# Лабораторная работа № 11

# Циклы и таймеры: Отображение информации с помощью цикла For ... Next

1. Запустите Visual Studio и создайте в папке **c:\vbnet03sbs\Гл.7** новый проект Windows Application (Приложение Windows) на Visual Basic с именем My For Loop. В Windows Forms Designer (Конструкторе Windows Forms) появится пустая форма. Прежде всего вы добавите в форму элемента управления Button, но на этот раз новым способом.
2. Дважды щелкните мышью в окне области элементов на элементе управления Button (он находится на закладке Windows Forms). Visual Studio поместит объект кнопки в верхний левый угол формы. В случае с элементом управления Button, а также и со многими другими, двойной щелчок мышью позволяет быстро создать в форме объект со стандартными размерами. Теперь можно перетащить этот объект кнопки в нужное место и настроить его свойства.
3. Перетащите объект кнопки вправо и поместите его в центре верхней части формы.
4. Откройте окно Properties (Свойства), а затем установите свойство Text кнопки на значение Цикл.
5. В области элементов щелкните дважды на элементе управления TextBox. Visual Studio создаст небольшое текстовое поле.
6. Установите свойство Multiline этого объекта текстового поля на значение True, а свойство ScrollBars на значениеVertical. Эти настройки подготавливают текстовое поле к отображению более чем одной строки текста.
7. Сделайте свойство Text объекта текстового поля пустым (empty).
8. Переместите текстовое поле под кнопку и увеличьте его так, чтобы оно заняло большую часть формы.
9. Щелкните дважды на кнопке **Цикл** формы. В Редакторе кода появится процедура события Button1\_Click.
10. Введите в этой процедуре следующие операторы программы:

Dim i As Integer

Dim Wrap As String

Wrap = Chr(13) & Chr(10)

For i = 1 To 10

TextBox1.Text = TextBox1.Text & "Строка " & i & Wrap

Next i

Эта процедура события объявляет две переменные - одну типа Integer (i), а вторую - типа String (Wrap), а затем присваивает второй переменной строковое значение, представляющее символ перевода строки. В терминах программирования символ перевода строки является эквивалентом нажатия на клавишу (Enter) на клавиатуре. Чтобы сделать его менее громоздким, я создал для этого символа специальную переменную, которая состоит из элементов "возврат каретки" (return) и "прокрутка строки" (*linefeed*).

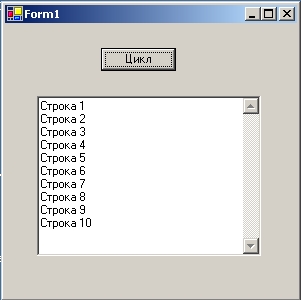
После объявления переменной и присвоения, я использую цикл For...Next для десятикратного отображения в объекте текстового поля строки "Строка X", где X - это текущее значение переменной-счетчика (другими словами, со "Строка 1" до "Строка 10"). Символы конкатенации строк (&) объединяют в текстовом поле части каждой строки воедино. В начале в объект добавляется все значение текстового поля, которое хранится в свойстве Text, так, что предыдущие строки при добавлении новых не удаляются. Затем для отображения новой строки и перевода курсора влево и на следующую строку объединяются строка "Строка", текущий номер строки и символ возврата каретки (Wrap). Оператор Next завершает цикл.

Обратите внимание, что Visual Studio автоматически добавляет оператор Next в конец цикла, когда вы вводите For в его начале. В данном случае я отредактировал оператор Next так, чтобы включить в него имя переменной i - это необязательное пояснение синтаксиса, которое мне нравится использовать.

1. Чтобы сохранить изменения, щелкните на кнопке Save All (Сохранить все) на стандартной панели инструментов. Теперь запустите эту программу.
2. Щелкните на кнопке Start (Начать) стандартной панели инструментов.
3. Щелкните на кнопке **Цикл**.

**Примечание**. Готовая программа For Loop находится в папке **c:\vbnet03sbs\ Гл.7\for loop**.

Цикл For...Next отображает в текстовом поле 10 строк, как показано ниже.



1. Снова щелкните на кнопке **Цикл**. Цикл For...Next отображает в форме еще 10 строк, и вы можете увидеть невидимые строки, используя вертикальную полосу прокрутки. При каждом повторении цикла он добавляет в объект текстового поля 10 новых строк.

**Совет**. Боитесь, что в объекте текстового поля не останется места? Если вы отображаете только 10 строк, это случится очень нескоро. Объект многострочного текстового поля имеет практическое ограничение на размер текста в 32 Кб. Для текстов большего объема или для форматирования попробуйте использовать элемент управления RichTextBox.

