



Урок 7. Оператор цикла с предусловием

Задачи урока: Содействовать формированию








Создавать условия для воспитания

Способствовать развитию

Учащиеся должны знать: алгоритмическую конструкцию «повторение»

Учащиеся должны уметь: реализовывать на языке программирования алгоритмы с повторением

ТЕХНОЛОГИЧЕСКАЯ КАРТА УРОКА

Этап урока	Вре- мя	Теория	Практика
1. Организацион-ный момент.	1	Вход в урок. Создание условий для положительной мотивации учащихся. Входная рефлексия.	
2. Проверка домашнего задания	2-8	Загрузка сервера тестирования: MyTestServer	
		Раздать файл:  Т 6.mtf (вопросов-, время- мин.)	
		 § (?), стр.	
		Загрузка сетевой программы: NetOp School	
3. Изучение нового материала	9-15	Вопросы:	
4. Закрепление знаний	16-2 5	 § (?), стр.	 ПР 7 №1 ⁰ ,2 ⁰
5. Контроль знаний	26-4 2	 ПР 7 №3-5	
		Загрузка сервера тестирования: MyTestServer	
		Раздать файл:  Т 7.mtf (вопросов-, время- мин.)	
		 К 7	

6. Итог урока	43-4 5	Подведение итогов урока. Выставление оценок в журнал и дневники. Ответы на вопросы.
7. Рефлексия		
8. Домашнее задание		§ 3, (? 4-6), упр. 5б. Завершение урока.

Тема12: «Основы алгоритмизации и программирования»

Урок 6: Запись на языке программирования. Реализация алгоритмов с повторением. Реализация алгоритмов с повторением (WHILE)

Цели урока:

- знакомство с циклами с предусловием;
- формирование умений и навыков записи оператора WHILE на языке программирования Pascal;
- создание условий для развития логического и алгоритмического мышления учащихся, развития памяти и внимания;
- формирование у учащихся компьютерной грамотности и потребности в приобретении знаний;
- привитие учащимся навыков самостоятельной работы;

План урока:

- I. Организационно-мотивирующий этап.
- II. Повторение + проверка знаний по пройденному материалу.
- III. Изучение нового материала
- IV. Закрепление пройденного материала (физкультминутка).
- V. Контрольно-оценочный этап.
- VI. Рефлексия.
- VII. Д/з.

Ход урока

I. Организационно-мотивирующий этап.

II. Повторение + проверка знаний по пройденному материалу.

1. Проверка домашнего задания.
2. Актуализация знаний.

Актуализация знаний и умений учащихся осуществляется во время фронтального опроса, в ходе которого им предстоит ответить на следующие вопросы:

1. Дайте определение циклического алгоритма.
2. Запишите оператор цикла FOR в общем виде.
3. Найдите и исправьте ошибки в записи оператора FOR: (слайды 2- 3)

for i:=5 to 1 do write(i, ' ');	for i:=1 downto 5 do write(i, ' ');	
x:=1; for i:=1 to 5 do write(x, ' '); x:=x+2;	for i:=1 to 5 do ; begin write(x, ' '); x:=x+2; end;	for i:=1 to 5 do begin write(x, ' '); x:=x+2; end.

4. Определите, сколько раз выполнится цикл (слайд 4):

- a) for i:=2 to 5 do k:=k + 1;
- б) for i:=5 to 5 do k:=k + 1;
- в) for i:=6 to 3 do k:=k + 1 ;
- г) for i:=5 downto 5 do k:=k + 1;
- д) for i:=5 downto 4 do k:=k + 1 ;
- е) for i:=i to i do k:=k + 1

III. Изучение нового материала

1. *Определение темы и целей урока.*

Тема сегодняшнего урока “ Реализация алгоритмов с повторением (WHILE)” (слайд 1). Первая часть темы для вас знакома. На предыдущих уроках вы учились строить алгоритмы с использованием команды повторения и реализовывать их на языке программирования. Однако приобретенных знаний нам недостаточно для решения задач и скоро вы в этом убедитесь.

Итак, цель нашего урока ...

