**Лабораторная работа № 6.**

**Реализация циклов.**

*(тема 2.5.)*

Цель работы: Изучить работу операторов циклов с предусловием, постусловием и счетчиком. Научиться решать задачи с использованием этих операторов.

**Общие сведения.**

**Оператор цикла с предусловием *While .. do*.**

Оператор предусловием *While .. do* состоит из заголовка и тела цикла. Ключевые слова *While* и *do* обозначают "до тех пор, пока" и "выполняй" соответственно.

Структура оператора:

*While <Условие выполнения цикла> do*

*Begin*

*{ Инструкции }*

*End;*

Условие выполнения цикла — это выражение логического типа. Тело цик­ла— оператор, чаще всего составной. Если тело цикла представляет собой одиночный оператор, то ключевые слова *Begin* и *End* лучше не писать, хотя их наличие не будет ошибкой.

Условие выполнения цикла производится в самом начале

оператора — если значение условия равно *true* (истина), то выполняются инструкции цикла, находящие­ся между *Begin* и *End* и снова вычисляется выражение <*Условия выполнения цикла>*. Так продолжается до тех пор, пока значение *Условия выполнения цикла* не станет равно *false* (ложь).

Для примера рассмотрим фрагмент программы суммирования чисел от 1 до 10:

*…*

*S := 0; I := 1;*

*While I := 10 do*

*S:= S + I ;*

*I := I + 1;*

*End;*

*…*

**Оператор цикла с постусловием *While .. do*.**

Оператор повтора *rереаt*: состоит из заголовка *(rереаt)*, *тела* и *условия* окончания *(until).* Ключевые слова *rереаt и*  *until* обозначают "повторяй" и "пока" соответственно.

*Repeat*

*{Инструкции}*

*Until < Условие выхода из цикла> ;*

*<Условие выхода из цикла>* — это выражение логического типа: *простое выражение отношения* или *сложное логическое выражение.*

Цикл работает так: *вначале* выполняется тело цикла — инструкции, которые находятся между *rереаt* и *until*, затемпроверяется значение *условия выхода из цикла*. В том случае, если оно равно *false* (ложь), т. е. не выполняется, — инструкции цикла повторяются еще раз. Так продолжается дот пор, пока условие не станет *true* (истина).

Для примера рассмотрим фрагмент программы суммирования чисел от 1 до 10:

*…*

*S := 0; I := 1;*

*Repeat*

*S:= S + I ;*

*I := I + 1;*

*While I := 11 ;*

*…*

**Оператор цикла с параметром For .. to .. do.**

Этот вид оператора цикла также называют циклом с параметром. В нем важ­ную роль играет переменная-параметр, которая на каждом шаге цикла авто­матически изменяет свое значение ровно на единицу — поэтому ее и назы­вают счетчиком.

Инструкцию *For* можно реализовать двумя способами.

**Вариант 1 (с увеличением счетчика):**

*For Счетчик := Начальное\_ Значение For Конечное\_Значение do*

*begin*

*{ Инструкции }*

*end;*

Ключевые слова *For* и  *to* обозначают "для", "выполняй" соответственно. Строка, содержащая *For .. to .. do*, называется заголовком цикла, оператор, стоящий после *do*, образует его тело. Очень часто тело цикла — составной оператор. Если тело цикла представлено одиночным оператором, то *Begin* и *End* не пишутся.

Инструкции между *Begin* и *End* выполняются такое число раз, какое опре­деляет выражение *[ (Конечное\_Значение — Начальное\_Значение) + 1]*.

Это соответствует всем значениям счетчика от начального до конечного включительно.

Если *Начальное\_Значение* больше, чем *Конечное\_Значение*, то инструкции между *Begin* и *End*  не выполняются ни разу. Это позволяет сделать вывод о том, что цикл с параметром является разновидностью цикла с предусловием.