1. Чтобы завершить выполнение программы, щелкните на кнопке **Закрыть** формы.

**Совет**. Если требуется исполнять набор операторов много раз, цикл For...Next значительно упростит ваш код и уменьшит общее число операторов, которые нужно вводить. В предыдущем примере цикл For...Next длиной в три строки выполнил ту же работу, что и его десятистрочный эквивалент без использования цикла.

# Лабораторная работа № 12

# Циклы и таймеры: Открытие файлов с помощью цикла For ... Next

1. В меню File (Файл) укажите на New (Создать), а затем щелкните на Project (Проект). Появится диалоговое окно New Project (Создать проект).
2. В папке **c:\vbnet03sbs\Гл.7** создайте новый проект Windows Application (Приложение Windows) на Visual Basic с именем **My For Loop Icons**. Будет создан новый проект, и в Windows Forms Designer (Конструктор Windows Forms) появится пустая форма.
3. Щелкните в окне области элементов на элементе управления PictureBox, расположенном на закладке Windows Forms, а затем нарисуйте в центре верхней половины формы объект вывода изображения средних размеров.
4. Щелкните в области элементов на элементе управления Button, а затем нарисуйте под полем вывода изображений широкую кнопку (чтобы уместился длинный текст).
5. Установите для этих двух объектов следующие свойства:

|  |  |  |
| --- | --- | --- |
| **Объект** | **Свойство** | **Установка** |
| PictureBox1 | BorderStyle | Fixed3D |
| SizeMode | StretchImage |
| Button1 | Text | "Показать четыре рожицы" |

1. Чтобы вывести на экран процедуру события для кнопки **Показать четыре рожицы**, дважды щелкните на этой кнопке. В Редакторе кода появится процедура события Button1\_Click.
2. Введите следующий цикл For...Next:

Dim i As Integer

For i = 1 To 4

PictureBox1.Image = System.Drawing.Image.FromFile\_

("c:\vbnet03sbs\Гл.7\face0" & i & ".ico")

MsgBox("Щелкните здесь, чтобы посмотреть следующую рожицу.")

Next

**Совет**. Метод FromFile в этой процедуре события слишком длинный, чтобы поместиться в одной строке этой книги, так что я разбил его на две с помощью пробела и символа продолжения строки (\_). Вы можете использовать этот символ в вашей программе везде, за исключением строковых выражений.

Для загрузки четырех значков из файлов, расположенных в папке **c:\vbnet03sbs\Гл.7** на вашем жестком диске, этот цикл использует метод FromFile. Имя файла создается с помощью переменной-счетчика и оператора конкатенации, который вы использовали ранее в этой лекции. Код

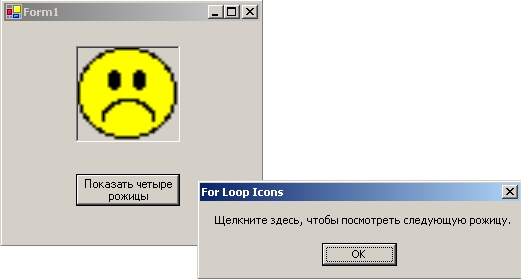
PictureBox1.Image = System.Drawing.Image.FromFile\_

("c:\vbnet03sbs\Гл.7\face0" & i & ".ico")

для создания четырех корректных имен файлов значков, хранящихся на жестком диске, объединяет путь, имя файла и расширение .ico. В этом примере вы загружаете в поле вывода изображений файлы face01.ico, face02.ico, face03.ico и face04.ico. Этот оператор работает потому, что несколько файлов в папке **c:\vbnet03sbs\Гл.7** имеют шаблон имен facexx.ico. Использование этого шаблона позволяет создать для этих имен файлов цикл For...Next.

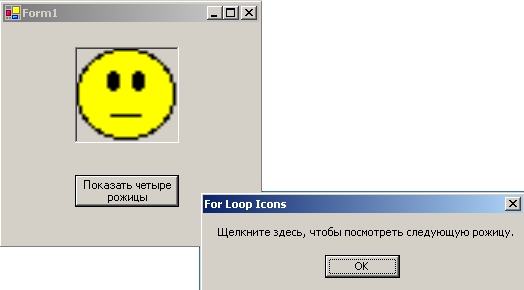
**Примечание**. Функция отображения сообщения ( MsgBox ) используется в первую очередь для того, чтобы замедлить работу и позволить вам увидеть, что происходит в цикле For...Next. В нормальном приложении вы, вероятно, не будете использовать такие функции (хотя такая возможность у вас есть).

