

Лабораторная работа №5. Операторы цикла и передачи управления

1 Цель и порядок работы

Цель работы – изучить операторы, используемые при организации программ циклических вычислительных процессов, получить практические навыки в составлении программ.

Порядок выполнения работы:

- ознакомиться с описанием лабораторной работы;
- получить задание у преподавателя, согласно своему варианту;
- написать программу и отладить ее на ЭВМ;

2 Краткая теория

2.1 Операторы цикла

Многokrратно повторяемые действия могут быть заданы с помощью операторов цикла. Язык C++ поддерживает три вида циклов:

- цикл с предусловием;
- цикл с постусловием;
- цикл с параметром (итерационный цикл).

2.1.1 Цикл с предусловием (while)

Цикла с предусловием задается при помощи оператора **while**. Он имеет следующую форму записи:

```
while (условие) тело_цикла ;
```

Условие – это скалярное выражение, определяющее условия продолжения выполнения операций. Оно принимает логическое значение "истина" (true или любое не нулевое или не пустое значение) или "ложь" (false, ноль или пустое значение (null)). Выполнение оператора повторяется до тех пор, пока значением условия является "истина".

Тело цикла не может быть описанием или определением. Это либо отдельный (в том числе пустой) оператор, либо блок (один или несколько операторов помещенных в фигурные скобки). Условие вычисляется заново перед каждой итерацией.

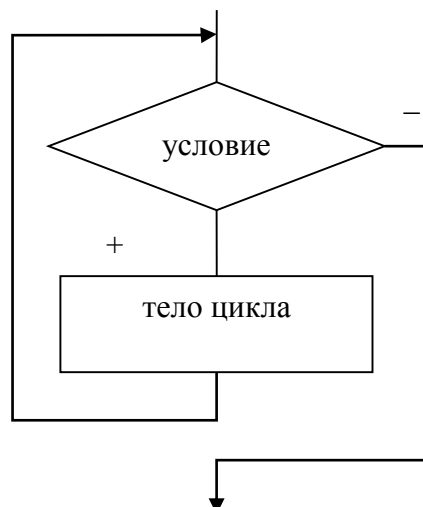


Рисунок 5.1 – Блок-схема оператора цикла с предусловием

Пример: подсчитать количество цифр входящих в запись натурального положительного числа.

```

int digits = 0;
while (N > 0)
{
    digits = digits + 1;
    N = N / 10;
}

```

2.1.2 Цикл с постусловием (do while)

Оператор цикла с постусловием начинается со служебного слова **do** и заканчивается служебным словом **while**, между которыми располагается тело цикла.

Синтаксис записи оператора:

```

do
    тело_цикла;
while (условие);

```

Отличие от предыдущей формы цикла **while** заключается в том, что условие проверяется после выполнения тела цикла.

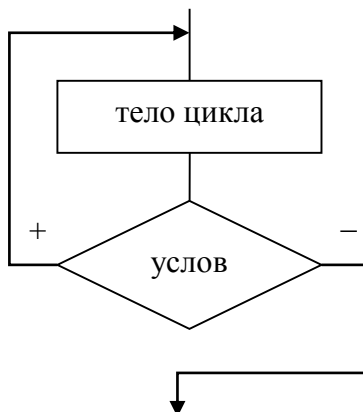


Рисунок 5.2 – Блок-схема оператора цикла с постусловием

Пример: вычислить произведение нечетных чисел от 1 до 10 включительно.

```

int i = 1, p = 1;
do
{
    p *= i; // умножим произведение на i (p = p * i)
    i += 2; // перейдем к следующему нечетному числу (i = i + 2)
} while (i < 10);

```

В операторах циклов **while** и **do while** можно использовать операторы **break** и **continue**, предназначенные для прекращения работы цикла и пропуска остатка цикла с переходом к следующей итерации соответственно.

2.1.3 Цикл с параметром (for)

Цикл с параметром или *итерационный цикл* реализует фундаментальный принцип вычислений в программировании – итерацию. Во время работы данного оператора, тело цикла будет повторяться, а повторение называют итерацией. Т.е. при этом выполняется перебор значений переменной, которая используется для работы операторов, входящих в тело цикла.

