Операторы цикла

Теория: Если требуется выполнить большое количество однотипных действий, то 20 раз писать одно и то же действие было бы глупо. Не говоря уже про 100 действий или 1000. Поэтому был придуман *оператор цикла*, который позволял, по заданному условию, выполнять любое количество однотипных действий. Операторы цикла делятся на 2 типа:

1. Цикл со счётчиком
2. Цикл без счётчика

Цикл со счётчиком

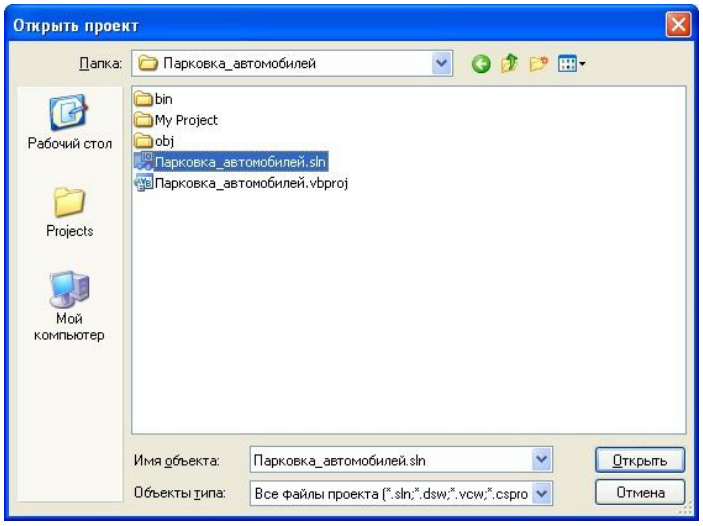
Используется он по следующей формуле:

For <переменная> = <начальное_значение> To <конечное_значение>
<действия >
Next <переменная>

, причём «переменная» должна быть уже объявлена и быть одинаковой в 1-й строке и в 3-й. «действия» выполняются ровно столько раз, сколько потребуется шагов от «начального_значения» до «конечного_значения».

Один такой шаг называется *Итерацией.* Все шаги называются *Циклом.*

Практика: Давайте откроем проект «Стоянка_автомобилей», созданный Лабораторной работе №3. Для этого запустите Visual Studio 2010, затем в меню «Файл» > Открыть > «Решение или проект». И выбрать файл проекта (значок содержит 10).



Появится текст предыдущей лабораторной работы.

```
Module Module1

    Sub Main()
        Dim текст As String
        Dim число As Integer
        Dim разрешено, запрещено, неверноеЗначение As String
        разрешено = "Парковка в данном месте сегодня разрешена"
        запрещено = "Парковка в данном месте сегодня запрещена"
        неверноеЗначение = "Такого числа в этом месяце не существует. Допустимые значения: от 1 до 30"
        Console.WriteLine("Введите сегодняшнее число: ")
        текст = Console.ReadLine()
        число = Integer.Parse(текст)
        If число < 1 Then
            Console.WriteLine(неверноеЗначение)
            Console.ReadKey()
            Return
        End If
        If число > 30 Then
            Console.WriteLine(неверноеЗначение)
            Console.ReadKey()
            Return
        End If
        If число Mod 2 = 0 Then
            Console.WriteLine(запрещено)
        Else
            Console.WriteLine(разрешено)
        End If
        Console.ReadKey()
    End Sub

End Module
```

В Лабораторной Работе №3 было неудобно несколько раз запускать программу, чтобы проверить несколько чисел. Давайте с помощью оператора цикла сделаем такую возможность. Для этого нам нужно весь код, который мы написали самостоятельно – «заключить» в оператор цикла.

Поставьте курсор между Sub Main() и Dim текст As String

```
Sub Main()
    |
    Dim текст As String
```

Однако прежде чем напечатать первую строку из формулы цикла со счётчиком, нам потребуется объявить переменную, которая будет выступать в качестве счётчика:

```
Dim счетчик As Integer
```

После того, как счётчик объявлен, мы можем приступать к формуле. Пусть наша программа будет выполняться 5 раз. Это значит, что начальное значение должно быть «1», а конечное «5».

```
For счетчик = 1 To 5
```