1. Чтобы сохранить изменения, щелкните на кнопке Save All (Сохранить все) на стандартной панели инструментов.
2. Чтобы запустить программу, щелкните на кнопке Start (Начать) на стандартной панели инструментов, а затем на кнопке**Показать четыре рожицы**. (Готовая программа For Loop Icons лежит в папке c:\vbnet03sbs\Гл.7\For loop icons.) ЦиклFor...Next загружает в область вывода изображений первую рожицу, а затем отображает окно сообщения.



**Примечание**. Если Visual Basic выдает сообщение об ошибке, проверьте код программы на предмет опечаток, а затем убедитесь, что файлы значков находятся по указанному вами в программе пути. Если вы установили файлы практических занятий этой книги в папку, отличную от папки по умолчанию, или переместили файлы значков после установки, то путь в этой процедуре события может оказаться неверным.

1. Чтобы показать следующую рожицу, щелкните на кнопке **OK**. Ваш экран будет выглядеть примерно так.



1. Щелкните на кнопке **OK** еще три раза, чтобы увидеть весь набор рожиц. Если хотите, то можете повторить всю последовательность.
2. Когда закончите, щелкните на кнопке Close формы, чтобы выйти из программы. Программа остановится, и вернется среда разработки.

# Лабораторная работа № 13

# Циклы и таймеры: Преобразование температур с помощью цикла Do

1. В меню File (Файл) укажите на New (Создать), а затем щелкните на Project (Проект). Появится диалоговое окно New Project (Создать проект).
2. В папке **c:\vbnet03sbs\Гл.7** создайте новый проект Windows Application (Приложение Windows) на Visual Basic с именем My Celsius Conversion. Будет создан новый проект и в Windows Forms Designer (Конструкторе Windows Forms) появится пустая форма. На этот раз вы поместите весь код программы в процедуре события Form1\_Load, так что при запуске приложения Visual Basic попросит вас ввести температуру по Фаренгейту. Для запроса данных используется функция InputBox, а для отображения преобразованных данных - функция MsgBox.
3. Сделайте двойной щелчок мышью на форме. В Редакторе кода появится процедура события Form1\_Load.
4. Введите в процедуре события Form1\_Load следующие операторы программы:
5. Dim FTemp, Celsius As Single
6. Dim strFTemp As String
7. Dim Prompt As String = "Введите температуру по Фаренгейту."
8. Do
9. strFTemp = InputBox(Prompt, "Пересчет температуры")
10. If strFTemp <> "" Then
11. FTemp = CSng(strFTemp)
12. Celsius = Int((FTemp + 40) \* 5 / 9 - 40)
13. MsgBox(Celsius, , "Температура по Цельсию")
14. End If
15. Loop While strFTemp <> ""

End

**Совет**. Убедитесь, что в конце процедуры события Form1\_Load вы добавили оператор End.

Этот код выполняет необходимые вычисления. Первая строка объявляет две переменные одинарной точности - FTemp иCelsius - которые хранят температуры по Фаренгейту и Цельсию соответственно. Вторая строка объявляет строковую переменную с именем strFTemp, которая хранит строковую версию температуры по Фаренгейту. Третья строка объявляется строковую переменную с именем Prompt, которая будет использоваться функцией InputBox, и присваивает ей начальное значение. Цикл Do запрашивает у пользователя температуру по Фаренгейту, преобразует ее в шкалу Цельсия, а затем отображает на экране с помощью функции MsgBox.

Значение, введенное пользователем, сохраняется в переменной strFTemp. Функция InputBox всегда возвращает значение строкового типа, даже если пользователь вводит число. Так как мы хотим выполнить с введенным значением математические вычисления, переменная strFTemp должна быть преобразована в число. Для преобразования строки в число одинарной точности используется функция CSng. CSng - это одна из нескольких функций, предназначенных для преобразования строки в другие типы данных. Затем преобразованное значение одинарной точности сохраняется в переменнойFTemp.