Оператор цикла состоит из заголовка цикла и тела цикла. Тело цикла – это оператор, который будет повторно выполняться. Заголовок – это ключевое слово `for`, после которого в круглых скобках записаны три выражения, разделенные точкой с запятой. Первое выражение (инициализация цикла) вычисляется один раз до начала выполнения цикла. Второе – это условие цикла. Тело цикла будет повторяться до тех пор, пока условие цикла истинно. Третье выражение, последовательность скалярных выражений разделенных запятыми ";", вычисляется после каждого повторения тела цикла.

```
for (инициализация; условие; модификация)
    тело_цикла;
```

Первое выражение обычно используется для установления начального значения переменных, управляющих циклом. Второе выражение – это выражение, определяющее условие, при котором тело цикла будет выполняться. Третье выражение определяет изменение переменных, управляющих циклом после каждого выполнения тела цикла.

Схема выполнения оператора **for**:

1. Вычисляется первое выражение.

2. Вычисляется второе выражение.

3.1. Если значения второго выражения отлично от нуля (истина), выполняется тело цикла, вычисляется третье выражение и осуществляется переход к пункту 2.

3.2. Если второе выражение равно нулю (ложь), то управление передается на оператор, следующий за оператором **for**.

Существенно то, что проверка условия всегда выполняется в начале цикла. Это значит, что тело цикла может ни разу не выполниться, если условие выполнения сразу будет ложным.

Предположим, нам нужно вычислить сумму всех целых чисел от 0 до 100. Для этого воспользуемся оператором цикла **for**:

```
int sum = 0;
int i;
for (i = 1; i <= 100; i = i + 1)    // заголовок цикла
    sum = sum + i;                 // тело цикла
```

Любое из трех выражений в заголовке цикла может быть опущено (в том числе и все три). То же самое можно записать следующим образом:

```
int sum = 0;
int i = 1;
for ( ; i <= 100; )
{
    sum = sum + i;
    i = i + 1;
}
```

Заметим, что вместо одного оператора в теле цикла записано несколько операторов, заключенных в фигурные скобки – блок.

Еще один вариант:

```
int sum = 0;
int i = 1;
for ( ; ; )
{
    if (i > 100)
        break;
    sum = sum + i;
}
```

```
        i = i + 1;
    }
```

В последнем примере был использован оператор **break**, который завершает выполнение цикла.

Еще одно полезное свойство цикла **for**: в первом выражении заголовка цикла можно объявить переменную. Эта переменная будет действительна только в пределах цикла.

2.2 Операторы передачи управления

К операторам передачи управления относят оператор безусловного перехода **goto**, оператор возврата из функции **return**, оператор выхода из цикла или переключателя **break** и оператор перехода к следующей итерации цикла **continue**.

2.2.1 Оператор безусловного перехода (goto)

Оператор безусловного перехода имеет вид:

goto идентификатор;

Идентификатор - имя метки расположенной в той же функции, где используется **goto**. Передача управления разрешается на любой помеченный оператор в теле функции. Запрещено перескакивать через описания, содержащие инициализацию объектов. Это ограничение не распространяется на вложенные блоки, которые можно обойти циклом. **Применение оператора goto необходимо свести к минимуму.** Если же применяете его, то придерживайтесь следующих правил:

- не входить внутрь блока извне;
- не входить внутрь условного оператора;
- не входить внутрь переключателя;
- не передавать управление внутрь цикла.

2.2.2 Оператор возврата из функции (return)

Оператор возврата из функции имеет вид:

return выражение ;

или

return ;

Выражение может быть только скалярным.

Например, функция вычисляет и возвращает куб значения своего аргумента:

```
float cube(float z) { return i*i*i; }
```

Выражение в операторе **return** не может присутствовать, если возвращенное функцией значение имеет тип **void**.

2.2.3 Оператор выхода из цикла (break)

Оператор **break** служит для принудительного выхода из цикла или переключателя. Он осуществляет передачу управления к следующему за циклом или переключателем оператору.

Например, если необходимо применить дополнительное условие выхода из цикла:

```
while(i < j)
{
    i++;
    if (i == j)
        break;
    j--;
}
```

Циклы и переключатели могут быть вложенными, **break** позволяет выйти только из самого внутреннего цикла или переключателя.

2.2.4 Оператор перехода к следующей итерации цикла (continue)

Еще одним оператором передачи управления является оператор перехода к к следующей итерации (оператор продолжения) **continue**. Он употребляется только в операторах цикла. С его помощью завершается текущая итерация (при этом пропускается остаток тела цикла) и начинается проверка условия дальнейшего продолжения цикла, т.е. условий начала следующей итерации.

