

**Тема занятия:  
Разветвляющийся  
алгоритм**

# Содержание:

- Определение
- Виды разветвляющихся алгоритмов
- Организация разветвляющихся алгоритмов в Turbo Pascal
- Решение задач с использованием разветвляющихся алгоритмов

# Разветвляющийся алгоритм

- это алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов

Например,

- Переход улицы по светофору (*Если свет зеленый, то переходи, иначе стой*)
- Определение числа корней в квадратном уравнении (*Если дискриминант отрицательный, то решений нет; если дискриминант равен нулю, то уравнение имеет один корень; если дискриминант положительный, то уравнение имеет два решения*)

# Виды развилок

## полная

Действия определены **и** в случае истинности условия, **и** в случае ложности

Например, алгоритм «Покупка»:  
Если товар в наличии,  
то купи его,  
иначе купи другой товар  
Конец ветвления.

Алгоритм «*Вычисление значения модуля числа*»: Если число неотрицательное, то запиши число, иначе запиши число с противоположным знаком  
Конец ветвления.

## неполная

Действия определены **либо** в случае истинности условия, **либо** в случае ложности

Например, алгоритм «*Покупка*»:  
Если товар в наличии,  
то купи его  
Конец ветвления.

Алгоритм «*Вычисление арифметического корня*»:  
Если число неотрицательное,  
то извлеки из него корень  
Конец ветвления.

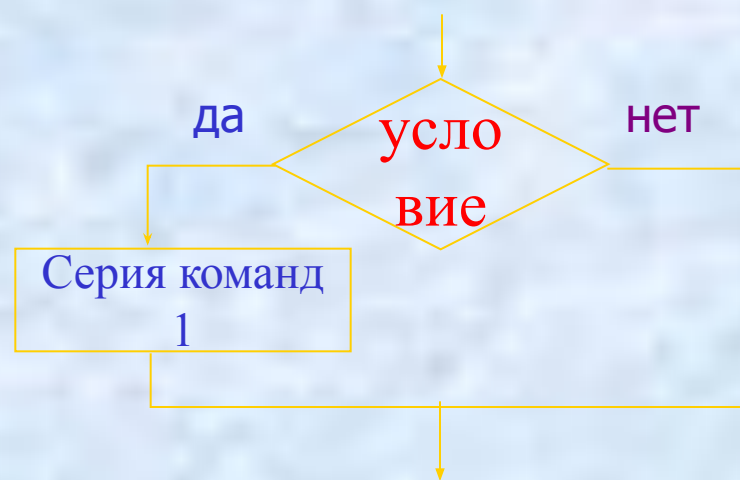
# Виды развилок

ПОЛНАЯ



Если **условие истинно**,  
то выполняется серия команд 1,  
иначе выполняется серия команд  
Конец ветвления.

НЕПОЛНАЯ



Если **условие истинно**,  
то выполняется серия команд 1  
Конец ветвления.

# Полная развилка

1. Проверить условие
2. Если условие истинно, то выполнить серию команд по ветви «ДА», если ложно - выполнить серию команд по ветви «НЕТ»

# Неполная развилка

1. Проверить условие
2. Если условие истинно, то выполнить серию команд по ветви «ДА»

# Примеры развилок

полная

да

Число  
неотрицательное

нет

Запиши число

Запиши число с  
противоположным  
знаком

неполная

да

Число  
неотрицательное

нет

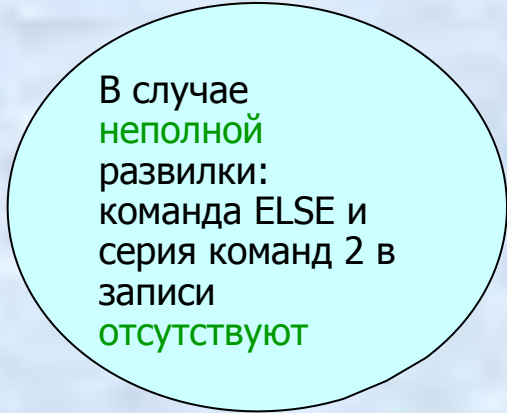
Извлеки  
корень  
из числа



# Организация разветвляющихся алгоритмов в Pascal

Для организации разветвленных алгоритмов в Pascal предусмотрена структура:

```
IF условие  
THEN серия команд 1  
ELSE серия команд 2;  
END;
```



В случае  
неполной  
развилки:  
команда ELSE и  
серия команд 2 в  
записи  
отсутствуют



# Организация разветвляющихся алгоритмов в Pascal

Например:

```
IF X >= 0  
    THEN Y := X  
    ELSE Y := -X;  
END.
```

```
IF X >= 0  
    THEN Y := SQRT(X);  
END.
```

