

«Информатика» (УМК Л.Л.Босова)

9 класс <http://www.lbz.ru/metodist/authors/informatika/3/eor9.php>

8 класс <http://www.lbz.ru/metodist/authors/informatika/3/eor8.php>

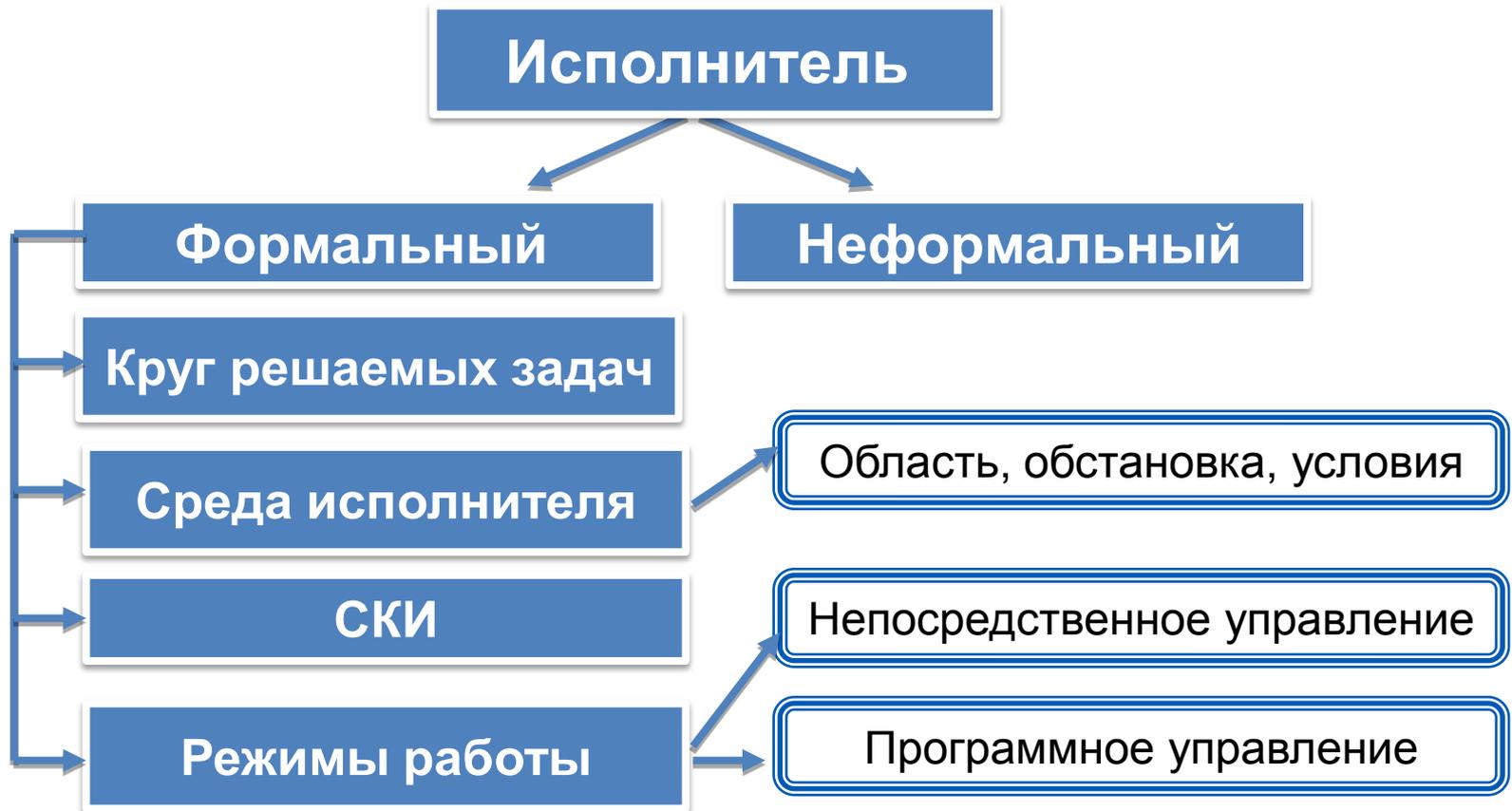
ОПОРНЫЙ КОНСПЕКТ по теме «Основы алгоритмизации. Алгоритмизация и программирование»



