

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Куижева Саида Казбековна
Должность: Ректор
Дата подписания: 03.08.2023 23:19:06
Уникальный программный ключ:
71183e1134ef9cfa69b206d480271b3c1a975e6f

МИНОБРНАУКИ РОССИИ

**Политехнический колледж филиала федерального государственного бюджетного
образовательного учреждения высшего образования «Майкопский государственный
технологический университет» в поселке Яблоновском**

МАТЕРИАЛЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

по теме: Алгоритмы и программы.

по дисциплине

ОП.16 Введение в специальность

специальность

09.02.07 Информационные системы и программирование

форма обучения

очная

квалификация выпускника

программист

Яблоновский, 2020

УДК 004.421 (07)

ББК 73

М-34

Одобрено предметной (цикловой) комиссией
информационных и математических дисциплин

Разработчик: Хуаде Р.А. - преподаватель первой категории политехнического колледжа филиала федерального государственного бюджетного образовательного учреждения высшего образования «Майкопский государственный технологический университет» в поселке Яблоновском

Алгоритмы и программы.

Блок-схемы алгоритмов. ГОСТ. Примеры

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»** [1]. Несмотря 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**.

Содержание:

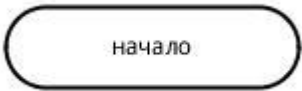
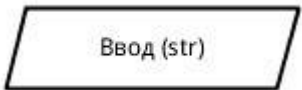
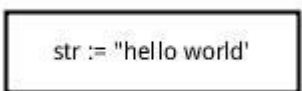
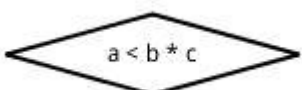
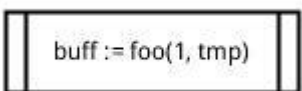
1. Элементы блок-схем алгоритмов
2. Примеры блок-схем
3. Нужны ли блок-схемы? Альтернативы

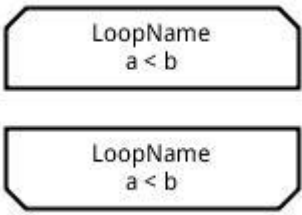
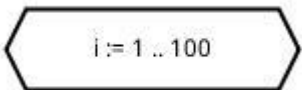

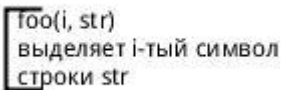
Элементы блок-схем алгоритмов

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.

 <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>
 <p>Выполнение операций над данными</p>	<p>В блоке операций обычно размещают одну или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций.</p>
 <p>Блок, иллюстрирующий ветвление алгоритма</p>	<p>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.</p>
	<p>Вызов внешних процедур и функций помещается в прямоугольник с дополнительными вертикальными линиями.</p>

<p>Вызов внешней процедуры</p>	
 <p>Начало и конец цикла</p>	<p>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).</p>
 <p>Подготовка данных</p>	<p>Символ «подготовка данных» в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком.</p>
 <p>Соединитель</p>	<p>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.</p>
 <p>Комментарий</p>	<p>Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией.</p>

