**Урок № 4. C# Теория**

**ПОВТОРЕНИЯ | ЦИКЛЫ**

* В программировании часто приходится использовать повторение для перебора элементов в коллекции или для выполнения одной и той же задачи снова и снова.
* Цикл while позволяет выполнить блок кода при истинности заданного условия цикла. Например, можно использовать цикл *while* для ввода данных пользователем до тех пор, пока пользователь не укажет, что вводить больше нечего. Цикл может продолжать запрашивать пользователя вводить значения до тех пор, пока он не решит завершить это действие, введя специальное значение. В этом случае специальное значение отвечает за завершение цикла.

|  |
| --- |
| **string** answer = Console.ReadLine();  **while** (answer != "Выход") *// "Выход" - специальное значение для завершения цикла*  {  *// тело цикла*  answer = Console.ReadLine();  } |

* Необходимо поместить оператор answer = PromptUser(); внутри скобок цикла. Отсутствие данного оператора в теле цикла приведет к бесконечному циклу, потому что значение условия цикла никогда не поменяется.

**ЦИКЛ DO..WHILE**

* Цикл do..while очень похож на цикл while, за исключением того, что цикл do обязательно выполнит тело цикла по крайней мере один раз. В то время как в цикле while, если условие ложно с самого начала, тело цикла никогда не выполнится.
* Использовать цикл do..while следует тогда, когда известно, что код будет выполняться только в ответ на запрос пользователя о предоставлении данных. В этом случае вы знаете, что приложению потребуется обработать хотя бы один ввод данных.

|  |
| --- |
| **do**  {  *// тело цикла*  answer = Console.ReadLine();  } **while** (answer != "Выход"); |

**Задания и лабораторные си шарп**

**Лабораторная работа 1. Цикл While**

**Выполнить:** Запросите у пользователя число (N). Создайте простой цикл while с N итерациями (повторениями), который отображает значения счетчика цикла.

**Пример выполнения:**

Введите число

3

Результат:

Текущее значение счетчика: 1

Текущее значение счетчика: 2

Текущее значение счетчика: 3

Введите число

1

Результат:

Текущее значение счетчика: 1

[Название проекта: Lesson\_4Lab1, название файла L4Lab1.cs]

 Алгоритм:

* + Запустите Visual Studio.
  + Создайте консольное приложение, назовите проект Lesson\_4Lab1.
  + В окне *Solution Explorer* (Обозреватель решений) найдите файл Program.cs и переименуйте его в L4Lab1.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. Переведите переменную N в целочисленное значение:

...

int N = Int32.Parse(Console.ReadLine());

...

* + Начнем со значения counter = 1. Создайте цикл while с N итерациями. Используйте переменную с именем counter в качестве счетчика цикла *while* (можно использовать быстрый ввод while + Tab (дважды))::

...

int counter = 1;

while (counter <= N)

{

counter++;

}

...

* + Условие цикла - если счетчик меньше N или равен N, то код тела цикла выполняется, мы выводим значение счетчика и затем увеличиваем его на 1.
  + Выводите значения счетчика внутри цикла:

...

int counter = 1;

while (counter <= N)

{

Console.WriteLine($"Текущее значение счетчика: {counter}");

counter++;

}

...

* + Используйте сочетание клавиш CTRL+F5, чтобы запустить приложение без отладки.
  + Это приведет к тому, что Visual Studio скомпилирует код и запустит приложение. Откроется окно консоли, в котором вам будет предложено ввести целочисленное значение.
  + Поэкспериментируйте с различными значениями, установив счетчик на значение больше N. Обратите внимание на выходные данные.
  + Отформатируйте код, нажав Ctrl+A, затем Ctrl+K, а затем Ctrl+F.
  + Не забудьте разместить текст задания в виде комментария перед программой.
  + Чтобы загрузить файл в систему *moodle*, найдите папку решения на компьютере (*d:\Projects\Lesson\_4Lab1\*) и загрузить файл *L4Lab1.CS*.

**Лабораторная работа 2. Цикл Do..while. Значения счетчика**

**Выполнить:** Создайте цикл Do..while с 5 итерациями (повторениями), который выводит значения счетчика цикла.

**Пример выполнения:**

Результат:

Текущее значение счетчика: 1

Текущее значение счетчика: 2

Текущее значение счетчика: 3

Текущее значение счетчика: 4

Текущее значение счетчика: 5

[Название проекта: Lesson\_4Lab2, название файла L4Lab2.cs]

 Алгоритм:

* + Запустите Visual Studio.
  + Создайте консольное приложение, назовите проект Lesson\_4Lab2.
  + В окне *Solution Explorer* (Обозреватель решений) найдите файл Program.cs и переименуйте его в L4Lab2.cs.
  + Установите курсор непосредственно после открытой скобки функции Main, щелкните *enter*, чтобы перейти на новую строку.
  + Начнем со счетчика counter = 1. Создайте цикл Do..While с 5 итерациями. Используйте переменную с именем counter в качестве счетчика цикла (для создания цикла можно воспользоваться быстрой вставкой кода do + Tab (дважды)):

...

int counter = 1;

do

{

counter++;

}

while (counter <= 5);

...

