

## Семинар 6. Циклы, часть 2

### 6.1 Цикл *do..while*

Этот цикл называется с циклом с постусловием.

```
do
{
    // несколько действий выполняемых подряд
    Оператор1;
    Оператор2;
    Оператор3;
} while (условие)
```

Разница с конструкцией `while` состоит лишь в том, что условие стоит в конце цикла, поэтому тело цикла будет выполнено, хотя бы один раз, вне зависимости от условия. В качестве примера цикла с постусловием приведем алгоритм, создающий пользовательское меню.

```
Int32 Choice = 0, Key = 0;
do{

    //Вывод меню на экран
    Console.Clear();
    Console.WriteLine("Уровень сложности:");
    Console.WriteLine(((Choice == 0) ? ">" : " ") +
        "Для слабаков");
    Console.WriteLine(((Choice == 1) ? ">" : " ") +
        "Реальный пацан");
    Console.WriteLine(((Choice == 2) ? ">" : " ") +
        "Супер продвинутый");
    Console.WriteLine(((Choice == 3) ? ">" : " ") +
        "Щемите монстры по лавкам");

    //Обработка ввода
    Key = (int)Console.ReadKey().Key;
    if (Key == 38) Choice--;           // кнопка вверх
    if (Key == 40) Choice++;           // кнопка вниз
    if (Key == 13) break;              // энтер

    //Обработка ошибки выхода за пределы
    if (Choice < 0) Choice = 3;
    if (Choice > 3) Choice = 0;

} while (Key != 27);
```

В строчке «`Key = (int)Console.ReadKey().Key;`» мы получаем код нажатой клавиши и выполняем явное преобразование полученного значения к целочисленному типу. У каждой кнопки на клавиатуре есть свой код, например, у кнопки вверх – 38, у кнопки вниз 40, у энтера – 13. Чтобы определить код любой клавиши, достаточно запустить следующий код и нажать эту клавишу.

```
Console.WriteLine((int)Console.ReadKey().Key);
Console.ReadLine();
```

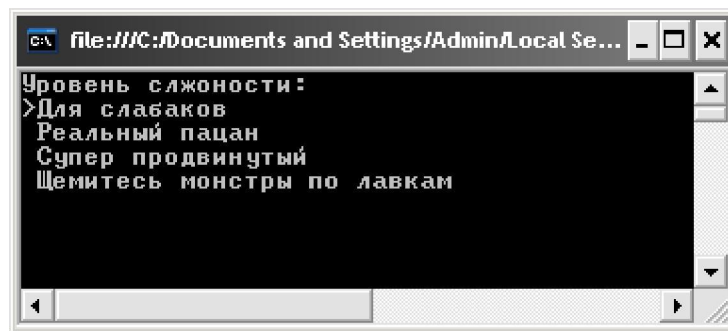


Рис. 6-1 Интерактивное меню в консоли, созданное с помощью do..while

## 6.2 Бесконечные циклы

Бесконечными называются такие циклы, условие выполнения которых никогда не сможет стать ложью. Выполняя бесконечный цикл, программа «зависает», и если программист не предусмотрел «черных выходов», прекратить ее выполнение средствами самой программы будет невозможно.

Примеры бесконечных циклов:  
while ( true ) действие;  
while ( 1 == 1 ) действие;  
do действие; while ( 5 < 6 )

И действительно, правда всегда будет правдой, единица всегда равна единице, а пять всегда меньше шести, даже после взрыва атомной бомбы .

Несмотря на кажущуюся бесполезность бесконечных циклов, они достаточно часто используются программистами. Как же выйти из бесконечного цикла? С помощью break! Такой выход из цикла будет нештатным, но почти единственно возможным для бесконечных циклов.

Для иллюстрации вышесказанного переделаем программу часовой, с использованием бесконечного цикла и оператора break.

```
String realpassword, userpassword;
realpassword = "secret";
while ( true )
{
    Console.WriteLine("Введите пароль");
    userpassword = Console.ReadLine();
    if (userpassword == realpassword) break;
    Console.WriteLine("Доступ запрещен");
}
Console.WriteLine("Доступ разрешен");
```

В этом примере мы запускаем бесконечный цикл и совершаем проверку правильности пароля сразу после его ввода. Если пароль окажется правильным – выходим из цикла оператором break. Если же пароль окажется неверным тело цикла будет выполнено еще раз и пользователю будет предложено ввести пароль еще раз.

