


Графический исполнитель.



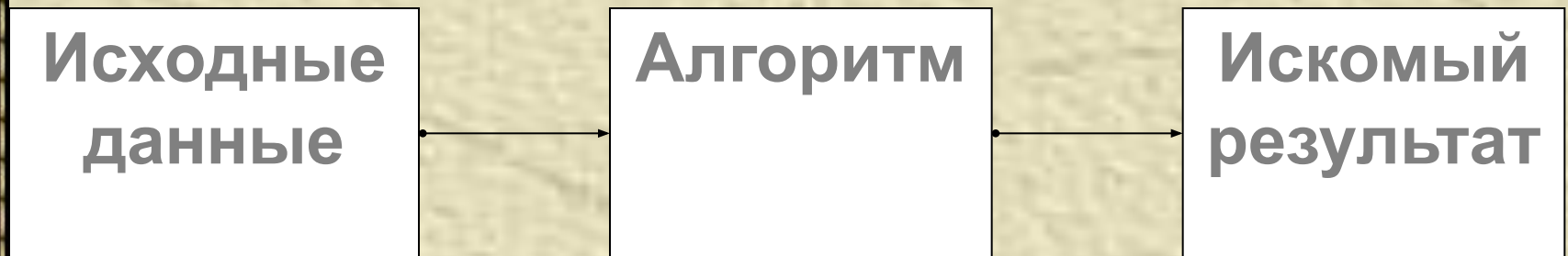


1. Алгоритм - это понятное и точное указание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

2. Разновидности алгоритмов:

- а) бытовые;
- б) вычислительные;
- в) информационные;
- г) управляющие.

3. Исполнитель – это объект, который будет исполнять алгоритм. Любой исполнитель обладает системой команд (СКИ).



4. Свойства алгоритма.

1) Дискретность – разбиение алгоритма на отдельные шаги (команды).

2) Понятность - каждая команда должна быть понятна исполнителю.

Свойства алгоритма.

3) Определенность
(детерминированность) – команды д.б. предельно четкими и однозначными.

4) Результативность – обязательное получение результата, т.е. правильно организованный алгоритм не может обрываться безрезультатно. Каждый алгоритм должен завершиться за конечное число шагов.

Свойства алгоритма.

5) **Корректность** – уверенность в том, что результат решения будет правильным для любых допустимых исходных данных.

6) **Массовость** – применение алгоритма многократно для различных наборов исходных данных.

Нахождение гипотенузы прямоугольного треугольника.

алг. Гипотенуза

нач

1. Возвести a в квадрат.
2. Возвести b в квадрат.
3. Сложить результаты действий 1 и 2.
4. Вычислить квадратный корень из результата 3 действия и принять его за значение c .

кон

Графический исполнитель Стрелочка

1. Среда исполнителя – это обстановка, в

которой **Строка меню** ИСПОЛНИТЕЛЬ.

The screenshot shows the graphical environment for the 'Стрелочка' (Arrow) executor. The interface includes a title bar with the program name and user information, a toolbar with execution and debugging controls, and a main workspace divided into three sections:

- Left Panel (System Commands):** A list of commands with their corresponding function keys: ШАГ (F2), ПРЫЖОК (F3), ПОВОРОТ (F4), ОПИСАНИЕ (F5), ВЫЗОВ (F6), ВЕТВЛЕНИЕ (F7), and ЦИКЛ (F8).
- Code Editor:** Contains the following text:

```
АЛГОРИТМ ПУТЬ_0
Дано: Исполнитель в т.А
Надо: Воспроизвести образец
НАЧ
КОН
```
- Grid:** A 12x12 grid with columns numbered 0-11 and rows numbered 0-11. A red arrow labeled 'А' is positioned at the top-left corner (0,0).

At the bottom left, a summary box shows the algorithm statistics:

Всего в алгоритме:	
команд0
процедур0
вызовов процедур0
ветвлений0
циклов0
Итого: строк5

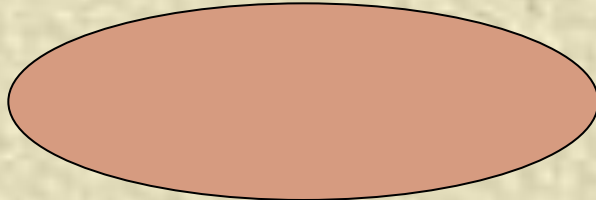
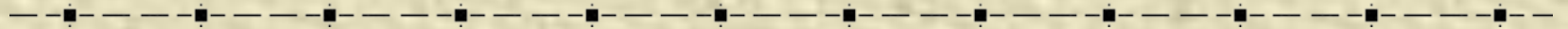
Пол