Цикл выполняется до тех пор, пока пользователь не щелкнет на кнопке Cancel (Отмена), или до тех пор, пока не нажмет на (Enter) или не щелкнет на **OK** при отсутствии в поле ввода какого-либо значения. Щелчок на кнопке Cancel (Отмена) или ввод пустого значения возвращают пустую строку (""). Цикл проверяет, не пуста ли строка, с помощью проверки условия While, стоящего в конце цикла. Оператор программы

Celsius = Int((FTemp + 40) \* 5 / 9 - 40)

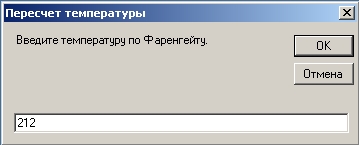
производит преобразование значений шкалы Фаренгейта в шкалу Цельсия. Этот оператор реализует стандартную формулу преобразования, но при этом использует функцию Int, чтобы возвратить в переменную Celsius значение, которое не содержит десятичных знаков. (Все, что находится справа от десятичной точки, отбрасывается.) Это отрезание приносит в жертву точность, но помогает вам избежать появления длинных нечитаемых значений типа 21.11111, которое является эквивалентом по Цельсию для температуры в 70 градусов по Фаренгейту.

1. Чтобы сохранить изменения, щелкните на кнопке Save All (Сохранить все) на стандартной панели инструментов.

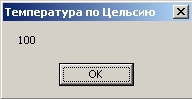
**Примечание**. Готовая программа Celsius Conversion находится в папке **c:\vbnet03sbs\Гл.7\celsius conversion**.

Теперь попробуйте запустить эту программу.

1. Щелкните на кнопке Start (Начать) стандартной панели инструментов. Программа запустится, и функция InputBox запросит у вас температуру по Фаренгейту.
2. Введите 212. Ваш экран должен выглядеть примерно так.



1. Щелкните на **OK**. Температура в 212 градусов по Фаренгейту преобразуется в 100 градусов по Цельсию.

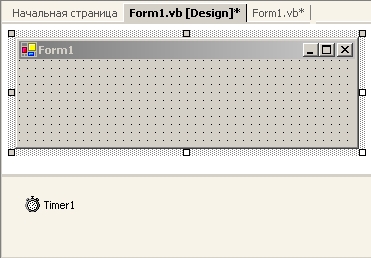


1. Щелкните на **OK**. Введите в поле ввода 72 и снова щелкните на **OK**. Температура в 72 градуса по Фаренгейту преобразуется в 22 градуса по Цельсию.
2. Щелкните на **OK**, а затем выйдите из программы, щелкнув на Cancel (Отмена) в диалоговом окне ввода. Программа закроется, и вернется среда разработки.

# Лабораторная работа № 13

**Циклы и таймеры: Создание программы Digital Clock**

1. В меню File (Файл) выберите New (Создать), а затем щелкните на Project (Проект). Появится диалоговое окно New Project (Создать проект).
2. В папке **c:\vbnet03sbs\Гл.7** создайте новый проект с именем **My Digital Clock**. Будет создан новый проект и в Windows Forms Designer (Конструкторе Windows Forms) появится пустая форма.
3. Измените размер формы так, чтобы она приняла вид небольшого прямоугольного окна (по ширине больше, чем по высоте). Ваши цифровые часы не должны занимать много места.
4. Дважды щелкните в окне области элементов на элементе управления Timer на закладке Windows Forms. Visual Studio создаст в области компонент под формой небольшой объект таймера, как показано ниже.



Некоторые элементы управления Visual Studio не имеют визуального представления на форме (см. ["Работа с меню и диалоговыми окнами"](http://www.intuit.ru/studies/courses/78/78/lecture/2364)), и когда создаются объекты этих элементов управления, они появляются в области компонент, расположенной под формой. Однако вы по-прежнему можете выбрать на этой специальной панели элементы управления и установить их свойства.

1. Щелкните в области элементов на элементе управления Label, а затем нарисуйте метку величиной почти во всю форму. Эта метка будет использоваться для отображения времени на часах, и она должна быть очень большой, чтобы в ней уместился шрифт размером в 24 пункта, который вы будете использовать.
2. Откройте окно Properties (Свойства) и установите следующие свойства формы и двух ее объектов.

|  |  |  |
| --- | --- | --- |
| **Объект** | **Свойство** | **Установка** |
| Label1 | Text | пустой (empty) |
| Font | Times New Roman, Bold (Жирный), 24-point (12 пунктов) |
| TextAlign | MiddleCenter |
| Timer1 | Enabled | True |
| Interval | 1000 |
| Form1 | Text | "Цифровые часы" |

