

АЛГОРИТМИЗА

ЦИИ И

ПРОГРАММИРО

ВАНИЯ

Алгоритм

Алгоритм – последовательность чётко определенных действий, выполнение которой ведёт к решению задачи не сам по себе, а предназначен для определенного исполнителя (человека, робота, компьютера, языка программирования и т.д.)

Своим свойством, характеризующим любого исполнителя, является то, что он умеет выполнять некоторые команды.

Алгоритм

Совокупность команд, которые данный исполнитель умеет выполнять, называется **системой команд исполителя**. командах исполнителя, который будет его **реализовывать**. Над которыми исполнитель может совершать действия, образуют так называемую **среду исполнителя**. Исходные данные и результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен

Свойства алгоритма

- **массовость** – алгоритм должен уметь решать не одну конкретную задачу, а
- **результативность** – алгоритм должен выдавать результат своей работы;
- **определенность** – алгоритм должен приводить к получению однозначного
- **дискретности** – заданным исходным **данным** из отдельных (дискретных) этапов;

Свойства алгоритма

- **последовательность** – на каждом шаге выполнения алгоритма должно быть точно известно, какой шаг будет
- **выполнимость** – каждая инструкция алгоритма должна быть понятна и
- **однозначность** (однозначность) – алгоритм обязательно должен завершаться.

Способы описания алгоритмов

- **Словесное описание представляет структуру алгоритма на естественном языке. Никаких правил составления словесного описания не существует. Запись алгоритма осуществляется в произвольной форме на естественном, например, русском языке. Этот способ описания строго не формализуем; допускает неоднозначность толкования при описании некоторых**

Способы описания алгоритмов

Алгоритм деления двух чисел:

- 1. Задать два числа, являющиеся делимым и делителем;**
- 2. Проверить, равняется ли делитель нулю;**
- 3. Если делитель не равен нулю, то найти частное, записать его в ответ;**
- 4. Если делитель равен нулю, то в ответ**

записать "нет решения"

Способы описания алгоритмов

- **Псевдокод** – описание структуры алгоритма на естественном, частично формализованном языке, позволяющее выявить основные этапы решения задачи, перед точной его записью на языке программирования. В псевдокоде используются некоторые формальные конструкции и общепринятая математическая символика.

алгоритмов

Строгих синтаксических правил для записи псевдокода не существует. Это облегчает запись алгоритма при проектировании и позволяет описать алгоритм, используя любой набор команд. Однако в псевдокоде обычно используются некоторые конструкции, присущие формальным языкам, что облегчает переход от псевдокода к записи алгоритма

на языке программирования Единого или

Способы описания алгоритмов

Базовые управляющие структуры псевдокода

Название структуры	Псевдокод
Присваивание	переменная = число
Ввод	ввод(переменная)
Вывод	вывод(переменная) вывод ("фраза")
Ветвление	если условие то действие1 иначе действие2
Повторение	пока условие начало пока действие конец пока

Способы описания алгоритмов

алг Нахождение частного двух чисел

начало

вывод ("задайте делимое и делитель")

ввод (делимое, делитель)

если делитель $\neq 0$

то частное = делимое / делитель

 вывод(частное)

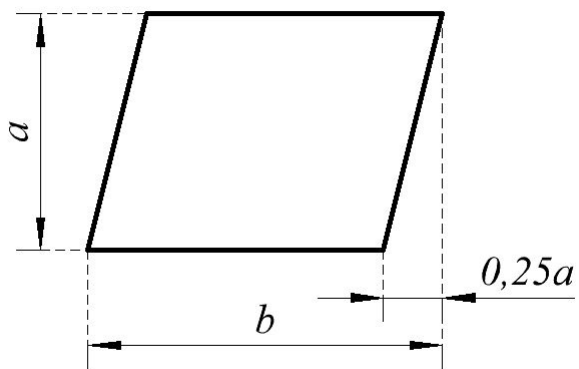
иначе вывод("нет решения")

кон алг Нахождение частного двух чисел

Способы описания алгоритмов

- **Блок-схема** – описание структуры алгоритма с помощью геометрических фигур с линиями-связями, показывающими порядок выполнения. Этот способ имеет ряд преимуществ. Благодаря наглядности, он обеспечивает «читаемость» алгоритма и явно отображает порядок: выполнения отдельных команд. В блок-схеме каждой формальной конструкции соответствует определенная

