**Урок № 3. C# Теория**

**ЦИКЛЫ**

 В Visual C# есть возможность использовать несколько видов циклов. Это for, while и do.

**ЦИКЛЫ FOR В СИ ШАРП**

 Цикл повторяет блок кода до тех пор, пока условие цикла не станет ложным. Каждое повторение — итерация цикла. Синтаксис:

for ([инициализация счетчика]; [условие цикла]; [изменение счётчика])

{

// повторяющийся код

}

 Блок[инициализация счетчика] используется, соответственно, для инициализации счетчика цикла for. Каждую итерацию цикл проверяет значение счетчика и сравнивает его с [условием цикла]. Если счетчик находится в указанном диапазоне, то цикл запускается на выполнение опять. Каждую итерацию блок [изменение счётчика] изменяет значение счетчика цикла, либо увеличивая, либо уменьшая его.

|  |
| --- |
| **for** (**int** i = 0 ; i < 10; i++)  {  *// тело цикла*  } |

В указанном примере:  
i = 0; это инициализация счетчика,  
i < 10; это условие цикла, и  
i++ это изменение счётчика.

**FOREACH ЦИКЛ**

Рассмотрим итерацию по коллекции или массиву значений. При использовании цикла For необходимо знать, сколько элементов находится в коллекции или массиве. Но иногда коллекции или массивы могут быть динамическими и не иметь размера во время компиляции. Если размер коллекции или массива изменяется во время выполнения, лучше использовать цикл foreach.  
Пример использования цикла foreach для итерации строкового массива:

|  |
| --- |
| **string**[] names = new **string**[10];  *// Process each name in the array.*  **foreach** (**string** name **in** names)  {  *// Code to execute.*  } |

**Задания и лабораторные си шарп**

**Лабораторная работа 1. Цикл for. Счетчик цикла**

**Выполнить:** Запросить пользователя ввести число (N). Создайте простой цикл for с N повторениями, который отображает значения счетчика цикла.

**Пример выполнения:**

Введите число

3

Результат:

Счетчик: 1

Счетчик: 2

Счетчик: 3

Введите число

1

Результат:

Счетчик: 1

[Название проекта: Lesson\_3Lab1, название файла L3Lab1.cs]

 Алгоритм:

* Запустите Visual Studio.
* Создайте консольное приложение, назовите проект Lesson\_3Lab1.
* В окне *Solution Explorer* (Обозреватель решений) найдите файл Program.cs и переименуйте его в L3Lab1.cs.
* Установите курсор непосредственно после открытой скобки функции Main, щелкните *enter*, чтобы перейти на новую строку.
* Попросите пользователя ввести число, используйте метод ReadLine() :

|  |
| --- |
| **static** **void** Main(**string**[] args)  {  Console.WriteLine("Введите число");  } |

* Присвойте введенное значение переменной N. Конвертируйте переменную N в целое значение:

|  |
| --- |
| ...  **int** N = Int32.Parse(Console.ReadLine());  ... |

* Создайте цикл *for* с N количеством итераций. В качестве счетчика цикла используйте переменную с идентификатором counter (для быстрого создания цикла используйте for + Tab(дважды))

...

for(int counter = 1; counter <= N; counter++)

{

...

}

...

* Output the values of the variable counter within the loop:

...

for(int counter = 1; counter <= N; counter++)

{

Console.WriteLine($"Счетчик: {counter}");

}

...

* Используйте горячие клавиши CTRL+F5 для запуска приложения без отладки.
* Откроется консольное приложение с запросом на ввод числа.
* Форматировать код с отступами можно, последовательно нажимая Ctrl+A затем Ctrl+K и Ctrl+F.
* Поэкспериментируйте, вводя различные значения и наблюдая результат.
* Не забудьте скопировать текст задания и оформить его в виде комментария до основного кода.

