1 курс

ПЛАН – КОНСПЕКТ
проведения занятий (лекционного, практического № 19) по дисциплине «Информатика»

Раздел 4. «Основы алгоритмизации и программирования.»

Тема 4.1: «Общие принципы построения базовых алгоритмических структур в среде программирования.»

часть 6

Подготовил: преподаватель
В.Н. Борисов

Рязань 2023

Вопросы занятия:

1. Алгоритмическая структура «Ветвление».
2. Оператор условного перехода.
3. Неполная и полная формы оператора условного перехода.
4. Программирование условного алгоритма (практическое занятие №19, теоретическая часть).

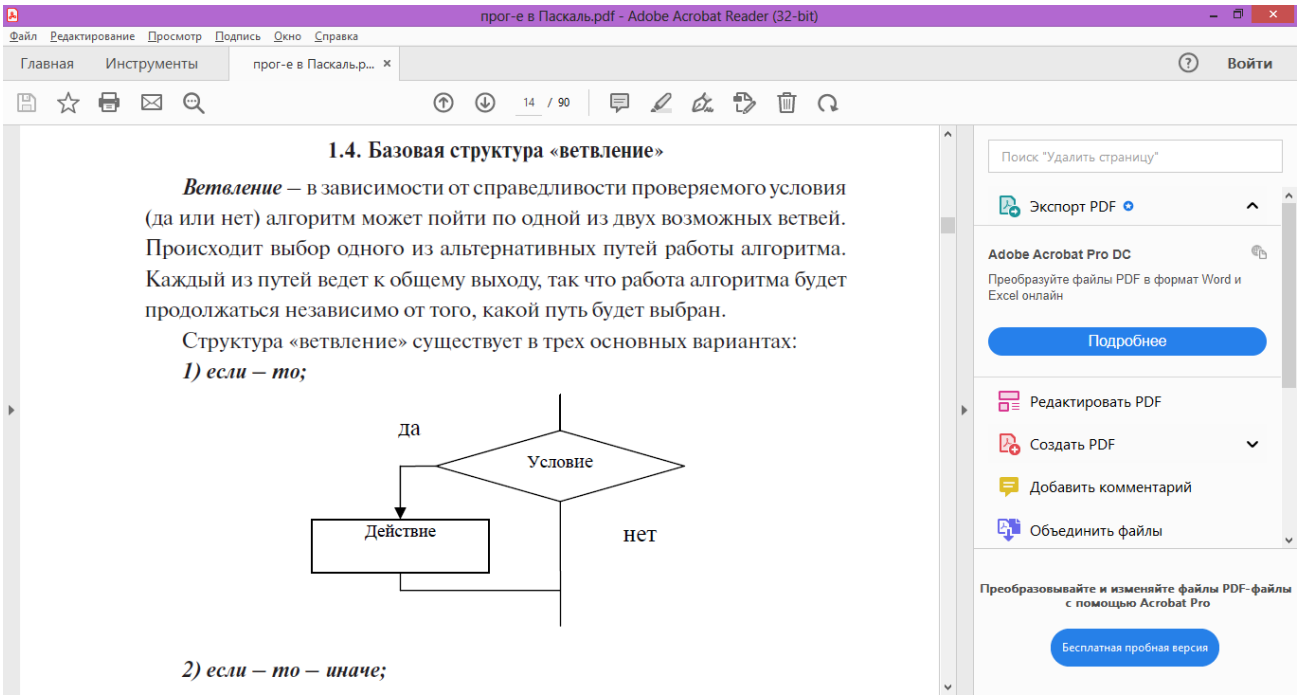
Время проведения занятий (лекционного, практического) – 4 часа.

Первый вопрос: Алгоритмическая структура «Ветвление».

Ветвление – в зависимости от справедливости проверяемого условия (да или нет) алгоритм может пойти по одной из двух возможных ветвей. Происходит выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу так, что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура «ветвление» существует в трех основных вариантах:

1) если – то;
2) если – то – иначе;
3) выбор.



