

A blue-tinted photograph of a chalkboard. The board has some faint, light-colored markings, possibly diagrams or text, which are out of focus. In the foreground, a wooden tray holds several pieces of white chalk. The overall scene is dimly lit, with a soft glow from the right side.

**Лекция №4**

# **Разветвленный алгоритм**

# План лекции

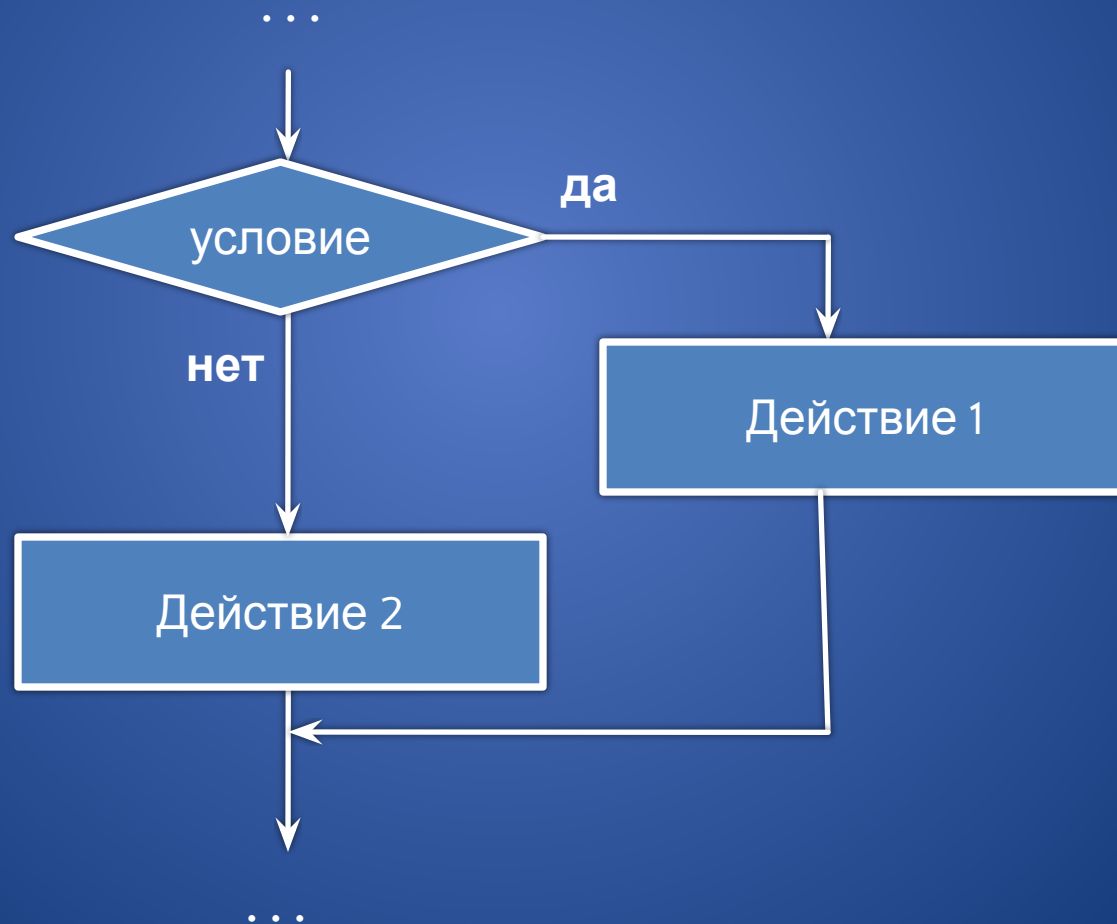
1. Разветвленный алгоритм
2. Примеры разветвленных алгоритмов
3. Условный оператор
4. Составной оператор
5. Вложенные условные операторы
6. Оператор выбора
7. Решение задач