1. **Совет**. Если вы хотите поместить в качестве фона для ваших часов какой-либо рисунок, установите в свойствеBackgroundImage объекта Form1 путь к графическому файлу.
2. Теперь вы должны написать программный код для таймера.
3. Дважды щелкните в области компонент на объекте таймера. В Редакторе кода появится процедура события Timer1\_Tick. Опытные программисты на Visual Basic 6 заметят, что эта процедура события с целью прояснения того, что она делает в программе, была переименована с Timer1\_Timer в Timer1\_Tick (это значит, что процедура события выполняется при каждом "тике" таймера).
4. Введите следующий оператор:

Label1.Text = TimeString

Этот оператор получает от системных часов текущее системное время и присваивает его свойству Text объекта Label1. (Если вы хотите отображать в часах кроме времени еще и дату, используйте вместо свойства TimeString свойствоSystem.DateTime.Now.) В этой программе требуется только один оператор, так как вы установили свойство Intervalтаймера с помощью окна Properties (Свойства). Все остальное обрабатывается объектом таймера.

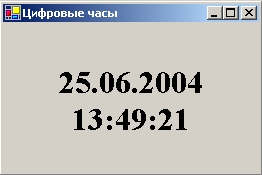
1. Чтобы сохранить изменения, щелкните на кнопке Save All (Сохранить все) на стандартной панели инструментов.

**Примечание**. Готовая программа Digital Clock находится в папке **c:\vbnet03sbs\Гл.7\digital clock**.

1. Чтобы запустить часы, щелкните на кнопке Start (Начать) на стандартной панели инструментов. Появятся часы, как показано на следующей иллюстрации (конечно, ваше время будет отличаться).



Если вы использовали свойство System.DateTime.Now, то вы также увидите на часах текущую дату, как показано ниже. Если ваш дисплей разбивается на две строки, как показано здесь, вы можете слегка увеличить форму.



1. Понаблюдайте некоторое время за часами. Visual Basic обновляет время каждую секунду.
2. Чтобы остановить часы, щелкните на кнопке **Закрыть** строки заголовка.

Программа Digital Clock настолько удобна, что ее можно скомпилировать в виде исполняемого файла и использовать на вашем компьютере. Ее можно настроить с помощью ваших собственных рисунков, текста и цветов.

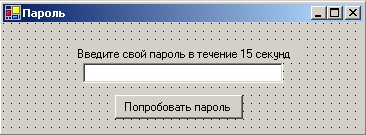
# Лабораторная работа № 14

#### Циклы и таймеры: Установка ограничения по времени на ввод пароля

1. В меню File (Файл) укажите на New (Создать), а затем щелкните на Project (Проект). Появится диалоговое окно New Project (Создать проект).
2. В папке **c:\vbnet03sbs\Гл.7** создайте новый проект с именем **My Timed Password**. Будет создан новый проект, и в Конструкторе Windows Forms появится пустая форма.
3. Придайте форме размеры небольшого прямоугольного окна, примерно с окно ввода.
4. Щелкните в окне области элементов на элементе управления TextBox, расположенном на закладке Windows Forms, а затем нарисуйте в центре формы текстовое поле для пароля.
5. Щелкните в области элементов на элементе управления Label, а затем нарисуйте над текстовым полем длинную метку.
6. Щелкните в области элементов на элементе управления Button, а затем нарисуйте под текстовым полем кнопку.
7. Дважды щелкните в области элементов на элементе управления Timer. Visual Studio добавит в область компонент, расположенную под формой, объект таймера.
8. Установите в программе свойства, приведенные в следующей таблице.

|  |  |  |
| --- | --- | --- |
| **Объект** | **Свойство** | **Установка** |
| Label1 | Text | "Введите свой пароль в течение 15 секунд" |
| TextBox1 | PasswordChar | "\*" |
| Text | пустой (empty) |
| Button1 | Text | "Попробовать пароль" |
| Timer1 | Enabled | True |
| Interval | 15000 |
| Form1 | Text | "Пароль" |

1. Настройка PasswordChar будет отображать при вводе пользователем пароля в текстовом поле символы звездочки (\*). Установка свойства Interval таймера на значение 15000 дает пользователю 15 секунд на то, чтобы ввести пароль и щелкнуть на кнопке Попробовать пароль. Установка свойства Enabled в значение True запустит таймер сразу, как только будет запущена программа. Если ваш таймер снова может понадобиться при дальнейшей работе программы, то вы можете отключить это свойство и включить его в какой-либо процедуре события.
2. Ваша форма должна выглядеть примерно так.



1. Сделайте двойной щелчок мышью в области компонент на объекте таймера, а затем введите в процедуре Timer1\_Tickсобытия следующие операторы:

MsgBox("Извините, время вышло.")