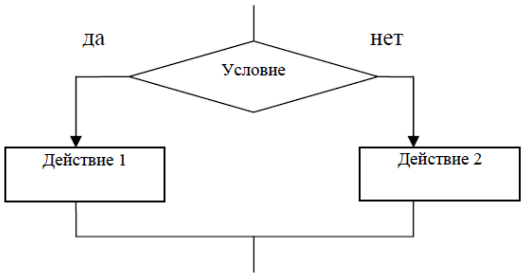
прог-е в Паскаль.pdf - Adobe Acrobat Reader (32-bit)

Файл Редактирование Просмотр Подпись Окно Справка

Главная Инструменты прог-е в Паскаль.р... Войти

14 / 90

2) если – то – иначе;



Поиск "Удалить страницу"

Экспорт PDF

Adobe Acrobat Pro DC

Преобразуйте файлы PDF в формат Word и Excel онлайн

Подробнее

Редактировать PDF

Создать PDF

Добавить комментарий

Объединить файлы

Преобразовывайте и изменяйте файлы PDF-файлы с помощью Acrobat Pro

Бесплатная пробная версия

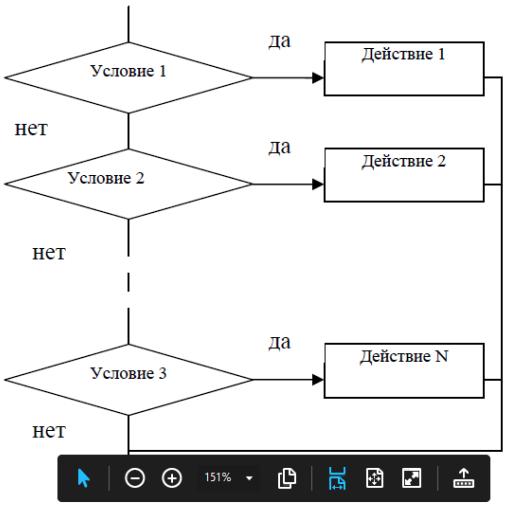
прог-е в Паскаль.pdf - Adobe Acrobat Reader (32-bit)

Файл Редактирование Просмотр Подпись Окно Справка

Главная Инструменты прог-е в Паскаль.р... Войти

15 / 90

3) выбор.



Поиск "Удалить страницу"

Экспорт PDF

Adobe Acrobat Pro DC

Преобразуйте файлы PDF в формат Word и Excel онлайн

Подробнее

Редактировать PDF

Создать PDF

Добавить комментарий

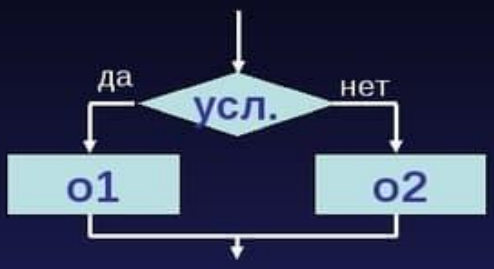
Объединить файлы

Преобразовывайте и изменяйте файлы PDF-файлы с помощью Acrobat Pro

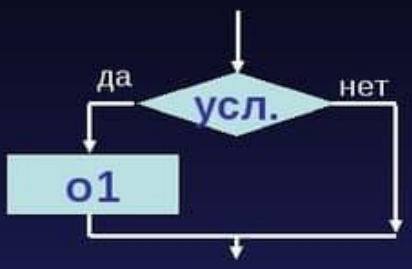
Бесплатная пробная версия

Второй вопрос: <u>Оператор условного перехода.</u>

Условный оператор на языке Pascal



полная



неполная

If <условие>
(если)
then <оператор1>
(то)
else <оператор2>;
(иначе)

If <условие>
then <оператор1>;

Условный оператор

Условные операторы в QBasic и Turbo Pascal 7.0 помогают нам осуществить "ветвление" программы, т.е. передать управление по условию.

