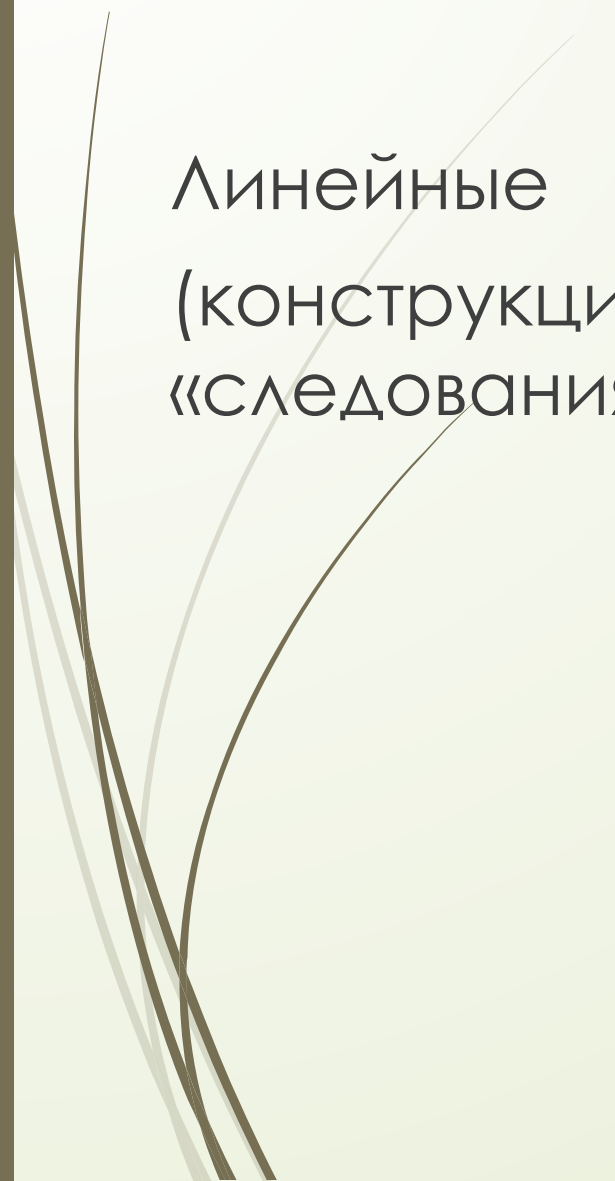


Какие бывают виды алгоритмов?



Линейные
(конструкция
«следования»)

Разветвляющиеся
(конструкция
«ветвление»)

Циклические
(конструкция
«повторение»)

Какой вид алгоритма использован в данной задаче

- Если друг на день рождения
 - Пригласил тебя к себе,
 - То оставь подарок дома —
 - Пригодится самому...
- Алгоритм с ветвлением
или
разветвляющийся алгоритм
 - А как вы поняли что это конструкция «ветвления»?
 - если ... то ...

Повторение

Собери структуру программы

- Структура программы
- **program** <имя программы>;
- **var** <описание используемых переменных>;
- **begin** <начало программного блока>
 <оператор >;
 if <условие> then <оператор1> else <оператор2>;
 <оператор >
- **end.**