Например, если необходимо найти сумму всех целых чисел от 0 до 100, которые не делятся на 7.

```
int sum = 0;
for (int i = 1; i <= 100; i = i+1)
{
    if ( i % 7 == 0)
        continue;
    sum = sum + i;
}
```

3 Контрольные вопросы

1. Какие виды операторов цикла существуют.
2. Опишите оператор цикла с предусловием.
3. Опишите оператор цикла с постусловием.
4. Опишите оператор цикла с параметром.
5. Перечислите операторы передачи управления.
6. Какое назначение оператора break.
7. Какое назначение оператора continue.
8. Какое назначение оператора return.
9. Какое назначение оператора goto.

4 Задание

1. Написать программу в соответствии с вариантом задания из пункта 5. Вариант определяется по последней цифре в номере студента в общем списке группы. И одному заданию на выбор задания из пунктов 6.1 и 6.2.
 - 1 вариант – 5.1.1, 5.1.11, 5.1.21, 5.2.1
 - 2 вариант – 5.1.2, 5.1.12, 5.1.22, 5.2.2
 - 3 вариант – 5.1.3, 5.1.13, 5.1.23, 5.2.3
 - 4 вариант – 5.1.4, 5.1.14, 5.1.24, 5.2.4
 - 5 вариант – 5.1.5, 5.1.15, 5.1.25, 5.2.5
 - 6 вариант – 5.1.6, 5.1.16, 5.1.26, 5.2.6
 - 7 вариант – 5.1.7, 5.1.17, 5.1.27, 5.2.7

8 вариант – 5.1.8, 5.1.18, 5.1.28, 5.2.8
 9 вариант – 5.1.9, 5.1.19, 5.1.29, 5.2.9
 0 вариант – 5.1.10, 5.1.20, 5.1.30, 5.2.10

2. Проверить работоспособность программ.
3. Отладить и протестировать программы.

5 Варианты заданий

5.1 Оператор цикла for

1. Составить программу возведения натурального числа в квадрат, используя следующую закономерность:

$$1^2 = 1$$

$$2^2 = 1 + 3$$

...

$$n^2 = 1 + 3 + 5 + 7 + \dots + (2n - 1).$$

2. Определить количество трехзначных чисел сумма цифр которых равна заданному числу N.
3. Найти количество делителей натурального числа. Сколько из них четные?
4. Среди двухзначных чисел найдите те, сумма квадратов цифр которых делится на 13.
5. Напишите программу поиска двухзначных чисел, таких, что если к сумме цифр этого числа прибавить квадрат этой суммы, получится это число.
6. Напишите программу поиска трехзначных чисел, таких, что сумма трех цифр на которые оканчивается квадрат этого числа, равняется этому числу.
7. Найдите сумму положительных четных чисел меньших 100.
8. Найдите сумму целых чисел из промежутка от A до B, кратных 4 (значения A и B вводятся с клавиатуры).
9. С клавиатуры последовательно вводятся числа n, a_1, a_2, \dots, a_n . Определить число членов a_k последовательности a_1, a_2, \dots, a_n , имеющих четные порядковые номера и являющихся нечетными числами.
10. Среди четырехзначных чисел выбрать те, у которых цифры различны.
11. Дано натуральное n. Написать программу вычисления a^n .
12. Дана функция $y = \text{tg}(x) + 1$. Написать программу для поиска максимального значения y в промежутке $x \in [2; 2.8]$ с шагом 0,1.
13. Составить алгоритм для расчета функции y при $x = 0..15$:

$$y = \begin{cases} x^3 - 3x, & \text{если } x \leq 1, \Delta x = 0,2; \\ \frac{(x-3)^3}{x}, & \text{если } x > 1, \Delta x = 0,5. \end{cases}$$
14. Дано целое число K. Найти сумму цифр числа K.
15. Программа ждет ввода числа и в зависимости от количества цифр в числе выдает сообщение об их разрядности: трехзначное, пятизначное и т.д.
16. Написать программу нахождения суммы чисел по правилу (n-нечетно): $a^1 + a^3 + a^5 + \dots + a^n$.
17. Дано целое число K. Найти произведение цифр числа K.
18. Дано вещественное число a, целое n. Вычислить: $a(a-1)(a-2)\dots(a-n)$.