Условный оператор имеет вид:

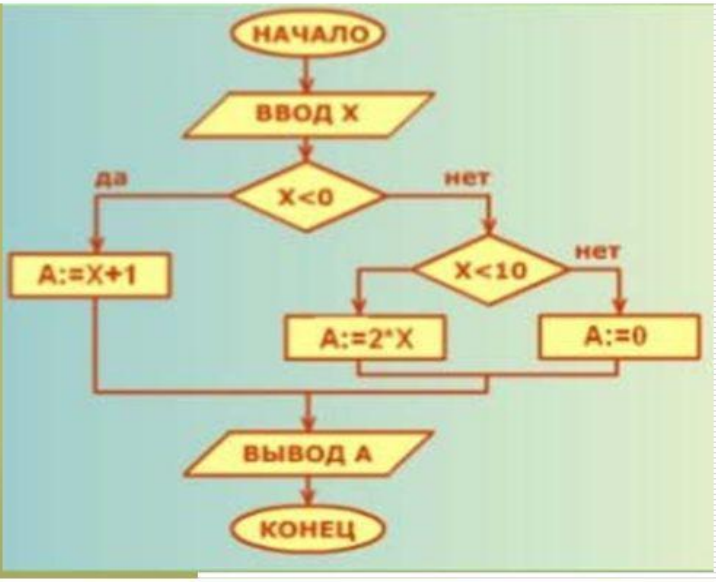
IF *условие* THEN <*операторы1*> [ELSE <*операторы2*]>

Выполнение условного оператора начинается с вычисления значения логического выражения, записанного в условии. Простые условия записываются в виде равенств или неравенств. Сложные условия составляют из простых с помощью логических операций.

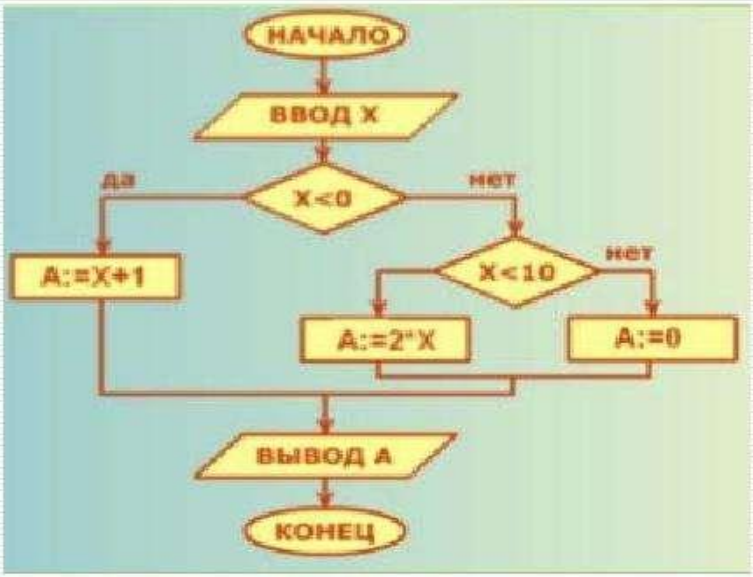
Если условие истинно, то выполняется <операторы1>, в противном случае -<операторы2>.

Программа, соответствующая этому алгоритму, выглядит так

```
Program Calculate_A;
Var a, x: integer;
BEGIN
   Writeln('Введите x');
   Readln(x);
   If (x<0)
      then a:=x+1
      else if (x<10)
            then a:=2*x
            else a:=0;
    Writeln('A = ',a);
Readln;
END.
```



Блок-схема, соответствующая этому алгоритму, выглядит так



$$A = \begin{cases} x+1, & x < 0 \\ 2 \cdot x, & 0 \le x < 10 \\ 0, & x \ge 10 \end{cases}$$

