The background is a vibrant blue with a digital theme. It features a network diagram on the left with four white nodes connected by lines. In the center, there are several lines of binary code (0s and 1s) in a light blue font. On the right, there is a target symbol consisting of three concentric white circles and a central white dot. At the bottom, there is a stylized globe with a grid pattern. The text "deposit photos" is repeated in a light blue font across the background.

Решение задач с использованием оператора циклов

по теме: «Решение задач с зованием оператора select case»

1. Формат оператора case имеет следующий вид

- a) Case <переключатель> of
<список значений 1> : <оператор 1>;
...
Else <оператор m>;
end;
- b) If <логическое выражение> then <оператор 1> else <оператор 2>.
- c) <переменная> := <выражение>.

ТЕСТ

по теме: «Решение задач с использованием оператора `select case`»

2. Может ли одно и то же значение переключателя повторяться в разных списках:

- а) да;
- б) нет;
- в) зависит от условия задачи.

ТЕСТ

по теме: «Решение задач с зованием оператора `select case`»

3. Какого типа переменную нельзя использовать в операторе `case` :

- a) `integer` ;
- b) `char`;
- c) `real`.

ТЕСТ

по теме: «Решение задач с зованием оператора `select case`»

4. Дополните: Оператор выбора `Case` является обобщением оператора:

- a) `Write`;
- b) `if`;
- c) `read`.

по теме: «Решение задач с использованием оператора `select case`»

5. Оператор выбора `Case` позволяет:
- a) присвоить определенное значение переменной;
 - b) выполнить одно из нескольких действий (2-4);
 - c) выполнить одно из нескольких действий (2 и более).

ТЕСТ

по теме: «Решение задач с использованием оператора `select case`»

6. Обязательным элементом завершающим запись оператора `Case` является:

- a) «end»
- b) «;»
- c) «end.»

Что такое цикл

Цикл - это повторение некоторой группы команд (тело цикла).

Посмотрим, как осуществить цикл на Паскале. Предположим, мы хотим, чтобы компьютер бесконечно повторял выполнение следующего фрагмента:

```
Write ('Это ');
```

```
Write ('тело ');
```

```
Write ('цикла');
```

```
Write (' ');
```

в результате чего на мониторе мы бы увидели:

```
Это тело цикла  Это тело цикла  Это тело цикла  Это тело цикла  . . . .
```

Большинство языков программирования (в том числе и Паскаль) устроены так, что операторы выполняются в том порядке, в котором они записаны. Это значит, что после оператора `Write ('Это ')` обязательно выполнится оператор `Write ('тело ')`, а после него - `Write ('цикла')`, а после него - `Write (' ')`.

Если нам нужно, чтобы после выполнения оператора Write (' ') Паскаль нарушал этот свой принцип последовательного выполнения операторов и выполнял бы оператор Write ('Это '). Если мы этого добьемся, то дальше все пойдет само собой, так как после Write ('Это ') Паскаль автоматически выполнит Write ('тело ') и так далее до бесконечности.

Если бы операторы Паскаля можно было писать по-русски, то для достижения нашей цели было бы естественно воспользоваться такой конструкцией:

метка m1: Write ('Это ');