# Разветвленный алгоритм

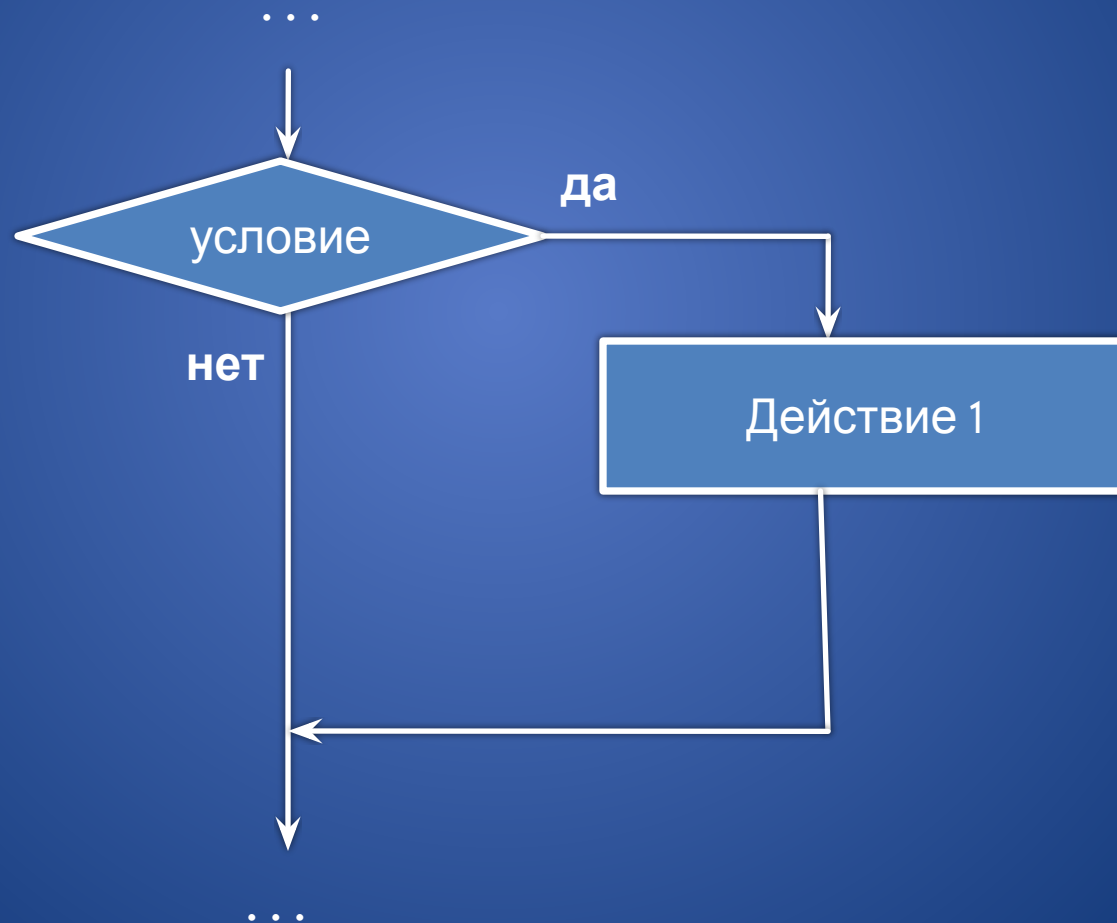
**Разветвленным алгоритмом** называется такой алгоритм, в котором выбирается один из нескольких возможных путей (вариантов) вычислительного процесса.

**Ветвью алгоритма** называется каждый подобный путь.

# Полная разветвленная алгоритмическая структура



# Неполная разветвленная алгоритмическая структура



# Условный оператор

Оператор `if` выбирает между двумя вариантами развития событий:

```
    if <условие>  
(если)  
    then <один_оператор>  
(тогда)  
    [else <один_оператор>];  
(иначе)
```

Обратите внимание, что перед словом `else` (когда оно присутствует, конечно же) символ `;` не ставится – ведь это разорвало бы оператор на две части.



# Условный оператор

Условный оператор `if` работает следующим образом:

- Сначала вычисляется значение `<условия>` – это может быть любое выражение, возвращающее значение типа `boolean`.
- Затем, если в результате получена "истина" (`true`), то выполняется оператор, стоящий после ключевого слова `then`, а если "ложь" (`false`) – без дополнительных проверок выполняется оператор, стоящий после ключевого слова `else`. Если же `else`-ветвь отсутствует, то не выполняется ничего.

