

Практика 20

Базовые алгоритмические структуры

1. **Цель работы:** сформировать понятие моделирования как метода познания, рассмотреть различные классификации моделей; изучить «информационная модель», научиться описывать информационные модели.
2. **Оборудование, приборы, аппаратура, материалы:** персональный компьютер с операционной системой Windows.
3. **Теоретические сведения для самоподготовки.**
Записать конспект в тетрадь.

Теоретический материал для самостоятельного изучения

Структуры алгоритмов

На прошлом уроке мы начали изучать алгоритмы. В жизни нам приходится решать множество задач, и для каждой из них можно составить множество алгоритмов. Но все они состоят из одинаковых структур, которые мы и рассмотрим на этом уроке.

В 1969 году нидерландский ученый Эдсгер Дийкстра доказал важную теорему. Суть ее в том, что для решения любой логической задачи можно составить алгоритм, используя лишь три алгоритмических структуры: следование, ветвление и повторение. Эти структуры называют базовыми.

Алгоритмическая конструкция «следование» (линейный алгоритм)

Самой простой структурой является «следование».

Следование — линейная последовательность действий, порядок выполнения которых соответствует порядку их записи.

Алгоритм реализован через последовательную алгоритмическую структуру, если все команды этого алгоритма выполняются один раз, причем в том порядке, в котором они записаны.

Алгоритм, основанный на конструкции «следование» называется **линейным алгоритмом**. Примером такого алгоритма может служить алгоритм вычисления дискриминанта квадратного уравнения, блок-схема которого приведена на рисунке 1.

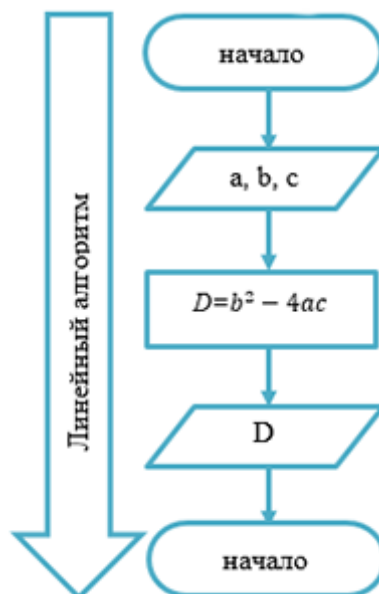


Рис. 1

Алгоритмическая конструкция «ветвление» (разветвляющийся алгоритм)

Следующей конструкцией является «**ветвление**».

Ветвление — алгоритмическая структура, в которой одна из двух последовательностей действий выполняется в зависимости от некоторого условия.

Она встречается, если действия алгоритма зависят от некоторого условия.

Алгоритм реализован через алгоритмическую конструкцию «ветвление», если от входных данных зависит, какие команды будут выполняться. Условие, которое выражает эту зависимость, фактически является вопросом, на который можно ответить либо «да», либо «нет».

Существуют **полная и неполная** формы ветвления.

В **полной форме** если условие выполняется, то алгоритм переходит к выполнению первой серии команд, а если не выполняется — то ко второй.

В **неполной форме** алгоритм выполняет серию команд только если условие истинно. В противном случае ничего не происходит.

Алгоритм, основанный на конструкции «ветвление» называется **разветвляющимся алгоритмом**. Примером такого алгоритма может служить алгоритм нахождения корней квадратного уравнения, блок-схема которого приведена на рисунке 2.

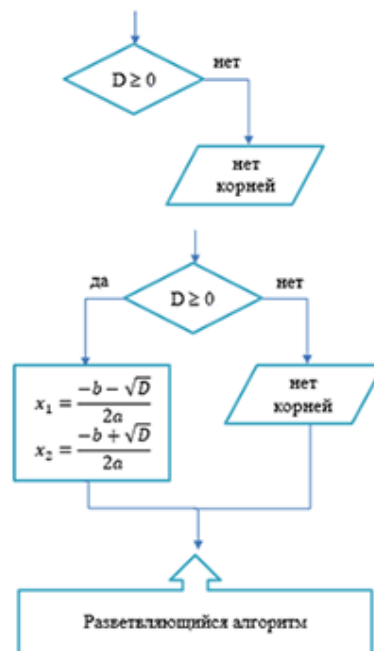


Рис. 2

Алгоритмическая конструкция «повторение» (циклический алгоритм)

И, наконец, последняя алгоритмическая конструкция — «**повторение**».

Повторение (или цикл) — это многократное выполнение одной и той же последовательности команд.

Алгоритм, содержащий конструкцию «повторение» называется **циклическим алгоритмом**.

Существует несколько разновидностей циклов: **цикл с предусловием, цикл с постусловием, цикл с параметром**

Алгоритм реализован с использованием алгоритмической конструкции «повторение», если некая группа подряд идущих шагов алгоритма (она называется телом цикла) может выполняться многократно в зависимости от входных данных.

Существует несколько разновидностей циклических алгоритмов.

Первый — цикл с заданным условием продолжения работы (цикл с предусловием или цикл-пока).

Второй — цикл с заданным условием окончания работы (цикл с постусловием или цикл-до).

И третий — цикл с заданным числом повторений (цикл с параметром).



Доказано, что при решении задач можно ограничиться только одним циклом — циклом с предусловием. Но в ряде случаев цикл с постусловием или цикл с параметром делают решение задачи легче.

Примером решения одной и той же задачи с помощью различных циклов может служить задача возведения некоторого числа a в натуральную степень n .

