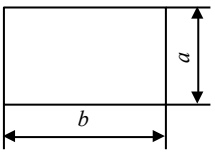
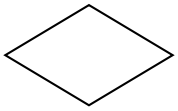
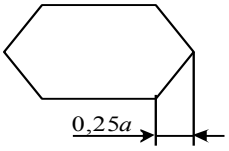
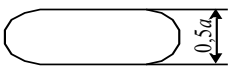
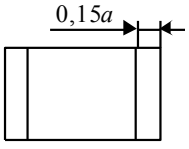
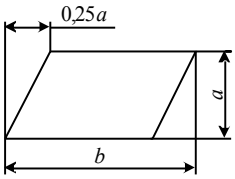
## Блок-схема алгоритма

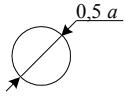
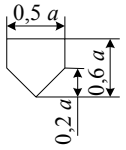
Блок-схема алгоритма – это графический способ его представления с элементами словесной записи. Каждое предписание алгоритма изображается с помощью плоской геометрической фигуры – блока. Отсюда название: блок-схема. Переходы от предписания к предписанию изображаются линиями связи – линиями потоков информации, а направление переходов – стрелками. Различным по типу выполняемых действий блокам соответствуют различные геометрические фигуры. Приняты определенные стандарты графических изображений блоков (таблица).

На территории Российской Федерации действует единая система программной документации (ЕСПД), частью которой является Государственный стандарт - ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем». Рассматриваемый ГОСТ практически полностью соответствует международному стандарту ISO 5807:1985.

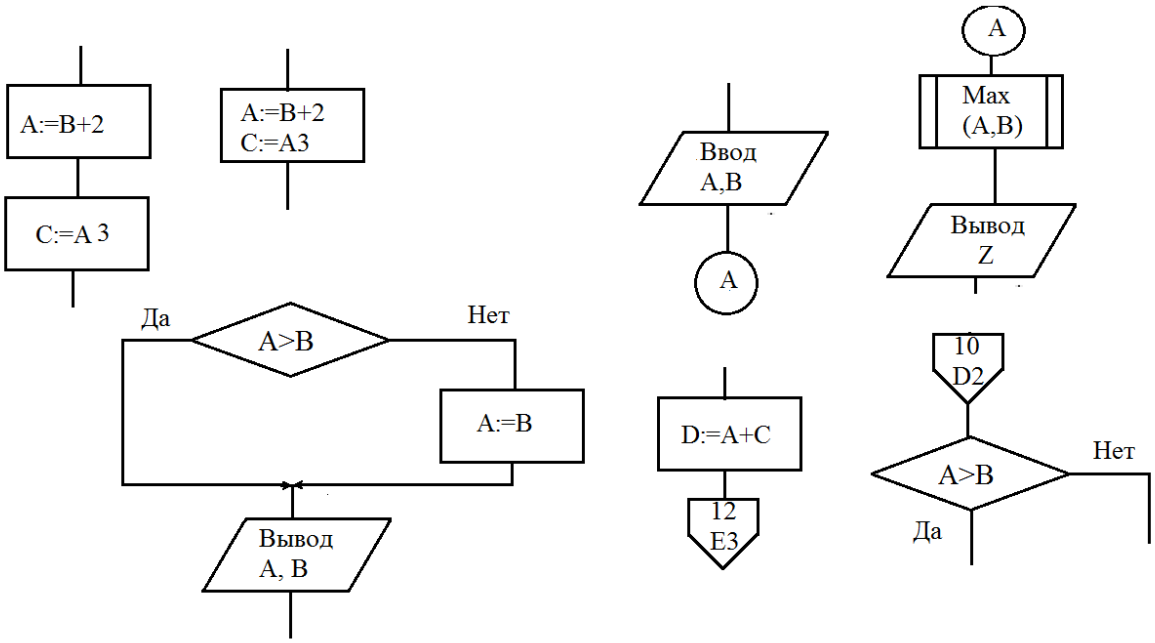
Общие правила построения блок-схем алгоритмов.

1. Для конкретизации содержания блока и уточнения выполняемого действия внутри блока помещаются краткие пояснения – словесные записи с элементами общепринятой математической символики

| Наименование символа | Обозначение и размеры | Функция |
|---|---|---|
| Процесс(вычислительный блок) |  | Выполнение операции или группы операций, в результате которых изменяются значение, форма представления или расположение данных |
| Решение (логический блок) |  | Выбор направления выполнения алгоритма в зависимости от некоторых условий |
| Модификация (заголовок цикла) |  | Выполнение операций по управлению циклом – повторением команды или группы команд алгоритма |
| Пуск-останов (начало-конец) |  | Начало или конец выполнения программы или подпрограммы |
| Предопределенный процесс (вызов подпрограммы) |  | Вызов и использование ранее созданных и отдельно описанных алгоритмов (подпрограмм) |
| Ввод/вывод |  | Общее обозначение ввода или вывода данных в алгоритме безотносительно к внешнему устройству |

| Наименование символа | Обозначение и размеры | Функция |
|---|---|---|
| Соединитель |   | Указание прерванной связи между блокам в пределах одной страницы |
| Межстраничный соединитель |   | Указание прерванной связи между блоками, расположенными на разных листах |

2. Основное направление потока информации в схемах может не отмечаться стрелками. Основное направление – сверху вниз и слева направо. Если очередность выполнения блоков не соответствует этому направлению, то возможно применение стрелок.




