

Урок 5

Темы урока:

- Что такое цикл?
- Цикл с параметром for
- Цикл с предусловием while
- Цикл с постусловием do while
- break, continue

Циклы

Очень часто в программировании нужно повторить многократно какой-то код или выражение. Например, нужно распечатать 10 раз выражение "Hello World." Это будет выглядеть следующим образом, если не использовать циклы:

```
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
Console.WriteLine("Hello World.");
```

Операторы цикла используются для организации многократно повторяющихся вычислений. К операторам цикла относятся: цикл с предусловием while, цикл с постусловием do while, цикл с параметром for и цикл перебора foreach.

Цикл с параметром for

Цикл с параметром имеет следующую структуру:

```
for ( инициализация; выражение; модификация)
{
    оператор;
    ...
}
```

} Тело цикла

Пример

```
for (int i = 1; i <= 10; i++)
{
    Console.WriteLine("Hello World.");
}
```

- **Инициализация** (`int i = 1`) используется для объявления и/или присвоения начальных значений величинам, используемым в цикле в качестве параметров (счетчиков). Инициализация выполняется один раз в начале исполнения цикла.
- **Выражение** (`i <= 10`) определяет условие выполнения цикла: если его результат истинен, цикл выполняется. Истинность выражения проверяется перед каждым выполнением тела цикла.
- **Модификация** (`i++`) выполняется после каждой итерации цикла и служит обычно для изменения параметров цикла.

Любая из частей оператора for (инициализация, выражение, модификация) может отсутствовать, но точку с запятой, определяющую позицию пропускаемой части, надо оставить.

Пример.

```
for (int i = 1; i <= 10;)
{
    Console.WriteLine("{0,-2}: Hello World.", i);
    i++;
}
```

```
C:\WINDOWS\system32\cmd.exe
1 : Hello World.
2 : Hello World.
3 : Hello World.
4 : Hello World.
5 : Hello World.
6 : Hello World.
7 : Hello World.
8 : Hello World.
9 : Hello World.
10: Hello World.
```

Цикл с предусловием *while*

Оператор цикла *while* организует выполнение оператора/операторов неизвестное заранее число раз. Формат цикла *while*:

```
while (условие)
{
    оператор;
    ...
}
```

Пример

```
int i = 1;
while (i <= 10)
{
    Console.WriteLine("{0,-2}: Hello World!", i);
    i++;
}
```

условие ($i \leq 10$) - выражение, истинность которого проверяется; перед каждым выполнением тела цикла анализируется значение условия: если оно истинно, то выполняется тело цикла, если условие ложно - цикл завершается и управление передается на оператор, следующий за телом цикла.

Если результат условия окажется ложным при первой проверке, то тело цикла не выполнится ни разу.



Важно!!! Условие во время работы цикла чаще всего не будет изменяться, тогда возможна ситуация заикливания, то есть невозможность выхода из цикла. Поэтому внутри тела должны находиться операторы (например, $i++$), приводящие к изменению условия, чтобы цикл мог корректно завершиться.

Пример заикливания:

```
while(true)
{
    //тело цикла
}
```

Цикл с постусловием *do while*

Оператор цикла *do while* также организует выполнение цикла неизвестное заранее число раз. Однако в отличие от цикла *while* условие завершения цикла проверяется после выполнения тела цикла. Формат цикла *do while*:

```
do
{
    оператор;
    ...
} while (условие);
```

Сначала выполняется тело цикла, а затем анализируется условие: если оно истинно, то выполняется опять тело цикла, если ложно - цикл завершается, и управление передается оператору, следующему за условием.

Так как условие проверяется после выполнения тела цикла, то в любом случае тело цикла выполнится **хотя бы один раз**.

В операторе `do while`, так же как и в операторе `while`, возможна ситуация заикливания в случае, если условие всегда будет оставаться истинным.

Пример.

```
int number = 0;
do
{
    Console.WriteLine("Enter a number greater than 10: ");
    number = Convert.ToInt32(Console.ReadLine());
} while (number < 10);
Console.WriteLine("Exit...");
```

Операторы `break` и `continue`

Оператор `break` используется внутри операторов ветвления и цикла для обеспечения перехода в точку программы, находящуюся непосредственно за оператором, внутри которого находится `break`.

Оператор перехода к следующей итерации цикла `continue` пропускает все операторы, оставшиеся до конца тела цикла, и передает управление на начало следующей итерации (повторение тела цикла).