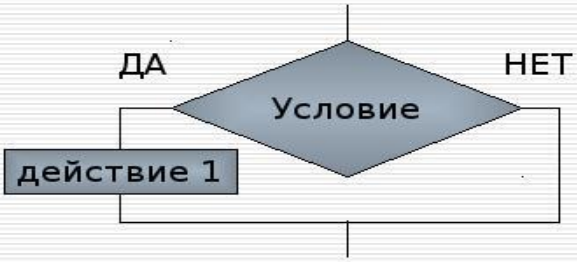
Составной условный оператор

Если в качестве оператора должна выполниться серия операторов, то они объединяются в операторные скобки **begin-end**.

```
IF условие THEN
        BEGIN действие1; действие 2; END
            ELSE
        BEGIN действие3; действие 4; END;
```

Третий вопрос: <u>Неполная и полная формы оператора условного перехода.</u>

Условный оператор (неполное ветвление)




```
IF условие THEN действие1;
```

Условный оператор (полное ветвление)

IF условие THEN действие1
ELSE действие2;

В качестве условий используются логические выражения, например, такие:

(C=D)

(a>b) and (a>c) or (a=d)

Условный оператор if…then…else полная форма

Синтаксис:

if <логическое выражение>
then <оператор1>
else <оператор2>;

Семантика:

1. Вычисляется значение **<логического выражения>**
2. Если **<логическое выражение>** истинно (TRUE), то выполняется оператор1, иначе выполняется оператор2.

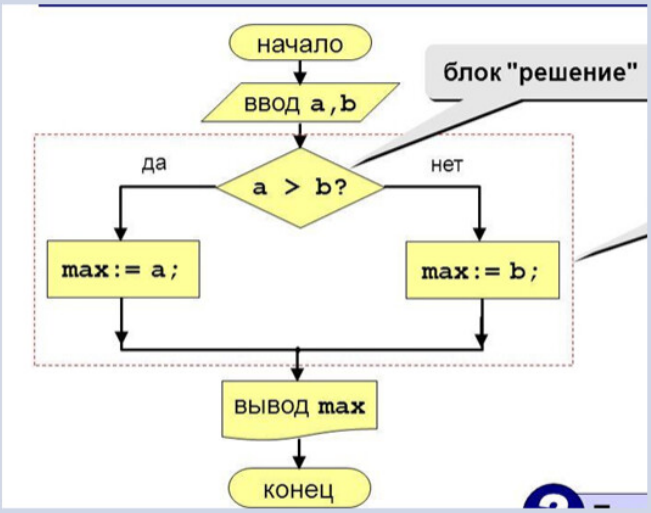
Пример:

If2. Дано целое число N. Если оно положительное, то прибавить к нему 1; если отрицательно вычесть из него 2. Вывести полученное число.

```
program if_2;
 var  n: integer;
 begin
 write ('введите целое число n=');
 readln  (n);
 if n>0
   then n:=n+1
   else n:=n-2;
 writeln ('n=',n);
 end.
```

Задача на Паскаль.
Условные операторы.

Задание №1

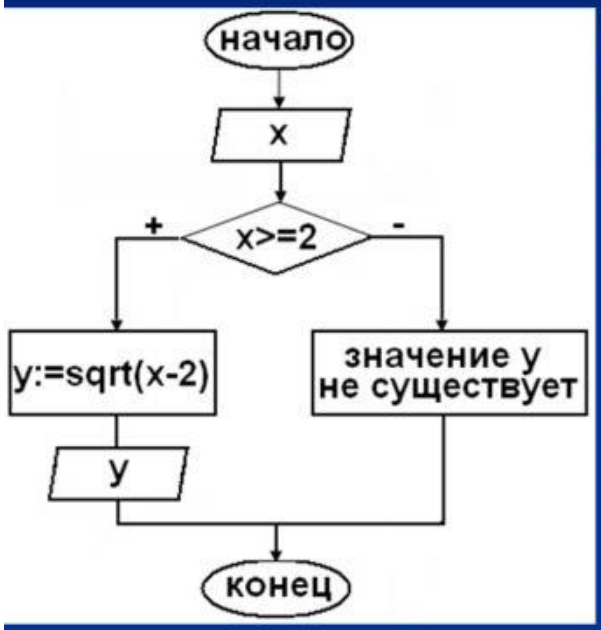
Вывести на экран наибольшее из двух чисел	Даны два числа. Вывести на экран то из них, которое больше
Program MaxOfTWo; Var **a,b:integer;** Begin **readln**(a,b) **if** a>b **then** begin **writeln**(a) end **else** begin **writeln**(b) end End.	