19. Составить алгоритм для расчета функции $y = \ln(\sin(x)+1)*0.15$ при изменении x от 0 до 12 с шагом $\Delta x=0.2$.
20. Написать программу нахождения суммы чисел по правилу (n-четно): $a^2+a^4+a^6+\dots+a^n$.
21. Вычислить $(1-\sin(0.1))+(1+\sin(0.2))+(1-\sin(0.3))+\dots+(1+\sin(1.0))$.
22. Дана функция $y=0.5+\sin(5x)$, причем x изменяется от 0 до 2π с шагом $\Delta x=\pi/6$.
23. Найти все натуральные числа от 1 до 1000, которые совпадают с последними разрядами своих квадратов, например: $25^2 = 625$, $76^2 = 5676$.
24. Дана функция $y=2x^3+1$ Написать программу для поиска максимального значения y в промежутке $x \in [-1;1]$ с шагом 0,2.
25. Составить программу вычисления суммы кубов чисел от 25 до 125.
26. Найти сумму целых положительных чисел из промежутка от A до B, кратных 3 (A и B вводятся с клавиатуры).
27. Найти сумму десяти чисел кратных 3.
28. Используя алгоритм Евклида, найти наименьшее общее кратное (НОК) m и n.
29. Даны два натуральных числа. Получить их наибольший общий делитель (НОД), используя алгоритм Евклида.
30. Найти сумму четных делителей натурального числа.

5.2 Оператор цикла с предусловием (while) и с постусловием (do while)

Задание необходимо выполнить в двух вариантах (используя два вида циклов)

1. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{n!}{n^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
2. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{1}{2^n} + \frac{1}{3^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
3. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{(-1)^{n-1}}{n^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
4. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{1}{(3n-2)(3n+1)}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
5. Найти сумму ряда с точностью ε , общий член которого равен $a_n = n^2 e^{-\sqrt{n}}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
6. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{n!}{2n!}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
7. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{3 \cdot n!}{(3n)!}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .

8. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{\ln(n!)}{n^2}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
9. Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{(2n-1)}{2^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .
10. 6.14 Найти сумму ряда с точностью ε , общий член которого равен $a_n = \frac{10^n}{n!}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε .

6 Дополнительные задания

6.1 Оператор цикла for

- Напечатайте таблицу умножения на 5. предпочтительно печатать $1 \times 5 = 5$, $2 \times 5 = 10$, а не просто 5, 10 и т.д.
- Напечатайте в столбик нечетные числа от 3 до 25.
- Выведите на экран таблицу значений синуса от 0 до 2π . В каждой строке должны стоять один аргумент и одно значение. Количество значений аргумента пусть задает пользователь.
- Напишите программу, которая вычисляет сумму квадратов чисел от 1 до N. Число N программа должна запрашивать у пользователя.
- Напишите программу, перемножающую целые числа без использования операции «*». Например, при умножении целых чисел $n \cdot m$ число m надо сложить само с собой n раз ($m+m+\dots+m$).
- Найдите сумму нечетных чисел от 1 до N. Число N программа должна запрашивать у пользователя.
- Выведите на экран последовательность сумм чисел от 1 до n. n меняется от 1 до 10. То есть первые члены последовательности: 1, 3 ($1+2$), 6 ($1+2+3$), 10 ($1+2+3+4$) и т.д.
- Найдите сумму 100 синусов от аргументов в диапазоне от 0 до 2π (из задачи 1.4).
- Вычислите сумму ряда: $1 + x + x^2 + x^3 + \dots + x^{10}$.
- Вычислите сумму ряда $s_n = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$

6.2 Оператор цикла с предусловием (while) и с постусловием (do while)

- Найдите сумму цифр произвольного натурального числа.
- Найдите самую большую цифру в записи произвольного целого числа.
- Составить программу, проверяющую, является ли заданное натуральное число совершенным, т.е. равным сумме своих положительных делителей, кроме самого этого числа.
- Задается натуральное число q равное площади некоторого прямоугольника. Напишите программу, определяющую количество прямоугольников имеющих площадь q, стороны которых выражены натуральными числами.
- Создайте программу, играющую с пользователем в орлянку. Программа должна спрашивать у пользователя орел или решка. Если пользователь вводит 0, то выбирает орла, 1 – решку, любое другое число – конец игры. Программа должна вести учет выигрышей и проигрышей и после каждого раунда сообщать пользователю о состоянии его счета. Пусть вначале на счету 1 рубль и ставка в

каждом коне тоже 1 рубль. Если денег у пользователя не осталось игра прекращается.

6. Вычислите золотое сечение по формуле

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}}$$

Сделайте 20 итераций. В каком знаке будут наблюдаться изменения, если сделать 30 итераций?

7. Пользователь вводит 10 чисел. Определить, образуют ли они возрастающую последовательность.