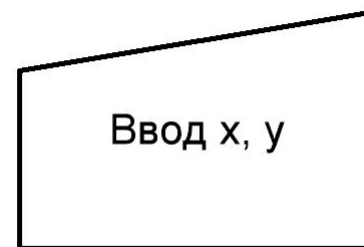
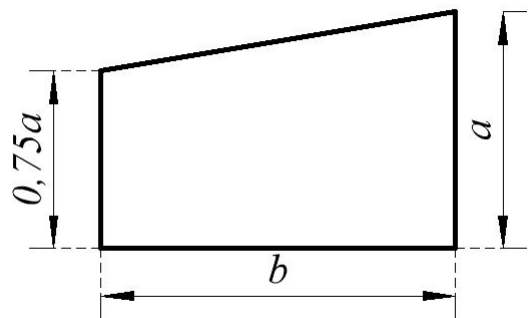
Способы описания алгоритмов



Символ отображает ввод/вывод данных, носитель данных не определен.

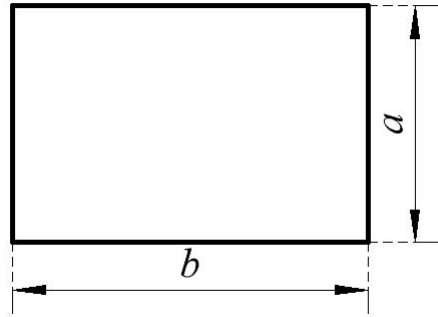
В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно

Способы описания алгоритмов



Символ отображает данные, вводимые вручную во время обработки с устройств любого типа (клавиатура, переключатели, кнопки, световое перо, полоски со штриховым кодом).