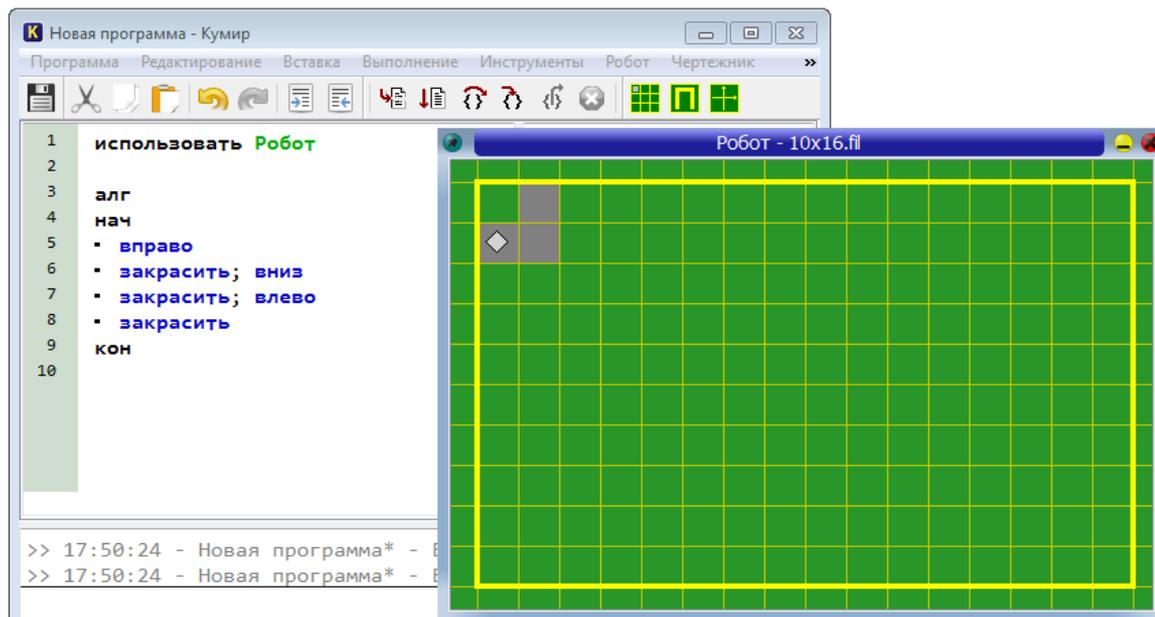
7, 8, 9 классы

Исполнитель алгоритма

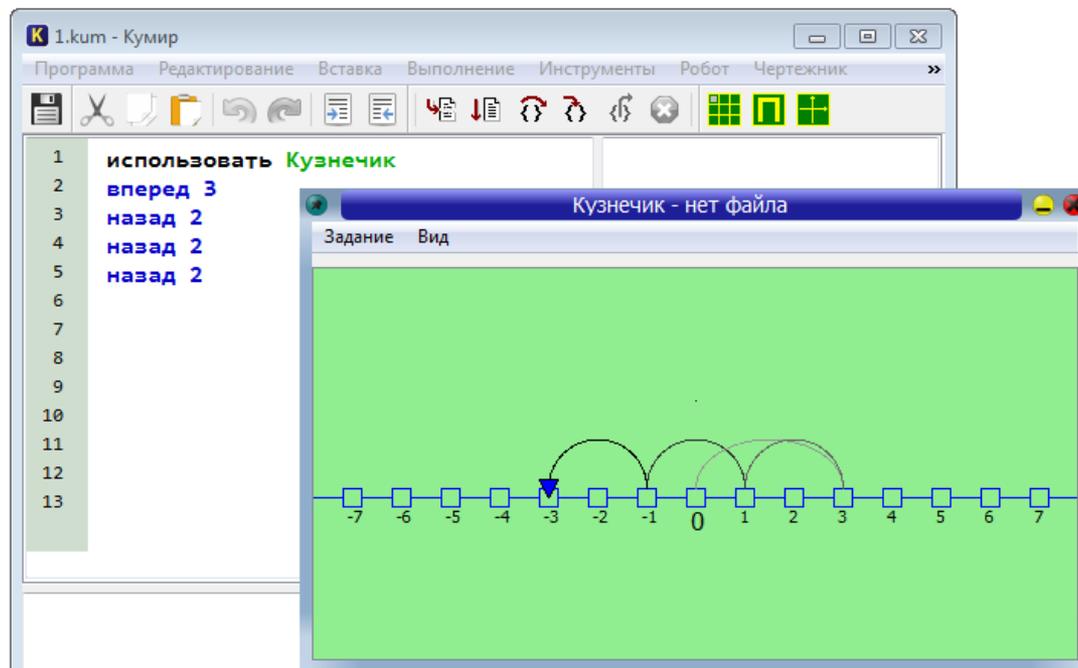
Исполнитель - это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.



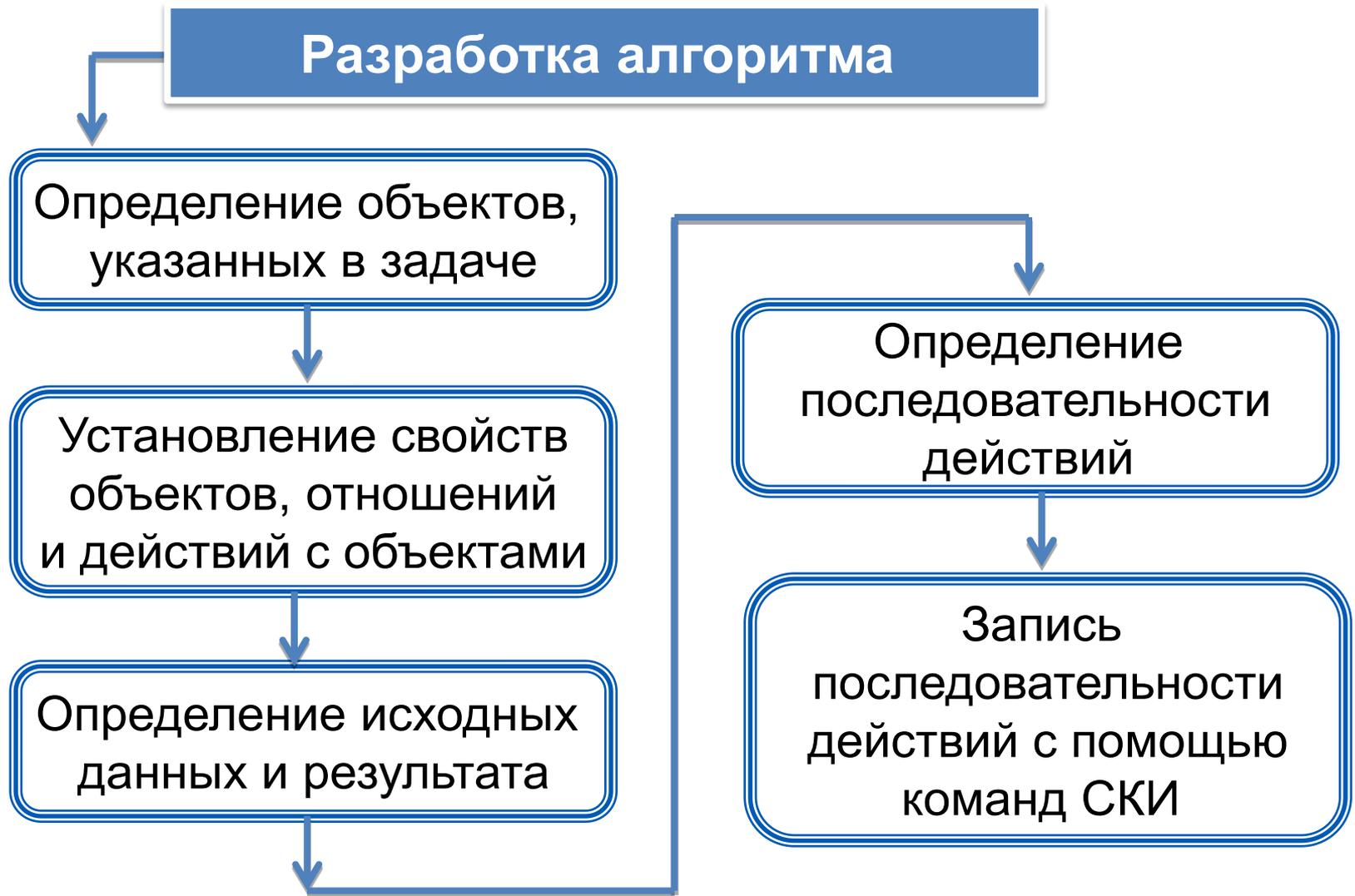
Исполнитель Робот



Исполнитель Кузнечик



Разработка алгоритма



Алгоритм – модель деятельности исполнителя алгоритмов

Свойства алгоритма

Свойства алгоритма

Дискретность

Путь решения задачи разделён на отдельные шаги

Понятность

Алгоритм состоит из команд, входящих в СИ

Определённость

Команды понимаются однозначно

Результативность

Обеспечивается получение ожидаемого результата

Массовость

Обеспечивается решение задач с различными исходными данными

АЛГОРИТМ - это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами:

- ***дискретности***
- ***понятности***
- ***определённости***
- ***результативности***
- ***массовости***

Возможность автоматизации деятельности человека

Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям.

Формальное исполнение алгоритма обеспечивает возможность автоматизации деятельности человека

Процесс решения задачи представляется в виде последовательности операций

Создается машина, способная выполнять эти операции в указанной последовательности

Человек освобождается от рутинной работы, выполнение которой поручается автомату



Способы записи алгоритмов



ИЗДАТЕЛЬСТВО

БИНОМ



Марков А.А. (1903—1979) установил, что алгоритмы должны содержать предписания двух видов:

1) ***функциональные операторы*** - предписания, направленные на непосредственное преобразование информации;

2) ***логические операторы*** - предписания, определяющие дальнейшее направление действий.

Именно эти операторы положены в основу большинства способов записи алгоритмов.

Основные способы записи алгоритма

Словесные

Словесное описание

Построчная запись

Обычный разговорный язык

Графические

Последовательность рисунков

Структурограмма

Блок-схема

Геометрические фигуры

На алгоритмических языках

Школьный алгоритмический язык

Язык программирования

Слова имеют заданный смысл и способ записи

Блок-схемы

В блок-схеме предписания изображаются с помощью различных геометрических фигур, а последовательность выполнения шагов указывается с помощью линий.



