

УВАЖАЕМЫЕ СТУДЕНТЫ!
ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с теорией и законспектировать лекцию не меньше трех листов, составить и ответить на вопросы.
2. Предоставит отчет конспекта лекции прислать в виде скриншота в течении трех дней .
3. Отправить преподавателю на почту v.vika2014@mail.ru и указать свою Ф.И.О, группу, и название дисциплины

Тема: Понятие блок-схемы. Основные элементы блок-схем.

Цель: ознакомиться и научиться составлять блок-схемы.

План

1. Элементы блок-схем алгоритмов
2. Примеры блок-схем

Блок-схемы алгоритмов.

Блок-схема — распространённый тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности.

Схема — это абстракция какого-либо процесса или системы, наглядно отображающая наиболее значимые части. Схемы широко применяются с древних времен до настоящего времени — чертежи древних пирамид, карты земель, принципиальные электрические схемы. Очевидно, древние мореплаватели хотели обмениваться картами и поэтому выработали единую систему обозначений и правил их выполнения. Аналогичные соглашения выработаны для изображения схем-алгоритмов и закреплены ГОСТ и международными стандартами.

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»**. Не смотря на то, что описанные в стандарте обозначения могут использоваться для изображения схем ресурсов системы, схем взаимодействия программ и т.п., в настоящей статье описана лишь разработка схем алгоритмов программ.

Рассматриваемый ГОСТ практически полностью соответствует международному стандарту **ISO 5807:1985**.


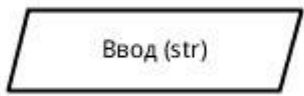
Элементы блок-схем алгоритмов


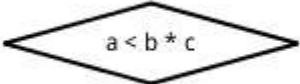
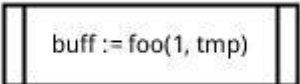
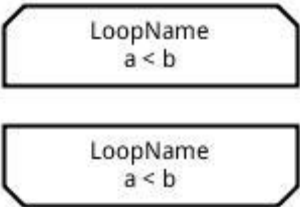
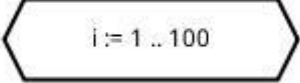

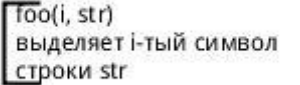
Блок-схема представляет собой совокупность символов, соответствующих этапам работы алгоритма и соединяющих их линий. Пунктирная линия используется для соединения символа с комментарием. Сплошная линия отражает зависимости по управлению между символами и может снабжаться стрелкой. Стрелку можно не указывать при направлении дуги слева направо и сверху вниз. Согласно п. 4.2.4, линии должны подходить к символу слева, либо сверху, а исходить снизу, либо справа.

Есть и другие типы линий, используемые, например, для изображения блок-схем параллельных алгоритмов, но в текущей статье они, как и ряд специфических символов, не рассматриваются. Рассмотрены лишь основные символы, которых всегда достаточно студентам.

Таблица 1

Элементы блок-схем алгоритмов

 <p>Терминатор начала и конца работы функции</p>	<p>Терминатором начинается и заканчивается любая функция. Тип возвращаемого значения и аргументов функции обычно указывается в комментариях к блоку терминатора.</p>
 <p>Операции ввода и вывода данных</p>	<p>В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно используется символ параллелограмма. Подробности ввода/вывода могут быть указаны в комментариях.</p>

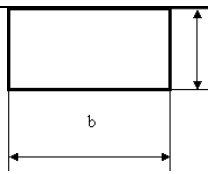
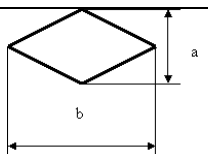
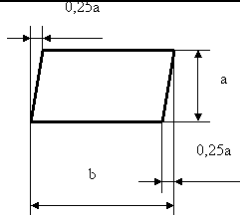
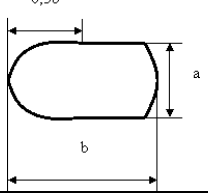
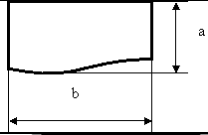

