

Программирование
разветвляющихся
алгоритмов.

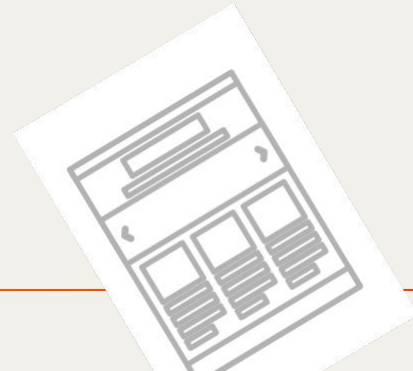
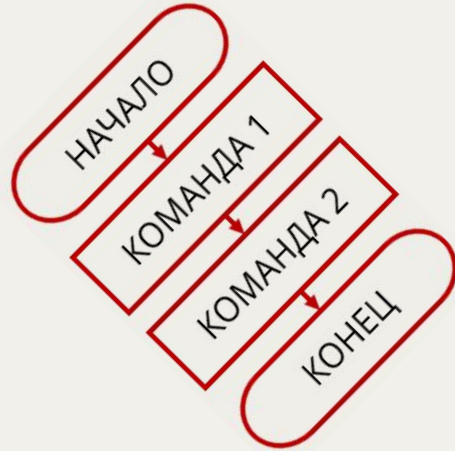
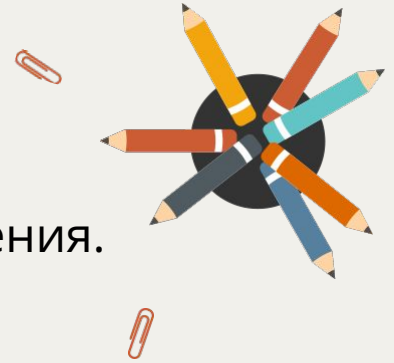
Простой и составной
условные операторы

Начала программирования



Линейные алгоритмы

Линейным называется алгоритм, в котором используется всего одна конструкция – следование. Он состоит из операторов, записанных последовательно в порядке их исполнения.





Типы данных в языке Pascal

Типы данных

Числовые:

- ✓ byte;
- ✓ integer;
- ✓ real.

Символьный:

- ✓ char.

Строковый:

- ✓ string.

Логический:

- ✓ boolean.



Программирование разветвляющихся алгоритмов

1

Определение
разветвляющегося
алгоритма.

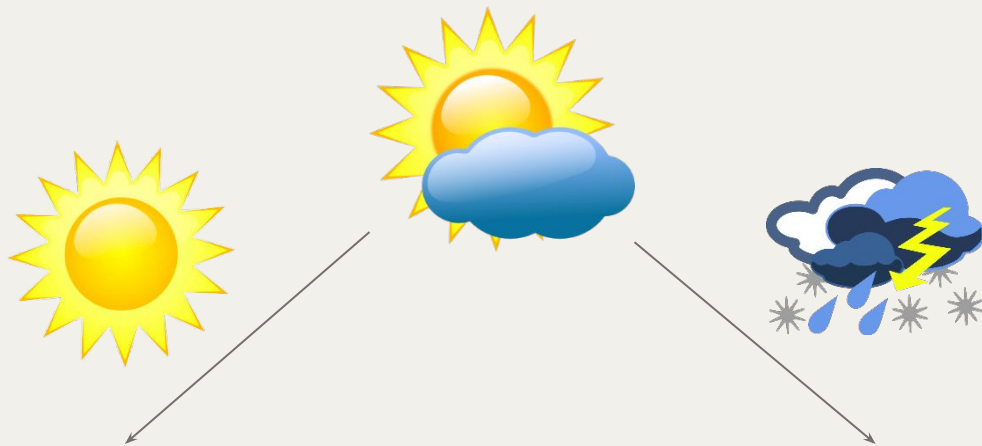
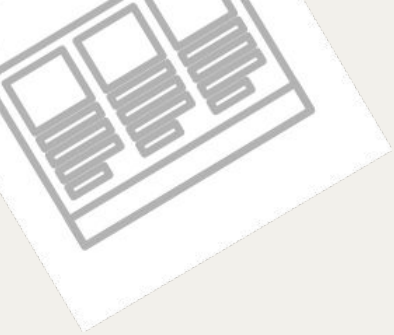
2

Запись
разветвляющегося
алгоритма в языке
Pascal.

