

**УО «МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ А.А. КУЛЕШОВА»
СОЦИАЛЬНО-ГУМАНИТАРНЫЙ КОЛЛЕДЖ**



**Дисциплина
«Конструирование программ и языки программирования»**

**Разработка программ с использованием операторов цикла
(2 часа)**

Методические рекомендации к лабораторной работе № 4

Могилев 2018

Понятие «Операторы цикла». Методические указания по лабораторной работе № 4 по дисциплине «Конструирование программ и языки программирования». Для учащихся 3 курса очной формы обучения специальности 2–40 01 01 «Программное обеспечение информационных технологий».

Оглавление

1 Цель работы	4
2 Краткие теоретические сведения	5
2.1 Цикл for	6
2.1.1. Прерывание цикла	6
2.1.2. Возобновление цикла	7
2.2 Цикл while	8
2.3 Цикл do while	8
2.4 Цикл foreach	9
3. Задания	11
4 Контрольные вопросы	13

1 Цель работы

выработать умение разрабатывать программы с использованием операторов цикла

2 Ход работы

1. Изучение теоретического материала.
2. Выполнение практических индивидуальных заданий по вариантам (вариант уточняйте у преподавателя).
3. Оформление отчета.
 - 3.1. Отчет оформляется индивидуально каждым студентом. Отчет должен содержать задание, алгоритм и листинг программы.
 - 3.2. Отчет по лабораторной работе выполняется на листах формата А4. В состав отчета входят:
 - 1) титульный лист;
 - 2) цель работы;
 - 3) текст индивидуального задания;
 - 4) выполнение индивидуального задания.
4. Контрольные вопросы.

3 Краткие теоретические сведения

Циклы применяются в программах C# для выполнения каких-либо повторяющихся действий.

3.1 Цикл for

Оператор for предназначен для повторного выполнения оператора или группы операторов заданное количество раз. Вот как выглядит этот оператор в общем виде:

```
for ( [Инициализация] ; [Условие] ; [Приращение] ) <Оператор>
```

Оператор [Инициализация] выполняется один раз перед началом цикла.

Перед каждой итерацией (т.е. перед каждым выполнением тела цикла <Оператор> проверяется [Условие]. И наконец, после каждой итерации выполняется оператор [Приращение].

Как правило, в цикле имеется переменная, играющая роль так называемой переменной цикла.

При каждой итерации переменная цикла изменяет свое значение в заданных пределах.

Начальное значение переменной цикла задается в программе до оператора for или в операторе [Инициализация]. Предельное значение переменной цикла определяется оператором приращения, а проверка ее текущего значения – в блоке [Условие].

Пример 1:

```
int i;
for (i = 0; i < 10; i++)
{
    System.Console.Write("{0}", i);
}
```

Здесь переменная i используется в качестве переменной цикла. Перед началом цикла ей присваивается нулевое значение. Перед каждой итерацией содержимое переменной i сравнивается с числом 10. Если i меньше 10, тело цикла выполняется один раз. В тело цикла помещен вызов метода Write, отображающий текущее значение переменной цикла на консоли.

После выполнения тела цикла значение i увеличивается на единицу в блоке приращения. Далее переменная цикла вновь сравнивается с числом 10. Когда значение i превысит 10, цикл завершится.

Таким образом, параметр цикла анализируется перед выполнением тела цикла, а модифицируется после его выполнения.

Вот что выведет на консоль приведенный выше фрагмент программы: 0123456789

3.1.1. Прерывание цикла

С помощью оператора break можно в любой момент прервать выполнение цикла. Например, в следующем фрагменте программы прерывается работа цикла, когда значение переменной i становится больше пяти:

```
for (i = 0; i < 10; i++)
{
```

```

        if (i > 5)
        {
            break;
        }
        System.Console.Write(" {0} ", i);
    }

```

В результате на консоль будут выведены цифры от 0 до 5: 0 1 2 3 4 5

Допускается опускать реализацию любого блока оператора for:

```

int i = 0;
for ( ; ; )
{
    if (i > 10)
    {
        break;
    }
    System.Console.Write(" {0} ", i);
    i++;
}

```

Это позволяет реализовывать любую необходимую логику циклической обработки.

При создании цикла обязательно нужно предусмотреть условие его завершения. Если же этого не сделать, цикл будет выполняться бесконечно. Программа при этом будет работать вхолостую на одном месте, или, как еще говорят, «зациклится».

