

Операторы цикла

Цель работы: усвоение принципов работы операторов цикла; построение программ, содержащих операторы цикла.

Краткие теоретические сведения

Цикл *for*

Синтаксис:

```
for ( Инициализация; Условие_Выполнения; Изменение )  
{  
    //Здесь инструкции цикла (тело цикла)  
}
```

Инициализация — инструкция инициализации счетчика циклов.

Условие_Выполнения — выражение, значение которого определяет условие выполнения инструкций цикла. Инструкции цикла выполняются до тех пор, пока *Условие_Выполнения* истинно, т. е. не равно нулю.

Изменение — инструкция изменения параметра цикла. Как правило, эта инструкция изменяет значение переменной, которая входит в *Условие_Выполнения*.

- инструкция **for** используется для организации циклов с фиксированным, известным во время разработки программы, числом повторений;
- количество повторений цикла определяется начальным значением переменной-счетчика и условием завершения цикла;
- переменная-счетчик должна быть целого (*int*) типа и может быть объявлена непосредственно в инструкции цикла

Пример 1: Программа выводит на экран числа от 1 до 10.

```
#include <stdio.h>  
void main()  
{  
    int counter;  
    for (counter=1; counter<=10; counter++)  
        printf("%d\n", counter);  
}
```

Пример 2: Напишите программу, которая суммирует последовательность целых чисел. Предположите, что первое целое число, считываемое с помощью *scanf*, определяет количество значений, которое осталось ввести. Ваша программа должна считывать только одно значение при каждом выполнении *scanf*. Типичной входной последовательностью могло бы быть

5 100 200 300 400 500

где 5 указывает, что должны суммироваться последующие 5 значений.

```
#include<stdio.h>
void main()
{
    int n,i,s=0,ch;
    //с помощью цикла for
    printf("Введите число ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        scanf("%d",&ch);
        s+=ch;
    }
    printf("Сумма равна %d\n", s);
}
```

Цикл while

Синтаксис:

```
while ( Условие_Выполнения ) {  
    // Инструкции цикла (тело цикла)  
}
```

Сначала проверяется значение выражения *Условие_Выполнения*. Если оно не равно нулю, т.е. условие истинно, то выполняйте инструкции цикла (тело цикла). Затем снова проверяется значение выражения *Условие_Выполнения*, и если оно не равно нулю, инструкции цикла выполняются еще раз. И так до тех пор, пока значение выражения *Условие_Выполнения* не станет равным нулю.

- число повторений инструкций цикла **while** определяется ходом выполнения программы;
- инструкции цикла **while** выполняются до тех пор, пока значение выражения, записанного после слова **while**, не станет ложным (равным нулю);
- после слова **while** надо записывать условие выполнения инструкций цикла;
- для завершения цикла **while** в теле цикла обязательно должны быть инструкции, выполнение которых влияет на условие завершения цикла;
- цикл **while** — это цикл с предусловием, т. е. возможна ситуация, при которой инструкции тела цикла ни разу не будут выполнены;
- цикл **while**, как правило, используется для организации приближенных вычислений, в задачах поиска и обработки данных, вводимых с клавиатуры или из файла.

Пример 3: Программа выводит на экран числа от 1 до 10.

```
#include <stdio.h>  
void main()  
{  
    int counter =1; /* инициализация */  
  
    while (counter <= 10) { /*условие повторения*/  
        printf("%d\n", counter);  
        ++counter;      /* приращение */  
    }  
}
```

Пример 4: Напишите программу, которая суммирует последовательность целых чисел. Предположите, что первое целое число, считываемое с помощью *scanf*, определяет количество значений, которое осталось ввести. Ваша программа должна считывать только одно значение при каждом выполнении *scanf*. Типичной входной последовательностью могло бы быть