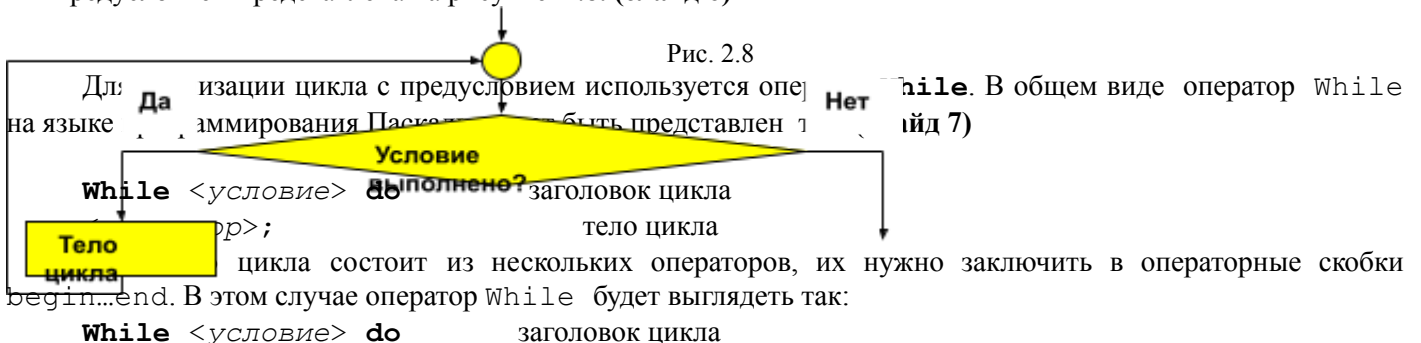
2)Решите следующую задачу: **Определить и вывести на экран цифры целого числа n.** (слайд 5)

Решая поставленную задачу, в процессе рассуждений учащиеся должны сделать вывод о том, что изученный ими ранее оператор цикла с параметром (оператор FOR) не позволяет решить задачу, поскольку заранее неизвестно количество повторений цикла. Учащиеся предполагают, что для решения поставленной задачи им нужно использовать какой-то другой оператор цикла.

3)Оператор цикла `for` отлично выполняет свои функции, когда число повторений действий заранее известно. Однако такая ситуация встречается в программировании далеко не всегда. Часто приходится решать задачи, когда число повторений действий в теле цикла неизвестно и определяется в ходе решения задачи. В этом случае применяют *цикл с условием*. В языке программирования Pascal имеется две разновидности цикла с условием:

- цикл с предварительным условием* - условие цикла проверяется перед выполнением тела цикла;
- цикл с последующим условием* - условие цикла проверяется после выполнения тела цикла.

Остановимся на цикле с предварительным условием (кратко, с предусловием). **Цикл с предусловием** – это цикл, который повторяется до тех пор, пока условие выполняется (истинно). Блок-схема цикла с предусловием представлена на рисунке 2.8. (слайд 6)



Begin

```

    <оператор_1>;           тело цикла
    <оператор_2>;
    .....
```

End;

Значение выражения *<условие>*, записанное после слова **while**, проверяется перед каждым выполнением оператора (операторов) тела цикла.

Если *<условие>* верно (истинно), выполняется тело цикла, и снова вычисляется и проверяется выражение *<условия>*. Если результат проверки *<условия>* неверен (ложный), происходит выход из цикла.

Сразу заметим, что в программе может быть такая ситуация, когда цикл вообще не выполняется ни разу.

Если *условие* всегда оказывается истинным, то может произойти *заикливание*:

```
while 2>1 do
```

```
    write(1);
```

Чтобы прервать заиклившуюся программу, следует использовать комбинацию клавиш **Ctrl-F2** или с помощью команды *Программа -> Завершить*.

Тело цикла **while** в **Pascal ABC** не может быть пустым. Это сделано для защиты от часто встречающейся у начинающих ошибки ставить после **do** точку с запятой.

Следует знать:

- число повторений операторов цикла **WHILE** определяется в ходе работы программы и как, правило, заранее неизвестно;
- в операторе цикла **WHILE** точка с запятой никогда не ставится после зарезервированного слова **DO**;
- для успешного завершения цикла **WHILE** в его теле обязательно должны присутствовать инструкции, оказывающие влияние на условие выполнения инструкций цикла;
- инструкции тела цикла **WHILE** *вообще могут быть не выполнены*, если проверяемое условие ложно с самого начала;
- исходя из последнего утверждения, цикл **WHILE** считают самым *универсальным* видом цикла.

4) Рассмотрим работу оператора **while ... do ...** на примере. Вспомним программу для вычисления суммы **N** первых натуральных чисел, записанных с помощью оператора цикла **FOR** (слайды 8-9)

Пример 1. Сумма N первых натуральных чисел

```

program summa;
var i,n,sum:integer;
begin
    write('n =');
    readln (n);
    sum:=0;
    for i:=1 to n do
        sum:=sum+i;
        writeln('Sum =',sum);
    end.
```

Изменим программу, используя оператор цикла **while**:

```

program summa;
var i,n,sum:integer;
begin
    write('n =');
    readln (n);
    sum:=0;
    i:=1;
    while i<=n do
    begin
        sum:=sum+i;
        i:=i+1;
    end;
    writeln('Sum =',sum);
end.
```

4) Таким образом, одну и ту же задачу можно решить разными способами, однако при решении некоторых задач нельзя обойтись без оператора цикла **while**. Вспомним задачу, которую мы сформулировали в начале урока. (на слайде 8 переход к слайду 5)

Пример 2. Определить и вывести на экран цифры целого числа n.