## 6.3 Вложенные циклы

Циклические конструкции можно вкладывать друг в друга, наподобие того, как мы вкладывали условные операторы if. Вложенные циклы довольно частое явление в программировании. Для иллюстрации рассмотрим программу, выводящую на экран таблицу умножения от 1 до 9.

```

Console.WriteLine("Таблица умножения \r\n");
for (int i = 1; i < 10; i++)
{
    for (int j = 1; j < 10; j++)
    {
        Console.Write("{0:00} ", i*j);
    }
    Console.WriteLine();
}

```

Цикл, печатающий цифры в строку сам десять раз вызывается более старшим циклом, который перебирает уже сами строки. Шаблон подстановки {0:00} означает, что нужно вывести первый из указанных параметров в виде двухзначного числа.

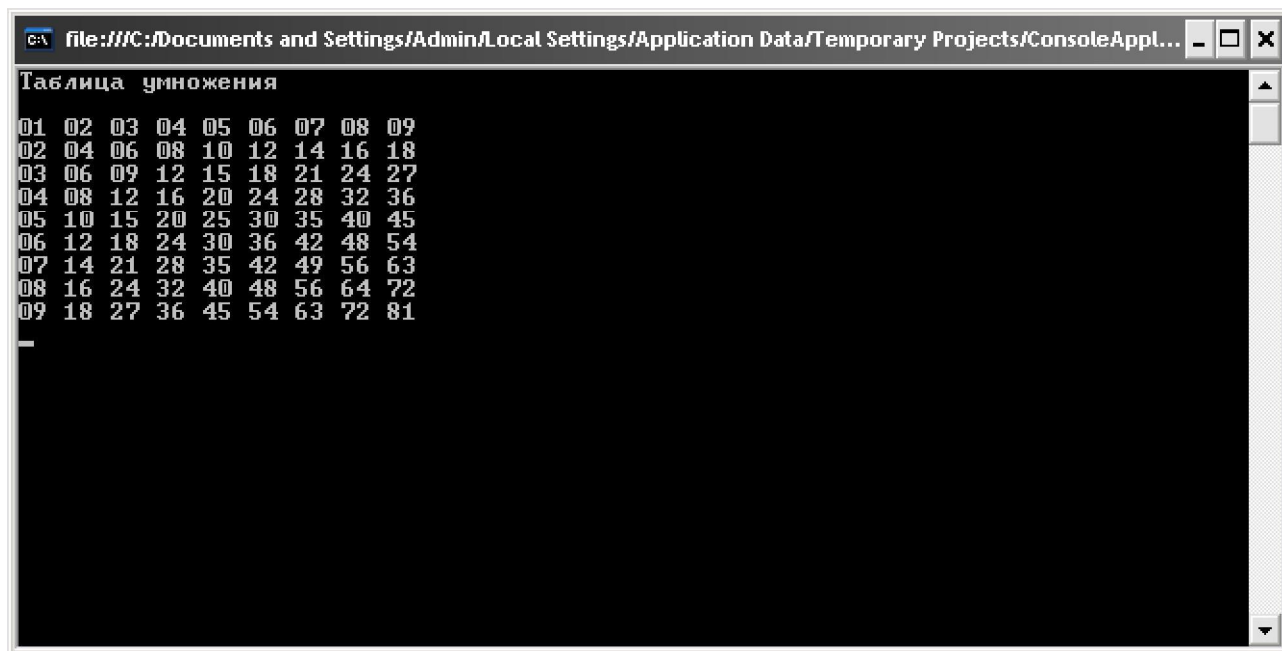


Рис. 6-2 Окно вывода программы «Таблица Умножения»

## 6.4 Примеры использования бесконечных и вложенных циклов (программа «Занудные окошки»)

Теперь, когда мы узнали так много нового, предлагаю размяться и применить новые знания к новым интересным задачкам. Скорее всего, ты видел хотя бы одну программу-прикол, которая выводит на экран окошко с надписью «Заккрыть» и кнопками «Да» и «Нет», допустим 100 раз. После того, как человек 100 раз нажмет на кнопку «Да», у него почти не будет шансов остановиться и нажать на кнопку «Нет», когда на 101-й раз окошко спросит его «Хочешь еще?».

Для того чтобы подопытный не смог выключить программу досрочно, нажав на крестик в углу, мы сделаем главное окно программы совершенно невидимым. Видимость окон настраивается в свойствах формы и называется Visible. Присвоив этому свойству значение false, мы получаем невидимую форму. Осталось научиться выводить окошки с кнопками. Эти окошки встроены в Windows и в языке C# вызываются с помощью операции MessageBox.Show(). Внутри скобок указывается текст в окне, надпись в заголовке, а также конфигурация окна, которая включает в себя такие мелочи как количество и тип кнопок, а также рисунок в окне.