Теперь мы должны закрыть оператор цикла с помощью своеобразной закрывающей скобки «Next». Установить этот «Next» нужно в конце выполнения написанного нами кода. Потому что именно эти действия мы хотим повторять несколько раз. Ставим Next счетчик сразу же после Console.ReadKey() в конце программы. Конечный вид программы должен выглядеть так:

```
    Sub Main()
        Dim счетчик As Integer
        For счетчик = 1 To 5
            Dim текст As String
            Dim число As Integer
            Dim разрешено, запрещено, неверноеЗначение As String
            разрешено = "Парковка в данном месте сегодня разрешена"
            запрещено = "Парковка в данном месте сегодня запрещена"
            неверноеЗначение = "Такого числа в этом месяце не существует. Допустимые значения: от 1 до 30"
            Console.WriteLine("Введите сегодняшнее число: ")
            текст = Console.ReadLine()
            число = Integer.Parse(текст)
            If число < 1 Then
                Console.WriteLine(неверноеЗначение)
                Console.ReadKey()
                Return
            End If
            If число > 30 Then
                Console.WriteLine(неверноеЗначение)
                Console.ReadKey()
                Return
            End If
            If число Mod 2 = 0 Then
                Console.WriteLine(запрещено)
            Else
                Console.WriteLine(разрешено)
            End If
            Console.ReadKey()
        Next счетчик
    End Sub
End Module
```

Давайте запустим программу с помощью клавиши «F5». Теперь мы можем целых 5 раз подряд вводить число, для того, чтобы узнать – разрешена парковка или нет. Сбивает с толку только команда Console.ReadKey() в конце программы. Лишний раз задерживающая выполнение программы и мешающая понять, когда можно вводить текст, а когда нам просто дают время посмотреть результаты. Давайте передвинем эту команду за пределы цикла, то есть после команды Next счетчик.

```
            If число Mod 2 = 0 Then
                Console.WriteLine(запрещено)
            Else
                Console.WriteLine(разрешено)
            End If
            Console.ReadKey()
        Next счетчик
        Console.ReadKey()
    End Sub
End Module
```

И уберите Console.ReadKey() и Return из операторов условия. Потому что нам теперь совсем не нужно покидать программу после неправильного ввода текста.

```
If число < 1 Then
    Console.WriteLine(неверноеЗначение)
    Console.ReadKey()
    Return
End If
If число > 30 Then
    Console.WriteLine(неверноеЗначение)
    Console.ReadKey()
    Return
End If
```

Запустите программу (F5). Программа должна 5 раз просить вас ввести число и потом выходить при нажатии любой клавиши. Покажите результат преподавателю.

Цикл с предусловием

Однако проблема в том, что мы заранее не знаем, сколько раз будет выполняться программа. Как раз для таких случаев и существует цикл с предусловием Do While.

Формула этого цикла выглядит следующим образом:

Do While <условие>

<действия_пока_условие_является_истиной>

Loop

Действия внутри цикла будут выполняться до тех пор, пока <условие> не станет ложью. Разумеется, что условие должно меняться тоже внутри цикла. Иначе цикл станет бесконечным.

Итак, приступим. Dim текст As String поставим перед оператором цикла For.

```
Sub Main()
    Dim счетчик As Integer
    Dim текст As String
    For счетчик = 1 To 5
        Dim текст As String
        Dim число As Integer
```

Теперь заменим. Цикл For на цикл Do While с условием True, а так как счётчик больше не нужен – уберём его:

```
Sub Main()
    Dim счетчик As Integer
    Dim текст As String
    For счетчик = 1 To 5
        Do While True
            Dim число As Integer
            Dim разрешено, запрещено, неверноеЗначение As String
            разрешено = "Парковка в данном месте сегодня разрешена"
            запрещено = "Парковка в данном месте сегодня запрещена"
            неверноеЗначение = "Такого числа в этом месяце не существует. Допустимые значения: от 1 до 30"
            Console.WriteLine("Введите сегодняшнее число: ")
            текст = Console.ReadLine()
            число = Integer.Parse(текст)
            If число < 1 Then
                Console.WriteLine(неверноеЗначение)
            End If
            If число > 30 Then
                Console.WriteLine(неверноеЗначение)
            End If
            If число Mod 2 = 0 Then
                Console.WriteLine(запрещено)
            Else
                Console.WriteLine(разрешено)
            End If
        Loop
    Next счетчик
    Console.ReadKey()
End Sub
```

Теперь так как вместо условия у нас True, то цикл будет выполняться бесконечно. Потому что True – всегда Истина.

Однако необходимо, чтобы мы могли остановить работу программы. Допустим когда мы вместо числа – просто нажмём Enter.