□ *Полная форма*

□ *Сокращённая форма*



Составной оператор. Вложенные ветвления

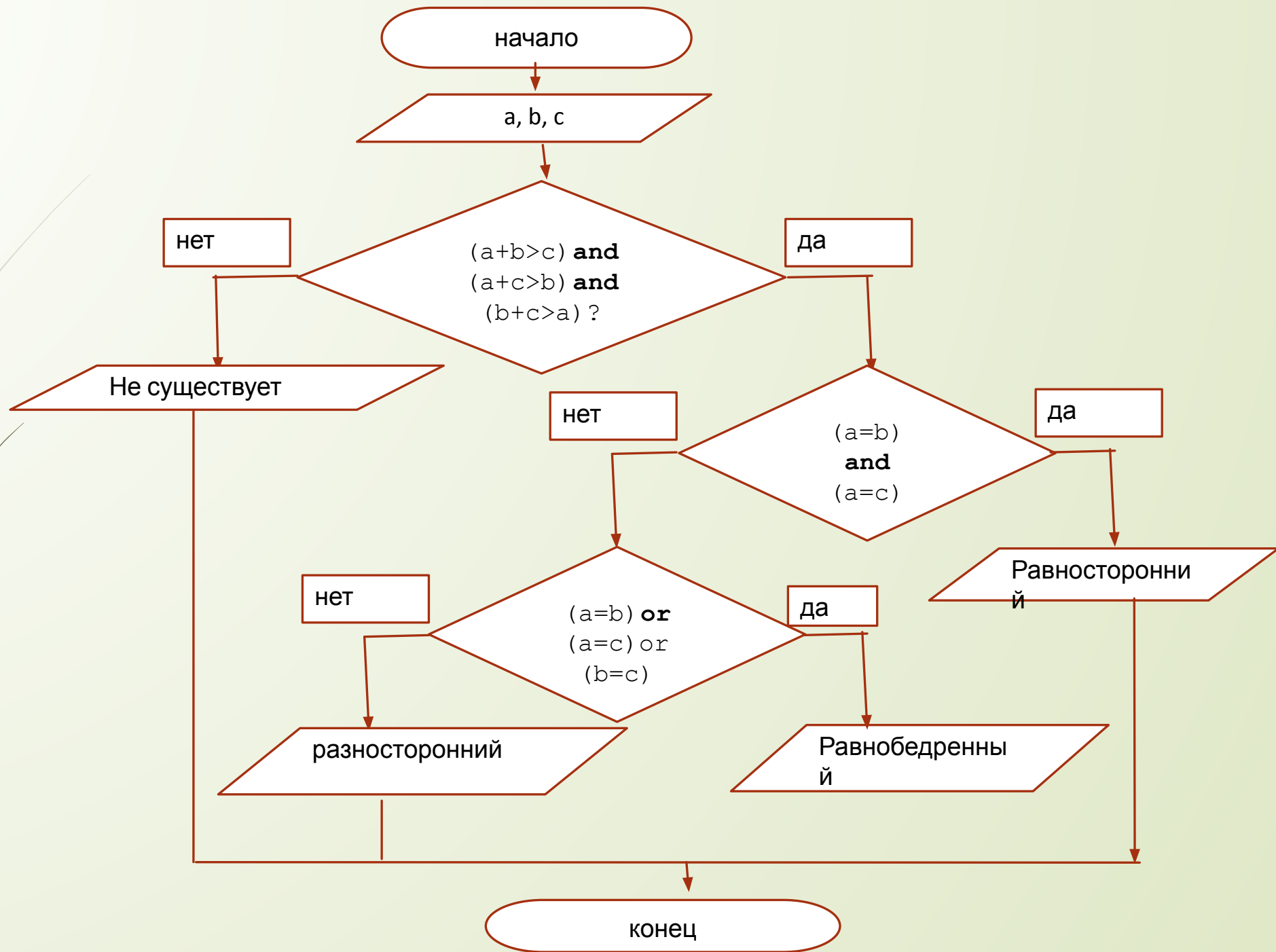
Составной оператор

- В условном операторе и **после** *then*, и **после** *else* можно использовать **только один оператор**.
- Если в условном операторе после *then* или после *else* нужно выполнить **несколько операторов**, то используют составной оператор – конструкцию вида:
begin <последовательность операторов> **end**



Вложенные ветвления

- **if** <условие1> **then**
 - if** <условие2>
 - then** <оператор1>
 - else** <оператор2>
 - else** <оператор3>;
- **else** всегда относится к ближайшему оператору **if**



Решение

```
□ program treugolnik;  
□ var  
□ a, b, c: integer;  
□ begin  
□ readln(a, b, c);  
□ if(a + b > c) and (a + c > b) and (b + c > a) then  
□ begin  
□ if (a = b) and (a = c) then writeln('равносторонний')  
□ else if (a = b) or (a = c) or (b = c) then writeln('равнобедренный')  
□ else writeln('разносторонний');  
□ end  
□ else writeln('не существует')  
□ end.
```




Итоги урока

- Назовите цель урока?
- Достигли ли вы цели урока?
- Что осталось для вас не ясным, не понятным?
- Зачем нужен оператор условия `if ... then ... else...`?
- Зачем нужен составной оператор?