 Выполнение операций над данными	В блоке операций обычно размещают одно или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций.
 Блок, иллюстрирующий ветвление алгоритма	Блок в виде ромба имеет один вход и несколько подписанных выходов. В случае, если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения — «да/нет». Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах — значения этой переменной.
 Вызов внешней процедуры	Вызов внешних процедур и функций помещается в прямоугольник с дополнительными вертикальными линиями.
 Начало и конец цикла	Символы начала и конца цикла содержат имя и условие. Условие может отсутствовать в одном из символов пары. Расположение условия, определяет тип оператора, соответствующего символам на языке высокого уровня — оператор с предусловием (while) или постусловием (do ... while).
 Подготовка данных	Символ «подготовка данных» в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком.
 Соединитель	В случае, если блок-схема не уместится на лист, используется символ соединителя, отражающий переход потока управления между листами. Символ может использоваться и на одном листе, если по каким-либо причинам тянуть линию не удобно.
 Комментарий	Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией.

Правила оформления блок – схем алгоритмов

Ввиду того, что часть последующего материала будет представлена с использованием блок – схем алгоритмов, необходимо рассмотреть основные правила их оформления. В таблице 2 показана форма и приведено содержание наиболее часто используемых блоков. Практически все блоки, приведенные в таблице, строятся на основе «базового» прямоугольника

размерами «а х в». Первый приведенный в таблице блок под названием «процесс» по – существу является «базовым» прямоугольником. Размер а = 10, 15, 20 ... мм, т.е. кратен пяти. Размер в = 1,5а, допускается в = 2а. Размер «а» выбирается в зависимости от масштаба блок – схемы и должен быть одинаковым для всех ее блоков.

Таблица 2

Форма и содержание блоков блок – схем алгоритмов		
Наименование	Обозначение	Функции
1	2	3
Процесс		Выполнение операций присваивания, например, $A = 0$, сложения с присваиванием, например, $C = A+B$, вычитания, умножения и т.д.
Решение		Выбор направления выполнения алгоритма (программы) в зависимости от некоторых переменных условий
Ввод - вывод		Ввод – вывод информации без указания (независимо) от типа устройства ввода или вывода
Дисплей		Ввод информации с дисплея (с клавиатуры), вывод информации на дисплей
Документ		Вывод информации на бумагу (на принтер)
Пуск - останов		Начало – конец алгоритма (программы)
Соединитель		Переход на блок номер 5 (номер блока приведен для примера)
Комментарий		-

Каждая блок – схема начинается блоком «Пуск», внутри которого пишется слово «Начало», и заканчивается блоком «Останов», внутри

которого пишется слово «Конец». Блоки соединяются линиями. Если линия, соединяющая блоки «приходит» в блок по направлению «сверху - вниз» или «слева – направо», то стрелка на ее конце не ставится (подразумевается). Если же по направлению «снизу – вверх» или «справа – налево», то стрелка на ее конце ставится обязательно.

Применение блока «Соединитель» позволяет значительно упростить блок – схему. Он применяется в том случае, когда линию, соединяющую блоку нужно вести на значительное расстояние и часто с пересечением других соединительных линий. Вместо этого достаточно после блока, из которого должна выходить соединительная линия, изобразить блок «Соединитель» и внутри его написать номер блока, в который эта линия должна прийти.

Блок «Комментарий» применяется в том случае, когда внутри какого – либо блока не удастся разместить (написать) всю необходимую информацию. В этом случае к линии, соединяющей блоки, перед блоком, для которого необходимо написать дополнительную информацию, посредством пунктирной линии присоединяется (справа или слева) блок «Комментарий». Информацию можно размещать по высоте - в пределах высоты данного блока, по ширине – до края страницы.