3

Простой и составной
условные
операторы.

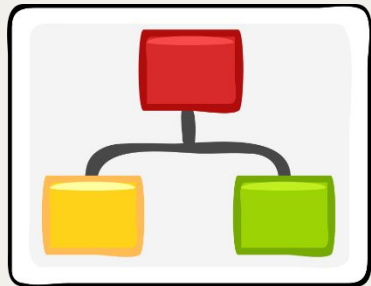
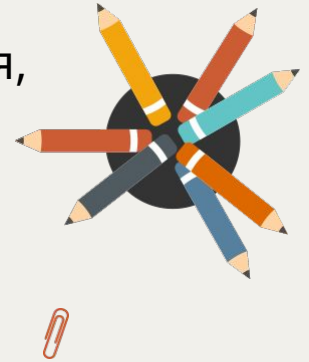
Ветвление



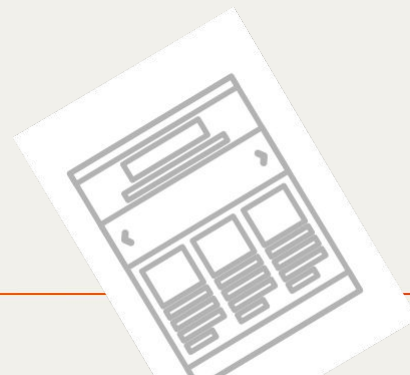
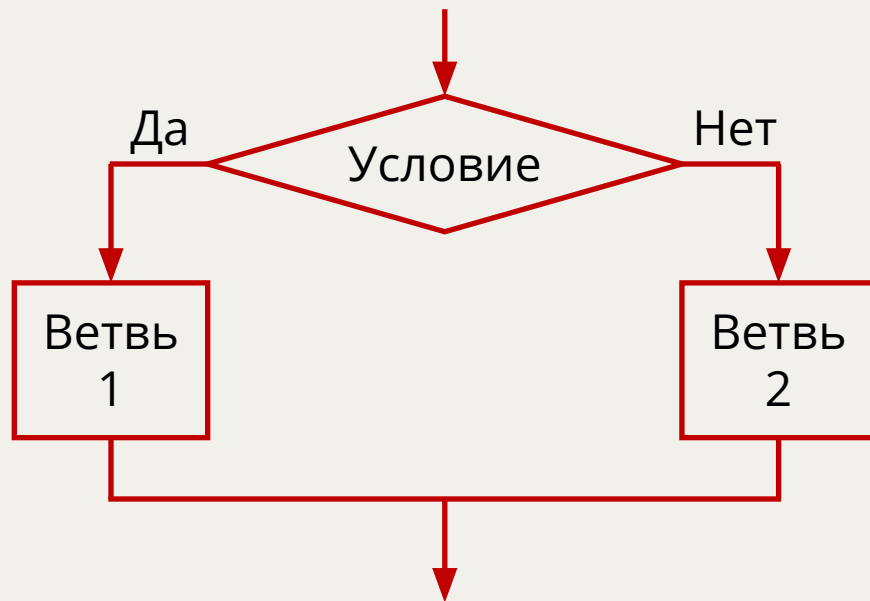
Разветвляющиеся алгоритмы

В **разветвляющихся алгоритмах**, помимо следования, используется **конструкция ветвления**.