5 100 200 300 400 500

где 5 указывает, что должны суммироваться последующие 5 значений.

```
#include<stdio.h>
void main()
{   int n,i,s=0,ch;
    //с помощью цикла for
    printf("Введите число ");
    scanf("%d",&n);
    i=1;
    s=0;
    while(i<=n)
    {   scanf("%d",&ch);
        s+=ch;
        i++;
    }
    printf("Сумма равна %d\n", s);
}
```

Пример 5. Написать программу, которая выводит таблицу квадратов первых *n* целых положительных четных чисел. Количество чисел должно вводиться во время работы программы. После вывода результата на экран снова должен появляться запрос количества чисел *n*, до тех пор, пока пользователь не введет 0 (когда введен 0, осуществляется выход из программы).

```
#include<stdio.h>
void main()
{
    int n,i;
    printf("Введите n ");
    scanf("%d",&n);
    while(n!=0) {
        for(i=2;i<=n;i+=2)
            printf(" %d  %d\n",i,i*i);
        printf("Введите n");
        scanf("%d",&n);
    }
}
```

Пример 6. Программа выводит на экран меню, в котором пользователь может выбрать одно из следующих действий введя соответствующее число: подсчитать сумму двух чисел, подсчитать разность двух чисел, выйти из программы. Отличие данного примера от примера 5 заключается в том, что программа будет выводить запрос выбора не один раз (как в примере 5), а до тех пор, пока пользователь не выберет выход из программы.

```
#include<stdio.h>
void main()
{
    int n, a,b;
    while(1) { //бесконечный цикл, условие цикла всегда верно
        printf("Меню:\n");
        printf("1 – подсчет суммы двух чисел\n ");
        printf("2 – подсчет разности двух чисел\n ");
        printf("3 – выход из программы\n ");
        printf("Ваш выбор? \n ");
        scanf("%d", &n);
        switch (n)
        {
            case 1:
                printf("Введите два числа\n ");
                scanf("%d%d", &a, &b);
                printf("Сумма равна %d\n ", a+b);
                break;
            case 2: printf("Введите два числа\n ");
                scanf("%d%d", &a, &b);
                printf("Разность равна %d\n ", a-b);
                break;
            case 3: return; break;
            default: printf("Неверное значение.");
        }
    }
}
```

Цикл *do...while*

Синтаксис:

```
do
{
    // Инструкции цикла (тело цикла)
}
while ( Условие_Повторения );
```

Сначала выполняются инструкции цикла (тело цикла), затем проверяется значение выражения *Условие_Повторения*, и если условие истинно, не равно нулю, то инструкции цикла выполняются еще раз. И так до тех пор, пока *Условие_Повторения* не станет ложным, т. е. равным нулю.

- число повторений инструкций цикла **do...while** определяется ходом выполнения программы;
- инструкции цикла **do...while** выполняются до тех пор, пока значение выражения, записанного после слова **while**, не станет ложным (равным нулю);
- после слова **while** надо записывать условие выполнения инструкций цикла;
- для завершения цикла **do...while** в теле цикла обязательно должны быть инструкции, выполнение которых влияет на условие завершения цикла;
- цикл **do...while** — это цикл с постусловием, т. е. инструкции тела цикла будут выполнены хотя бы один раз;
- цикл **do...while**, как правило, используется для организации приближенных вычислений, в задачах поиска и обработки данных, вводимых с клавиатуры или из файла.

Пример 7. В программе структура *do...while* используется для вывода чисел от 1 до 10. Обратите внимание, что к управляющей переменной *counter* при проверке условия продолжения цикла применяется операция преинкремента. Обратите также внимание на фигурные скобки, заключающие тело структуры *do/while* (состоящее из одного оператора) подсчитать разность.

```
#include <stdio.h>
void main ()
{
    int counter = 1;
    do {
        printf("%d ", counter);
    } while (++counter <= 10);
}
```

Практическая часть

Упражнение 1

Создать файл проекта и разработать Си-программу в соответствии с вариантом, составить графическую диаграмму алгоритма программы.