Графический исполнитель Стрелочка

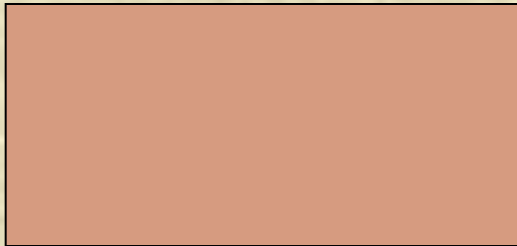
3. Система команд исполнителя (СКИ):

- 1) Шаг - перемещение Стрелочки вперед с рисованием линии;
- 2) Поворот – поворот Стрелочки на 90 градусов против часовой стрелки;
- 3) Прыжок – перемещение Стрелочки на один шаг вперед без рисования линии.

Блок-схемы алгоритмов.



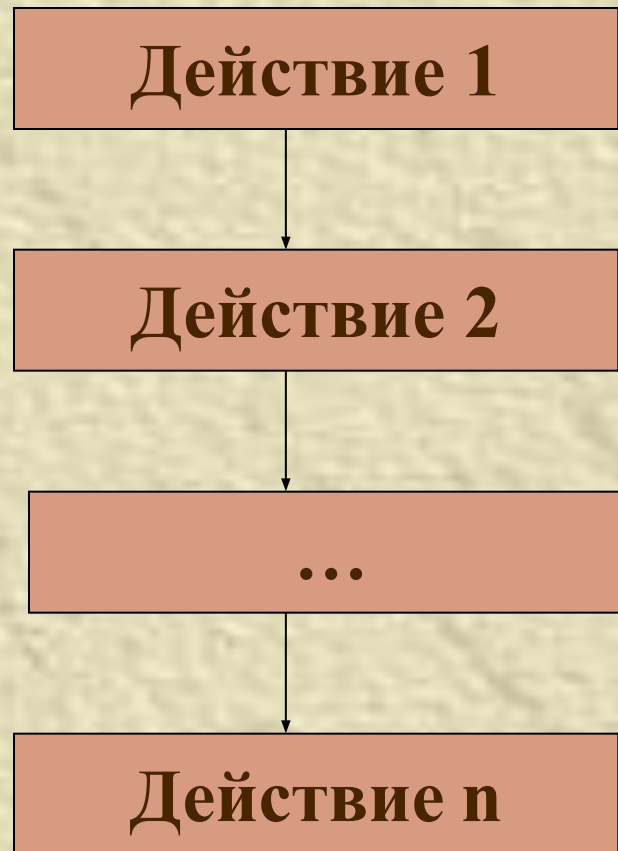
Начало или конец
алгоритма



Простая команда

Линейная алгоритмическая структура.

Базовая структура
«СЛЕДОВАНИЕ»
образуется
последовательностью
действий, следующих
одно за другим