Вложенные условные операторы

Возможно использование в качестве оператор1 или оператор 2 других условных операторов:


```
if <условие1> then
        if <условие2> then <оператор1>
                else <оператор2>;
```

Задача1. Составить программу на языке программирования Turbo Pascal: найти значение функции y= $\sqrt{x-2}$




```
program znach_funk;
uses crt;
var x,y:real;
begin clrscr;
  writeln('vvedite x');
  readln(x);
  if x>=2 then begin
                 y:=sqrt(x-2);
                 writeln('y=',y:3:2)
               end
  else writeln('значение y не
   существует');
  readln;
end.
```

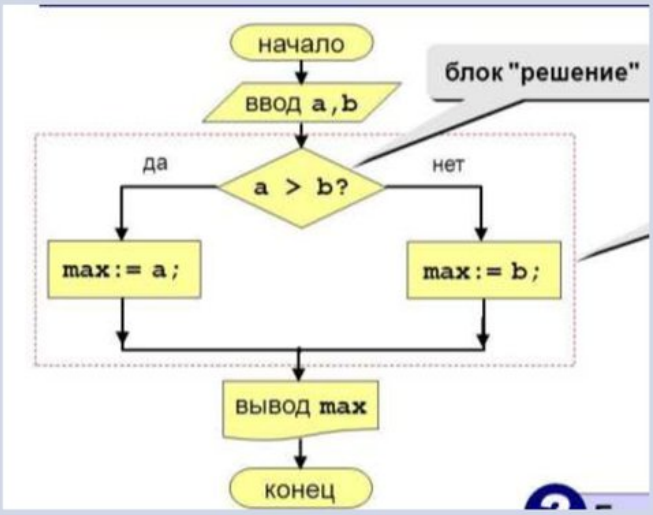
14 из 20

MyShared

Четвертый вопрос: <u>Программирование условного алгоритма (практическое занятие №19, теоретическая часть).</u>

Задача на Паскаль.
Условные операторы.

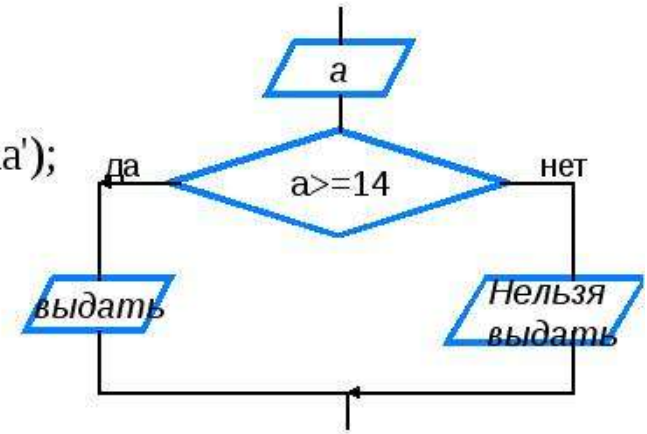
Задание №1

Вывести на экран наибольшее из двух чисел	Даны два числа. Вывести на экран то из них, которое больше
Program MaxOfTWo; Var **a,b:integer;** Begin **readln**(a,b) **if** a>b **then** begin **writeln**(a) end **else** begin **writeln**(b) end End.	

