

Задание для студентов гр. ИС 3 26.10.2020 (4 часа – 2 пары)

Дисциплина «Основы алгоритмизации и программирования»

Преподаватель Токарская М.С.

Почта для обратной связи: maya_tok@mail.ru

Тел. 89147174421 – WhatsApp – если есть вопросы.

Все задания отправлять на почту!!!!

Задание:

Выполнить лабораторную работу в среде программирования Турбо Паскаль. Для отчета выслать архив под именем Фамилия_вариант (напр. Иванов_3) который содержит письменные ответы на контрольные вопросы (номер вопроса и ответ, напр. 2 – а или 3- цикл) и файл с программой (не скриншоты, а программу!!!)

Лабораторная работа «Организация работы программ циклической структуры»

Цель работы: научиться работать в среде программирования Турбо Паскаль, составлять программы циклической структуры, используя цикл с предусловием, постусловием и цикл с параметром.

План работы:

1. Повторить теоретический материал по лекциям.
2. При необходимости ознакомиться с кратким теоретическим материалом и приведенным примерами;
3. Выполнить тест;
4. Выполнить задание ЛР по вариантам;

Краткий материал.

Цикл – это многократно выполняемый участок алгоритма или программы. Использование циклов приводит к уменьшению длины программы, сокращению времени на составление и отладку.

В Паскале имеем три вида оператора цикла (рис. 20):

1. Оператор цикла с предусловием – **while**.
2. Оператор цикла с постусловием – **repeat**.
3. Оператор цикла с параметром (со счетчиком) – **for**.

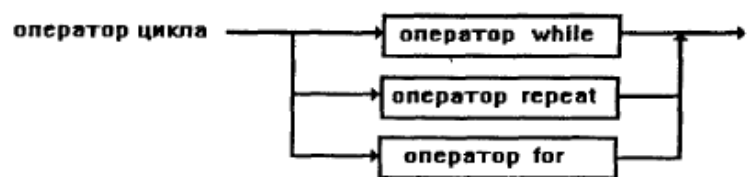


Рис. 20. Три вида оператора цикла

Чаще всего вид оператора определяется не единственным образом. Наиболее универсальным является оператор цикла с предусловием. Простой или составной оператор, содержащийся в цикле, называется *телом цикла*, а переменная, изменяющаяся в нем, – *параметром цикла*. В одном цикле может быть несколько параметров.

Условиями циклов служат логические константы, переменные или выражения с логическим результатом. Чтобы избежать бесконечного повторе-

ния ("защипливания"), необходимо изменять в теле цикла хотя бы один параметр, входящий в логическое выражение.

В циклах часто вычисляются суммы и произведения. До сих пор во всех наших программах переменные получали свое значение в результате вычислений или при вводе с клавиатуры. Установка начальных значений переменных обязательно должна предшествовать их использованию в циклах. В противном случае при первом выполнении цикла в операции может участвовать произвольное значение переменной. Более того, эта операция требует тщательного обдумывания и подбора.

2. Оператор цикла с предусловием WHILE ... DO

Этот оператор организует так называемый цикл «ПОКА», или цикл с предусловием, так как проверка проводится перед началом очередной итерации (неизвестно количество повторений цикла):

WHILE < Условие > **DO**
< тело цикла >;

Оператор **while** позволяет многократно выполнять одни и те же действия в зависимости от некоторого < Условия >, которое записывается между ключевыми словами **while** и **do** и должно быть выражением булевого типа, т. е. принимать только значения **True** или **False**.

Цикл выполняется, пока < Условие > истинно. Как только < Условие > нарушается, выполнение цикла завершается. Блок-схема цикла **while** приведена на рис. 21.