**Задание 1:**

**Выполнить:** Выведите в консоль последовательность чисел -3 0 3 6 9 12 15 18 21 24. Выполните задание, используя цикл FOR. *Изменение счетчика* должно соответствовать увеличению его на **3**.  
    
**Указание:** Изменение шага цикла на **3**, можно выполнить следующим образом:

counter+=3

[Название проекта: Lesson\_3Task1, название файла L3Task1.cs]

**Пример выполнения:**

Последовательность : -3 0 3 6 9 12 15 18 21 24

**Задание 2:**

**Выполнить:** Выведите в консоль следующую последовательность: 1 2 3 4 . . . 99 100 99 . . . 3 2 1.  
    
**Указание 1:** Создайте два цикла for: один - для последовательности 1 2 3 4 . . . 99 100 и второй - для последовательности 99 . . . 3 2 1 (с шагом цикла i-- для счетчика второго цикла).  
<="" strong="" style="box-sizing: border-box;">Чтобы вывести все значения в одну строку, следует использовать метод:

|  |
| --- |
| Console.Write($"{i} "); |

**Пример выполнения:**

Последовательность : 1 2 3 4 5 . . . 99 100 99 . . . 4 3 2 1

[Название проекта: Lesson\_3Task2, название файла L3Task2.cs]

**Задание 3:**

**Выполнить:** Запросите ввести 10 целых чисел. Необходимо вывести количество положительных и отрицательных чисел среди введенных.  
    
**Указание 1:** Для ввода чисел создайте цикл for. В теле цикла проверяйте, положительное ли число введено или отрицательное. Используйте две переменных-счетчика (увеличивающихся на единицу) для подсчета количества положительных и отрицательных чисел.

**Указание 2:** Не забудьте конвертировать переменную с вводимым числом в тип integer (Int32.Parse(...)).  
   
**Пример выполнения:**

1 -5 -12 2 3 9 -1 9 5 -8 => количество положительных = 6, количество отрицательных = 4

[Название проекта: Lesson\_3Task3, название файла L3Task3.cs]

**Задание 4:**

**Выполнить:** Запросите ввести 10 целых чисел. Выведите сумму введенных чисел.  
    
**Указание 1:** Создайте цикл for для ввода чисел. Для подсчета суммы используйте переменную с идентификатором sum.

**Указание 2:** Не забудьте преобразовать переменную с числами в тип integer (Int32.Parse(...)).

**The result example:**

1 -5 -12 2 3 9 -1 9 5 -8 => сумма = 3

[Название проекта: Lesson\_3task4, название файла L3Task4.cs]

**Задание 5:**

**Выполнить:** Программа должна считать сумму 10 чисел следующей последовательности: 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19 (числа последовательности НЕ вводятся, а получаются в цикле).  
    
**Указание:** Для подсчета суммы используйте переменную сидентификатором sum.

[Название проекта: Lesson\_3task5, название файла L3Task5.cs]  
    
**Пример выполнения:**

1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19 => сумма = 100

**Задание 6:**

**Выполнить:** Вводятся 10 вещественных чисел. Вывести их произведение.  
    
**Указание 1:** Для ввода чисел используйте цикл for. Для подсчета произведения используйте переменную с идентификатором product.

**Указание 2:** Не забудьте инициализировать переменную product значением вещественного типа (double product = 1.0;). Переменную для вводимых чисел конвертируйте также в вещественный тип (Double.Parse(Console.ReadLine())).  
   
[Название проекта: Lesson\_3Task6, название файла L3Task6.cs]  
   
**Пример выполнения:**

1,1 2,4 5,1 7,2 6,4 8,1 6,7 3,2 3,3 2,4 => произведение = 853338,921998746

**Лабораторная работа 2. Вложенные циклы. Простые числа**

**Выполнить:** Используйте вложенный цикл for для нахождения всех простых чисел, меньших **100** (начиная с **2**).  
    
**Пример выполнения:**

Простые числа:

2

3

5

7

11

...

