

Алгоритмы и величины Структура --- алгоритмов

Работа по решению любой задачи с использованием компьютера делится на следующие этапы:

- 1. Постановка задачи.**
- 2. Формализация задачи.**
- 3. Построение алгоритма.**
- 4. Составление программы на языке программирования.**
- 5. Отладка и тестирование программы.**
- 6. Проведение расчетов и анализ полученных результатов.**

Программист должен обладать следующими знаниями и навыками:

- уметь строить алгоритмы;
- знать языки программирования;
- уметь работать в соответствующей системе программирования.

Понятие алгоритма

Одним из фундаментальных понятий в информатике является понятие алгоритма. Происхождение самого термина «алгоритм» связано с математикой. Это слово происходит от *Algorithmi* – латинского написания имени Мухаммеда аль-Хорезми (787-850), выдающегося математика средневекового Востока.

Алгоритм – это последовательность команд управления каким-либо исполнителем.

В школьном курсе информатики с понятием алгоритма, с методами построения алгоритмов ученики знакомятся на примерах учебных исполнителей: Робота, Черепахи, Чертежника и т.д. Эти исполнители ничего не вычисляют. Они создают рисунки на экране, перемещаются в лабиринтах, перетаскивают предметы с места на место. Таких исполнителей принято называть исполнителями, работающими в обстановке.

В разделе информатики под названием «Программирование» изучаются методы программного управления работой компьютера. Следовательно, в качестве исполнителя выступает компьютер. Компьютер работает с величинами — различными информационными объектами: числами, символами, кодами и т. п. Поэтому алгоритмы, предназначенные для управления компьютером, принято называть алгоритмами работы с величинами.

Данные и величины

Совокупность величин, с которыми работает компьютер, принято называть **данными**. По отношению к программе данные делятся на **исходные**, **результаты** (окончательные данные) и **промежуточные**, которые получаются в процессе вычислений.



Пример

При решении квадратного уравнения

$$ax^2 + bx + c = 0$$

исходными данными являются коэффициенты a , b , c , результатами – корнями уравнения x_1 , x_2 , промежуточным данным – дискриминант уравнения $D = b^2 - 4ac$.

Для успешного освоения программирования необходимо усвоить следующее правило: **всякая величина занимает свое определенное место в памяти компьютера (иногда говорят – ячейку памяти).**

У всякой величины имеются три основных свойства: **имя, значение и тип.**

В алгоритмах и языках программирования величины делятся на **константы и переменные.**

Константа – неизменная величина, и в алгоритме она представляется собственным значением, например: 15, 34.7, ‘k’, true и т.д.

Переменные величины могут изменять свои значения в ходе выполнения программы и представляются символическими именами – идентификаторами, например: X, S2, cod15.

Любая константа, как и переменная, занимает ячейку памяти, а значение этих величин определяется двоичным кодом в этой ячейке.

Типы величин — типы данных

В любой язык входит минимально необходимый набор основных типов данных, к которому относятся: **целый, вещественный, логический и символьный типы.**

С типом величины связаны три ее характеристики: **множество допустимых значений, множество допустимых операций, форма внутреннего представления.**

Тип	Значения	Операции	Внутреннее представление
Целый	Целые положительные и отрицательные числа в некотором диапазоне. Примеры: 23, -12, 387	Арифметические операции с целыми числами: +, -, *, целочисленное деление и остаток от деления. Операции отношений (<, >, = и др.)	Формат с фиксированной запятой
Вещественный	Любые (целые и дробные) числа в некотором диапазоне. Примеры: 2.5, -0.01, 45.0, $3.6 \cdot 10^9$	Арифметические операции: +, -, *, /. Операции отношений	Формат с плавающей запятой
Логический	true (истина) false (ложь)	Логические операции: И (and), ИЛИ (or), НЕ (not). Операции отношений	1 бит: 1 — true; 0 — false
Символьный	Любые символы компьютерного алфавита. Примеры: 'a', '5', '+', '\$'	Операции отношений	Коды таблицы символьной кодировки. 1 символ — 1 байт

Есть еще один вариант классификации данных – классификация по структуре. Данные делятся на **простые** и **структурированные**. Для простых величин (их еще называют скалярными) справедливо утверждение: одна величина – одно значение, для структурированных: одна величина – множество значений. К структурированным величинам относятся массивы, строки, множества и т.д.