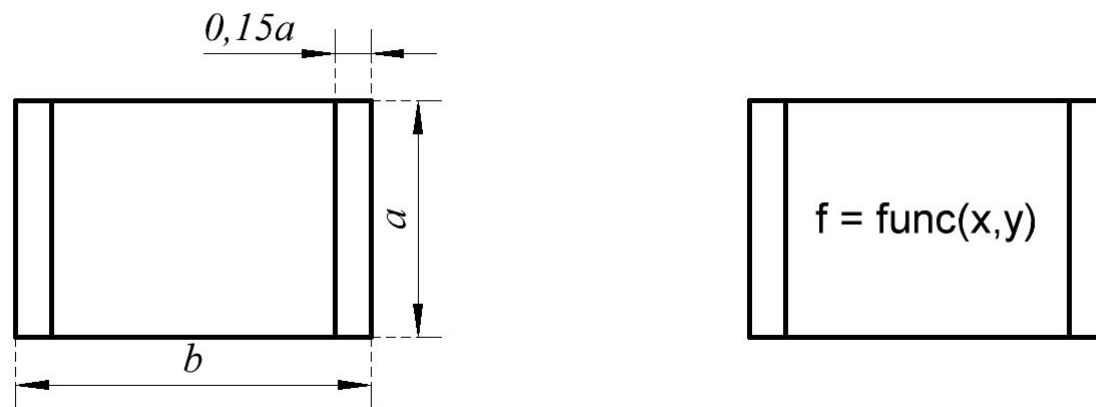
Способы описания алгоритмов



$$\begin{aligned}x &= 2 \\ y &= x - 4\end{aligned}$$

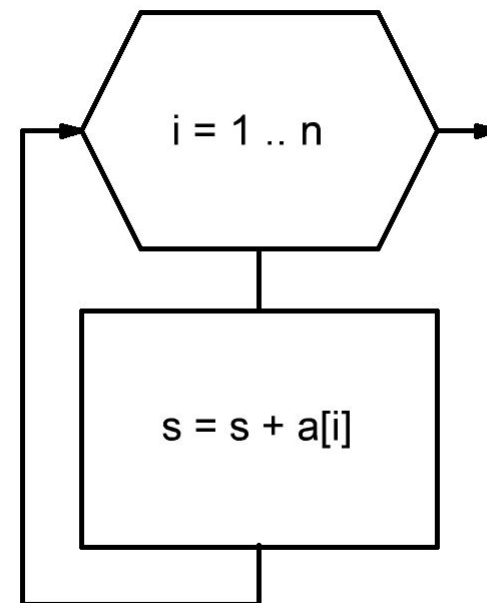
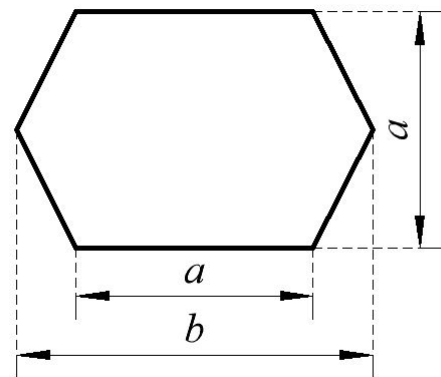
Символ изображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации). Символы обычно представляют собой блок, одно из которых является жестко заданным и привязанным к потоку, а остальные являются внешними функциями.


Способы описания алгоритмов



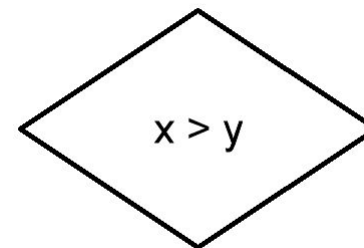
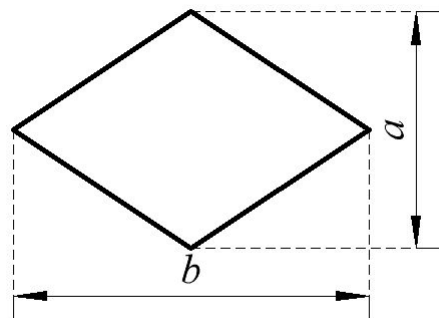
Символ отображает predetermined процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле).

Способы описания алгоритмов



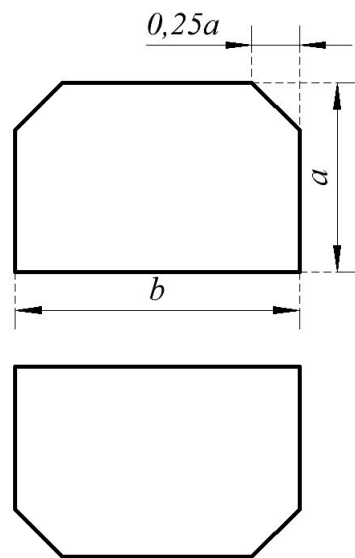
Символ  отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация адреса для задания цикла и т.д.)

Способы описания алгоритмов

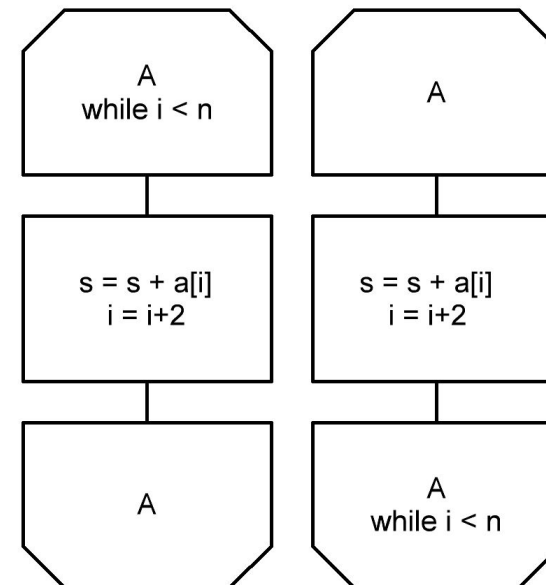


Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа.

Способы описания алгоритмов

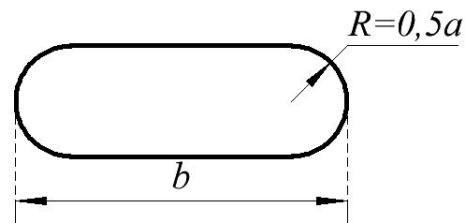


Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют



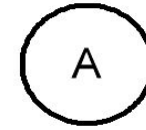
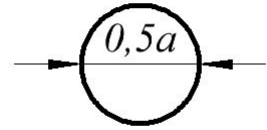
завершения. Они помещаются внутри символа и идентифицируются в конце в зависимости от расположения операции, используемой для описания цикла с проверяющей операцией, пред- и постусловием.