97

[Название проекта: Lesson\_3Lab2, название файла L3Lab2.cs]

 Алгоритм:

* Выполните первые три пункта точно так же, как вы их сделали для предыдущей лабораторной работы. Назовите проект в соответствии с заданием.
* Выведите в консоль фразу *"Простые числа: "*:

|  |
| --- |
| **static** **void** Main(**string**[] args)  {  Console.WriteLine("Простые числа: ");  } |

* Объявите две переменные, которые будут использованы в качестве счетчиков циклов:

...

int outer;

int inner;

...

* Создайте внешний цикл for со счетчиком outer. Этот цикл будет просматривать все числа в диапазоне [2,99] (**можно использовать сочетание клавиш для быстрого создания кода for + Tab(дважды)**):

...

for (outer = 2; outer < 100; outer++)

{

...

}

...

* Создайте вложенный цикл for (внутри предыдущего) со счетчиком inner. Данный цикл будет проверять каждое число диапазона, является ли оно простым. Если оно является таковым, то его значение выводится в консоль:

...

for (outer = 2; outer < 100; outer++)

{

for (inner = 2; inner <= (outer / 2); inner++)

{

if ((outer % inner) == 0) break; // if factor found, not prime

}

if (inner > (outer / inner))

{

Console.WriteLine("{0} is prime", outer);

}

}

...

Как проверить, является ли число простым?

* Простое число - это натуральное число большее **1**, которое не может быть получено путем умножения двух меньших натуральных чисел.
* Например, **5** - это *простое* число, потому что его можно получить только выполнив следующие произведения: 1 × 5 или 5 × 1
* А вот strong>6 можно получить произведением двух чисел - 2 × 3, которые меньше, чем **6**.
* Все составные числа, которые могут быть разложены на множители, представлены произведением простых чисел; то есть все их множители — простые числа. Ибо, если найти множитель, который не является простым числом, он всегда может быть разложен и представлен двумя или более простыми числами.
* Запустите приложение без отладки CTRL+F5.
* Не забудьте скопировать текст задания и оформить его в виде комментария до основного кода.

**Задание 7:**

**Выполнить:** Для каждого x в интервале **[2;8]** найдите значение функции z(x,y) = xy. Переменная y изменяется в интервале **[2;5]**.  
    
**Указание 1:** Создайте два цикла for (один из них - вложенный). Переменнаяx должна изменяться во внешнем цикле, тогда как переменнаяy должна меняться во внутреннем.  
**Указание 2:** Чтобы посчитать степень числа, необходимо подключить класс Math:

|  |
| --- |
| **using** **static** System.Math;  ...  Math.Pow(x,y); |

**Пример выполнения:**

z(x,y) = 2^2 = 4

z(x,y) = 2^3 = 8

z(x,y) = 2^4 = 16

z(x,y) = 3^2 = 9

z(x,y) = 3^3 = 27

z(x,y) = 3^4 = 81

z(x,y) = 4^2 = 16

z(x,y) = 4^3 = 64

z(x,y) = 4^4 = 256

z(x,y) = 5^2 = 25

z(x,y) = 5^3 = 125

z(x,y) = 5^4 = 625

... и т.д.

[Название проекта: Lesson\_3Task7, название файла L3Task7.cs]

**Задание 8:**

**Выполнить:** Для каждого x в интервале **[30;33]** найдите значение функции z(x,y) = x - y. Переменная y изменяется в интервале **[1;5]**.  
    
**Указание:** Создайте два цикла for (один из них - вложенный). Переменнаяx должна изменяться во внешнем цикле, тогда как переменнаяy должна меняться во внутреннем.

**Пример выполнения:**

z(x,y) = 30-1=29

z(x,y) = 30-2=28

z(x,y) = 30-3=27

z(x,y) = 30-4=26

z(x,y) = 30-5=25

z(x,y) = 31-1=30

... и т.д.

[Название проекта: Lesson\_3Task8, название файла L3Task8.cs]