Вызов окна с кнопками «Да» и «Нет» записывается так:

```

MessageBox.Show("Заккрыть?", "Кнопочки",
    MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);

```

Узнать, какая кнопка была нажата пользователем, можно сравнив результат, возвращенный окном, со специальными переменными вроде DialogResult.Yes или DialogResult.No.

```

if (DialogResult.Yes ==
    MessageBox(«да или нет», MessageBoxButtons.YesNo))
    // действие, если нажата кнопка Да

```

В программе мы безжалостно запустим бесконечный цикл, в который вложим цикл, показывающий 20 (пока) окошек с кнопочками подряд. Выходом из бесконечного цикла будет только нажатие в окошке «Хочешь еще?» кнопки «Нет».

Создай проект программы с графическим интерфейсом и сделай двойной щелчок по пустой форме. В открывшемся окне редактирования кода, найди конструкцию «private void Form1\_Load(object sender, EventArgs e) { }» и вставь внутрь фигурных скобок строки:

```

this.Visible = false; //делаем окно невидимым
while (true)           //начинаем бесконечный цикл
{
    //этот цикл 20 раз выводит окно «заккрыть»
    for (int i=0; i < 20; i++)
        MessageBox("Заккрыть окошко? ", "Сообщение",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Asterisk);
    //прерываем бесконечный цикл, если нажата кнопка «No»
    if (DialogResult.No ==
        MessageBox("Еще кнопочек?", "Сообщение",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Asterisk))
        break;
}

```

Запусти программу и потестируй. Перед тем как показать программу друзьям, измени параметр цикла с 20 до 100 или больше, им точно мало не покажется!

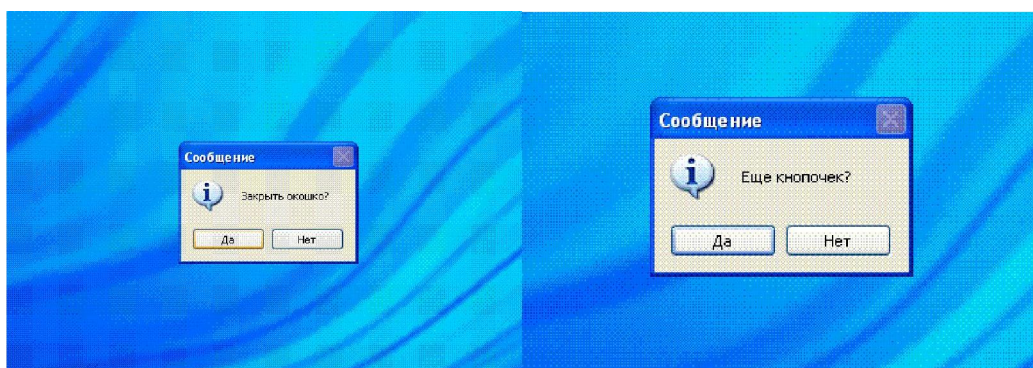


Рис. 6-3 Окошки программы «Кнопочки»

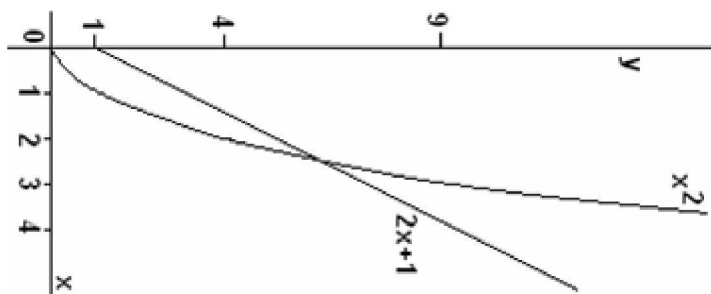


### Задания

**#1** Напиши программу, которая будет запрашивать у пользователя пароль до тех пор, пока он не введет правильный, или не напишет слово exit. Если пароль неправильный, должно появиться приветственное сообщение.

**#2** Напиши программу, которая выводит на экране в консольном режиме «шахматную доску» из символов.

**#3** Написать программу, которая заполняет таблицу для построения функций  $y_1 = 2x + 1$  и  $y_2 = x^2$ , при  $x = 0, 1, 2, 3, 4$ .



**#4** Напиши программу, которая выводит с помощью цикла бланк дневника

Пн	Вт	Ср	Чт	Пт
1	_____	_____	_____	_____
2	_____	_____	_____	_____
3	_____	_____	_____	_____
4	_____	_____	_____	_____
5	_____	_____	_____	_____
6	_____	_____	_____	_____

**#5** Создай игру, в которой два персонажа сражаются друг с другом (бетмен и черепашки-ниндзя). Бить можно по очереди в одну из трех зон (голова, корпус, ноги). У каждого персонажа разная уязвимость зон и количество жизни, отнимаемое в случае поражения зоны (уязвимость ног 2 случая из 5, но отнимает только 10% от жизни, уязвимость головы 1 случай из 5, но отнимает 25%). Игра продолжается, пока один из персонажей не побеждает.