Компьютер – исполнитель алгоритмов

Исполнителем является комплекс компьютера + Система программирования (СП).

Программист составляет программу на том языке, на который ориентирована СП.



Алгоритм решения любой задачи на компьютере может быть составлен из команд:

- присваивания;
- ввода;
- вывода;
- обращения к вспомогательному алгоритму;
- цикла;
- ветвления.

Линейные вычислительные алгоритмы

Пример

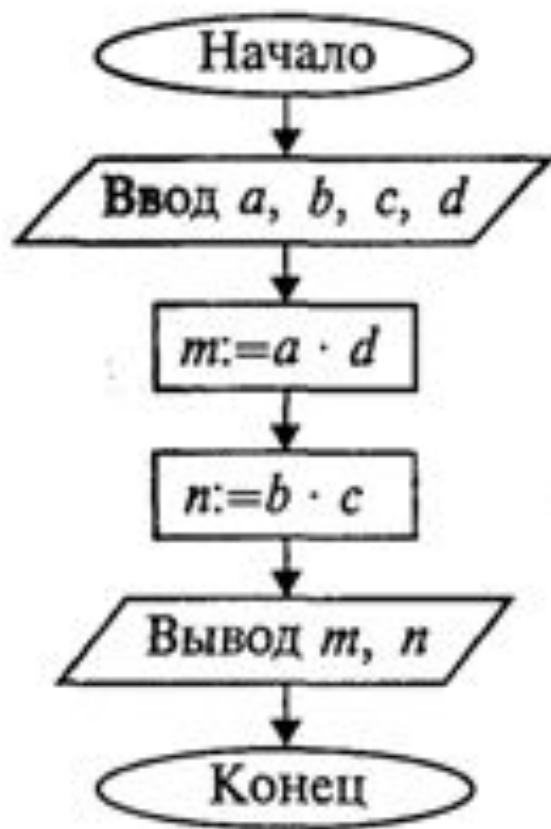
В школьном учебнике математики правила деления обыкновенных дробей описаны так:

1. Числитель первой дроби умножить на знаменатель второй дроби.
2. Знаменатель первой дроби умножить на числитель второй дроби.
3. Записать дробь, числитель которой есть результат выполнения пункта 1, а знаменатель – результат выполнения пункта 2.

В алгебраической форме это выглядит следующим образом:

$$\frac{a}{b} : \frac{c}{d} = \frac{a \cdot d}{b \cdot c} = \frac{m}{n}.$$

Исходными данными являются целочисленные переменные a , b , c , d . Результатом – также целые величины тип.



```
алг Деление дробей
нач
    цел  $a, b, c, d, m, n$ 
    ввод  $a, b, c, d$ 
     $m := a * d$ 
     $n := b * c$ 
    вывод  $m, n$ 
кон
```

Формат команды присваивания следующий:

переменная:=выражение

Знак «:=» нужно читать как «присвоить».

Команда присваивания обозначает следующие действия, выполняемые компьютером:

1. Вычисляется выражение.
2. Полученное значение присваивается переменной.

В приведенном выше алгоритме присутствуют две команды присваивания. В блок-схемах команда присваивания записывается в прямоугольнике. Такой блок называется вычислительным блоком.

В приведенном алгоритме присутствует команда ввода:

ввод a, b, c, d

В блок-схеме команда ввода записывается в параллелограмме — блоке ввода-вывода. При выполнении данной команды процессор прерывает работу и ожидает действий пользователя. Пользователь должен набрать на устройстве ввода (клавиатуре) значения вводимых переменных и нажать на клавишу ввода Enter. Значения следует вводить в том же порядке, в каком соответствующие переменные расположены в списке ввода.