Вот пример цикла, из которого нет выхода:

```

for (i = 0; ; )    //   Зацикливание!
{
    i++;
    System.Console.Write("{0} ", i);
}

```

Здесь не предусмотрели проверку значения переменной цикла, поэтому программа будет постоянно выводить на консоль возрастающие значения, пока вы не прервете ее работу. Кстати, это можно сделать, нажав комбинацию клавиш Control-C или закрыв консольное окно.

3.1.2. Возобновление цикла

В отличие от оператора **break**, прерывающего цикл, оператор **continue** позволяет возобновить выполнение цикла с самого начала.

Вот как он используется:

```

for (i = 0; ; i++)
{
    System.Console.Write("{0} ", i);
    if (i < 9)
    {
        continue;
    }
}

```

```

    }
    else
    {
        break;
    }

```

Если в ходе выполнения цикла значение переменной *i* не достигло девяти, цикл возобновляет свою работу с самого, начала (т. е. с вывода значения переменной цикла на консоль). Когда указанное значение будет достигнуто, выполнение цикла прервется оператором `break`.

3.2 Цикл `while`

Оператор `while` проверяет условие завершения цикла перед выполнением тела цикла:

```

i = 0;
while (i < 10)
{
    System.Console.Write("{0} ", i); i++;
}

```

В отличие от оператора `for` оператор `while` никак не изменяет значения переменной цикла, поэтому мы должны позаботиться об этом сами.

Перед тем как приступить к выполнению цикла, устанавливается начальное значение параметра цикла *i*, равное нулю. После выполнения тела цикла необходимо самостоятельно изменять значение параметра цикла, увеличивая его на единицу.

Цикл будет прерван, как только значение переменной *i* превысит 10.

В цикле `while` можно использовать описанные ранее операторы прерывания цикла `break` и возобновления цикла `continue`.

Следующий цикл будет выполняться бесконечно:

```

while (true)
{
    System.Console.Write("{0}", i);
    i++;
}

```

3.3 Цикл `do while`

Оператор `do` используется вместе с ключевым словом `while`. При этом условие завершения цикла проверяется после выполнения его тела:

```

i = 0;
do
{
    System.Console.Write("{0}", i);
    i++;
} while (i < 10);

```

Как только это значение достигнет 10, цикл будет прерван.

Аналогично циклу `while` цикл `do` допускает прерывание оператором `break` и возобновление оператором `continue`.

3.4 Цикл `foreach`

Для обработки таких типов данных, как массивы и контейнеры, язык `C#` предлагает очень удобный оператор `foreach`, для которого нет аналога в языках программирования `C` и `C++`.

Так как изучение этих типов данных еще впереди, то вернемся к изучению этого цикла позже.

Примеры использования циклов.

Листинг 1

```
using System;
namespace Iteration
{
    class IterationApp
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Цикл for\n");
            int i;
            for (i = 0; i < 10; i++)
            {
                Console.WriteLine(" {0} ", i);
            }
            Console.WriteLine("\n\nЦикл for (вариант2)");
            for (i = 0; i < 10; i++)
            {
                if (i > 5)
                {
                    break;
                }
                Console.WriteLine("{0} ", i);
            }
            Console.WriteLine("\n\nЦикл for (вариант3)");
            for (i = 0; ; i++)
            {
                Console.WriteLine("{0} ", i);
                if (i < 9)
                {
                    continue;
                }
                else
                {
                    break;
                }
            }
            Console.WriteLine("\n\nЦикл while");
            i = 0;
            while (i < 10)
```

```

        {
            Console.Write("{0} ", i);
            i++;
        }
        Console.WriteLine("\n\nЦикл do");
        i = 0;
        do
        {
            Console.Write("{0} ", i);
            i++;
        } while (i < 10);
        Console.ReadKey();
    }
}

```

4 Задания

1. Выполнить задания по вариантам. Дано натуральное n . Вычислить значение суммы.(с помощью циклов FOR, WHILE)

1. $\frac{1}{1} + \frac{1}{2} + \dots;$

2. $\left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots;$

3. $\frac{1}{1^5} + \frac{1}{2^5} + \dots +;$

4. $-\frac{1}{3} + \frac{1}{5} + \dots +;$

5. $1 + \frac{1}{2} + \frac{1}{4} + \dots;$

6. $\frac{1}{1*2} - \frac{1}{2*3} + \dots;$

7. $\frac{1}{3^2} + \frac{1}{5} + \dots;$

8. $\frac{1!}{1} + \frac{2!}{1+\frac{1}{2}} + \dots;$

9. $n \left\{ \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}; \right.$

10. $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots;$

11. $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots;$

12. $\sin 1 + \sin 2 + \dots;$

13. $\frac{1}{3} + \frac{1}{8} + \dots;$

14. $1 + 3 + 6 + \dots;$

15. $(-1)^n.$

2. Составьте программу табулирования функции $y(x)$, выведите на экран значения x и $y(x)$. Нужный вариант задания выберите из нижеприведенного списка по указанию преподавателя. Откорректируйте элементы управления в форме в соответствии со своим вариантом задания.