# Составной оператор

Составной оператор – объединение нескольких операторов в одну группу. Группа операторов внутри составного оператора заключается в операторные скобки (begin–end).

```
begin  
    оператор 1;  
    оператор 2;  
end;
```



# Составной условный оператор

Если в качестве оператора должна выполняться серия операторов, то они заключаются в операторные скобки `begin-end`. Конструкция `Begin ... End` называется составным оператором.

```
if <логическое выражение>
  then
    begin
      оператор 1;
      оператор 2;
      ...
    end
  else
    begin
      оператор 1;
      оператор 2;
      ...
    end;
end;
```

# Пример разветвленного алгоритма

**Задача 1.** Вычислить отношение двух целых чисел.

Var

a,b : integer;

ratio : real;

Begin

writeln('Введите два числа: ');

read (a,b);

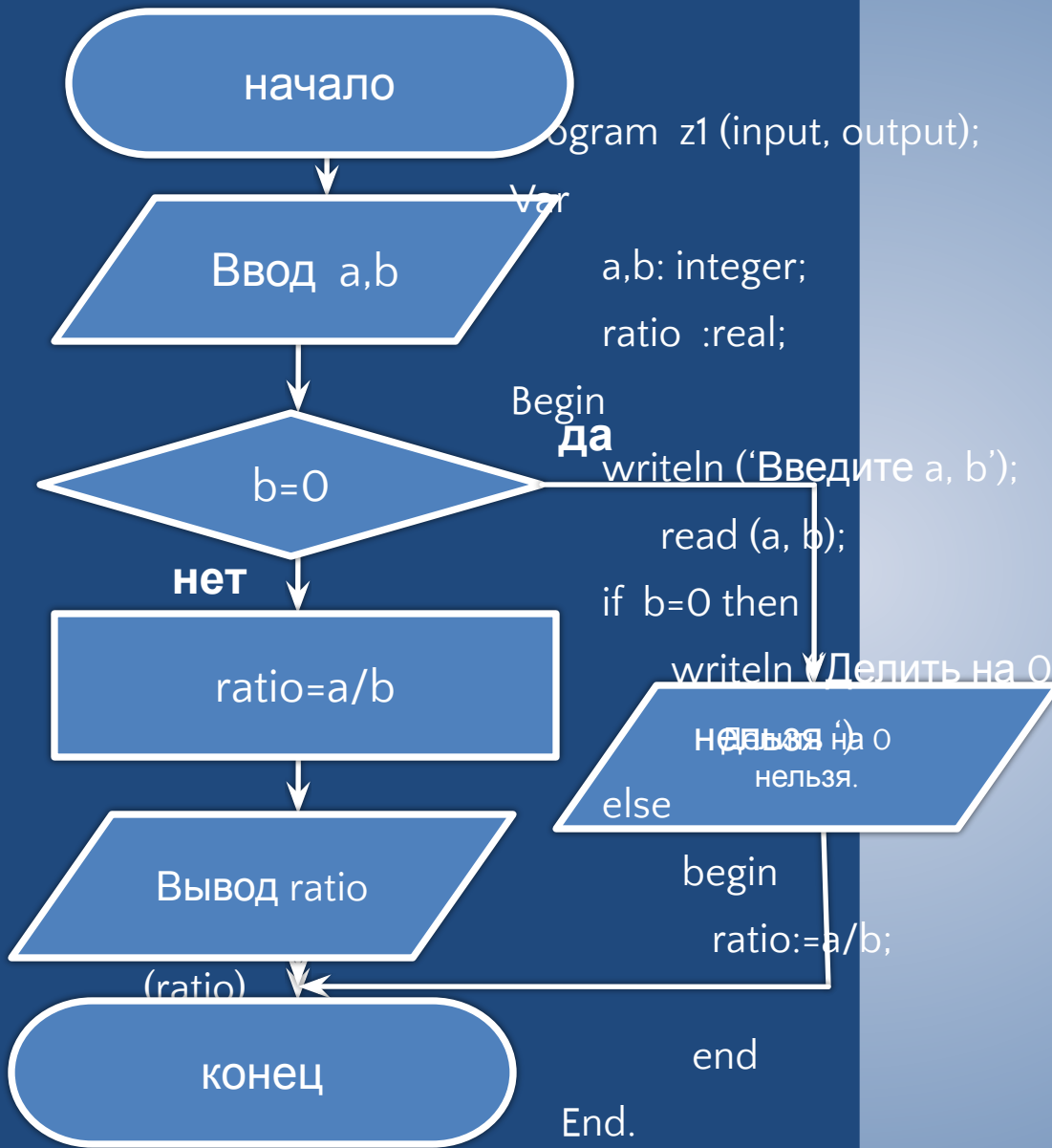
ratio:= a / b;

