText(100, 150, "Введи название цвета фона на английском.")  
color = Controls.AddTextBox(190, 100)  
Controls.TextTyped = ChangeColor  
Sub ChangeColor  
    GraphicsWindow.BackgroundColor = Controls.GetTextBoxText(color)  
EndSub
```





Поздравляем! Вы изучили следующее.

- ✚ Использование различных свойств объекта **Controls**.
- ✚ Использование различных операций объекта **Controls**.
- ✚ Использование событий элементов управления для кнопок и текстовых полей в программе.

Продемонстрируйте свои знания

Написание программы для отображения простой формы и выполнения следующих действий.

- ❖ Добавление текстовых полей для запроса имени, адреса, номер телефона и адреса электронной почты пользователя.
- ❖ Добавление к форме кнопки **Отправить**.
- ❖ После ввода пользователем данных и нажатия кнопки **Отправить** отображается соответствующее сообщение.