Например, выполнение цикла

*For I := 10 to 14 do write ( I : 3 ) ;*

выведет на экран последовательность чисел в виде:

*10 11 12 13 14*

**Вариант 2 (с уменьшением счетчика):**

*For Счетчик := Начальное\_Значение downto Конечное\_Значение do*

*begin*

*{ Инструкции }*

*end;*

Инструкции между *begin* и *end* выполняются такое число раз, какое определяет выражение *[ (Конечное\_Значение — Начальное\_Значение) + 1 ]* .

Если *Начальное\_Значение*  меньше, чем *Конечное\_Значение*, то инструкции между *begin* и *end* не выполняются ни разу.

Например, выполнение цикла

*For I := 14 downto 10 do write ( I : 3 ) ;*

выведет на экран последовательность чисел в виде:

*14 13 12 11 10*

Если переменная – счетчик имеет символьный тип char, то оператор

*For I := ‘a’ do ‘e’ do write ( ch : 2 ) ;*

выведет на экран последовательность букв в виде:  *a b c d e .*

А оператор

*For I := ‘e’ do ‘a’ do write ( ch : 2 ) ;*

выведет на экран последовательность букв в виде: *e d c b a .*

**Задания для выполнения.**

1. Внимательно прочитайте инструкцию.
2. Загрузите среду *ТР*.
3. Составьте программы для решения задач своего варианта. Каждую задачу решите с использованием всех трех операторов цикла.
4. Запустите программы на выполнение. Если возникнут ошибки, отредактируйте тексты программ.
5. Тексты программ перепишите в отчет вместе с исходными данными и результатами.
6. Результаты сравните и проанализируйте.
7. Письменно ответьте на контрольные вопросы.

**Контрольные вопросы.**

1. Что такое цикл?
2. Обоснуйте необходимость применения цикла.
3. Какие разновидности циклов вам известны?
4. Приведите блок – схему работы оператора цикла с предусловием.
5. Приведите блок – схему работы оператора цикла с постусловием.
6. Приведите блок – схемы работы оператора цикла с параметром.
7. Какой из операторов цикла является более универсальным? Почему?
8. Чем отличаются циклы с предусловием и постусловием от цикла с параметром?
9. Укажите в каких примерах более удобно использовать оператор цикла с параметром, в отличие от цикла с предусловием и постусловием.

**Варианты заданий.**

|  |
| --- |
| Вариант 1.  Решите задачи тремя способами:   1. Напечатайте квадраты всех натуральных чисел от 10 до 30. 2. Вычислить произведение натуральных чисел от 12 до 38. |
| Вариант 2.  Решите задачи тремя способами:   1. Найдите произведение N произвольных чисел. 2. Найдите сумму кубов всех двузначных чисел. |
| Вариант 3.  Решите задачи тремя способами:   1. Напечатайте таблицу квадратов всех натуральных чисел до 50. 2. Напечатайте таблицу значений функции *y = x3 + 3* при изменении *x* от 2 до 12 с шагом 2. |
| Вариант 4.  Решите задачи тремя способами:   1. Составьте таблицу умножения для числа 9. 2. Вычислите значение функции *y = 3x + 9* на отрезке (-3 ; 3) с шагом 0.5. |
| Вариант 5.  Решите задачи тремя способами:   1. Найдите сумму квадратов всех чисел от 1 до 20 с ша­гом 1. 2. Найдите сумму квадратов всех двузначных чисел. |
| Вариант 6.  Решите задачи тремя способами:   1. Найдите сумму произвольных десяти чисел. 2. Вычислить произведение натуральных чисел от 40 до 78. |
| Вариант 7.  Решите задачи тремя способами:   1. Вычислите значение функции V= x + 1 на отрезке (0,10) с шагом 0,3. 2. Напечатайте таблицу значений функции *y = x7 + 10* при изменении *x* от 20 до 40 с шагом 4. |
| Вариант 8.  Решите задачи тремя способами:   1. Вычислите факториал числа *N (N!* =, 1\*2\*3..\*N). 2. Вычислите значение функции *y = 9x + 12* на отрезке (-10 ; 10) с шагом 0.9. |