Блок начала или конца алгоритма



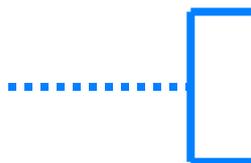
Блок ввода или вывода данных



Блок обработки данных



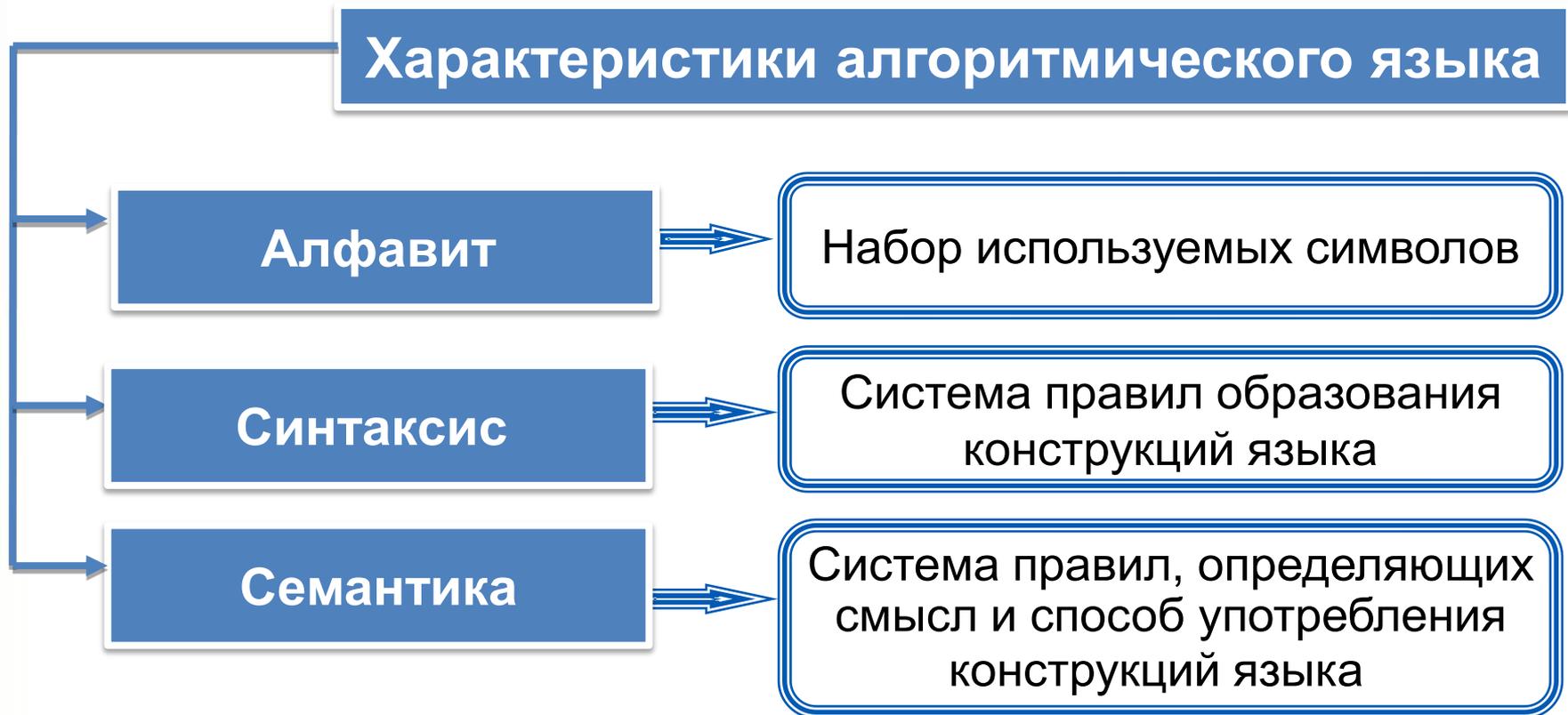
Блок проверки условия



Блок пояснительных записей

Алгоритмические языки

Алгоритмические языки - формальные языки, предназначенные для записи алгоритмов.



Общий вид программы на школьном алгоритмическом языке:

алг <название алгоритма>

нач

<последовательность команд>

кон

```
1  вещь длина, ширина
2  длина := 10
3  ширина := 15
4  алг
5  нач
6  · вещь S
7  · S := длина*ширина
8  · вывод "Площадь равна ", S
9  кон
10
11
```

длина=10.0
ширина=15.0
S=150.0

Площадь равна 150
>> 15:10:31 - Новая программа* - Выполнение завершено

Алгоритм для исполнителя Водолей

алг переливания

нач

наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л

наполнить сосуд ёмкостью 5 л из сосуда ёмкостью 8 л

вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л

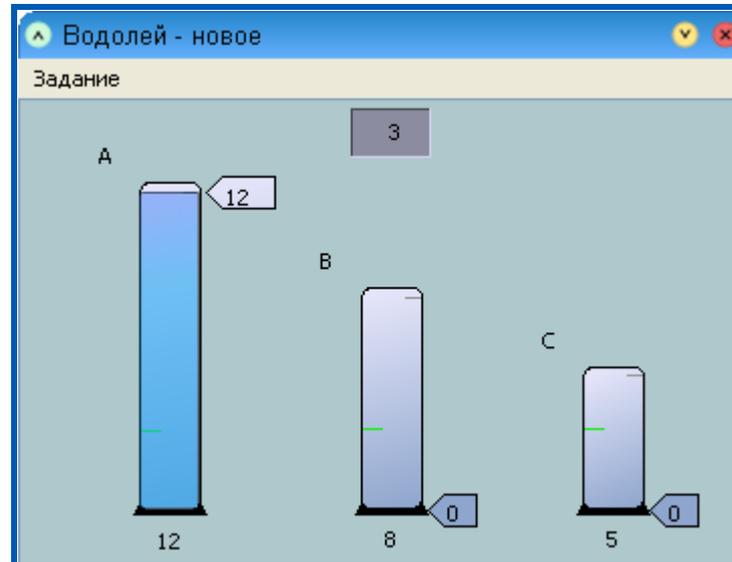
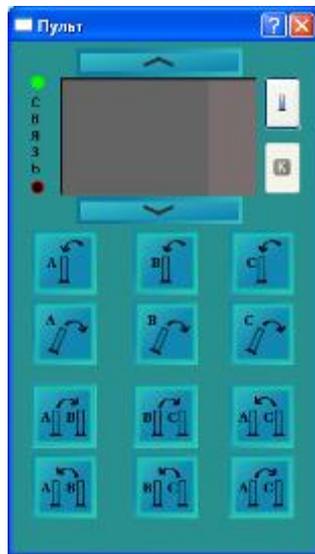
вылить всё из сосуда ёмкостью 8 л в сосуд ёмкостью 5 л

наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л

долить из сосуда ёмкостью 8 л сосуд ёмкостью 5 л

вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л

КОН





ОБЪЕКТЫ АЛГОРИТМОВ



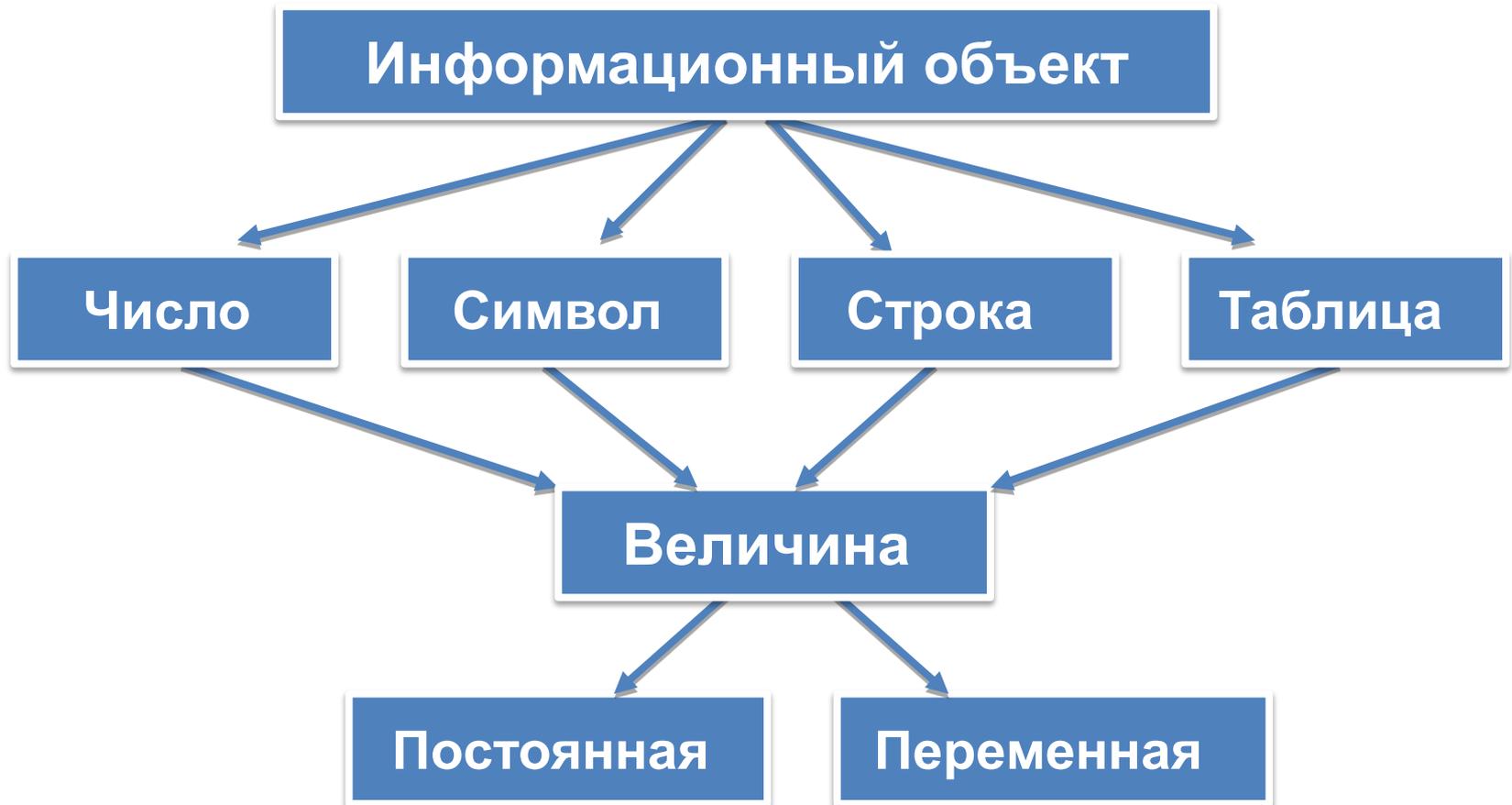
ИЗДАТЕЛЬСТВО

БИНОМ

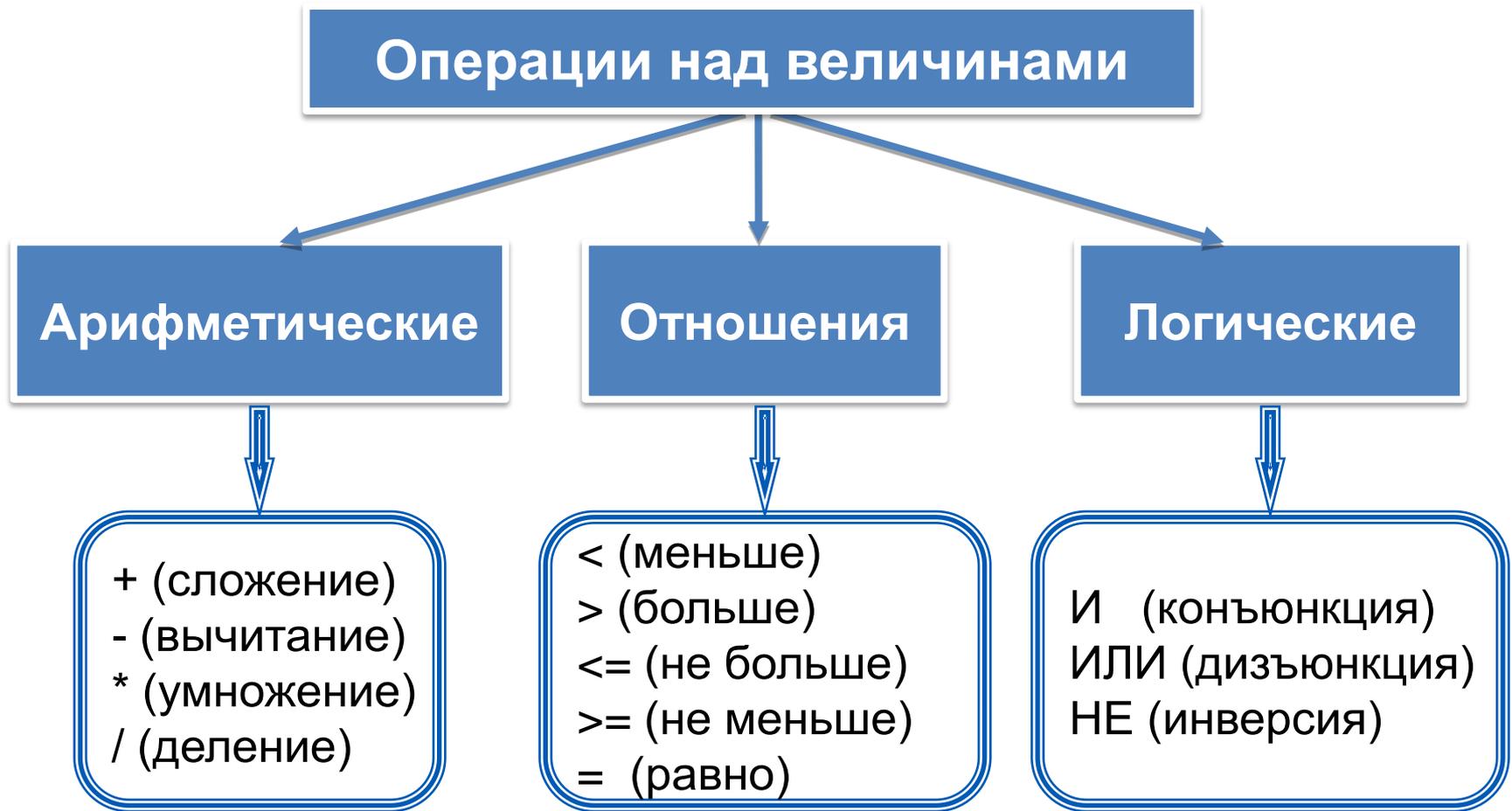
Величины

Алгоритмы описывают последовательность действий над некоторыми **информационными объектами**.