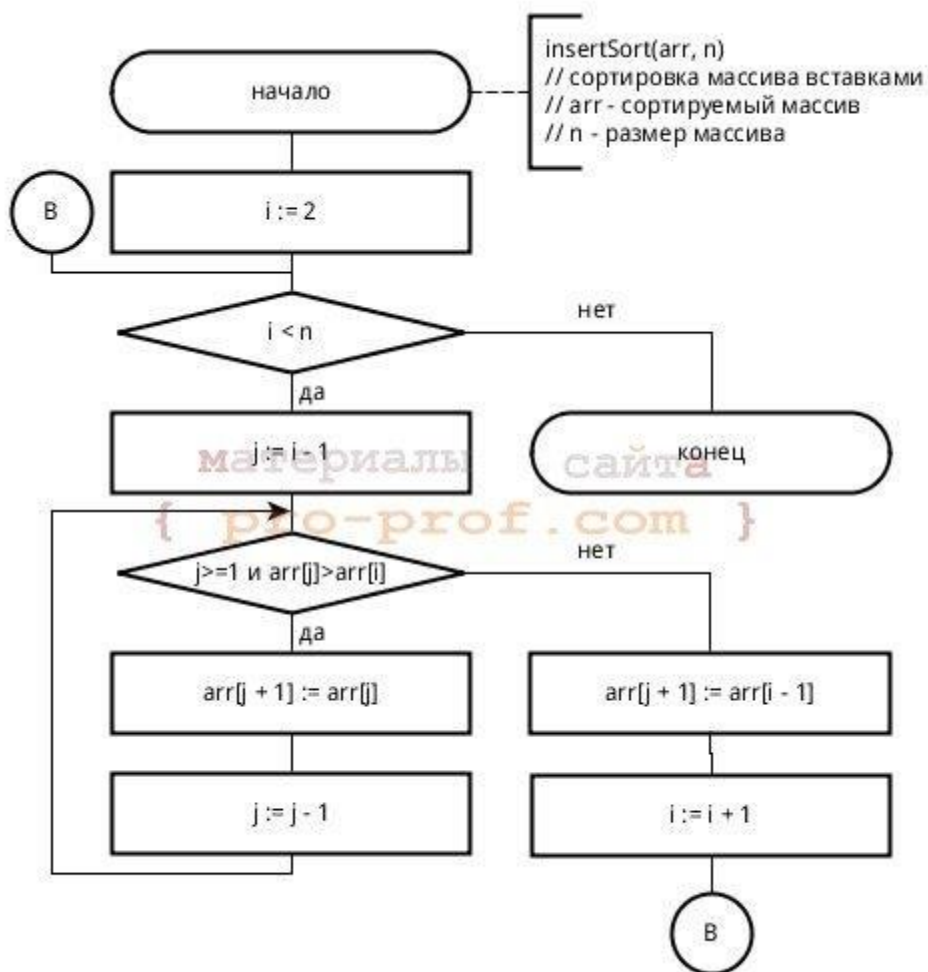
Примеры блок-схем

В качестве примеров, построены блок-схемы очень простых алгоритмов сортировки, при этом акцент сделан на различные реализации циклов, т.к. у студенты делают наибольшее число ошибок именно в этой части.

Сортировка вставками

Массив в алгоритме *сортировки вставками* разделяется на отсортированную и еще не обработанную части. Изначально отсортированная часть состоит из одного элемента, и постепенно увеличивается.

На каждом шаге алгоритма выбирается первый элемент необработанной части массива и вставляется в отсортированную так, чтобы в ней сохранялся требуемый порядок следования элементов. Вставка может выполняться как в конец массива, так и в середину. При вставке в середину необходимо сдвинуть все элементы, расположенные «правее» позиции вставки на один элемент вправо. В алгоритме используется два цикла — в первом выбираются элементы необработанной части, а во втором осуществляется вставка.



Блок-схема алгоритма сортировки вставками

В приведенной блок-схеме для организации цикла используется символ ветвления. В главном цикле ($i < n$) перебираются элементы необработанной части массива. Если все элементы обработаны — алгоритм завершает работу, в противном случае выполняется поиск позиции для вставки i -того элемента. Искомая позиция будет сохранена в переменной j в результате выполнения