Пример.

```
Console.WriteLine("Demonstrating the use of break.\n");
for (int x = 1; x < 10; x++)
{
    if (x == 5)
    {
        break;
    }
    Console.WriteLine("Number " + x);
}
Console.WriteLine("\nDemonstrating the use of continue.\n");
for (int x = 1; x < 10; x++)
{
    if (x == 5)
    {
        continue;
    }
    Console.WriteLine("Number " + x);
}
```

```
C:\WINDOWS\system32\cmd.exe
Demonstrating the use of break.
Number 1
Number 2
Number 3
Number 4

Demonstrating the use of continue.
Number 1
Number 2
Number 3
Number 4
Number 6
Number 7
Number 8
Number 9
```

Задание

1. Выведите на экран 10 случайных чисел от 1 до 100. Посчитайте сумму этих чисел.
2. **Угадай число.** Напишите программу, которая "задумывает" число в диапазоне от 1 до 10 и предлагает пользователю угадать. Распечатайте количество попыток.
3. **Игра Крепс**

Одной из наиболее популярных азартных игр считается игра в кости, известная как «крепс», в которую играют и в казино, и в глухих закоулках по всему миру. Правила игры просты:

Игрок бросает две кости. Каждая кость имеет шесть граней. Эти грани помечены как 1, 2, 3, 4, 5 и 6. После броска вычисляется сумма цифр двух верхних граней. Если сумма после первого броска равна 7 или 11, игрок выиграл. Если после первого броска сумма равна 2, 3 или 12 (это называется «крепс»), игрок проигрывает (т. е. выигрывает «банк»). Если после первого броска сумма равна 4, 5, 6, 8, 9 или 10, то эта сумма становится «очками» игрока. Чтобы выиграть, игрок должен продолжать бросать кости до тех пор, пока не выпадет сумма, равная его очкам. Игрок проигрывает, если во время этих бросков ему выпадет сумма 7.

Пример. Игра 21

```

static void Main(string[] args)
{
    int compCard;
    int gamerCard;
    int sumGamer = 0;
    int sumComp = 0;
    char suit;
    Console.BackgroundColor = ConsoleColor.DarkGreen;
    Console.Clear();
    Console.CursorVisible = false;
    Random rnd = new Random();
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("\n\n***** Games 21 *****\n\n");
    Console.WriteLine("{0,15}{1,30}", "You: ", "Computer: \n");
    while (sumGamer <= 21 && sumComp <= 21)
    {
        int ch = rnd.Next(0, 2); //0..1
        gamerCard = ch == 0 ? rnd.Next(6, 12) /*6..11*/ : rnd.Next(2, 5) /*2..4*/;
        sumGamer = sumGamer + gamerCard;
        suit = Convert.ToChar(rnd.Next(3, 7));
        if ((int)suit == 3 || (int)suit == 4)
        {
            Console.ForegroundColor = ConsoleColor.Red;
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.Black;
        }
        System.Threading.Thread.Sleep(1000);
        switch (gamerCard)
        {
            case 2: Console.Write("{0,8}{1}\t({2})", 'V', suit, sumGamer); break;
            case 3: Console.Write("{0,8}{1}\t({2})", 'D', suit, sumGamer); break;
            case 4: Console.Write("{0,8}{1}\t({2})", 'K', suit, sumGamer); break;
            case 11: Console.Write("{0,8}{1}\t({2})", 'T', suit, sumGamer); break;
            default: Console.Write("{0,8}{1}\t({2})", gamerCard, suit, sumGamer); break;
        }
        ch = rnd.Next(0, 2); //0..1
        compCard = ch == 0 ? rnd.Next(6, 12) /*6..11*/ : rnd.Next(2, 5) /*2..4*/;
        sumComp = sumComp + compCard;
        suit = Convert.ToChar(rnd.Next(3, 7));
        if ((int)suit == 3 || (int)suit == 4)
        {
            Console.ForegroundColor = ConsoleColor.Red;
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.Black;
        }
        System.Threading.Thread.Sleep(1000);
        switch (compCard)
        {
            case 2: Console.WriteLine("\t{0,10}{1}\t({2})", 'V', suit, sumComp); break;
            case 3: Console.WriteLine("\t{0,10}{1}\t({2})", 'D', suit, sumComp); break;
            case 4: Console.WriteLine("\t{0,10}{1}\t({2})", 'K', suit, sumComp); break;
            case 11: Console.WriteLine("\t{0,10}{1}\t({2})", 'T', suit, sumComp); break;
            default: Console.WriteLine("\t{0,10}{1}\t({2})", compCard, suit, sumComp); break;
        }
    }
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("\n\n*****\n\n");
    Console.SetCursorPosition(Console.CursorLeft + 15, Console.CursorTop);
    if (sumGamer <= 21)
    {
        Console.WriteLine("You winner!!!");
    }
    if (sumComp <= 21)
    {
        Console.WriteLine("Computer winner!!!");
    }
    if (sumGamer > 21 && sumComp > 21)
    {
        for (int i = 0; i < 20; i++)
        {
            Console.Write(".");
            System.Threading.Thread.Sleep(100);
        }
    }
    Console.ReadLine();
}
}

```