Величина в информатике – это отдельный информационный объект.

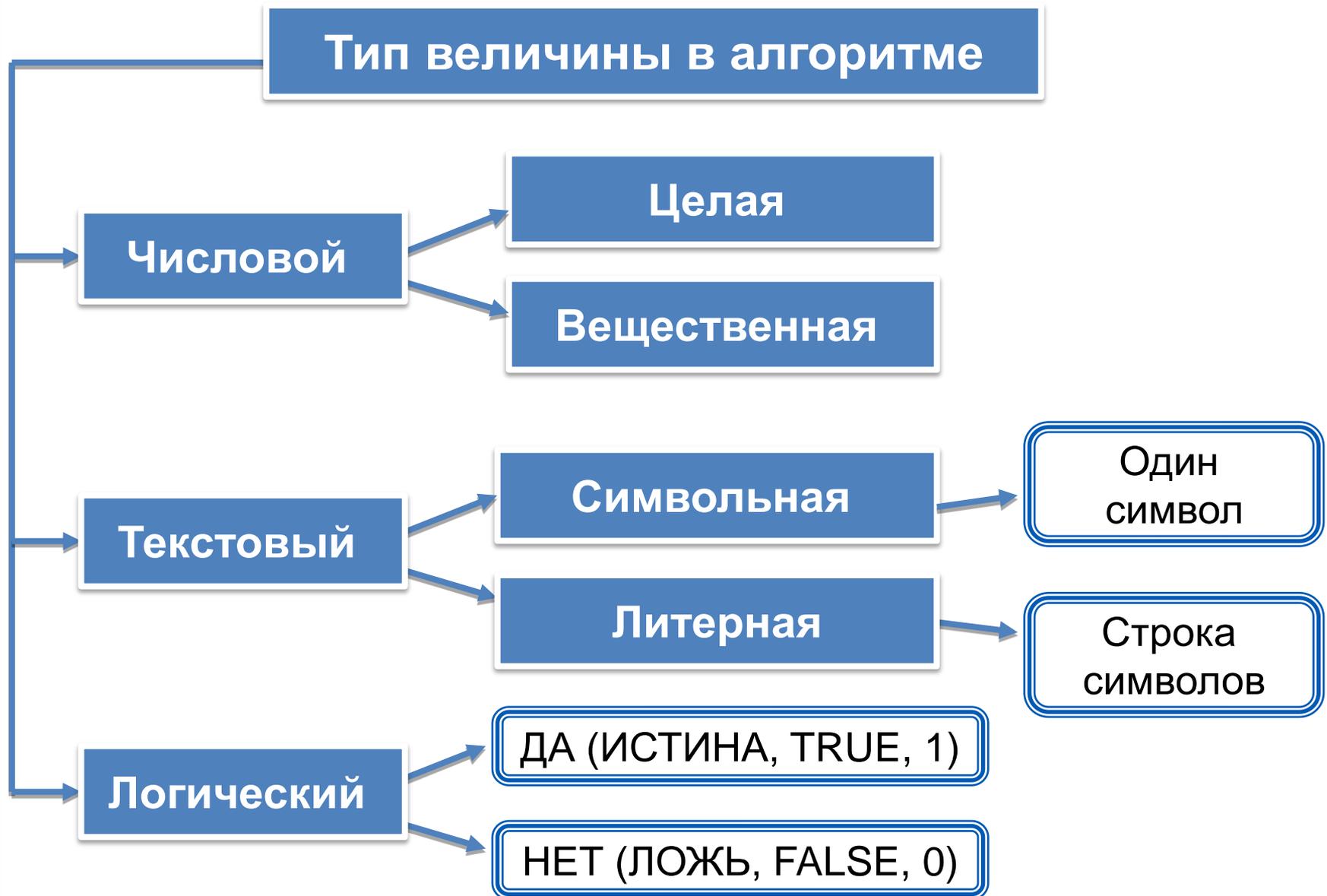


Операции над величинами

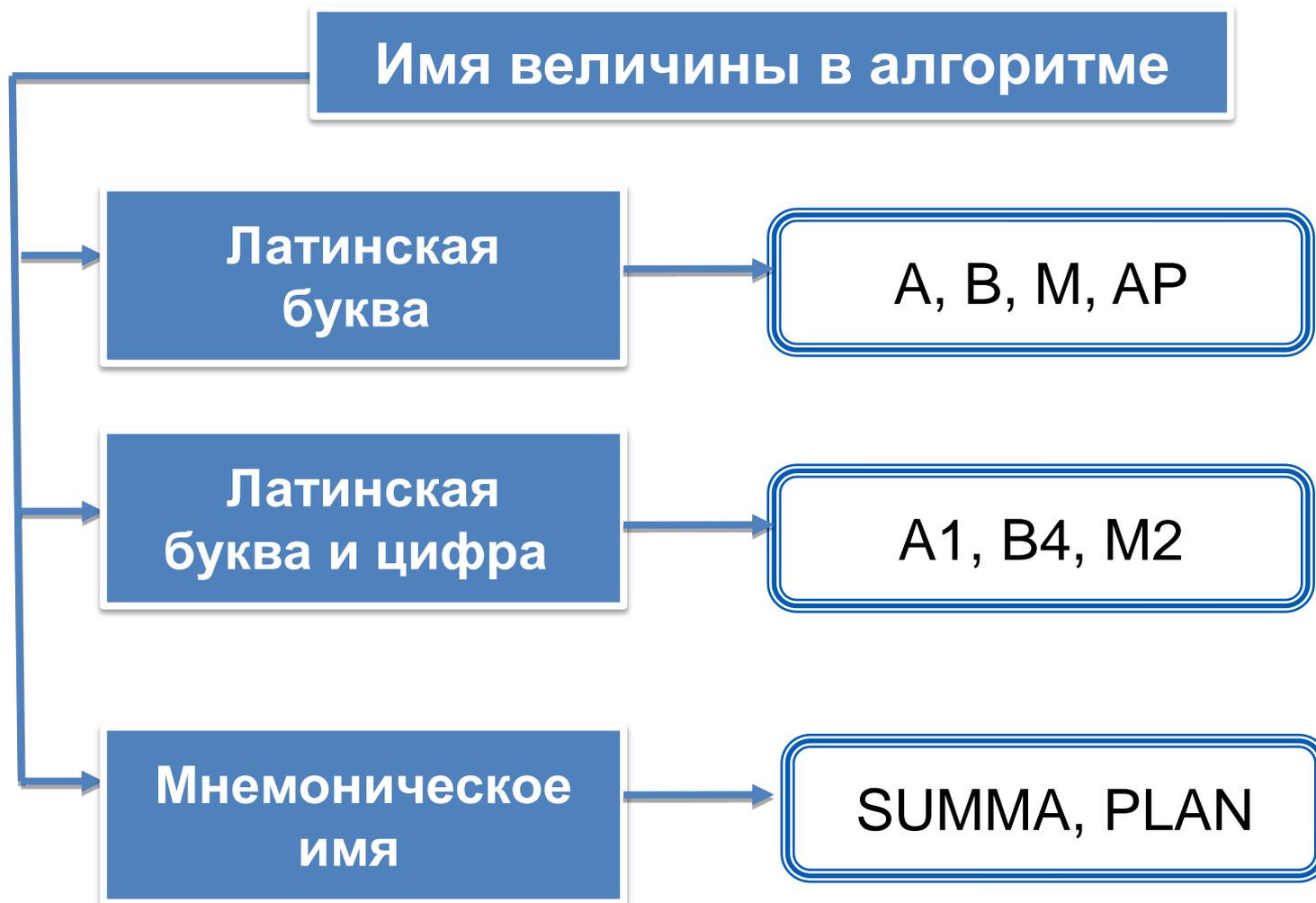


Операнды - объекты, над которыми выполняют операции.