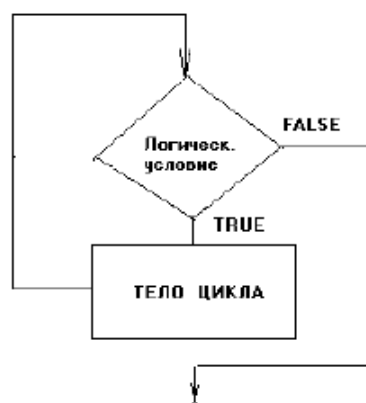


Рис. 21. Цикл «Пока» с предусловием

Общую структуру оператора цикла с предусловием можно представить одним из двух следующих способов (табл. 11):

Случай 1. Если тело цикла состоит из одного оператора, то операторные скобки **Begin** – **End** можно не ставить:

```
while < Условие > do
```

```
Оператор;
```

Случай 2. Если тело цикла состоит из нескольких операторов, то операторные скобки **Begin** – **End** необходимо ставить:

```
while < Условие > do
```

```
begin Оператор;
```

```
Оператор;
```

```
Оператор;
```

```
end;
```

Задача 1

При вводе осуществить проверку длины вводимой последовательности.

Program Lab5_1;

Var

 N: integer;

BEGIN {1}

 ReadLn (N);

 While N <= 0 Do

 Begin {2}

 WriteLn ('Введите длину последовательности N = ');

 ReadLn (N);

 End; {2}

 WriteLn ('Длина последовательности N = ', N);

END. {1}

Пояснения к задаче 1

До первого входа в цикл переменной N уже должно быть присвоено начальное значение соответственно условию решаемой задачи.

N – длина последовательности, что влечет за собой необходимость выполнения логического условия $N > 0$ (длина не может быть отрицательной). До тех пор пока мы будем вводить отрицательное или равное нулю число, цикл будет работать и требовать «Введите длину последовательности N = ».

Так как в цикле имеем два оператора, необходимо ставить операторные скобки Begin – End. В фигурных скобках {2} стоит номер операторных скобок. Напоминаем, что в фигурных скобках ставится комментарий, служащий для пояснения работы программы. Комментарий не вызывает никаких действий в программе, и его можно убрать без последствий для работы программы.

3. Оператор цикла REPEAT ... UNTIL

Этот оператор организует так называемый цикл «ДО», или цикл с постусловием, так как проверка логического < Условия > завершения цикла осуществляется после каждого выполнения тела цикла (количество повторений цикла неизвестно). Вид цикла:

REPEAT (Повторять)

< тело цикла >

UNTIL (До тех пор)

< условие >;

Оператор цикла с постусловием состоит из ключевого слова **repeat**, за которым следуют операторы тела цикла, и замыкающего ключевого слова **until**, после которого указывается < Условие > окончания цикла.

Логическое < Условие > выполнения цикла должно быть False, и как только < Условие > станет True (истинным), цикл прерывается.

Блок-схема цикла **repeat** приведена на рис. 22. Упрощенно общую структуру оператора **repeat** можно представить так:

REPEAT

Оператор;

Оператор;

.....

Оператор;

UNTIL < Условие >;

Оператор **repeat** не требует использования операторных скобок **Begin – End**, когда в его теле необходимо записать более одного оператора. Кроме того, оператор **repeat** является не столь универсальным, как оператор **while**, поскольку его логическое < Условие > располагается в конце тела цикла и операторы его тела будут всегда выполняться как минимум один раз. Выражение, используемое в качестве логического < Условия > окончания цикла, должно давать результат булевского типа.

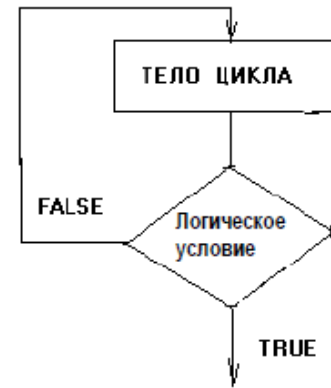


Рис. 22. Цикл «До» с постусловием

Задача 3

При вводе осуществить проверку длины вводимой последовательности.

Program Lab5_3;

Var

N: integer;

BEGIN {1}

Repeat

WriteLn ('Введите длину последовательности N = ');

ReadLn (N);

Until N > 0;

WriteLn ('Длина последовательности N = ', N);

END. {1}

Пояснения к задаче 3

Как и в задаче 1, длина последовательности обозначена буквой N и должна быть положительной.