End

Первый оператор отображает сообщение, указывающее, что период времени истек, а второй останавливает программу. Visual Basic выполняет эту процедуру события в том случае, если интервал таймера достигает 15 секунд, а правильный пароль не был введен.

1. Отобразите форму, сделайте двойной щелчок мышью на объекте кнопки, а затем введите в процедуре событияButton1\_Click следующие операторы:

If TextBox1.Text = "secret" Then

Timer1.Enabled = False

MsgBox("Добро пожаловать в систему!")

End

Else

MsgBox("Извините, я вас не знаю.")

End If

Этот код программы проверяет, равен ли введенный в текстовом поле пароль слову "secret". Если да, то таймер отключается, отображается *приветственное сообщение* и программа завершается. Более полезная программа может в этом месте продолжить работать. Если введенный пароль неверен, пользователь получает уведомление в окне сообщения и получает еще одну попытку ввести пароль. Но у пользователя есть только 15 секунд, чтобы это сделать!

1. Чтобы сохранить изменения, щелкните на кнопке Save All (Сохранить все) на панели инструментов.

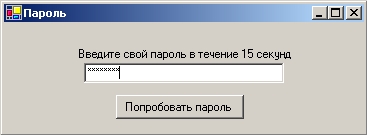
**Примечание**. Готовая программа *Timed Password* лежит в папке **c:\vbnet03sbs\Гл.7\timed password**.

#### 

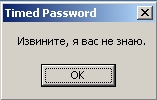
# Лабораторная работа № 15

#### Циклы и таймеры: Тестирование программы Timed Password

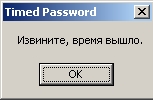
1. Чтобы запустить программу, щелкните на кнопке Start (Начать). Программа запустится, и 15-секундные часы начнут тикать.
2. Введите в текстовое поле слово "откройся". Символы звездочки скроют текст вашего ввода, как показано здесь.



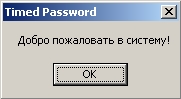
1. Щелкните на кнопке **Попробовать пароль**. На экране появится окно сообщения, уведомляющее вас о некорректном ответе.



1. Щелкните на **OK** и подождите, пока истечет период "регистрации". Программа отобразить сообщение об истечении времени.



1. Щелкните на **OK**, чтобы закрыть программу.
2. Снова запустите программу, введите в текстовом поле secret (правильный пароль), а затем щелкните на **Попробовать пароль**.
3. Программа отобразит такое сообщение.



1. Щелкните на **OK**, чтобы закрыть программу. Появится среда разработки Visual Basic. Существуют различные способы практического применения объектов таймера. Как и в случае с циклами For...Next и Do, вы можете использовать объекты таймера для повторного выполнения команд или процедур столько раз, сколько потребуется в вашей программе.

### Краткая справка по лекции

|  |  |
| --- | --- |
| **Чтобы** | **Сделайте следующее** |
| Выполнить группу операторов программы заданное количество раз | Добавьте свои операторы между операторами цикла For и Next. Например:  Dim i As Integer  For i = 1 To 10  MsgBox("Нажмите ОК!")  Next i |
| Использовать с операторами заданную последовательность чисел | Поместите свои операторы в цикл For...Next и используйте для определения последовательности чисел ключевые слова To и Step. Например:  Dim i As Integer  For i = 2 To 8 Step 2  TextBox1.Text = TextBox1.Text & i  Next i |
| Избежать бесконечного цикла Do | Убедитесь, что цикл имеет проверяемое условие, которое может принять значение False. |
| Досрочно выйти из цикла For \dots Next | Используйте оператор программы Exit For. Например:  Dim InpName As String  Dim i As Integer  For i = 1 To 10  InpName = InputBox("Имя?")  If InpName = "Вася" Then Exit For  TextBox1.Text = InpName  Next i |
| Выполнять группу операторов программы до тех пор, пока не выполнится заданное условие | Поместите эти операторы между операторами Do и Loop. Например:  Dim Query As String = ""  Do While Query <> "Yes"  Query = InputBox("Вася?")  If Query = "Yes" Then MsgBox("Привет")  Loop |
| Выполнять цикл до тех пор, пока условие не станет равно True | Используйте цикл Do с ключевым словом Until. Например:  Dim GiveIn As String  Do  GiveIn = InputBox("Say 'Uncle'")  Loop Until GiveIn = "Uncle" |
| Выполнять цикл в вашей программе в течение указанного интервала времени | Используйте элемент управления Timer. |