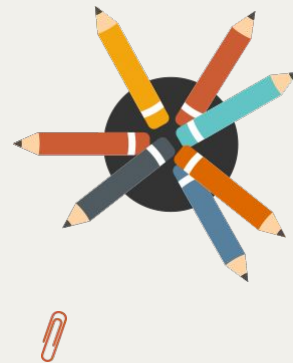
Ветвление – это алгоритмическая конструкция, в которой в зависимости от некоторого условия происходит исполнение одной из двух последовательностей команд (**ветвей**).



Блок-схема разветвляющегося алгоритма



Запись разветвляющегося алгоритма в языке Pascal



Формы записи
условного
оператора

Сокращённая:

```
if <условие>  
then <оператор 1>;
```

Полная:

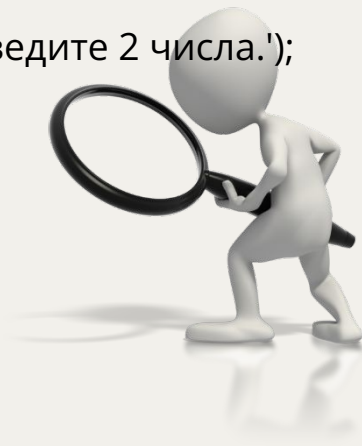
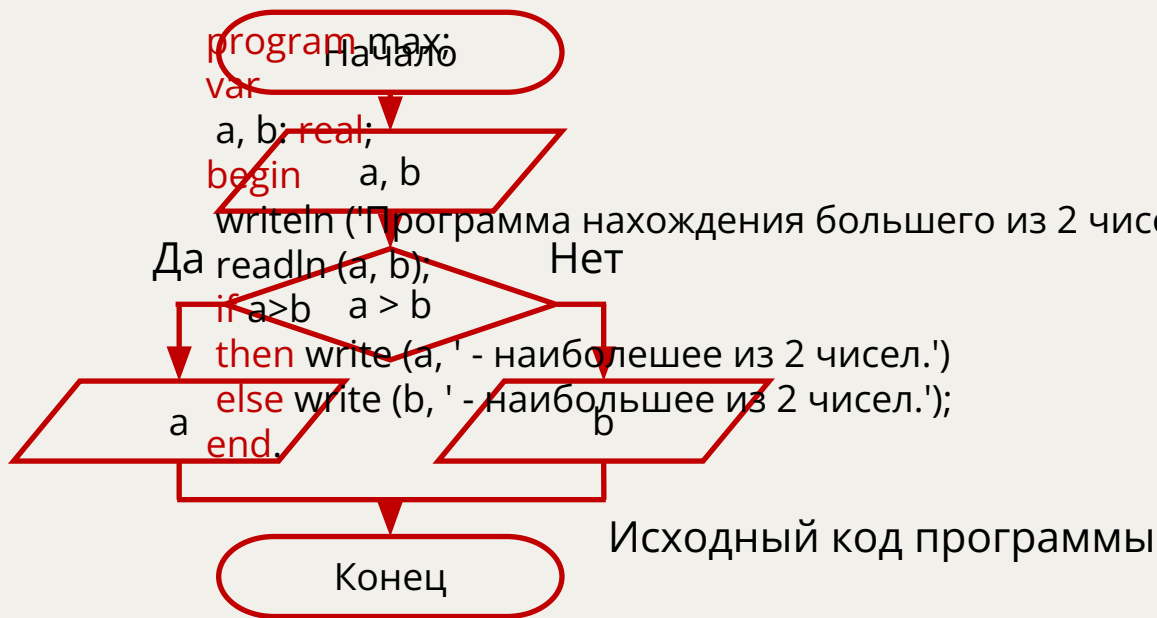
```
if <условие>  
then <оператор 1>  
else <оператор 2>;
```


Задача

Найти наибольшее из двух чисел, введённых пользователем.
Если числа равны между собой, вывести любое из них.