Полученные компьютером результаты решения задачи должны быть сообщены пользователю. Для этих целей предназначена команда вывода:

ВЫВОД m, n

С помощью этой команды результаты выводятся на экран или на устройство печати на бумагу.

Команда	a	b
$a := 1$	1	—
$b := 2 \cdot a$	1	2
$a := b$	2	2
$b := a + b$	2	4

Этот пример иллюстрирует три основных свойства команды присваивания:

- пока переменной не присвоено значение, она остается неопределенной;
- значение, присвоенное переменной, сохраняется в ней вплоть до выполнения следующей команды присваивания этой переменной;
- новое значение, присваиваемое переменной, заменяет ее предыдущее значение.

Система основных понятий

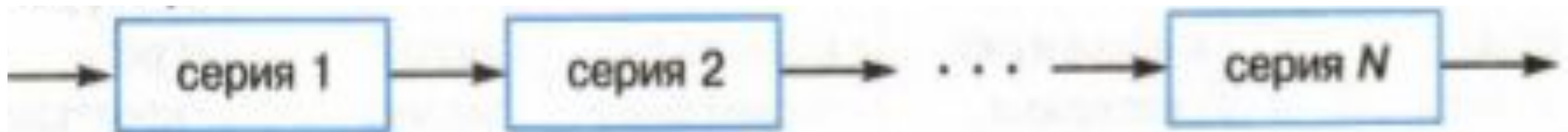
Алгоритмы и величины				
<p>Этапы решения задачи на компьютере</p> <ol style="list-style-type: none"> 1. Постановка задачи. 2. Формализация задачи. 3. Построение алгоритма. 4. Составление программы на языке программирования. 5. Отладка и тестирование программы. 6. Проведение расчетов и анализ полученных результатов 				
<p>Компьютер + система программирования = исполнитель алгоритмов обработки данных</p>				
Система команд исполнителя	Классификация данных			
<ul style="list-style-type: none"> - присваивание; - ввод; - вывод; - ветвление; - цикл; - обращение к подпрограмме 	<p>По отношению к алгоритму:</p> <ul style="list-style-type: none"> - исходные; - промежуточные; - итоговые (результаты) 	<p>По значениям:</p> <ul style="list-style-type: none"> - константы; - переменные 	<p>По типам:</p> <ul style="list-style-type: none"> - целые; - вещественные; - логические; - символьные 	<p>По структуре:</p> <ul style="list-style-type: none"> - простые; - структурированные

Базовые структуры алгоритмов

В 1969 году известным голландским ученым-программистом Э. В. Дейкстрой было доказано, что алгоритм для решения любой логической задачи можно составить только из структур следование, ветвление, цикл. Их называют базовыми алгоритмическими структурами. Методика программирования, основанная на этой теореме, называется структурным программированием.

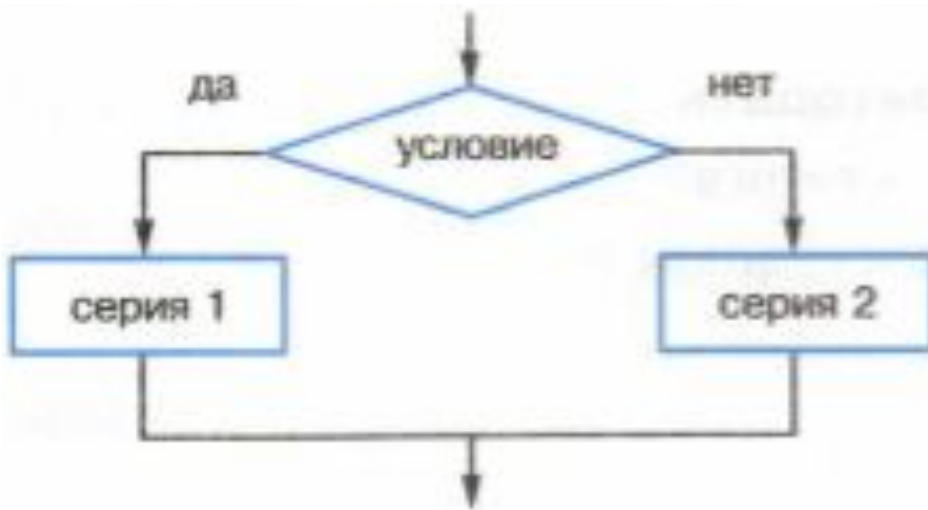
Следование (линейный алгоритм)