3. По отношению к блоку линии могут быть входящими и выходящими. Количество входящих линий принципиально не ограничено. Количество выходящих линий регламентировано и зависит от типа блока. Например, логический блок должен иметь не менее двух выходящих линий, каждая из которых соответствует одному из возможных направлений вычислений. Блок модификации должен иметь две выходящие линии, одна соответствует повторению цикла, вторая – его окончанию.

4. Допускается разрывать линии потока информации, размещая на обоих концах разрыва специальный символ «соединитель». В пределах одной страницы используется символ обычного соединителя, во внутреннем поле которого помещается маркировка разрыва либо отдельной буквой, либо буквенно-цифровой координатой блока, к которому подходит линия потока. Если схема располагается на нескольких листах, переход линий потока с одного листа на другой обозначается

с помощью символа «межстраничный соединитель». При этом на листе с блоком-источником соединитель содержит номер листа и координаты блока-приемника, а на листе с блоком-приемником – номер листа и координаты блока-источника.

5. Нумерация блоков осуществляется либо в левом верхнем углу блока в разрыве его контура, либо рядом слева от блока. Принцип нумерации может быть различным, наиболее простой – сквозная нумерация. Блоки начала и конца не нумеруются.

6. Для блоков приняты следующие размеры: a = 10, 15, 20 мм; b = 1,5a. Если необходимо увеличить размер блока, то допускается увеличение на число, кратное пяти. Необходимо выдерживать минимальное расстояние 3 мм между параллельными линиями потоков и 5 мм между остальными символами.

С помощью блок-схем можно изображать самые различные алгоритмы, например, линейной, разветвляющейся и циклической структур.

Блок-схемы являются исключительно простым и наглядным способом представления алгоритмов. Их очень полезно использовать при разработке общей структуры алгоритма, чтобы отчетливо представить себе алгоритм в целом и проследить все логические связи между его отдельными частями, проверить все ли возможные варианты решения поставленной задачи нашли в нем отражение. Блок-схемы не накладывают никаких ограничений на степень детализации в изображении алгоритма. Степень детализации определяется программистом. Однако следует помнить, что схемы общего характера мало информативны, а излишне подробные схемы проигрывают в наглядности. При разработке сложных алгоритмов, чтобы понять сущность выполняемых действий и выявить основные связи, алгоритм записывается в виде общей схемы, состоящей из крупных блоков. Блоки соответствуют основным шагам алгоритма (вводу данных, обработке, выводу данных). Затем крупные блоки разбивают на более мелкие, составляя более детальные схемы, позволяющие проверить их логическую правильность.

**Псевдокод**

**Псевдоко́д -** компактный, зачастую неформальный язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные для понимания алгоритма подробности и специфический синтаксис. Предназначен для представления алгоритма человеку, а не для компьютерной трансляции и последующего исполнения программы.

Главная цель использования псевдокода -обеспечить понимание алгоритма человеком, сделать описание более воспринимаемым, чем исходный код на языке программирования. Псевдокод широко используется в учебниках и научно-технических публикациях, а также на начальных стадиях разработки компьютерных программ. Блок-схемы можно рассматривать как графическую альтернативу псевдокоду.

В отличие от языков программирования, на синтаксис псевдокода не устанавливается стандартов, и автор может применять свой оригинальный псевдокод. На практике обычно заимствуют нужные им конструкции из одного или

нескольких широко известных и распространённых языков программирования. Сейчас обычно заимствуют элементы синтаксиса таких языков как Паскаль, Си, Python, Java, а в более старых публикациях часто использовался Алгол.

Из псевдокода исключаются технические элементы, такие как описание переменных, системно-зависимый код, операции выделения и освобождения памяти, если только они не являются существенными элементами рассматриваемого алгоритма. Математические выражения часто включаются в псевдокод в том виде, как их принято записывать в математике, а не в языках программирования, а некоторые фрагменты псевдокода могут быть фразами естественного языка (русского, английского и т. д.).

Таким образом - псевдокод это неформальное высокоуровневое описание компьютерной программы или алгоритма. Чтобы написанная вами программа могла выполняться, псевдокод должен быть преобразован в обычный код, написанный на каком-либо языке программирования.

Для начального приобретения навыков написания псевдокода можно использовать псевдокод, описанный в [XX], основные служебные слова которого приведены в таблице 1.

| | | | | |
|---|---|---|---|---|
| **алг** (алгоритм) | **сим** (символьный) | **дано** | **для** | **да** |
| **арг** (аргумент) | **лит** (литерный) | **надо** | **от** | **нет** |
| **рез** (результат) | **лог** (логический) | **если** | **до** | **при** |
| **нач** (начало) | **таб**(таблица) | **то** | **знач** | **выбор** |
| **кон** (конец) | **нц** (начало цикла) | **иначе** | **и** | **ввод** |
| **цел** (целый) | **кц** (конец цикла) | **все** | **или** | **вывод** |
| **вещ** (вещественный) | **длин** (длина) | **пока** | **не** | **утв** |

Пример. Написать псевдокод.
Даны два произвольных числа. Найти максимальное число.
**алг** Поиск максимального числа.
**вещ** a,b
**ввод** a,b \\ ввод числе a и b.
**если** a ≠ b **то**
**если** a <b **то** рез «b-максимальное число»
**иначе рез** «a-максимальное число»
**иначе рез** «числа равны».
**кон**

Пример.
Написать псевдокод нахождения суммы всех элементов заданной матрицы.
**алг** Сумма элементов
**цел** N,M
**вещ** a, S
**ввод** массив a из N*M элементов
S=0
**цикл от** i=1 **до** N \\ суммирование элементов матрицы

**нц**
**цикл от** j =1 **до** M
**нц**
S=S+a[i.j]
**кц**
**кц**
**рез** S
**кон**