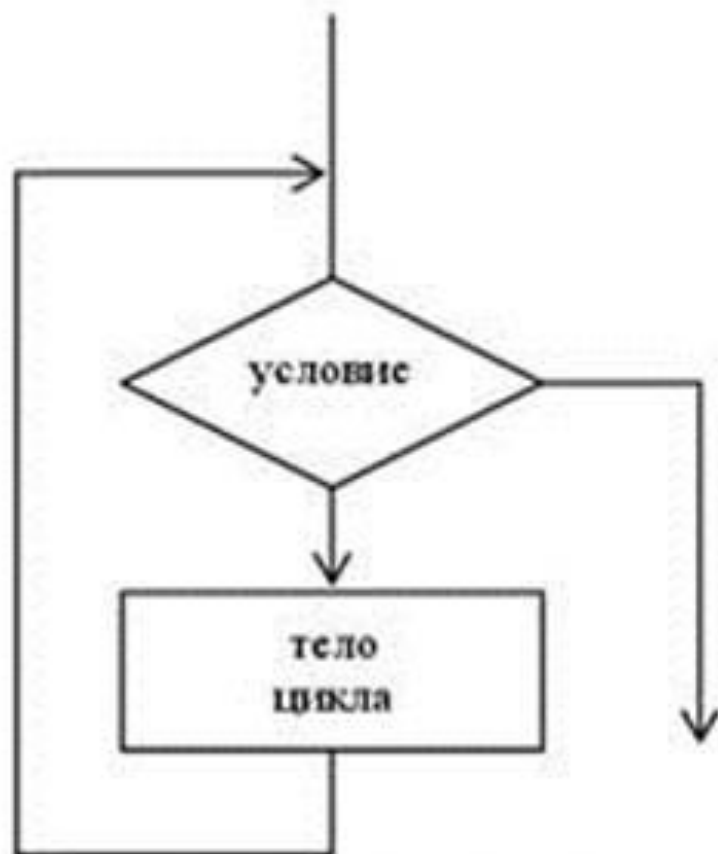
Write ('тело ');

Write ('цикла');

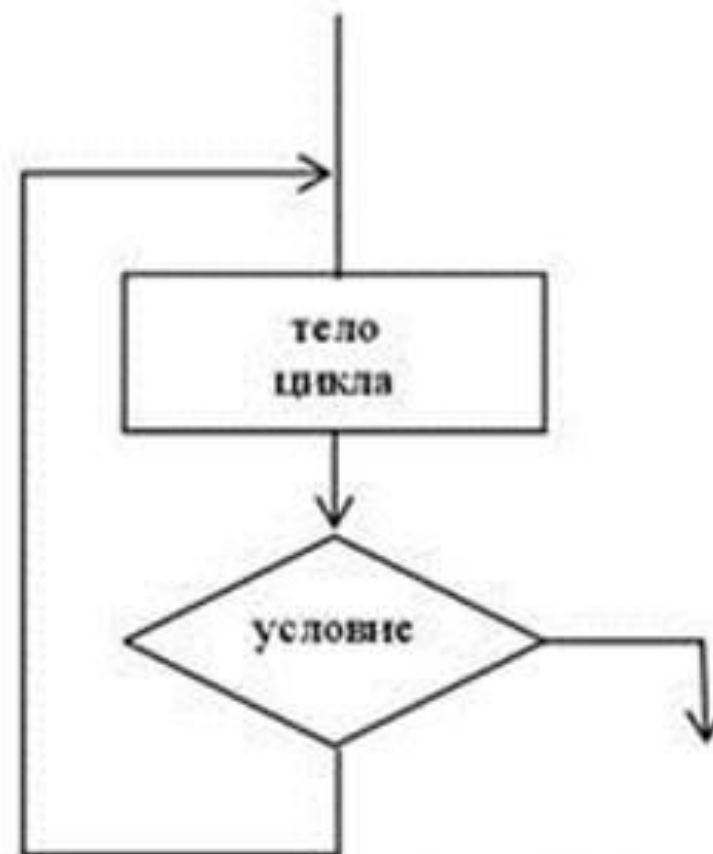
Write (' ');

иди к оператору, помеченному меткой m1

Виды блок-схем для оператора циклов



Цикл с предусловием



Цикл с постусловием

Оператор GOTO

А вот как этот фрагмент выглядит реально на Паскале:

```
m1: Write ('Это ');  
      Write ('тело ');  
      Write ('цикла');  
      Write (' ');  
      GOTO m1
```

Здесь **GOTO** – оператор перехода, читается " 'гоуту", переводится "иди к", m1 - метка.