# Вопросы для проверки

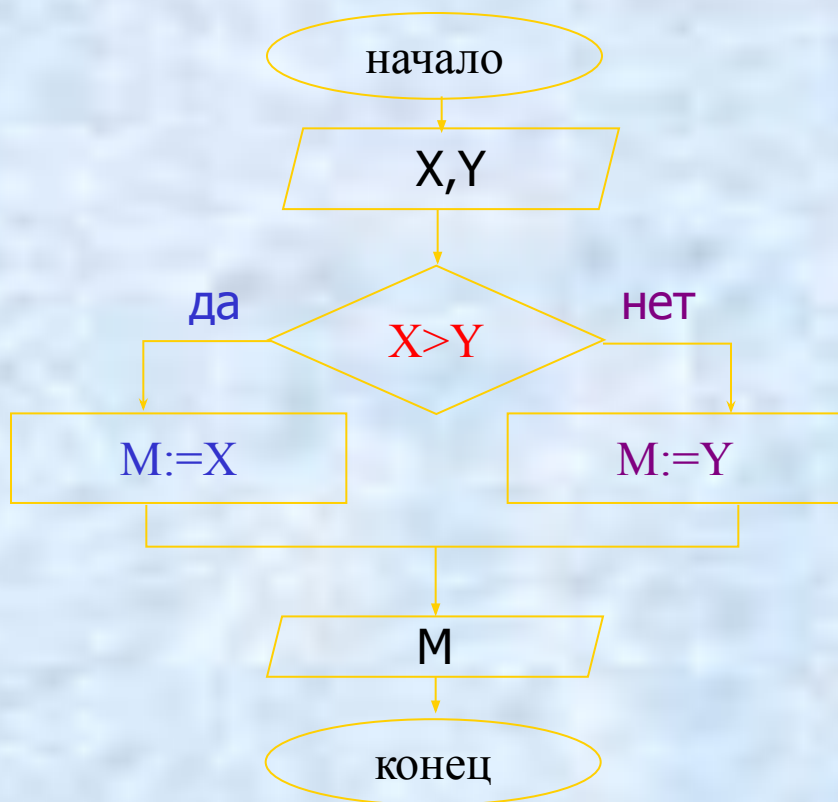
- Какие алгоритмы называются разветвленными?
- Приведите 3-4 примера разветвленных алгоритмов.
- Какие виды развилок бывают?
- Приведите примеры алгоритмов полной и неполной разветвленной структуры.
- Какого типа структура описана в пословице «Нет друга – ищи, а нашел – береги»?

# Вопросы для проверки

- Какие команды служат для записи разветвленных алгоритмов в Pascal?
- Какие формы записи предусмотрены для записи разветвлений?
- Когда удобно использовать строчную или блочную форму записи разветвленных алгоритмов?
- Чем отличается запись полной развилки от неполной?
- Добавьте недостающие слова  
*«Если условие истинно, то выполняются действия, записанные после команды ..., в противном случае выполняются действия, записанные после команды ...»*

# Решение задач

Даны два числа  $X$  и  $Y$ .  
Вывести на экран наибольшее из заданных чисел.

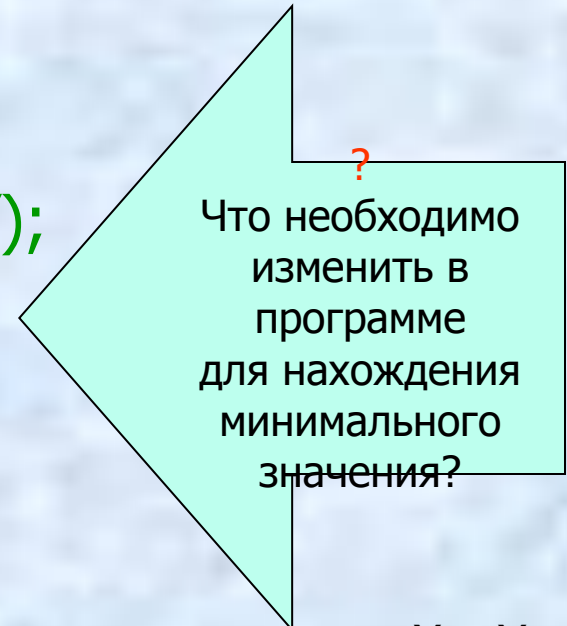


шаг	X	Y	M	условие	комментарий
1	4	2			ВВОД
2				4 > 2 да	
3			4		
4			4		ВЫВОД
1	3	5			ВВОД
2				3 > 5 нет	
3		5			
4		5			ВЫВОД

# Решение задач

```
PROGRAM maximum;  
VAR x, y, m: integer;  
BEGIN  
  WRITELN('введите числа x и y');  
  READLN (x,y);  
  IF x>y  
    THEN m:=x  
    ELSE m:=y;  
  WRITELN ('наибольшее',m);  
END
```

1. В данном примере задаются два числа – переменные X и Y
2. Проверяется условие  $X > Y$
3. Если условие окажется истинным (т.е. число X больше числа Y), то переменной M будет присвоено значение переменной X, иначе (т.е. число X не больше числа Y) - переменной M будет присвоено значение переменной Y
4. Значение переменной M выводится на экран



# Задания для проверки

По заданным фрагментам программ определите значение переменной M

- 1) A:=2;  
IF A>0 THEN M:=1 ELSE M:=0;
- 2) B:=0;  
IF B=0 THEN M:=0 ELSE M:=1;
- 3) C:=SQRT(16);  
IF C<8 THEN M:=1 ELSE M:=-1;
- 4) S:=-1;  
IF S>=0 THEN M:=SQRT(S) ELSE M:=S\*2;
- 5) R:=-5;  
IF R=ABS(R) THEN M:=R ELSE M:=0;

1) Ответ: 1. Так как условие «A>0» для A, равного 2, истинно, то выполняется действие, записанное после команды THEN: «M:=1»

2)

3)

4)

5)

# Задания для проверки

*Запишите условие на языке Pascal*

1. Число А положительное
2. Число В неотрицательное
3. Число Н принадлежит [5;15]
4. Число М двузначное
5. Число С кратно 5
6. Число Р четное

*Ответ*

1.  $A > 0$
- 2.
3.  $(N \geq 5) \text{ and } (N \leq 15)$
- 4.
5.  $(C \bmod 5) = 0$
- 6.

**ВНИМАНИЕ!**

Для записи сложных условий используется логическая операция

**and**



Спасибо за внимание!!!!