writeln ('Отношение = ',ratio:6:2);

End.

Когда вы введете 0 для второго значения, программа напечатает сообщение об ошибке

(Divide by zero – Деление на ноль) и остановится.



# Полная разветвленная структура

**Задача 2.** Вывести на экран большее из двух данных чисел.

```
Program z2;  
  Var  
    x, y : integer; {вводимые числа}  
  Begin  
    writeln('Введите 2 числа '); {вводим два целых числа через  
пробел}  
    readln(x,y);  
    if x>y  
    then  
      writeln (x) {если x больше y, то выводим x}  
    else  
      writeln (y) {иначе выводим y}  
    readln;  
  End.
```

# Неполная разветвленная структура

**Задача 3.** Составить программу, которая, если введенное число отрицательное меняет его на противоположное.

```
Program Chisla;  
  Var  
    x : integer; {вводимое число}  
  Begin  
    writeln('Введите число '); {вводим целое число}  
    readln(x);  
    if x<0  
      then  
        x:=-x;  
        writeln (x);  
        readln;  
  End.
```

# Вложенные условные операторы

**Задача 4.** Даны целые числа  $a$ ,  $b$ ,  $c$ . Если  $a \leq b \leq c$ , то все числа заменить их квадратами, если  $a > b > c$ , то каждое число заменить наибольшим из них, в противном случае сменить знак каждого числа.

```
Program z4;
Var
  a, b, c : integer;
Begin
  writeln('Введите числа a, b, c');
  readln(a,b,c);
  if (a<=b) and (b<=c)
  then
    begin
      a:=sqr(a);
      b:=sqr(b);
      c:=sqr(c);
    end
  else
    if (a>b) and (b>c)
    then
      begin
        b:=a;
        c:=a;
      end
    else
      begin
        a:=-a;
        b:=-b;
        c:=-c;
      end
    end
  writeln(a,b,c);
  readln;
End.
```



# Оператор выбора

Оператор `case` позволяет сделать выбор между несколькими вариантами:

```
case <переключатель> of
```

```
    <список_констант> : <один_оператор>;
```

```
    [<список_констант> : <один_оператор>;]
```

```
    [<список_констант> : <один_оператор>;]
```

```
    [else <один_оператор>;]
```

```
end;
```

# Оператор выбора

Существуют дополнительные правила, относящиеся к структуре этого оператора:

- Переключатель должен относиться только к порядковому типу данных, но не к типу `longint`.
- Переключатель может быть переменной или выражением.
- Список констант может задаваться как явным перечислением, так и интервалом или их объединением.
- Повторение констант не допускается.
- Тип переключателя и типы всех констант должны быть совместимыми

# Оператор выбора

Пример оператора выбора:

```
case symbol of
```

```
'a'..'z', 'A'..'Z' : writeln('Это латинская буква');
```

```
'а'..'я', 'А'..'Я' : writeln('Это русская буква');
```

```
'0'..'9' :      writeln('Это цифра');
```

```
' ',#10,#13,#26 : writeln('Это пробельный символ');
```

```
else      writeln('Это служебный символ');
```

```
end;
```