Для этого мы должны заменить True на условие – если текст пустой (то есть ничего не введено), то не выполнять действия. Давайте попробуем объяснить это компьютеру. Пустая строка на языке компьютера может выглядеть так “” или String.Empty. Итак, если текст <> “” , то выполняем действия, иначе – выходим.

```
Do While текст <> ""
```

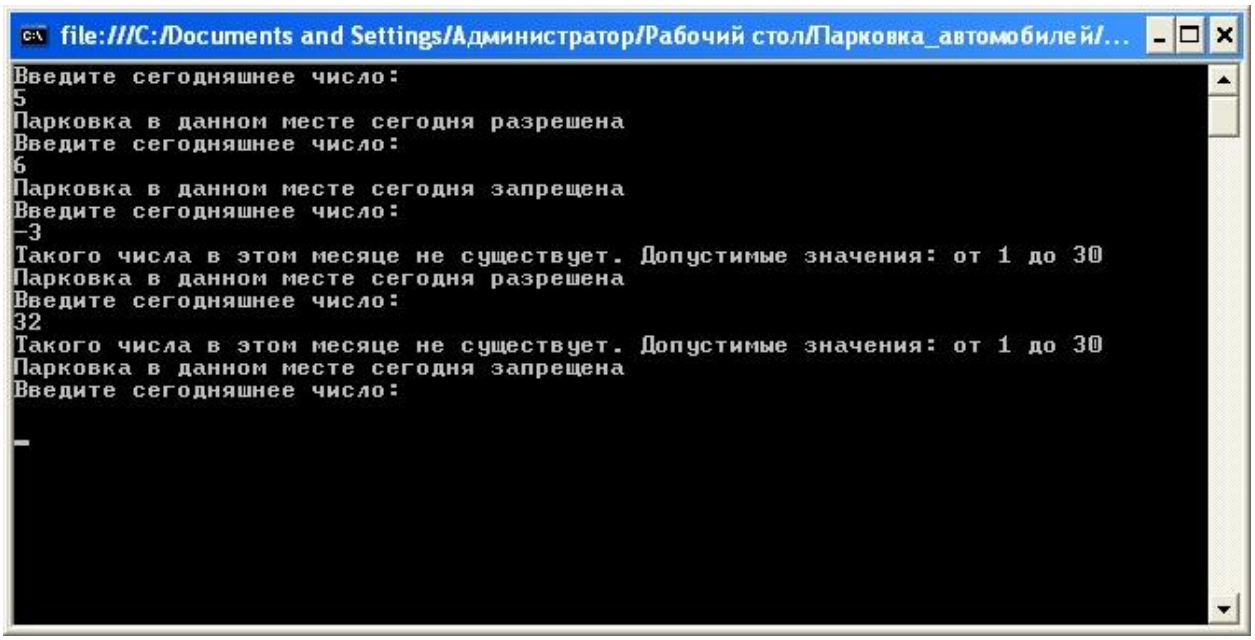
Однако текст уже в первый раз должен содержать в себе значение. Поэтому перед проверкой мы помещаем в текст значение введённое с клавиатуры и не забываем, конечно же, спросить об этом:

```
Sub Main()
    Dim текст As String
    Console.WriteLine("Введите сегодняшнее число: ")
    текст = Console.ReadLine()
    Do While текст <> ""
        Dim число As Integer
```

А чтобы это значение не спрашивалось два раза – переместим Console.WriteLine("Введите сегодняшнее число: ") и текст = Console.ReadLine() в самый конец цикла.

```
Do While текст <> ""
    Dim число As Integer
    Dim разрешено, запрещено, неверноеЗначение As String
    разрешено = "Парковка в данном месте сегодня разрешена"
    запрещено = "Парковка в данном месте сегодня запрещена"
    неверноеЗначение = "Такого числа в этом месяце не существует. Допустимые значения: от 1 до 30"
    Console.WriteLine("Введите сегодняшнее число: ")
    текст = Console.ReadLine()
    число = Integer.Parse(текст)
    If число < 1 Then
        Console.WriteLine(неверноеЗначение)
    End If
    If число > 30 Then
        Console.WriteLine(неверноеЗначение)
    End If
    If число Mod 2 = 0 Then
        Console.WriteLine(запрещено)
    Else
        Console.WriteLine(разрешено)
    End If
    Console.WriteLine("Введите сегодняшнее число: ")
    текст = Console.ReadLine()
Loop
```

Всё. Теперь если нажать F5, то программа должна спрашивать число до тех пор, пока не будет нажат просто Enter.



Покажите результат преподавателю.