Способы описания алгоритмов



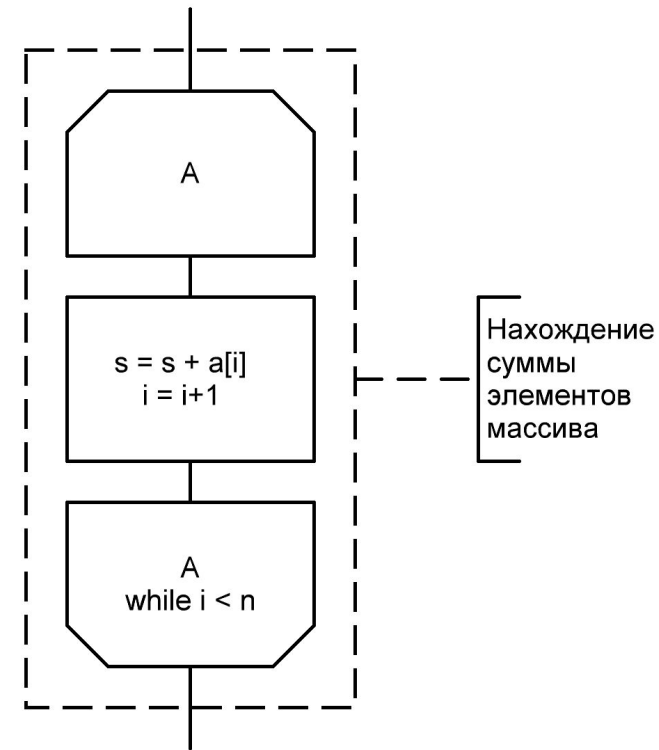
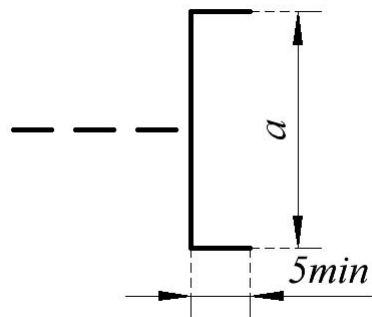
Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).

Способы описания алгоритмов



Разрыв линий потока

Способы описания алгоритмов



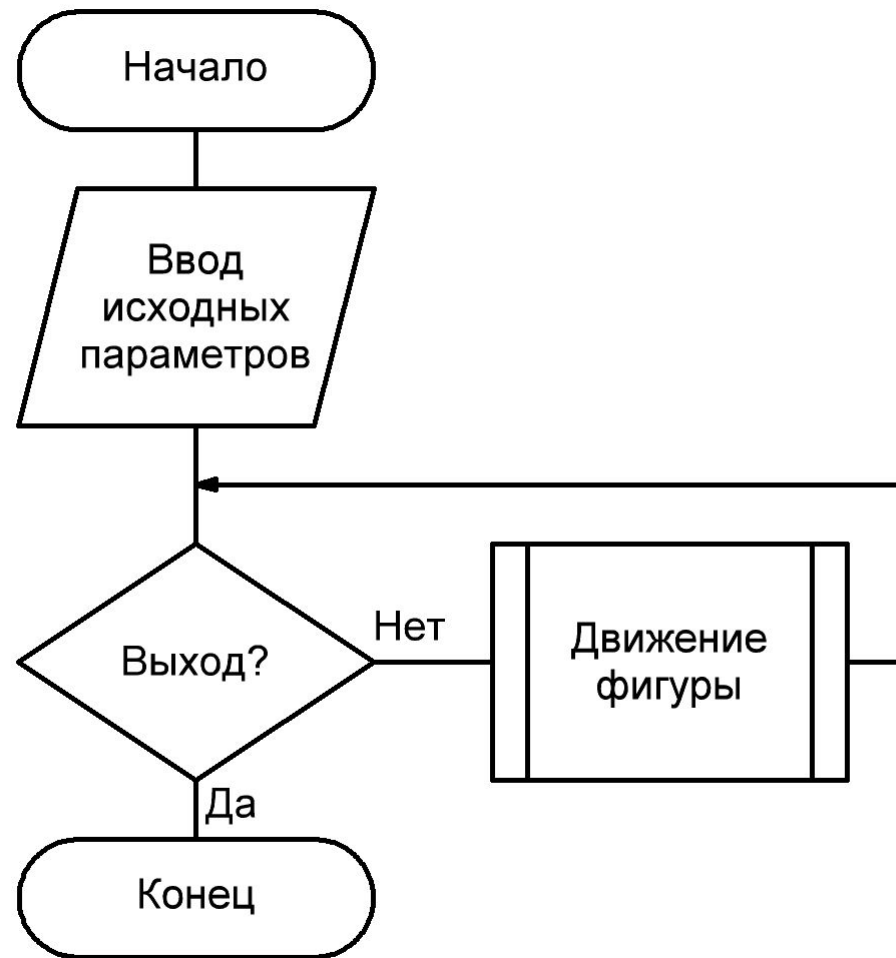
Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в