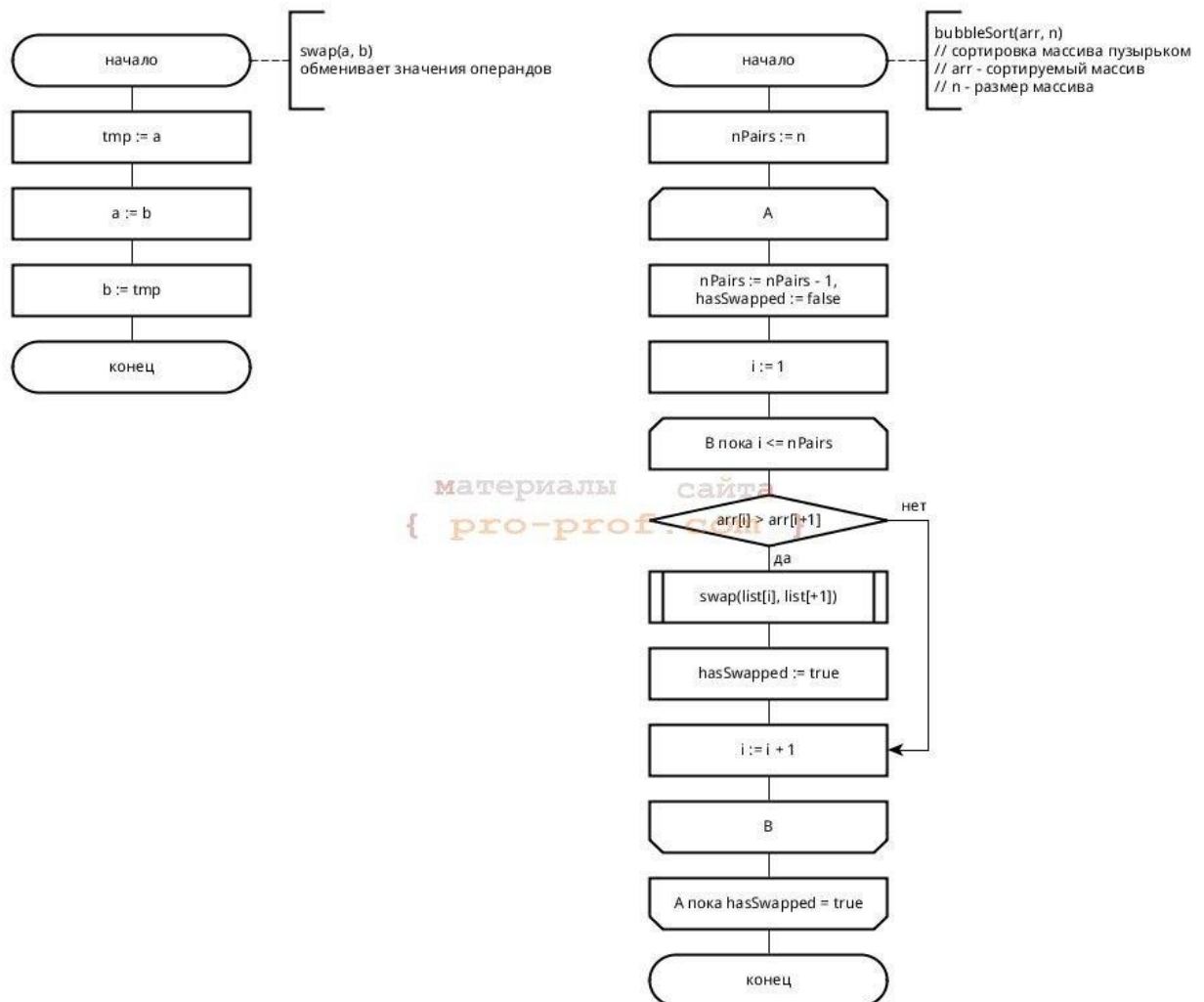
внутреннего цикла, осуществляющем сдвиг элементов до тех пор, пока не будет найден элемент, значение которого меньше *i-того*.

На **блок-схеме** показано каким образом может использоваться символ перехода — его можно использовать не только для соединения частей схем, размещенных на разных листах, но и для сокращения количества линий. В ряде случаев это позволяет избежать пересечения линий и упрощает восприятие алгоритма.

Сортировка пузырьком

Сортировка пузырьком, как и *сортировка вставками*, использует два цикла. Во вложенном цикле выполняется попарное сравнение элементов и, в случае нарушения порядка их следования, перестановка. В результате выполнения одной итерации внутреннего цикла, максимальный элемент гарантированно будет смещен в конец массива. Внешний цикл выполняется до тех пор, пока весь массив не будет отсортирован.