Линейным алгоритмом называется алгоритм, в котором все действия (операции) выполняются один раз и последовательно друг за другом.



Ветвление

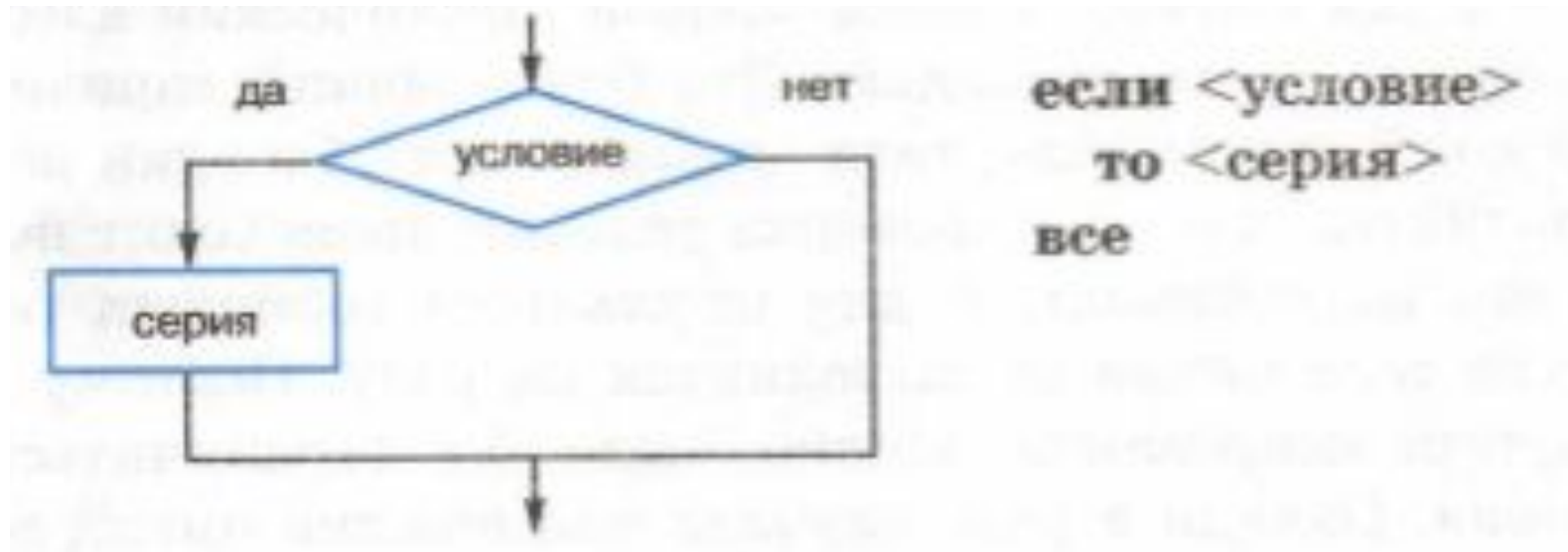
Ветвление – алгоритмическая альтернатива.



```
если <условие>  
то <серия 1>  
иначе <серия 2>  
все
```