Решение. Разберем решение этой задачи на частном примере.

Найдем цифры числа 4538. Для этого надо найти остаток от деления 4538 на 10 с помощью операции нахождения остатка от деления целых чисел (**mod**): $4538 \bmod 10 = 8$, получим последнюю цифру числа (она же является первой справа).

Выдаем сообщение: "1 - я цифра справа равна 8".

После этого выполним целочисленное деление заданного числа 4538 на 10, получим 453 (остаток отбрасывается): $4538 \text{ div } 10 = 453$.

Далее процесс повторяем:

2 - й раз; $453 \bmod 10 = 3$

2 - я цифра справа равна 3,

$453 \text{ div } 10 = 45$,

3 - й раз; $45 \bmod 10 = 5$,

3 - я цифра справа равна 5,

$45 \text{ div } 10 = 4$,

4 - й раз; $4 \bmod 10 = 4$,

4 - я цифра справа равна 4,

$4 \text{ div } 10 = 0$.

Обратите внимание! Процесс будет продолжаться *пока* число *n* не равно нулю. Как только оно станет равно нулю *цикл* заканчивается.

В программе еще надо указывать, какая по счету цифра в числе справа. Для этого надо завести еще одну переменную в качестве счетчика. Эта переменная каждый цикл должна увеличиваться на 1.

Программа (переход к слайду 10)

```
Program Problem1; {Опред. и вывод на экран цифр числа.}
var n, p, i : integer;
begin
  write('Введите натуральное число n <= 32767 '); readln(n);
  i := 1;
  while n > 0 do
    begin
      p := n mod 10;
      writeln(i, ' - я цифра справа равна ', p);
      n := n div 10;
      i := i + 1;
    end;
  end.
```

IV . Закрепление пройденного материала (физкультминутка)

- Введите эту программу и выполните на ПК.
- На основе данной программы составьте программу, которая подсчитывает сумму цифр заданного числа (самостоятельно).

Program Problem2; {Сумма цифр числа}

```
var n, p,s:integer;
begin
  write('Введите натуральное число n <= 32767 ');
  readln(n);
  s:=0;
  while n > 0 do
    begin
      p := n mod 10;
      s:=s+p;
      n := n div 10;
    end;
    writeln(' s= ', s);
  end.
```

V. Контрольно-оценочный этап.

Выполнение теста.

1. Команда while означает
 - а) пока;
 - б) до свидания;

в) если;

г) привет.

2. Когда окончится выполнение цикла:

```
while a<b do
```

```
a:=a+1;
```

а) когда а станет больше b;

б) сразу закончится;

в) когда а станет равно b;

г) цикл не закончится.

3. Установите соответствие:

1. While

2. do

3. for

4. to

а) для;

б) пока;

в) до;

г) делать.

4. Какая из команд не является командой алгоритма с циклом?

а) Do;

б) While;

в) For;

г) else.

5. Когда применяется оператор с циклом?

а) когда в алгоритме много раз повторяются одни и те же действия;

б) когда команды алгоритма идут одна за другой и выполняются по одному разу;

в) когда в зависимости от условия мы поступаем так или иначе.

6. Определите, завершится ли когда-нибудь цикл с условием $x*x+1>0.8$

а) завершится;

б) не завершится.

VI. Рефлексия.

1) В каких случаях используется оператор цикла WHILE?

2) Почему цикл, записанный с помощью оператора WHILE, называют циклом с предусловием?

3) Можно ли данный оператор цикла назвать универсальным?

4) Было ли вам комфортно сегодня на уроке?

5) Что нового вы узнали на сегодняшнем уроке?

6) Решение какой задачи было для вас наиболее сложным?

7) С какими именно трудностями вы столкнулись при решении этой задачи?

VIII. Задание на дом. (слайд 12)

1. §3, п. 3.3.

2. Выполнить упражнения:

1). Может ли завершиться выполнение цикла while, условие которого выглядит так: $a*a+b*b+2>1$?

2) Сколько звездочек будет напечатано в результате выполнения приведенного фрагмента программы:

```
i:=1; readln(n);  
while i<=n do  
  begin  
    write('*');  
    i:=i+2;  
  end;
```

при а) n=5; б) n=0 в) n=1 ?