символе комментария связаны с

составления блок-схем

- При передаче управления в схеме «снизу-вверх» или «справа-налево» линии
- ~~Каждый блок~~ блок должен иметь определённое количество входов и выходов.
- Не допускается передача управления «в никуда», т. е. «источник» передачи управления и «получатель» должны быть
- ~~Блок~~ необходимо размещать таким образом, чтобы линии передачи управления не пересекались.

Способы описания алгоритмов



Способы описания алгоритмов

- **Программа** – описание структуры алгоритма на алгоритмическом языке программирования.
- На практике исполнителями алгоритмов выступают компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на «понятном» ему языке, такой формализованный язык называют языком программирования.

конструкции

- **линейные (последовательные),**
- **разветвляющиеся,**
- **циклические**
- **рекурсивные.**

Основные понятия программирования

- **Программа** – набор команд на языке, понятном исполнителю, реализующий
- **Первый алгоритм** на языке высокого уровня называется **исходным текстом** и
- **хранится в текстовом файле**. программа записанная в машинных кодах исполняемая процессором (файлы, содержащие исполняемый код имеют расширение .exe).

Этапы разработки программы

- **постановка задачи;**
- **проектирование программы;**
- **построение модели;**
- **разработка алгоритма;**
- **написание программы;**
- **отладка программы;**
- **тестирование программы;**
- **документирование.**

Отладка программы

Отладка программы - это процесс обнаружения и исправления ошибок. Программные ошибки можно разделить на два класса: синтаксические и алгоритмические (логические). Синтаксические ошибки выявляются в процессе компиляции программы - это наиболее простые с точки зрения исправления ошибки. Для обнаружения

логически ошибок требуется этап

Трансляция программы

Трансляция программы – преобразование программы, представленной на одном из языков программирования, в программу на другом языке

Транслятор – программа или техническое средство, выполняющее трансляцию

Обратная трансляция – преобразование программы, представленной на одном из языков программирования, в коды на машинно-ориентированном языке, которые

Трансляция программы

Интерпретация – процесс
непосредственного покомандного
выполнения программы, «на лету».

программирования

- **Структурная:** программа представляет собой структуру, построенную из трёх
- **Функциональная** как конструкция вычисление результатов функций от исходных данных и результатов других функций,
- **Безопасная** состоит из программирование основанное на автоматическом доказательстве теорем (логического вывода информации на основе заданных

программирования

- **Процедурная:** программирование основано на концепции вызова процедуры. Процедуры, алгоритмические блоки содержащие последовательность шагов. В ходе выполнения программы любая процедура может быть вызвана из любой точки, включая саму данную процедуру.

программирования

- **Объектно-ориентированная:** разработка совокупности объектов, указывающих друг другу что делать. Объект содержит как данные так и процедуры их обработки
- **Аспектно-ориентированная.** основана на идее разделения кода по функциональности.

Типы языков

- *Алгоритмические языки*
- *Скриптовые (сценарные) имеющие дело с готовыми компонентами.*
 - *Командно-сценарные batch language языки командных оболочек (sh, VB Script).*
 - *Прикладные сценарные языки управляют документами или приложениями (JavaScript, VBA).*
- *Языки разметки (тегированные языки). Встраивание специальных кодов (тегов) в обычный текст для структурирования,*