В логическое < Условие > ставим $N > 0$, и до тех пор, пока будет вводиться отрицательное или нулевое значение последовательности, цикл **repeat** будет работать и требовать: «Введите длину последовательности N = ». Цикл выполняется до тех пор, пока пользователь будет вводить отрицательное число или ноль, и прервется при вводе положительного числа.

Нижняя граница оператора определена словом **until**, и нет надобности в операторных скобках Begin – End. Но если поставить операторные скобки, то это не будет ошибкой.

Задача 4

Вычислить значение функции $Y = X^2$ при $X = 8$; $X = 6$; $X = 4$; $X = 2$.

PROGRAM Lab5_4;

Uses Crt

Var X,Y: real;

Begin

ClrScr;

X := 8;

WriteLn('Табулирование функции');

WriteLn(' X Y ');

WriteLn('-----');

Repeat

Y := X * X;

WriteLn(X : 4 : 2, Y : 6 : 2);

X := X - 2;

Until X = 0;

ReadLn;

End.

4. Оператор цикла с параметром (со счетчиком) FOR ... DO

Оператор цикла со счетчиком `for` подходит для программирования только таких циклических фрагментов, в которых до выполнения цикла известны начальное и конечное значения счетчика повторений цикла. Это обуславливает его менее широкую сферу применения, чем более универсальных операторов `while` и `repeat`. Однако в тех случаях, когда его использование возможно, оператор `for` благодаря своей высокой наглядности имеет несомненное преимущество перед операторами циклов с условиями.

Схематично цикл со счетчиком изображается следующим образом (рис. 23).

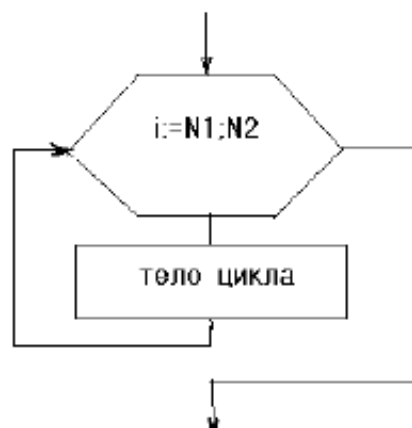


Рис. 23. Цикл со счетчиком

Упрощенно общую структуру оператора цикла `for` можно представить одним из следующих способов:

1. Если счетчик при выполнении цикла наращивает свое значение:

FOR I := N1 TO N2 DO

< тело цикла >;

где I – управляющая переменная цикла, которая должна быть перечисляемого типа (целая, символьная и т. д.);

N1 – начальное значение управляющей переменной цикла;

N2 – конечное значение управляющей переменной цикла.

Шаг изменения управляющей переменной равен + 1.

Чтобы оператор выполнился хотя бы один раз, N1 (начальное значение счетчика) должно быть не больше, чем N2 (конечное значение счетчика).

2. Если счетчик при выполнении цикла уменьшает свое значение, то при программировании может быть использован другой оператор цикла:

FOR I := N2 DOWNTO N1 DO

< тело цикла >;

где I – управляющая переменная цикла, которая должна быть перечисляемого типа (целая, символьная и т. д.);

N2 – начальное значение управляющей переменной цикла;

N1 – конечное значение управляющей переменной цикла.

Шаг изменения управляющей переменной равен -1.

Чтобы оператор выполнился хотя бы один раз, нужно, чтобы N2 (начальное значение счетчика) было не меньше, чем N1 (конечное значение счетчика).

Цикл `for` выполняется заданное число раз. Чтобы прервать выполнение досрочно, необходимо увеличить I до конечного значения.

Задача 5

Вычислить значения двух величин по формулам:

$A = 2I$ и $B = 2I + 1$, где I меняется от 1 до 4 с шагом 1.

Program Lab5_5

Var

I, B, A : Integer;

Begin {1}

For I := 1 To 4 Do

Begin {2}

A := 2 * I;

B := 2 * I + 1;

WriteLn (A : 3, B : 3);

End; {2}

End. {1}

Пояснения к задаче 5

I – управляющая переменная циклом (по ее значению цикл начинается и завершается) принимает значения: $I = 1, I = 2, I = 3, I = 4$ (шаг = 1).