Типы величин

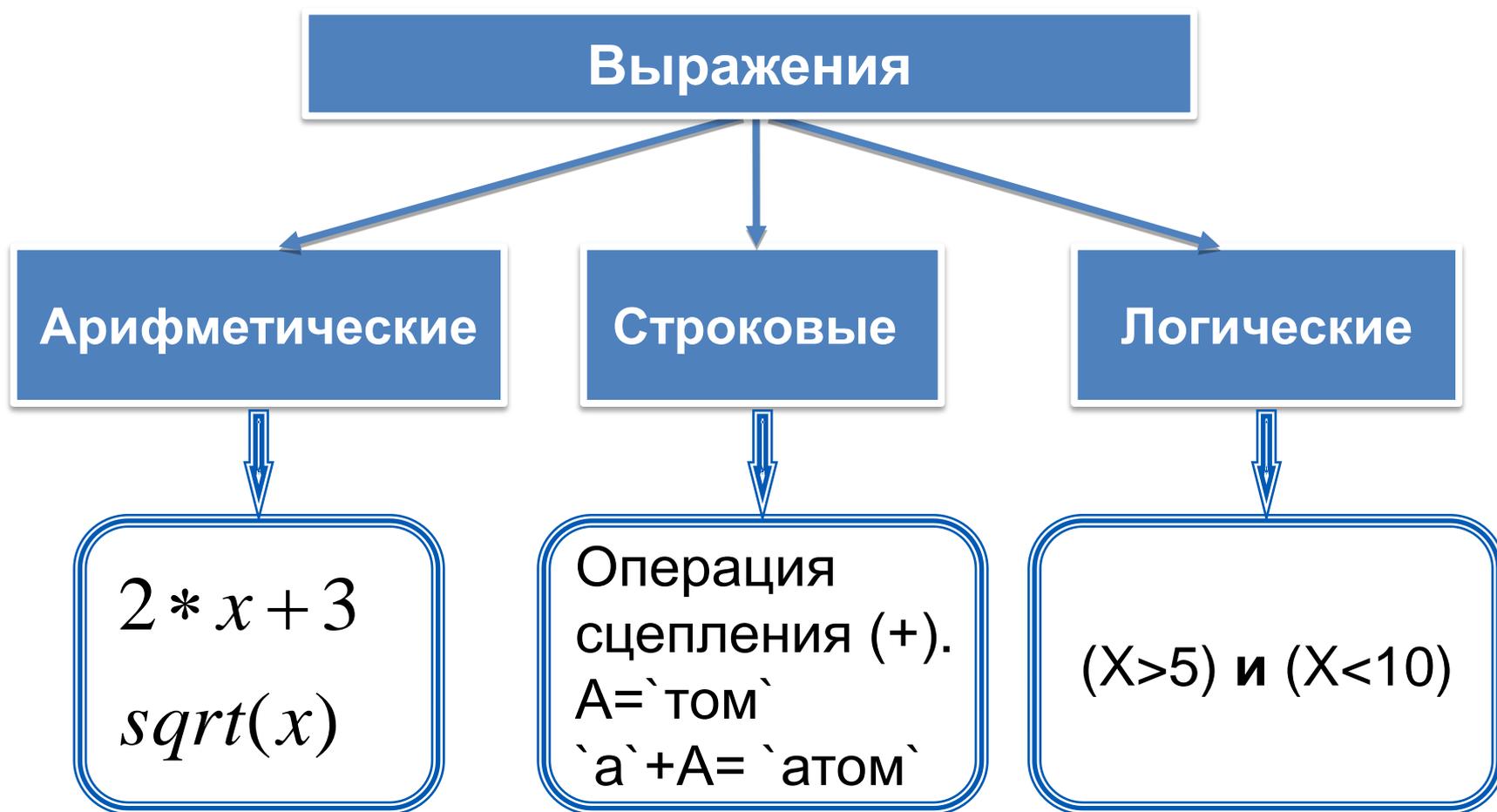


Имя величины



Выражения

Выражение - языковая конструкция для вычисления значения с помощью одного или нескольких операндов.



Команда присваивания

<имя переменной>:= <выражение>

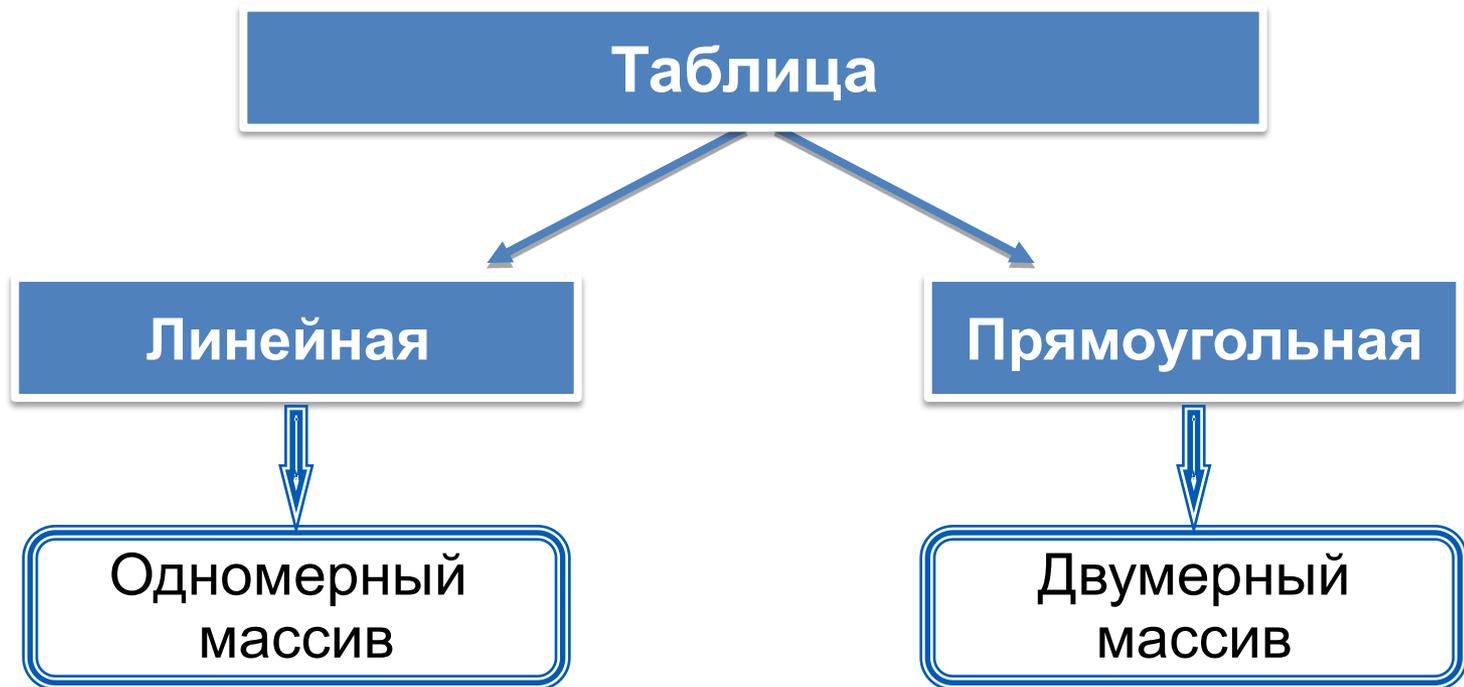
Свойства присваивания

Пока переменной не присвоено значение, она остаётся неопределённой

Значение, присвоенное переменной, сохраняется до следующего присваивания

Если переменной присваивается новое значение, то предыдущее её значение теряется

Таблица (массив) - набор некоторого числа однотипных элементов, которым присвоено одно имя. Положение элемента в таблице однозначно определяется его индексами.





АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ



ИЗДАТЕЛЬСТВО

БИНОМ

Основные алгоритмические конструкции

Для записи любого алгоритма достаточно трёх основных алгоритмических конструкций:

- следования,
- ветвления,
- повторения.

(Э. Дейкстра)

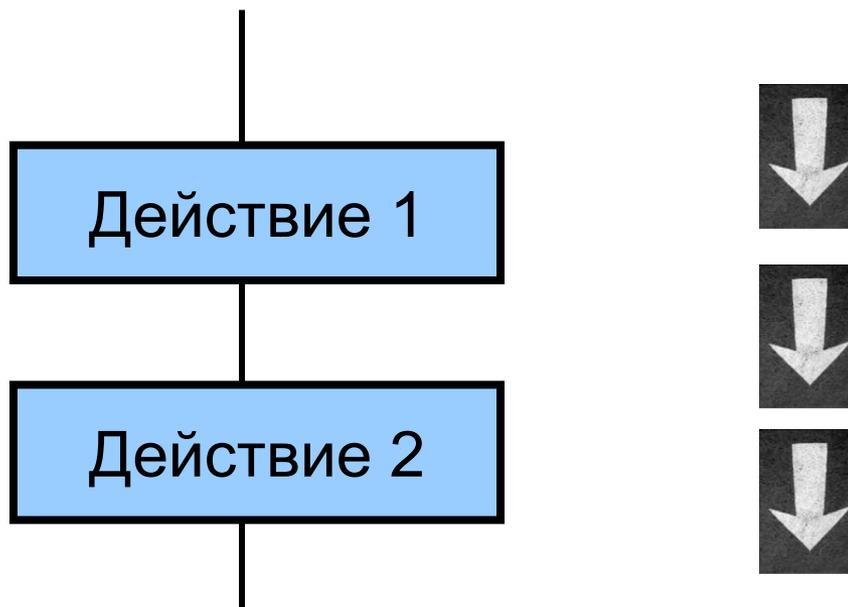


Эдсгер Вибе Дейкстра (1930–2002).
Выдающийся нидерландский учёный,
идеи которого оказали огромное
влияние на развитие компьютерной
индустрии.

Следование

Следование - алгоритмическая конструкция, отображающая естественный, последовательный порядок действий.

Алгоритмы, в которых используется только структура «следование», называются **линейными алгоритмами**.



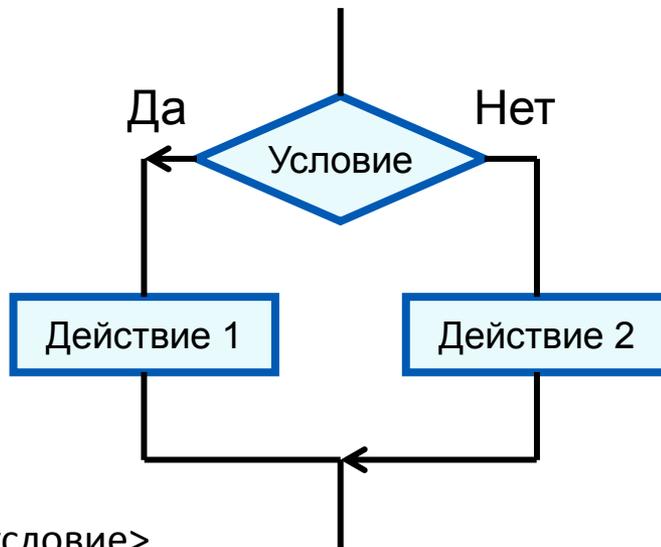
Алгоритмическая структура «следование»

Ветвление

Ветвление - алгоритмическая конструкция, в которой в зависимости от результата проверки условия (да или нет) предусмотрен выбор одной из двух последовательностей действий (ветвей).

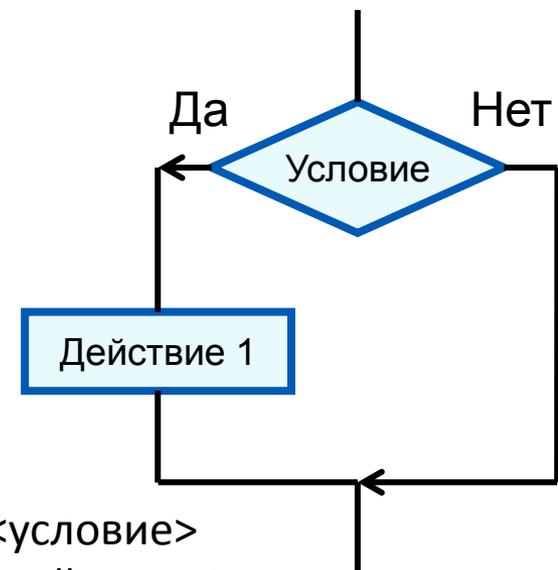
Алгоритмы, в основе которых лежит структура «ветвление», называют **разветвляющимися**.