Ветвление

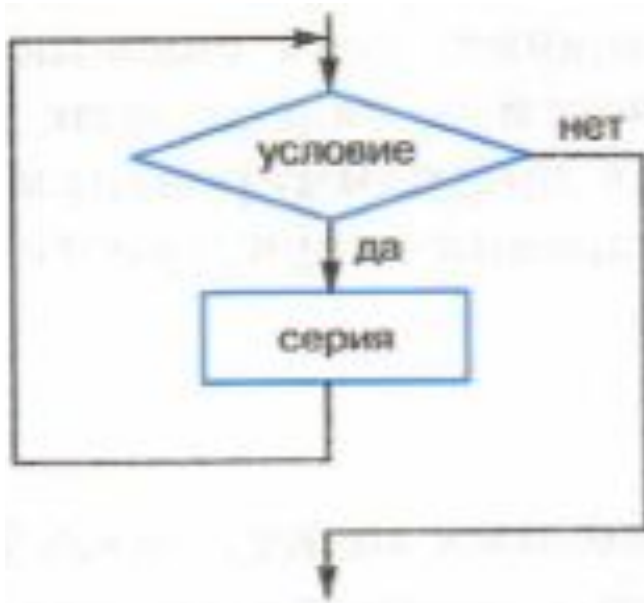
Ветвление – алгоритмическая альтернатива.



Циклический алгоритм

Цикл – повторение некоторой группы действий по условию.

Цикл с предусловием: цикл-пока



```
если <условие>  
нц  
  <серия>  
все
```

Циклический алгоритм

Цикл – повторение некоторой группы действий по условию.

Цикл с постусловием: цикл-до



повторить
<серия>
до <условие>

Комбинации базовых структур

Сложный алгоритм состоит из соединенных между собой базовых структур. Соединяться эти структуры могут двумя способами: *последовательным и вложенным.*

Наглядность структуре описания алгоритма на АЯ придает *структуризация внешнего вида текста.* Основной используемый для этого прием — сдвиги строк, которые должны подчиняться следующим правилам:

- конструкции одного уровня вложенности записываются на одном вертикальном уровне (начинаются с одной позиции в строке);
- вложенная конструкция записывается смещенной по строке на несколько позиций вправо относительно внешней для нее конструкции.

Комбинации базовых структур

Структурное программирование — это не только форма описания алгоритма и программы, но это еще и *способ мышления программиста*. Размышляя над алгоритмом, нужно стремиться составлять его из стандартных структур. Если использовать строительную аналогию, то структурная методика построения алгоритма подобна сборке здания из стандартных секций, в отличие от складывания по кирпичику.

Вопросы и задания

1. Перечислите основные базовые алгоритмические структуры и покажите способы их отображения на блок-схемах и в АЯ.
2. Какой алгоритм называется структурным?
3. Нарисуйте блок-схемы и напишите на АЯ два варианта алгоритма решения задачи: выбрать из двух числовых величин наибольшее значение. Первый вариант — с полным ветвлением, второй вариант — с неполным ветвлением.
4. Нарисуйте блок-схемы и напишите на АЯ два варианта алгоритма решения задачи: выбрать из трех числовых величин наименьшее значение. Первый вариант — с вложенными ветвлениями, второй вариант — с последовательными ветвлениями.

Вопросы и задания

5. Для данного натурального числа N требуется вычислить сумму: $1 + 1/2 + 1/3 + \dots + 1/N$. Постройте блок-схемы и напишите на АЯ два варианта алгоритма: с циклом-до и с циклом-пока.
6. Какую структуру будет иметь алгоритм решения следующей задачи? Дано целое положительное число N . Если N — четное, то вычислить $N! = 1 \cdot 2 \cdot \dots \cdot N$. Если N — нечетное, то вычислить сумму: $1 + 2 + \dots + N$. Составьте блок-схему алгоритма решения и опишите его на АЯ.

Домашнее задание

§ 12, 13, задания стр. 98