Метка – это произвольное имя или произвольное не слишком большое целое положительное число. Оператор GOTO можно писать в любых местах программы и метку можно ставить перед любым оператором, заставляя компьютер таким образом перескакивать в программе откуда угодно куда угодно.

Метка должна отделяться от оператора двоеточием.

Как мы знаем, переменная, встречающаяся в программе, должна быть описана выше BEGIN после слова VAR. Метки, встречающиеся в программе, тоже должны быть описаны выше BEGIN после слова LABEL (читается "лэйбл", переводится "метка").

Вот наша программа полностью:

```
LABEL m1;  
BEGIN  
  m1:  Write ('Это ');  
        Write ('тело ');  
        Write ('цикла');  
        Write (' ');  
        goto m1  
END.
```

Группа операторов, выполняющихся многократно, называется **телом цикла**. У нас это все операторы, начиная с Write ('Это ') и кончая GOTO m1.

Поэтапное решение задачи

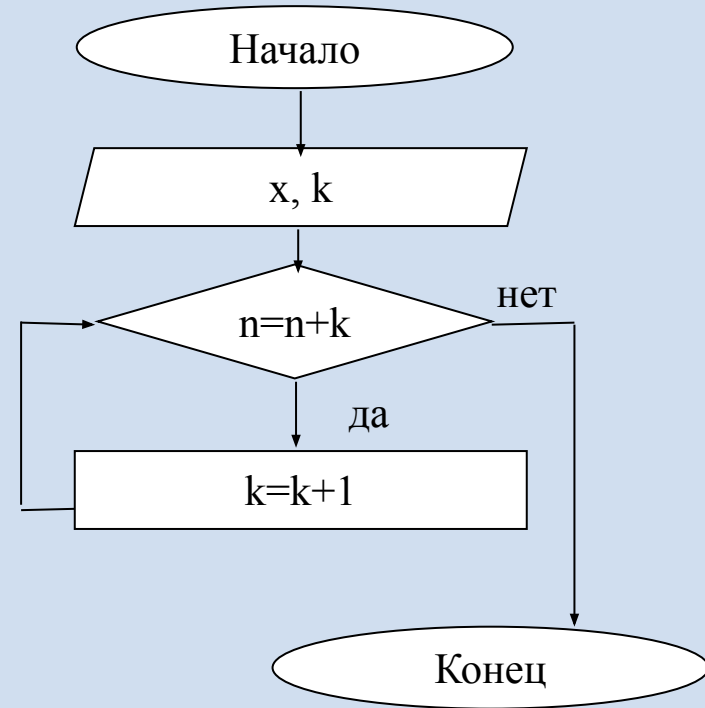
Задание 1. Напишите программу, которая после вывода сообщения «Считаем зайцев» начала бы бесконечный счет «зайцев» со значения 5 с непостоянным интер-валом, который бы каждый раз увеличивался на 1 (то есть: 5, 6, 8, 11 и т. д.).

Этап решения	Описание
1. Постановка задачи – <i>определение исходных данных и искомых результатов</i>	Дано $n=5$. Нужно задать цикл с переменной k , если $k = k+1$.
2. Формализация – <i>переход к задаче обработки некоторый знаковой системы (математ. задаче)</i>	$n = n+k$ $k = k+1$

Этап решения

Описание

3. Построение алгоритма – *определение структуры алгоритма, последовательность команд*



Этап решения	Описание
4. Составление программы – запись и отладка программы на языке программирования	<pre>PROGRAM Here; LABEL m1, met5; VAR n,k : Integer; BEGIN n:=5; WriteLn('Считаем зайцев'); met5: Write(n); n:=n+k; goto m1; n:=n+1; m1: Write(' зайцев '); ReadLn; k:=k+1; goto met5; WriteLn('Посчитали зайцев') END.</pre>
5. Тестирование	Протестируйте программу на Турбо Паскале 7.0