Обозначим:

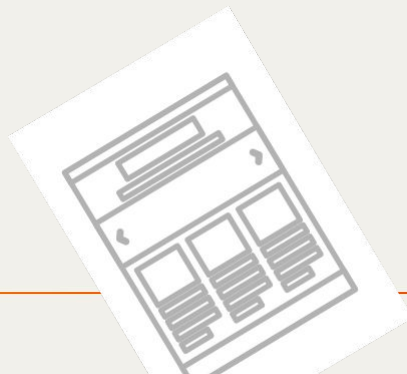
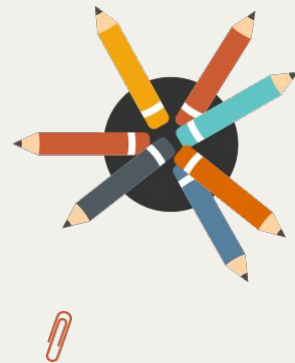
a и **b** – числа введённые пользователем.



Составной условный оператор

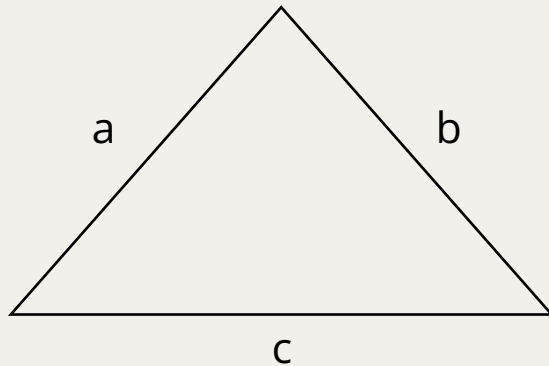
```
if <условие>  
then begin  
    <оператор 1>  
    <оператор 2>  
end  
else begin  
    <оператор 3>  
    <оператор 4>  
end;
```

Вложенный
условный
оператор



Задача

Три отрезка заданы своими длинами.
Определить, образуют ли эти отрезки треугольник,
и если образуют, то какой:
остроугольный, прямоугольный или тупоугольный.

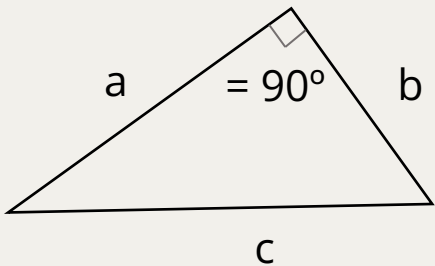


$$c < a + b$$

Задача

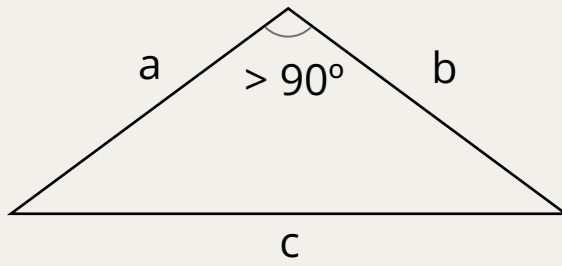
Три отрезка заданы своими длинами.
Определить, образуют ли эти отрезки треугольник,
и если образуют, то какой:
остроугольный, прямоугольный или тупоугольный.