* + Условие цикла - если счетчик меньше N или равен N - выполнится уже после вывода первого значения счетчика. В цикле выводите значения счетчика:
  + Output the values of the variable counter within the loop:

...

int counter = 1;

do

{

Console.WriteLine($"Текущее значение счетчика: {counter}");

counter++;

}

while (counter <= 5);

...

* + Используйте сочетание клавиш CTRL+F5, чтобы запустить приложение без отладки.
  + Это приведет к тому, что Visual Studio скомпилирует код и запустит приложение.
  + Поэкспериментируйте с различными значениями, установив счетчик в другое значение. Обратите внимание на выходные данные.
  + Отформатируйте код, нажав Ctrl+A, затем Ctrl+K, а затем Ctrl+F.
  + Не забудьте разместить текст задания в виде комментария перед программой.
  + Чтобы загрузить файл в систему *moodle*, найдите папку решения на компьютере (*d:\Projects\Lesson\_4Lab2\*) и загрузить файл *L4Lab2.CS*.

**Задание 1:**

**Выполнить:** Выведите последовательность *3 5 7 9 ... 21* (от *3* до *21* с шагом = *2*). Сделайте это дважды: используя while и do..while цикл.  
    
**Указание 1:** В одном проекте создайте два цикла одновременно: while и do..while. Счетчики цикла должны иметь разные идентификаторы (counter1, counter2).  
    
**Указание 2:** Для вывода значений используйте метод:

Console.Write(...);

**Пример выполнения:**

3 5 7 9 11 13 15 17 19 21

[Название проекта: Lesson\_4Task1, название файла L4Task1.cs]

**Задание 2:**

**Выполнить:** Выведите последовательность *15 12 9 6 3 0* (от *15* до *0* с шагом = *-3*). Сделайте это дважды: используя цикл while и do..while.  
    
**Указание:** В одном проекте создайте два цикла одновременно: while и do..while. Счетчики цикла должны иметь разные идентификаторы (counter1, counter2).  
    
   
**Пример выполнения:**

15 12 9 6 3 0

[Название проекта: Lesson\_4Task2, название файла L4Task2.cs]

**Задание 3:**

**Выполнить:** Вычислите умножение 2-значных четных чисел в интервале *[10;20]* (*10 \* 12 \* 14 \* 16 \* 18 \* 20*). Выполните задание с помощью цикла while.  
    
**Указание:** Для вычисления умножения необходимо использовать переменную с идентификатором product. Начните со значения product = 1.

**Пример выполнения:**

10 \* 12 \* 14 \* 16 \* 18 \* 20 = 9676800

[Название проекта: Lesson\_4Task3, название файла L4Task3.cs]

**Задание 4:**

**Выполнить:** Вводятся *5* вещественных чисел. Вычислите сложение введенных чисел. Выполните задание с помощью цикла while.  
    
**Указание 1:** For real numbers and for variable sum you have to use the variable of double type:

double sum = 0;

double numb;

**Указание 2:** Для вещественных чисел и для переменной суммы следует использовать переменную вещественного типа double:

numb = Double.Parse(Console.ReadLine());

**Пример выполнения:**

Введите 5 вещественных чисел

1.1 4.2 2.1 9.3 2.5

Сумма введенных чисел = 19.2

[Название проекта: Lesson\_4Task4, название файла L4Task4.cs]

**Лабораторная работа 3. Цикл Do..while**

**Выполнить:** Вводится последовательность целых чисел. Признаком завершения последовательности является введенное число *0* (если введено *0*, то ввод последовательности завершен). Требуется найти минимальное среди чисел последовательности. Для выполнения задания используйте цикл do..while. Не допускается использование стандартной функции min.