Условный оператор

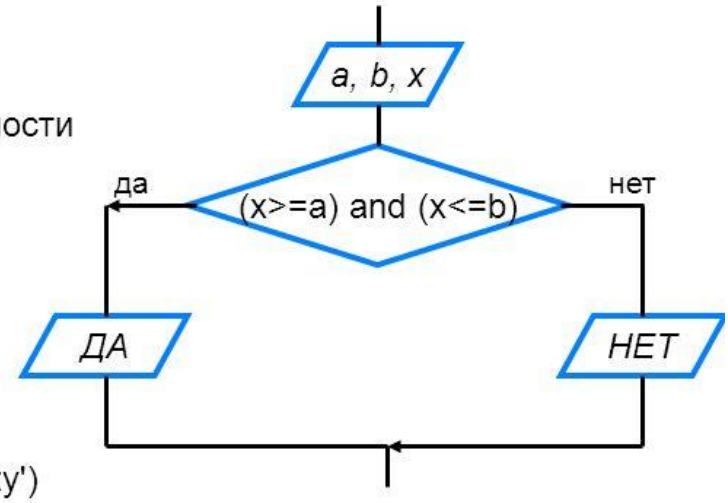
Пример 1: Составить программу для выдачи паспорта ребенку впервые (паспорт выдается при достижении 14 лет).

```
program n_1;
 var a: real;
begin
 writeln ('Введите возраст ребенка');
readln (a);
if (a>=14) then
 writeln ('Выдать паспорт')
 else
writeln ('Паспорт выдавать нельзя')
end.
```



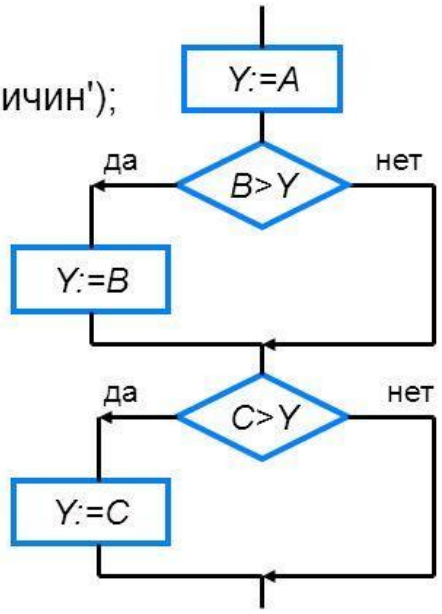
Условный оператор

```
program n_9;
  var x, a, b: real;
begin
  writeln ('Определение принадлежности
                точки отрезку');
  write ('Введите a, b>>');
  readln (a, b);
  write ('Введите x>>');
  readln (x);
  if (x>=a) and (x<=b) then
  writeln ('Точка принадлежит отрезку')
  else writeln ('Точка не принадлежит отрезку')
end.
```



Неполная форма условного оператора

```
program n_10;
  var y, a, b, c: integer;
begin
  writeln ('Нахождение наибольшей из трёх величин');
  write ('Введите a, b, c>>');
  readln (a, b, c);
  y:=a;
  if (b>y) then y:=b;
  if (c>y) then y:=c;
  writeln ('y=', y)
end.
```



Примеры решения задач (записать в тетрадь)

Задача 2. Даны три величины A, B, C. Переменной У присвоить значение большей из них.

Разберемся в условии задачи:

Понятно, что значения нужно сравнить. Предположим, что наибольшим является значение А – сохраним его в переменной У. Затем сравним с У переменную В. Если В больше по значению – сохрани его в У. Аналогично поступим с переменной С. Выводим значение У.

```
program Z2;
  var
     y, a, b, c: integer;
begin
  writeln ('Нахождение max(A, B, C)' );
  write ('Введите a, b, c>>');
  readln (a, b, c);
  y:=a;
  if (b>y)
     then y:=b;
  if (c>y)
     then y:=c;
  writeln ('y=', y);
  readln;
end.
```

Задача решена с использованием двух кратких форм условного оператора