# Оператор выбора

**Задача 5.** Написать программу преобразования цифр в слова.

```
Program z5;
  Var
    a, b, c : integer;
  Begin
    writeln('Введите цифру ');
    readln(a);
    case a of
      0 : writeln ('ноль');
      1 : writeln ('один');
      2 : writeln ('два');
      3 : writeln ('три');
      4 : writeln ('четыре');
      5 : writeln ('пять');
      6 : writeln ('шесть');
      7 : writeln ('семь');
      8 : writeln ('восемь');
      9 : writeln ('девять')
    else writeln ('Это число не является цифрой');
    end;
    readln;
  End.
```

# Решение задач

**Задача 6.** Чтобы получить одну молекулу серной кислоты  $H_2SO_4$  нужно 2 атома водорода (H), 1 атом серы (S) и 4 атома кислорода (O). В химическом реакторе имеется A атомов водорода, B атомов кислорода и C атомов серы. Составьте алгоритм или напишите программу, вычисляющую, сколько молекул серной кислоты может получиться из такого количества атомов?



# Решение задач

Алгоритм решения этой задачи может быть записан следующим образом:

Переменные: тип – целый A, B, C, M1, M2, K;

Ввести A, D, C;

Вычислить целую часть от деления A на 2

$M1 = \text{цел. часть } (A/2);$

Вычислить целую часть от деления B на 4

$M2 = \text{цел. часть } (B/4);$

Найти минимальное из трех чисел C, M1, M2

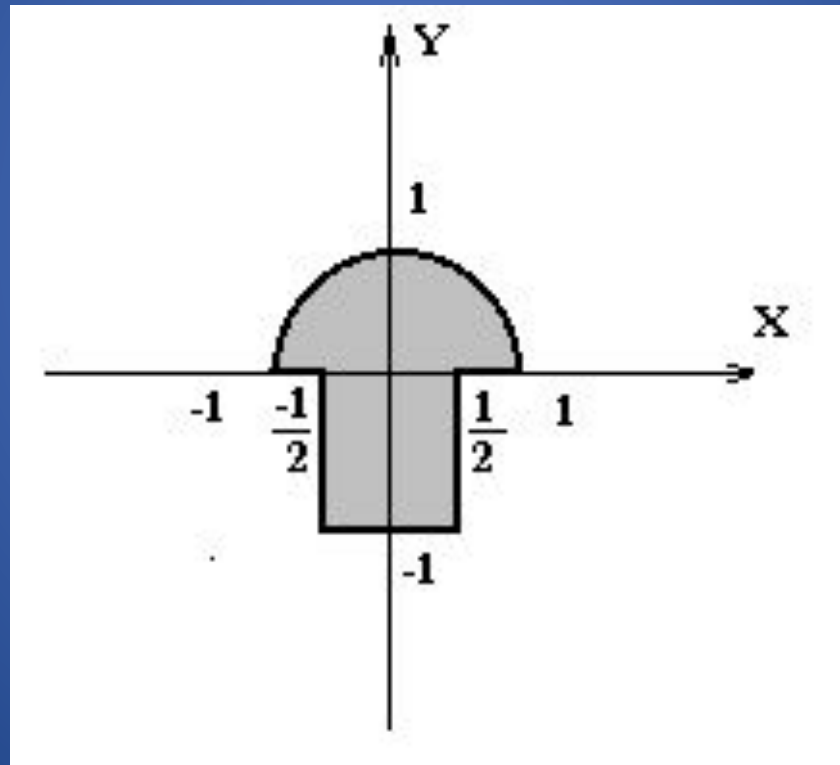
$K = \min(C, M1, M2);$

Выдать на печать: В редакторе образуется K молекул кислоты.



# Решение задач

Задача 7. Определить принадлежит ли точка  $A(X_0, Y_0)$  данной фигуре.



# Решение задач

**Задача 8.** Напишите программу, которая вычисляет стоимость междугородного телефонного разговора (цена одной минуты определяется расстоянием до города, в котором находится абонент). Исходными данными для программы являются код города и длительность разговора. Ниже приведены коды некоторых городов.

Город	Код	Цена минуты(руб.)
Владивосток	423	2,2
Москва	095	1,0
Мурманск	815	1,2
Самара	846	1,4