**Прямоугольный
треугольник**



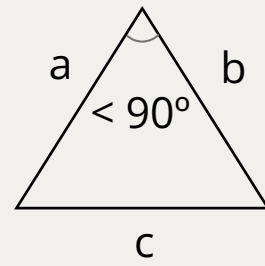
$$c^2 = a^2 + b^2$$

**Тупоугольный
треугольник**



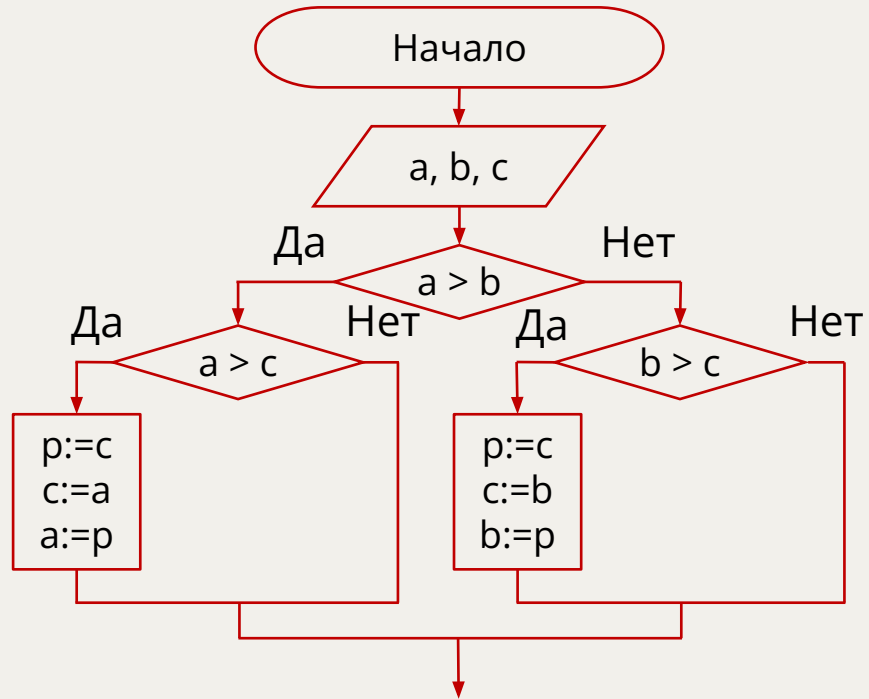
$$c^2 > a^2 + b^2$$

**Остроугольный
треугольник**

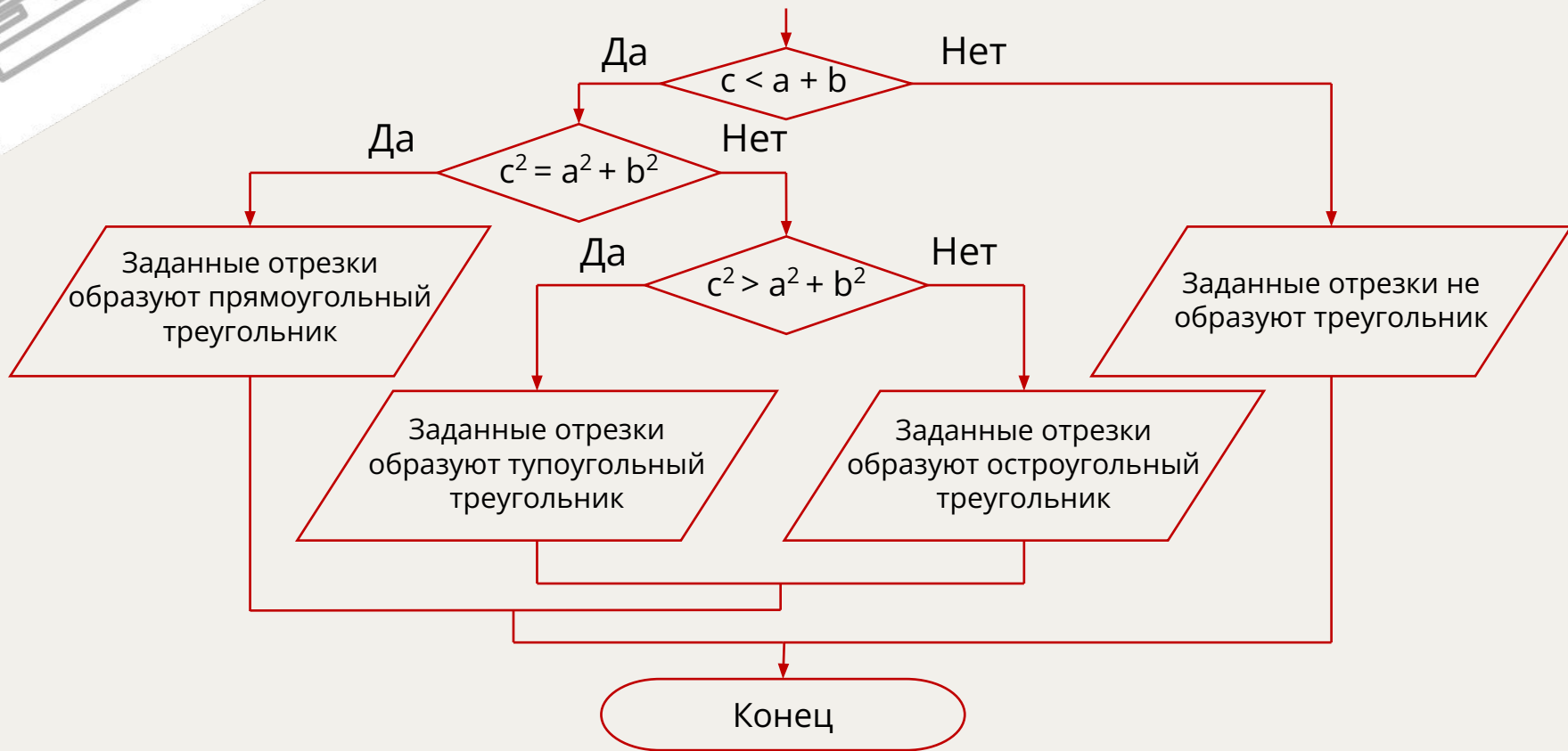


$$c^2 < a^2 + b^2$$

Блок-схема алгоритма:



Блок-схема алгоритма:



Написание программы

```
program treugolnik;
var
  a, b, c, p: real;
begin
  writeln ('Программа проверки того, образуют
ли 3 заданных отрезка треугольник. Введите
длины отрезков.');
```

writeln ('Программа проверки того, образуют ли 3 заданных отрезка треугольник. Введите длины отрезков.');

```
  readln (a, b, c);
  if a>b
  then if a>c
  then begin
    p:=c;
    c:=a;
    a:=p;
  end
  else if b>c
  then begin
```

```
    p:=c;
    c:=b;
    b:=p;
  end;
  if c<a+b
  then if sqr(c)=sqr(a)+sqr(b)
  then write ('Заданные отрезки образуют
прямоугольный треугольник.')
```

if c<a+b then if sqr(c)=sqr(a)+sqr(b) then write ('Заданные отрезки образуют прямоугольный треугольник.')

```
  else if sqr(c)>sqr(a)+sqr(b)
  then write ('Заданные отрезки образуют
тупоугольный треугольник.')
```

else if sqr(c)>sqr(a)+sqr(b) then write ('Заданные отрезки образуют тупоугольный треугольник.')

```
  else write ('Заданные отрезки образуют
остроугольный треугольник.')
```

else write ('Заданные отрезки не образуют треугольник.');

```
  end.
```

Исходный код программы

Программирование разветвляющихся алгоритмов.

Простой и составной условные операторы

В **разветвляющемся** алгоритме используются ветвления. **Ветвление** – это алгоритмическая конструкция, в которой при определённом условии выполняется одна из двух последовательностей действий или ветвей.

Запись условного оператора:

```
if <условие>  
then <оператор 1>  
else <оператор 2>;
```

Краткая форма записи условного оператора:

```
if <условие>  
then <оператор 1>;
```


Программирование разветвляющихся алгоритмов.

Простой и составной условные операторы

Составной условный оператор:

```
if <условие>  
then begin  
  <оператор 1>;  
  <оператор 2>;  
  ...  
end  
else begin  
  <оператор 3>;  
  <оператор 4>;  
  ...  
end;
```