**Пример выполнения:**

Введите последовательность чисел и закончите ввод нулем

3 5 1 0

минимальное значение 1

[Название проекта: Lesson\_4Lab3, название файла L4Lab3.cs]

 Алгоритм:

* + Запустите Visual Studio.
  + Создайте консольное приложение, назовите проект Lesson\_4Lab3.
  + В окне *Solution Explorer* (Обозреватель решений) найдите файл Program.cs и переименуйте его в L4Lab3.cs.
  + Попросите пользователя ввести числа:

|  |
| --- |
| ...  Console.WriteLine("Введите последовательность чисел и закончите ввод нулем");  ... |

* + Объявите переменную для вводимых чисел (numb). Инициализируйте переменную для хранения минимума (min) максимальным числом среди типа int:

...

int numb;

int min= int.MaxValue;

...

* + Добавьте цикл do..while с условием для проверки, не равно ли введенное число нулю:

...

do

{

...

}

while (numb != 0);

...

* + Условие цикла проверяет, не равно ли значение переменной numb нулю, если не равно, то выполняется тело цикла.
  + В теле цикла используйте метод ReadLine() для ввода чисел. Не забывайте преобразовать значение в целое число:

...

numb = Int32.Parse(Console.ReadLine());

...

* + После этого вы должны проверить введенное число, меньше ли оно минимального и не равно ли нулю (потому что *0* означает конец последовательности):

...

numb = Int32.Parse(Console.ReadLine());

if (numb < min && numb!=0)

{

min = numb;

}

...

* + Теперь промоделируем ситуацию, что введенное значение было строковым, и в таком случае возникнет исключение. Чтобы избежать ошибки введите конструкцию try..catch:

try

{

do

{

numb = Int32.Parse(Console.ReadLine());

if (numb < min && numb != 0)

{

min = numb;

}

}

while (numb != 0);

}

catch (**FormatException** e)

{

Console.WriteLine(e.Message + " Повторите ввод");

}

* + Запустите программу и введите строковое значение. Посмотрите на результут.
  + Выведите минимальное значение (min) после тела цикла:

...

Console.WriteLine($"минимальное значение {min}");

...

* + Используйте горячие клавиши CTRL+F5 для запуска приложения без отладки.
  + 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 5:**

**Выполнить:** Вводится последовательность целых чисел. Признаком завершения последовательности является введенное число 0 (если введено 0, то ввод чисел завершен). Выведите максимальное число последовательности и его порядковый номер в последовательности. Для выполнения программы используйте цикл Do..while. Не допускается использование стандартной функции max.

**Пример выполнения:**

Введите последовательность чисел и закончите ввод нулем

3 5 1 0

максимальное число 5, его порядковый номер 2

[Название проекта: Lesson\_4Task5, название файла L4Task5.cs]

**Дополнительные задания**

**Дополнительное задание 1:**

**Выполнить:** Вычислите значение функции y = 4(x-3)6-7(x-3)3 + 2 для введенного значения x. Используйте вспомогательную переменную для хранения выражения (x-3). Сделайте программу для 3-х введенных значений x (используйте цикл).  
    
**Указание:** Добавьте к проекту классы console и math:

|  |
| --- |
| **using** **static** System.Console;  **using** **static** System.Math; |
|  |

    
**Пример выполнения:**

Введите x

2

x=2;y=13

+++++

Введите x

3

x=3;y=2

+++++

Введите x

4

x=4;y=-1

[Название проекта: Lesson\_4ExTask1, название файла L4ExTask1.cs]

**Дополнительное задание 2:**

**Выполнить:** Даны координаты шахматного поля x и y (целые числа в диапазоне *1-8*). Известно, что нижний левый квадрат доски (*(1, 1)*) черный, проверьте, правда ли, что: *"этот квадрат белый"* (используя введенные координаты). Нельзя использовать условный оператор.  
    
**Указание:** Добавьте к проекту классы console и math:

|  |
| --- |
| **using** **static** System.Console;  **using** **static** System.Math; |

**Пример выполнения:**

введите x (1-8)

3

введите y (1-8)

4

этот квадрат белый: true

введите x (1-8)

3

введите y (1-8)

5

этот квадрат белый: false

[Название проекта: Lesson\_4ExTask2, название файла L4ExTask2.cs]

**Дополнительное задание 3:**

**Выполнить:** Создайте поле шахматной доски размером *8 х 8*. Используйте X и O для представления квадратов. Создайте соответствующую вложенную циклическую структуру для вывода символов в сетке *8 x 8* с помощью метода Write() или WriteLine(), когда это необходимо. Используйте условную конструкцию, чтобы гарантировать, что альтернативные строки начинаются с противоположных символов.  
    
   
**Пример выполнения:**

XOXOXOXO

OXOXOXOX

XOXOXOXO

OXOXOXOX

XOXOXOXO

OXOXOXOX

XOXOXOXO

OXOXOXOX

[Название проекта: Lesson\_4ExTask3, название файла L4ExTask2.cs]