1. Сколько слагаемых должно быть в сумме $1+1/2+1/3+1/4+\dots+1/n$, чтобы эта сумма оказалась больше 5 ?
2. Сумма 10000 руб. положена в сберегательный банк под 3% годовых (процент капитализированный). Составить алгоритм, определяющий через какой промежуток времени первоначальная сумма увеличится в 2 раза.
3. В 1626г. индейцы продали остров за 20 долларов. Если бы эти деньги были помещены в банк под 4% годовых (процент капитализированный), то какова была бы стоимость капитала сегодня?
4. Сумма R руб. положена в банк под 4% годовых (процент капитализированный). Составить алгоритм, определяющий через какой промежуток времени сумма достигнет M руб. ($M>R$).
5. Население города ежегодно увеличивается на $1/n$ наличного состава жителей, где n-натуральное число. Через сколько лет население города утроится.
6. Можно ли разменять m руб. на рублёвые, трёхрублёвые, пятирублёвые купюры так, чтобы получить всего 10 купюр. ($10<m<50$)
7. Составить алгоритм поиска четырёхзначного числа, начинающегося с единицы и такого, что если переставить эту цифру в конец записи числа, то получится число, в три раза большее искомого.
8. Искомое число больше 400 и меньше 500. Составить алгоритм поиска этого числа, если сумма его цифр равна 9 и оно равняется $47/36$ числа, изображённого теми же цифрами, но в обратном порядке.
9. Имеются контейнеры двух видов: по 130кг и 160кг. Можно ли полностью загрузить ими грузовик грузоподъёмностью 3т.
10. Сумма цифр двузначного числа равна 11. Если к этому числу прибавить 27, то получится число, записанное теми же цифрами, но в обратном порядке. Составить алгоритм поиска этого числа, если оно существует.

Упражнение 2

Создать файл проекта и разработать Си-программу в соответствии с вариантом, составить графическую диаграмму алгоритма программы.

1. Составить алгоритм, определяющий количество способов, какими задуманное число $n > 1$ можно представить в виде суммы $n = i^3 + j^3$, считая, что перестановка слагаемых нового способа не даёт.
2. Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него есть простое число (число 121 – полный квадрат, т.к. $121 = 11 * 11$, а 11 – простое число)
3. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50коп., 20коп., 5коп. и 2коп.
4. Имеются два сосуда. В первом сосуде находится С1 литров воды, во втором - С2 литров воды. Из первого сосуда переливают половину воды во второй сосуд, затем из второго переливают половину в первый сосуд, и так далее. Сколько воды окажется в обоих сосудах после 12 переливаний.
5. Составить алгоритм вычисления числа Пи по формуле Грегори, взяв 500 членов ряда:
$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots$$
6. N человек играют в следующую игру: стоя в кругу они начинают считалку. Счёт идёт до числа М. Игрок, на которого падает счёт М, выбывает, а считалка начинается сначала со следующего по кругу игрока. Выигрывает тот, кто остался последним в кругу. Считалка начинается с игрока Т, составить алгоритм для определения выигравшего игрока и первой пятёрки выбывших игроков.
7. Три приятеля были свидетелями нарушения правил дорожного движения. Номер автомобиля - четырехзначное число - никто не запомнил. Из их показаний следует, что номер делится на 2, на 7 и на 11, в записи номера участвуют только две цифры, сумма цифр номера равна 30. Составьте алгоритм и программу для определения номера автомашины.
8. Число а возводят в квадрат и результат увеличивают на 1. Полученное число снова возводят в квадрат и увеличивают на 1. Этот процесс продолжается до тех пор, пока не будет получено число Х, большее миллиона. Найти число Х.
9. Найти сумму S и произведение P:
 - а) четных чисел от 1 до n
 - б) нечетных чисел от 1 до n
 - в) чисел, кратных 3, от 1 до n.
10. Найти сумму:
 - а) квадратов первых n натуральных чисел
 - б) кубов первых n натуральных чисел
 - в) квадратов четных чисел из первых n натуральных чисел
 - г) кубов четных чисел из первых n натуральных чисел.