Выполняются операторы циклической части: значения A и B вычисляются по формуле и принимают следующие значения в процессе выполнения программы (табл. 12). Всего цикл проработает четыре раза. При $I = 4$ последний раз идет вычисление A, B и управление передается за пределы цикла.

Таблица 12

I – управляющая переменная циклом, A, B – выражения	Проходы цикла			
	Первый	Второй	Третий	Четвертый
I	1	2	3	4
A := 2 * I	2	4	6	8
B := 2 * I + 1	3	5	7	9

Задания для практической работы

Вариант 1.

№	ФИО	№	ФИО
1.	Антипенко А.	9	Русяев Р.
2.	Вылегжанина Т.	10	Табула П.
3.	Грушенец А.	11	Тишкин В.
4.	Корниенко Р.	12	Хохрин Е.
5.	Красикова А.	13	Шевченко Д.
6.	Куринный М.	14	Шеина В.
7.	Лаврихин Д.	15	Щерба Е.
8.	Лобякова В.		

Составить программу планирования закупки товара в магазине на сумму, не превышающую заданную величину.

{Указание: ввести обозначения для цены, количества товаров, предельной суммы закупки, общей стоимости покупки.

Начальное значение общей стоимости покупки (s) равно нулю. Значение предельной суммы считывается с клавиатуры. Необходимо повторять запрос цены и количества выбранного товара, вычислять его стоимость, суммировать ее с общей стоимостью и выводить результат на экран до тех пор, пока она не превысит предельную сумму p . В этом случае на экран нужно вывести сообщение о превышении)

Вариант 2.

3. Написать программу, которая определяет максимальное число из введенной с клавиатуры последовательности положительных чисел (длина последовательности неограниченна). Ниже представлен рекомендуемый вид экрана во время работы программы (данные, введенные пользователем, выделены полужирным шрифтом).

Определение максимального числа последовательности положительных чисел.

Вводите после стрелки числа. Для завершения ввода введите ноль.

—> **56**

—> **75**

—> **43**

—> **0**

Максимальное число: 75

Вариант 3

В группе N студентов. Определить максимальный рост студента, вводя его по очереди с клавиатуры.

Вариант 4.

Написать программу, которая выводит на экран таблицу квадратов первых десяти целых нечетных чисел в следующем виде:

Число	1	3	5	7	9	И т.д.				
Квадрат	1	9	25	49	81					

Вариант 5.

Написать программу, которая вычисляет сумму первых пяти N чисел, введенных с клавиатуры. Количество чисел N – программа запрашивает у пользователя.

Вариант 6.

Даны два целых числа A и B ($A > B$). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество N этих чисел.

Вариант 7.

Написать программу, которая выводит таблицу квадратов первых десяти целых положительных чисел в следующем виде:

Число	1	2	3	4	5	6	7	8	9	10
Квадрат	1	4	9	16	25	36	49	64	81	100

Вариант 8.

Составить программу для вычисления произведения вводимых с клавиатуры чисел до первого отрицательного.

Вариант 9.

Составить таблицу значений функции $y = x^2 - 2$. Вывести значения в виде таблицы.

Вариант 10.

Составить таблицу значений функции $y = \sqrt{x} - 2$, где X - целое положительное четное число. Вывести значения в виде таблицы.

Вариант 11.

Дана последовательность чисел

Вариант 12.

Написать программу, которая выводит на экран таблицу умножения на 6.

Вариант 13.

Составить программу для вывода на экран таблицы кубов N чисел, введенных с клавиатуры.

Вариант 14.

Вывести на экран кубы чисел от 4 до 9.

Вариант 15.

Из чисел от 10 до 99 вывести те, сумма цифр которых равна S ($0 < S < 18$).

Контрольные вопросы:

7. Контрольные вопросы

1. Как называются программы, в основе которых лежит структура повторения?
2. Сколько операторов цикла вам известно?
3. Как называется оператор, синтаксическая диаграмма которого приведена на рис. 25?



Рис. 25

4. Какой тип выражения в операторе цикла с предусловием?
5. Верно ли, что в теле цикла с предусловием должен находиться один оператор?