Полная форма ветвления



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

Неполная форма ветвления



```
если <условие>  
  то <действие 1>  
все
```

Операции сравнения

$A < B$ A меньше B

$A \leq B$ A меньше или равно B

$A = B$ A равно B

$A > B$ A больше B

$A \geq B$ A больше или равно B

$A \neq B$ A не равно B

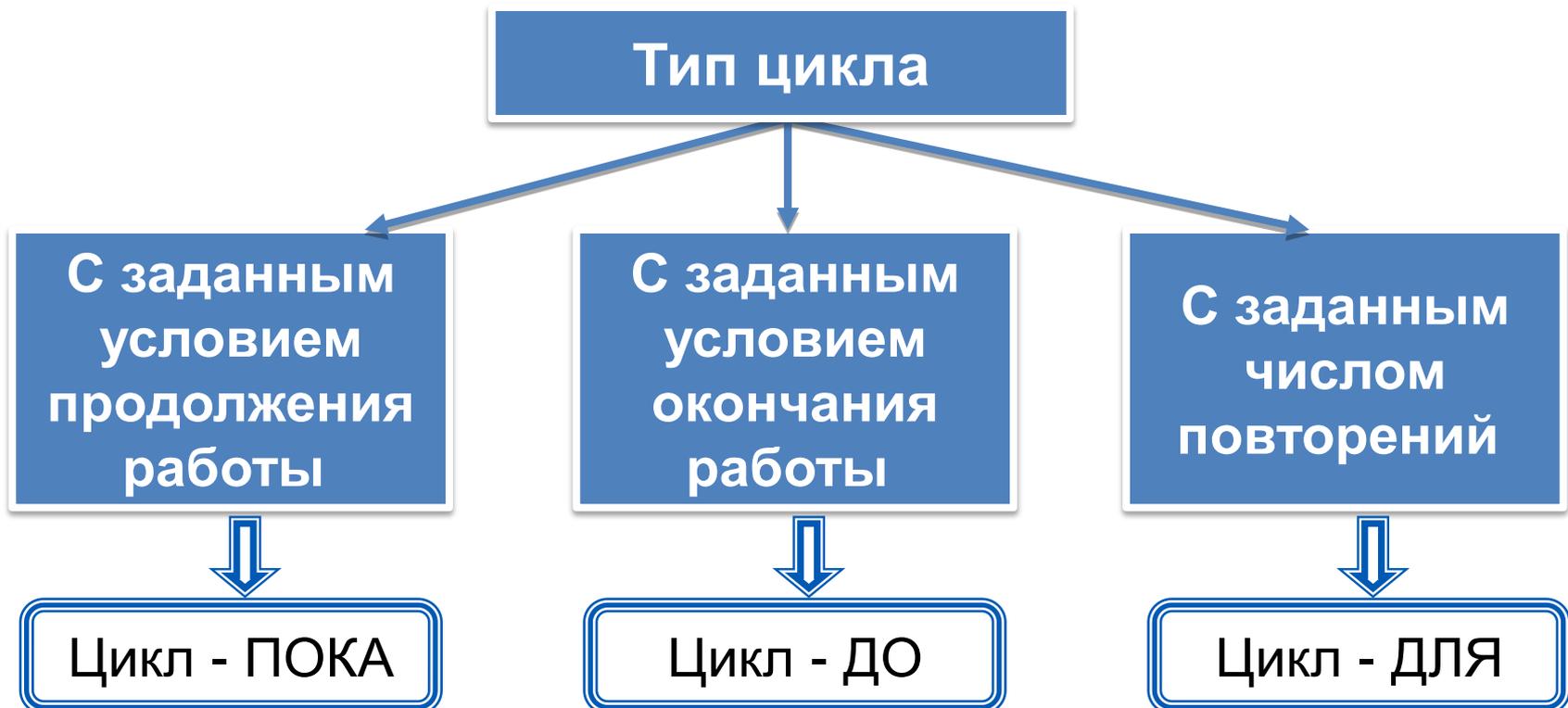


Повторение

Повторение - алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно.

Алгоритмы, содержащие конструкцию «повторение», называют **циклическими** или **циклами**.

Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.



Типы циклов



Могут быть

Заданы условия
продолжения работы

Пока есть кирпич

Заданы условия
окончания работы

*Пока не наступит
ночь*

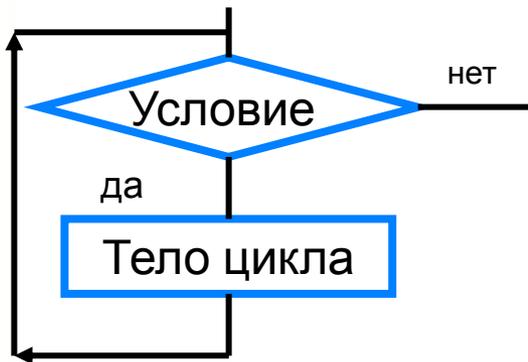
Задано число
повторений

Ровно 100 кирпичей

Цикл с заданным условием продолжения работы

*цикл-ПОКА,
цикл с предусловием*

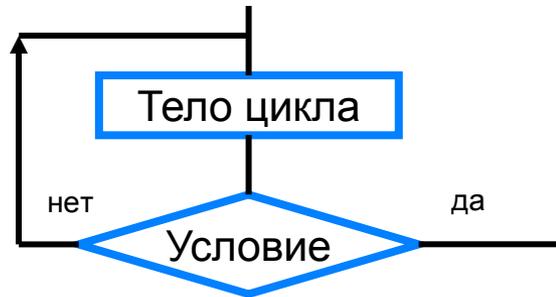
нц пока <условие>
<тело цикла
(последовательность
действий)>
кц



Цикл с заданным условием окончания работы

*цикл-ДО,
цикл с постусловием*

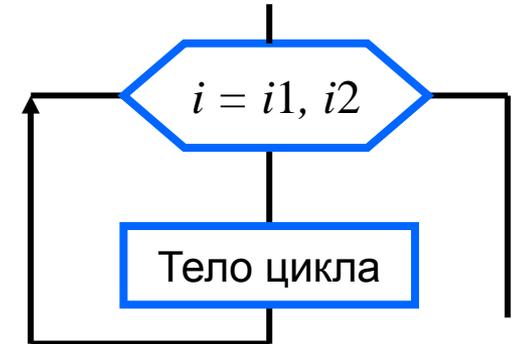
нц
<тело_цикла
(последовательность
действий)>
кц при <условие>



Цикл с заданным числом повторений

*цикл-ДЛЯ,
цикл с параметром*

нц для i от i_1 до i_2 шаг R
<тело_цикла
(последовательность
действий)>
кц





РЕШЕНИЕ ЗАДАЧ НА КОМПЬЮТЕРЕ



ИЗДАТЕЛЬСТВО

БИНОМ

Этапы решения задач на компьютере





Массив

Массив - это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элемента в массиве.

Одномерный массив



Решение разнообразных задач, связанных с обработкой массивов, базируется на решении таких типовых задач, как:

- суммирование элементов массива;
- поиск элемента с заданными свойствами;
- сортировка массива.

```
var <имя_массива>: array [<мин_знач_индекса> ..  
<макс_знач_индекса>] of тип_элементов;
```

Вычисление суммы элементов массива

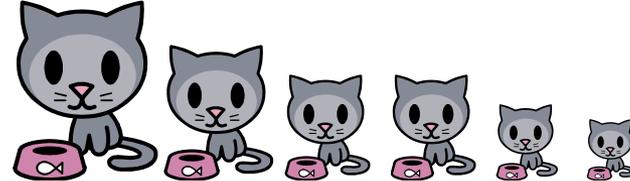
Суммирование элементов массива осуществляется за счёт поочерёдного добавления слагаемых:

Определяется ячейка памяти (переменная s), в которой будет последовательно накапливаться результат суммирования

Переменной s присваивается начальное значение 0 - число, не влияющее на результат сложения

Для каждого элемента массива из переменной s считывается её текущее значение и складывается со значением элемента массива; полученный результат присваивается переменной s .

Сортировка массива



Сортировка элементов массива по невозрастанию выбором осуществляется следующим образом:

1. В массиве выбирается максимальный элемент

2. Максимальный и первый элемент меняются местами (первый элемент считается отсортированным)

3. В неотсортированной части массива снова выбирается максимальный элемент; он меняется местами с первым неотсортированным элементом массива

Действия пункта 3 повторяются с неотсортированными элементами массива, пока не останется один неотсортированный элемент (минимальный)

Нахождение наибольшего элемента в стопке карточек с записанными числами:

1) Взять верхнюю карточку, записать на доске (запомнить) число как наибольшее.

2) Взять следующую карточку, сравнить числа. Если на карточке число больше, то записать это число.

Повторить действия, описанные в пункте 2 для всех оставшихся карточек

! При организации поиска наибольшего элемента массива правильнее искать его индекс.