1) $y = 10^{-2}bc / x + \cos \sqrt{a^3x},$
 $x_0 = -1.5; x_k = 3.5; dx = 0.5;$
 $a = -1.25; b = -1.5; c = 0.75;$

2) $y = 1.2(a-b)^3 e^{x^2} + x,$
 $x_0 = -0.75; x_k = -1.5; dx = -0.05;$
 $a = 1.5; b = 1.2;$

3) $y = 10^{-1}ax^3 \operatorname{tg}(a - bx),$
 $x_0 = -0.5; x_k = 2.5; dx = 0.05;$
 $a = 10.2; b = 1.25;$

4) $y = ax^3 + \cos^2(x^3 - b),$
 $x_0 = 5.3; x_k = 10.3; dx = 0.25;$
 $a = 1.35; b = -6.25;$

- 5) $y = x^4 + \cos(2 + x^3 - d)$,
 $x_0 = 4.6; x_k = 5.8; dx = 0.2$;
 $d = 1.3$;
- 6) $y = x^2 + \operatorname{tg}(5x + b/x)$,
 $x_0 = -1.5; x_k = -2.5; dx = -0.5$;
 $b = -0.8$;
- 7) $y = 9(x + 15\sqrt{x^3 + b^3})$,
 $x_0 = -2.4; x_k = 1; dx = 0.2$;
 $b = 2.5$;
- 8) $y = 9x^4 + \sin(57.2 + x)$,
 $x_0 = -0.75; x_k = -2.05; dx = -0.2$;
- 9) $y = 0.0025bx^3 + \sqrt{x + e^{0.82}}$,
 $x_0 = -1; x_k = 4; dx = 0.5$;
 $b = 2.3$;
- 10) $y = x \cdot \sin(\sqrt{x + b - 0.0084})$,
 $x_0 = -2.05; x_k = -3.05; dx = -0.2$;
 $b = 3.4$;
- 11) $y = x + \sqrt{x^3 + a - be^x}$,
 $x_0 = -4; x_k = -6.2; dx = -0.2$;
 $a = 0.1$;
- 12) $y = 9(x^3 + b^3)\operatorname{tg}x$,
 $x_0 = 1; x_k = 2.2; dx = 0.2$;
 $b = 3.2$;
- 13) $y = |x - b|^{1/2} / |b^3 - x^3|^{3/2} + \ln|x - b|$,
 $x_0 = -0.73; x_k = -1.73; dx = -0.1$;
 $b = -2$;
- 14) $y = (x^{5/2} - b)\ln(x^2 + 12.7)$,
 $x_0 = 0.25; x_k = 5.2; dx = 0.3$;
 $b = 0.8$;
- 15) $y = 10^{-3}|x|^{5/2} + \ln|x + b|$,
 $x_0 = 1.75; x_k = -2.5; dx = -0.25$;
 $b = 35.4$;
- 16) $y = 15.28|x|^{-3/2} + \cos(\ln|x| + b)$,
 $x_0 = 1.23; x_k = -2.4; dx = -0.3$;
 $b = 12.6$;
- 17) $y = 0.00084(\ln|x|^{5/4} + b)/(x^2 + 3.82)$,
 $x_0 = -2.35; x_k = -2; dx = 0.05$;
 $b = 74.2$;
- 18) $y = 0.8 \cdot 10^{-5}(x^3 + b^3)^{7/6}$,
 $x_0 = -0.05; x_k = 0.15; dx = 0.01$;
 $b = 6.74$;
- 19) $y = (\ln(\sin(x^3 + 0.0025)))^{3/2} + 0.8 \cdot 10^{-3}$,
 $x_0 = 0.12; x_k = 0.64; dx = 0.2$;
- 20) $y = a + x^{2/3} \cos(x + e^x)$,
 $x_0 = 5.62; x_k = 15.62; dx = 0.5$;
 $a = 0.41$

5 Контрольные вопросы

1. Для чего применяются циклы?
2. Как записывается и как работает оператор FOR?
3. Для организации каких циклов применим оператор FOR?
4. Как записывается и как работает оператор WHILE? DO ... WHILE?
5. В чем отличие оператора WHILE от оператора DO ... WHILE?