6. Как сделать, чтобы в теле цикла было несколько операторов?
7. Когда проверяется истинность выражения в цикле while?
8. Верно ли, что истинность выражения в цикле с предусловием является условием продолжения цикла?
9. Сколько раз выполнится оператор в теле цикла while, если с самого начала значение выражения равно false?
10. Верно ли, что цикл while используется при вычислении всякого рода сумм и произведений, когда заранее не известно число повторений?
11. Как называется оператор, синтаксическая диаграмма которого приведена на рис. 26?

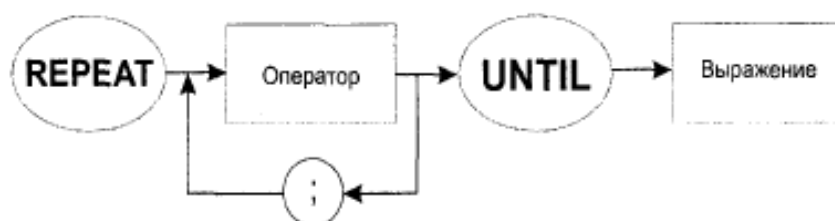


Рис. 26

12. Сколько операторов можно записать между ключевыми словами repeat и until?
13. Когда проверяется истинность выражения в операторе цикла repeat?
14. Какой тип имеет выражение в операторе цикла с постусловием?
15. Почему в цикле repeat оператор тела цикла всегда будет выполнен хотя бы один раз?
16. Верно ли, что истинность выражения в цикле repeat является условием окончания цикла?
17. С помощью каких служебных слов записывается цикл с параметром?
18. Верно ли, что цикл с параметром применяется в тех случаях, когда заранее известно число повторений?
19. Пусть цикл записан в виде
for I := N1 to N2 do S;
Тогда:
 - а) как называется N1?
 - б) как называется N2?
 - в) как называется I?
20. Сколько раз выполнится оператор S, если $N1 = N2$, а цикл записан в виде for I := N1 to N2 do S?
21. Сколько раз выполнится оператор S и можно ли изменить значение переменной I внутри цикла, если $N1 > N2$, а цикл записан в виде for I := N1 to N2 do S?
22. Сколько раз выполнится оператор S, если $N1 = N2$, а цикл записан в виде for I := N1 downto N2 do S?
23. Сколько раз выполнится оператор S, если $N1 < N2$, а цикл записан в виде for I := N1 downto N2 do S?
24. Верно ли, что в качестве параметра цикла for используется простая переменная, описанная в текущем блоке?

25. Какой тип должен иметь параметр цикла в цикле `for`?
26. Какой тип должны иметь начальное и конечное значения в цикле `for`?
27. Чему равно значение параметра цикла `for` после завершения цикла?
28. Можно ли войти в тело цикла `for`, минуя его заголовок?
29. Можно ли войти в тело цикла `while`, минуя его заголовок?
30. Можно ли войти в тело цикла `repeat`, минуя его заголовок?
31. Допустим ли выход из тела цикла `for`?
32. Допустим ли выход из тела цикла `while`?
33. Допустим ли выход из тела цикла `repeat`?
34. Какие циклы называются вложенными?
35. Верно ли, что цикл, содержащий в себе другой цикл, называется внешним?
36. Какой цикл называется внутренним?
37. Какому условию должны удовлетворять вложенные циклы?
38. Может ли внешний цикл быть циклом `for`, а внутренний – циклом `repeat`?
39. Могут ли внутренний и внешний циклы быть циклами разных видов?
40. Могут ли внутренний и внешний циклы быть циклами одного вида?
41. Может ли вещественная переменная быть параметром цикла `for`?
42. Можно ли в качестве параметров вложенных циклов `for` использовать одну и ту же переменную?
43. Какие утверждения являются верными (проиллюстрируйте ответ соответствующими примерами):
 - а) любой цикл `for` можно записать при помощи цикла `while`;
 - б) любой цикл `while` можно записать при помощи цикла `for`;
 - в) не любой цикл `for` можно записать при помощи цикла `while`;
 - г) не любой цикл `while` можно записать при помощи цикла `for`.