Поэтапное решение задачи

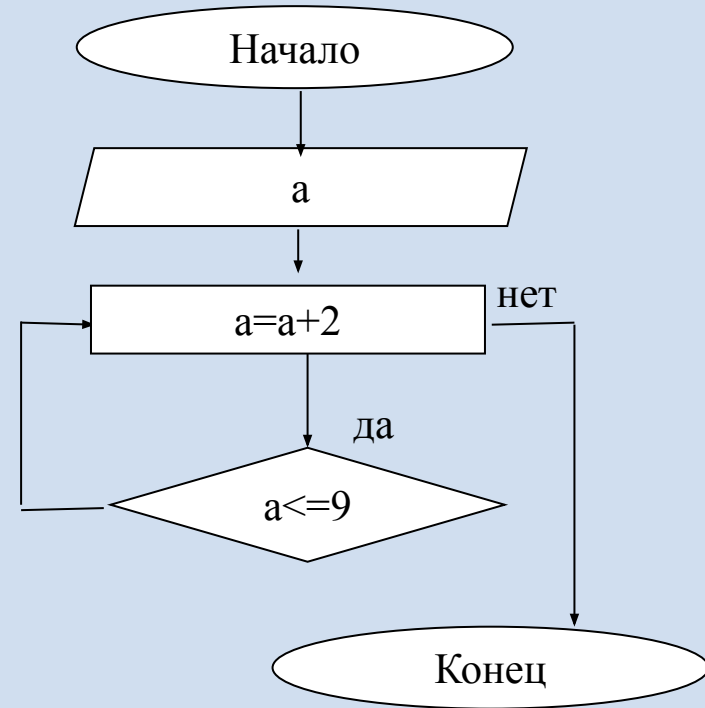
Задание 3. Напишите программу, которая после вывода сообщения «Начало счета» начала бы счет от 3 до 9 с интервалом 2, а после выводила бы сообщение «Конец счета».

Этап решения	Описание
1. Постановка задачи – <i>определение исходных данных и искомых результатов</i>	Дано $a=3$. Нужно задать цикл от 3 до 9 с интервалом $=2$.
2. Формализация – <i>переход к задаче обработки некоторой знаковой системы (математ. задаче)</i>	$a = a + 2$ $a \leq 9$

Этап решения

3. Построение алгоритма – *определение структуры алгоритма, последовательность команд*

Описание



Этап решения	Описание
4. Составление программы – запись и отладка программы на языке программирования	<pre>PROGRAM Counting2; LABEL m; VAR a : Integer; BEGIN Write('Начало счета '); a:=3; m: Write(f, ' '); a:=a+2; if a<=9 then goto m; Write(' Конеч счета') END.</pre>
5. Тестирование	Протестируйте программу на Турбо Паскале 7.0

Инструкция к выполнению практической работы № 8

Тема: Решение задач с использованием оператора циклов.

Цель: научиться составлять программы, используя оператор циклов.

Ход работы

Задание 1. Напишите программу, которая после вывода сообщения «Считаем зайцев» начала бы бесконечный счет «зайцев» со значения 5 с не-постоянным интервалом, который бы каждый раз увеличивался на 1 (то есть: 5, 6, 8, 11 и т. д.).

Задание 2. Напишите программу, которая после вывода сообщения «Начинаем обратный отсчет» начала бы вести отсчет в обратном направлении со значения 1000 с интервалом -1.

Задание 3. Напишите программу, которая после вывода сообщения «Начало счета» начала бы счет от 3 до 9 с интервалом 2, а после выводила бы сообщение «Конец счета».

Контрольные вопросы:

1. Что такое цикл?
2. Для чего предназначен оператор циклов `goto`?
3. Какое значение имеет «метка» для оператора `goto`?
Какое правило написания метки?
4. Дайте определение «тело цикла»?
5. Запишите форму оператора `goto` внутри оператора `if`?

Домашнее задание

Семакин И. Г. Информатика. Базовый уровень: учебник для 10 класса. С. 143-163.