На блок-схеме показано использование символов начала и конца цикла. Условие внешнего цикла (A) проверяется в конце (*с постусловием*), он работает до тех пор, пока переменная *hasSwapped* имеет значение *true*. Внутренний цикл использует *предусловие* для перебора пар сравниваемых элементов. В случае, если элементы расположены в неправильном порядке, выполняется их перестановка посредством вызова *внешней процедуры* (*swap*). Для того, чтобы было понятно назначение внешней процедуры и порядок следования ее аргументов, необходимо писать комментарии. В случае, если функция возвращает значение, комментарий может быть написан к символу терминатору конца.



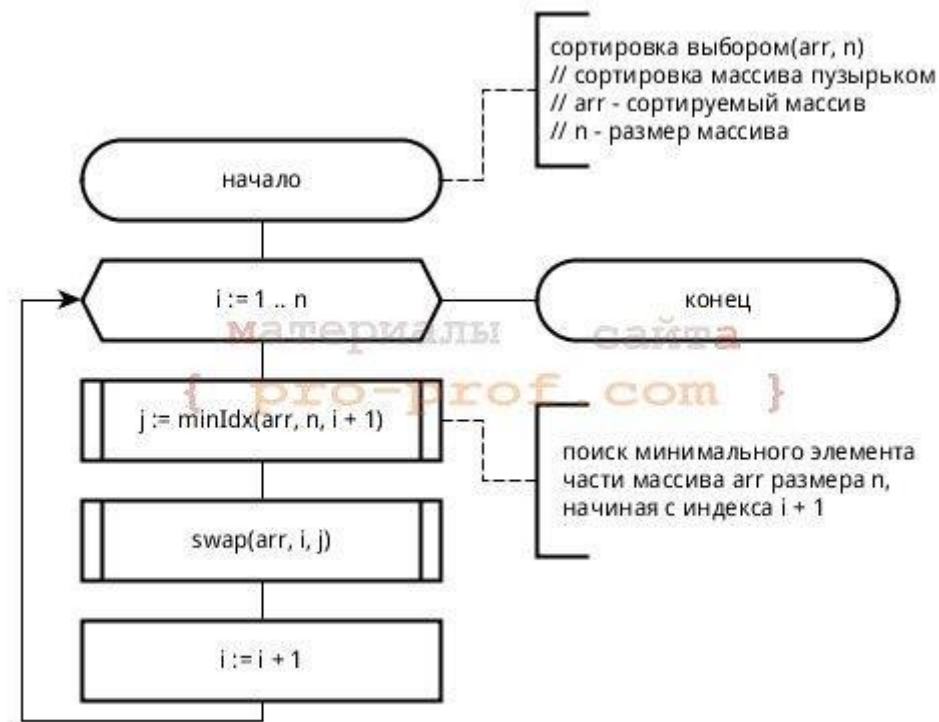
Блок-схема алгоритма сортировки пузырьком

Сортировка выбором

В *сортировке выбором* массив разделяется на отсортированную и необработанную части. Изначально отсортированная часть пустая, но постепенно она увеличивается. Алгоритм производит поиск минимального элемента необработанной части и меняет его местами с первым элементом той же части, после чего считается, что первый элемент обработан (отсортированная часть увеличивается).

На блок-схеме приведен пример использования блока «подготовка», а также показано, что в ряде случаев можно описывать алгоритм более «укрупнённо» (не вдаваясь в детали). К сортировке выбором не имеют отношения детали реализации *поиска индекса минимального элемента массива*, поэтому они могут быть описаны символом вызова внешней процедуры. Если блок-схема алгоритма внешней процедуры отсутствует, не

помешает написать к символу вызова комментарий, исключением могут быть функции с говорящими названиями типа *swap*, *sort*,



Блок-схема сортировки выбором

Часть студентов традиционно пытается рисовать блок-схемы в *Microsoft Word*, но это оказывается сложно и не удобно. Например, в *MS Word* нет стандартного блока для терминатора начала и конца алгоритма (прямоугольник со скругленными краями, а не овал). Наиболее удобными, на мой взгляд, являются утилиты *MS Visio* и *yEd*, обе они позволяют гораздо больше, чем строить блок-схемы (например рисовать диаграммы UML), но первая является платной и работает только под *Windows*, вторая бесплатная и кроссплатформенная. Все блок-схемы в этой статье выполнены с использованием *yEd*.