Пример

алг Рисование квадрата

нач

1. шаг

2. поворот

3. шаг

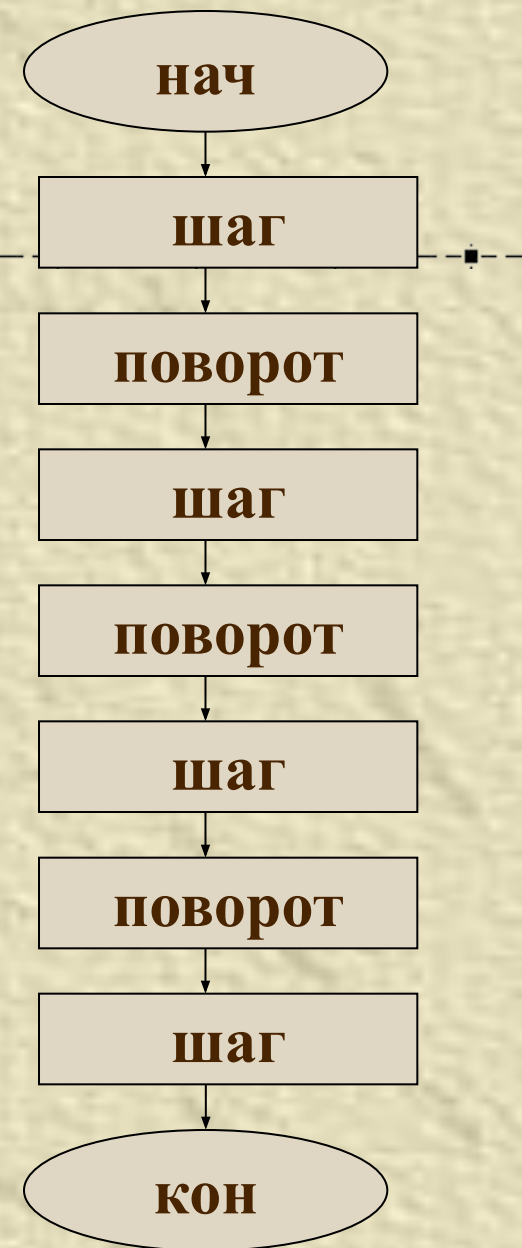
4. поворот

5. шаг

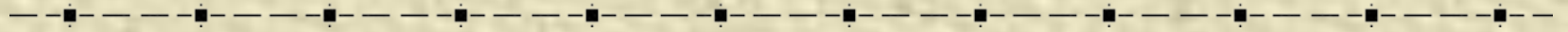
6. поворот

7. шаг

кон



Циклические алгоритмы



Циклическим называется алгоритм, в котором результат достигается после многократного исполнения одних и тех же действий.

Циклические алгоритмы

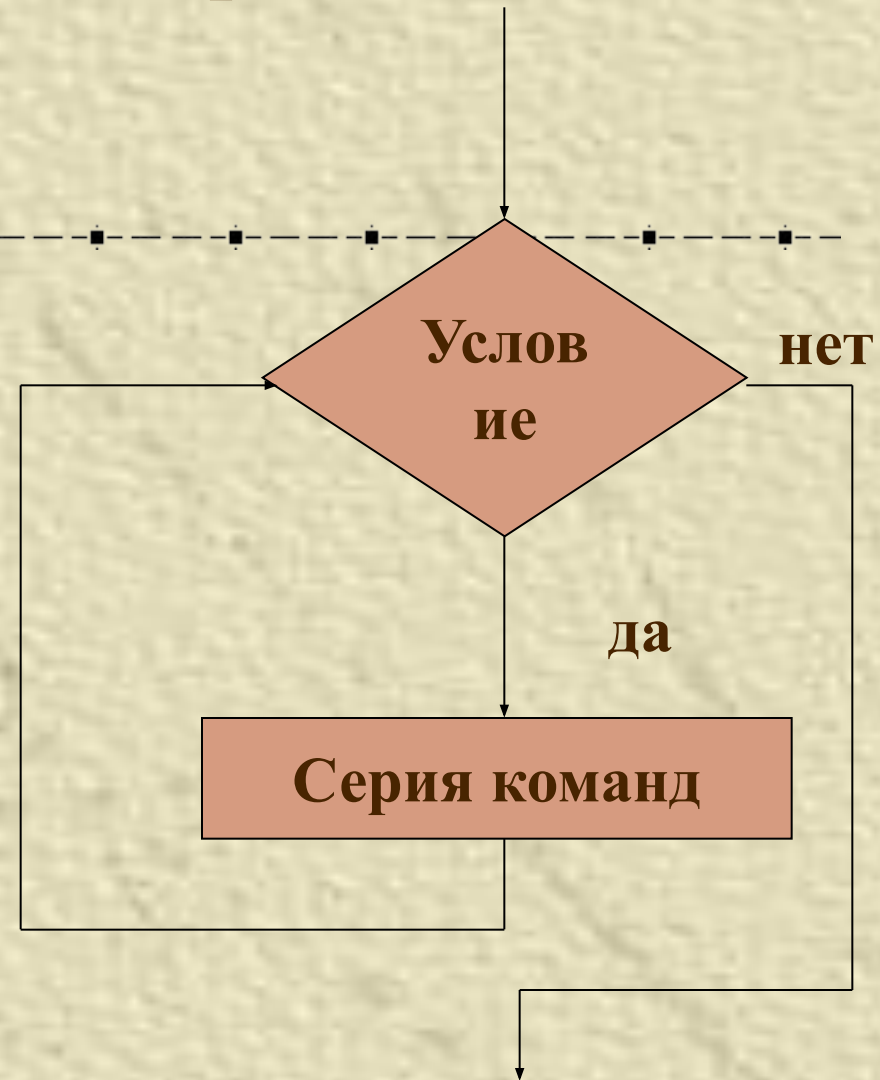
Пока <Условие>

нц

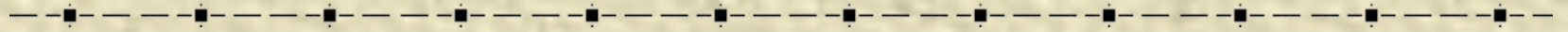
<тело цикла>

кц

Тело цикла – это серия команд, которая выполняется в цикле.



Вспомогательные алгоритмы



Алгоритм, по которому решается подзадача из основной задачи и который, как правило, выполняется многократно, называется **вспомогательным алгоритмом.**

В языках программирования их называют **подпрограммами** или **процедурами.**

Вызов процедуры.