3) – 4) Сколько слагаемых должно быть в сумме $1+2+3+4+\dots$, чтобы эта сумма оказалась а) больше 20; б) больше некоторого данного числа N. Составьте программы и блок-схемы.

5) Составьте программу, которая вычисляет сумму натуральных чисел на отрезке $[5, 10]$ с помощью оператора While.

Общий вид оператора цикла с предусловием

1) While <условие> do
 <оператор>; - Тело цикла

2) While <условие> do
 begin
 <оператор 1>;
 <оператор 2>;
 ...
 <оператор n>;
 end; - Тело цикла

слайд 4

Работа цикла While

Тело цикла выполняется до тех пор, пока условие ИСТИННО, если условие ЛОЖЬ цикл завершается.

Пример 1 (Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы)

```
S:=0;  
i:=1;  
While i<=4 do  
  begin  
    S:=s+2*i;  
    i:=i+1;  
  end;
```

1 шаг

1<=4 (истинно)

S:=0+2*1=2;

i:=1+1=2;

2 шаг

2<=4 (истинно)

S:=2+2*2=6;

i:=2+1=3;

3 шаг

3<=4 (истинно)

S:=6+2*3=12;

i:=3+1=4;

4 шаг

4<=4 (истинно)

S:=12+2*4=20;

i:=4+1=5;

5 шаг

5<=4 (ЛОЖЬ)

Ответ S=20

Для закрепления рассматриваем несколько примеров с циклом While. Предлагаю обсудить в парах решения данных примеров.

слайд 5

Пример 2 Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы

```
S:=0;  
i:=1;  
While i<=4 do  
begin  
S:=s+2*i;  
i:=i+2;  
end;
```

Условие истинно при $i=1$ и $i=3$

Значит тело цикла выполнится 2 раза

S=8

слайд 6

Пример 3 Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы

```
S:=0;  
i:=1;  
While i<=4 do  
begin  
S:=s+2*i;  
i:=i+4;  
end;
```

Условие истинно при $i=1$

Значит тело цикла выполнится 1 раз

S=2

слайд 7

Пример 4 Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы

```
S:=0;  
i:=5;  
While i>3 do  
begin  
S:=s+i;  
i:=i -1;  
end;
```

Условие истинно при $i=5$ и $i=4$

Значит тело цикла выполнится 2 раза

S=9

слайд 8

Пример 5 Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы

```
S:=0;  
i:=5;  
While i<=3 do  
begin  
S:=s+i;  
i:=i +1;  
end;
```

Условие ложно

Значит тело цикла не выполнится ни разу

S=0

слайд 9

Пример 6 Сколько раз выполнится тело цикла и чему будет равно значение S после выполнения фрагмента программы

```
S:=0;  
i:=1;  
While i<=5 do  
begin  
S:=s+i;  
i:=i -1;  
end;
```

Условие всегда истинно

Тело цикла будет выполняться до бесконечности.

Произойдет зависание программы или заикливание

А сейчас рассмотрим задачу, решение которой мы решили на прошлом уроке.

слайд 10

Задача 1 Составить программу нахождения частичной суммы ряда натуральных чисел

$S=1+2+3+4+\dots+n$

Предлагаю подумать и дописать пропущенный фрагмент программы, используя цикл While, используя интерактивную доску.

<pre>Program Z1; Var S, n, i : integer; Begin clrscr; Writeln ('Введите количество слагаемых'); Read(n); S:=0; For i:=1 to n do S:=S+i; Writeln ('Сумма =',S); End.</pre>	<pre>Program Z2; Var S, n, i : integer; Begin clrscr; Writeln ('Введите количество слагаемых'); Read(n); S:=0; Writeln ('Сумма =',S); End.</pre>
---	---

ФИЗКУЛЬТМИНИТКА (на выбор учителя)

3. Контрольно – коррекционный этап (15 мин).

слайд 11

На этом этапе ученикам предлагается на выбор 1 из 3 задач. Перед самостоятельной работой разбираем все 3 задачи.

Далее проводится самопроверка и самоанализ результатов самостоятельной деятельности.

4.Рефлексивно – диагностирующий этап (2 мин).

- Что на уроке вы узнали для себя нового?
- До каких пор выполняются тело цикла в операторе While?
- Что необходимо помнить, когда работаем с циклом While?
- Всегда ли можно оформить цикл с помощью оператора FOR?
- В каких случаях лучше использовать оператор While?
- В чём основное отличие оператора While от оператора FOR?
- А как вы думаете, что мы будем делать на следующем уроке?

5. Организация домашней работы (1 мин.)

Опорный конспект. Оформить все задачи из опорного конспекта в тетради