делай <имя процедуры>

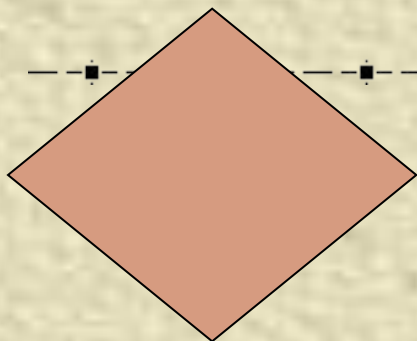
Описание процедуры.

процедура <имя процедуры>

<команды>

конец процедуры

Блок-схемы алгоритмов.



Проверка условия

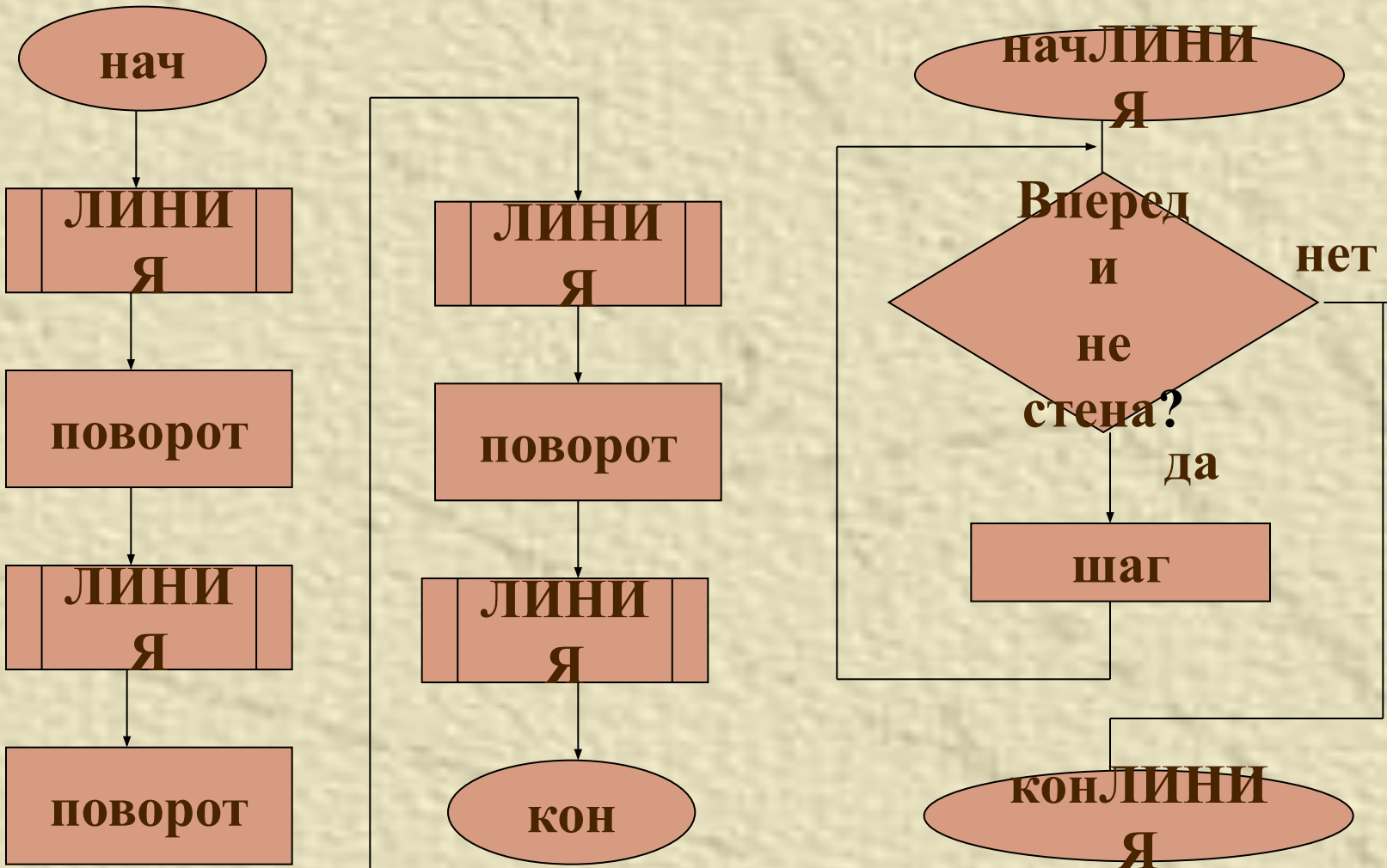


**Обращение к
вспомогательному
алгоритму (процедуре)**

Алгоритм РАМКА

Основной алгоритм

Процедура



Алгоритм РАЗЛИНОВКА