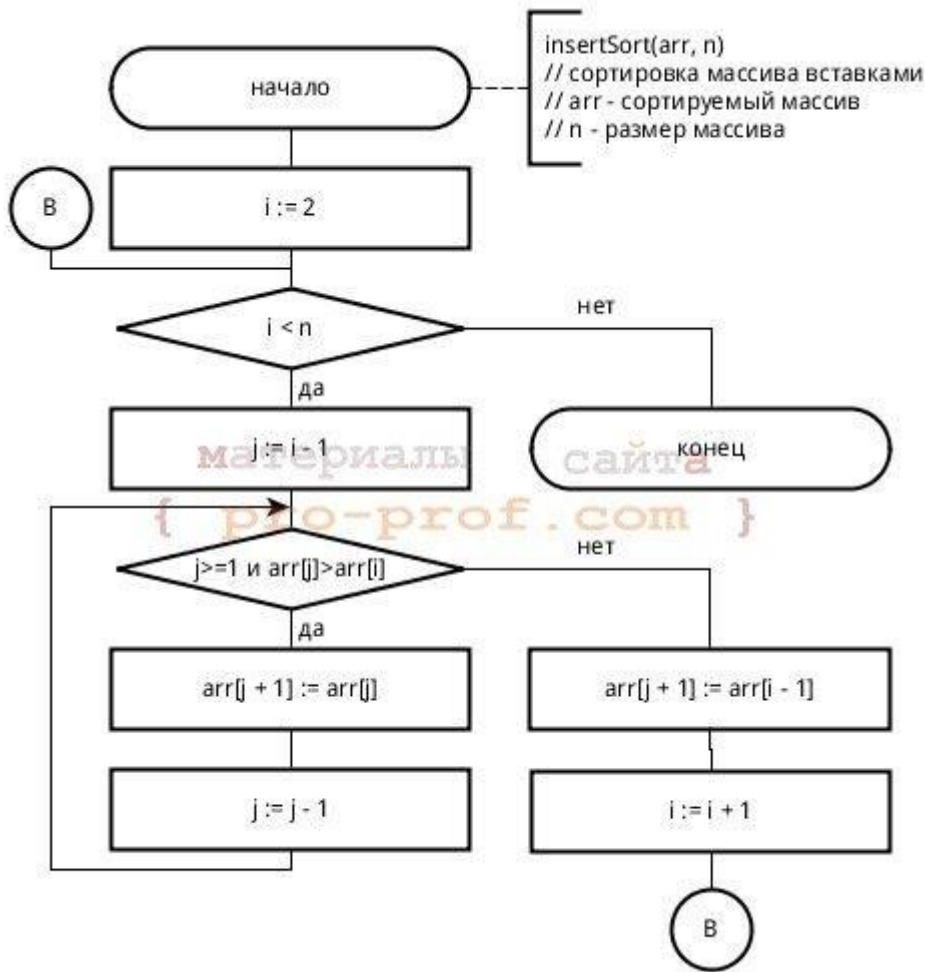
Примеры блок-схем

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.

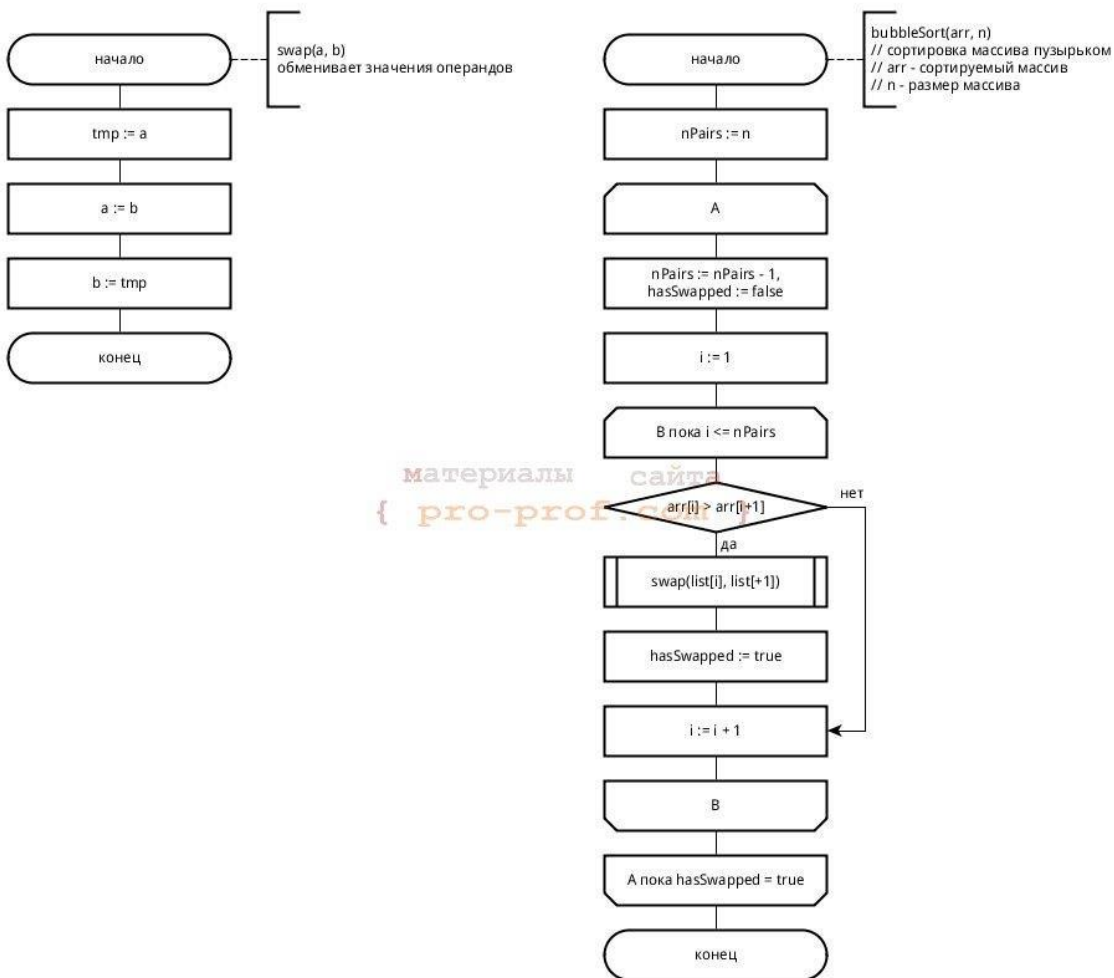


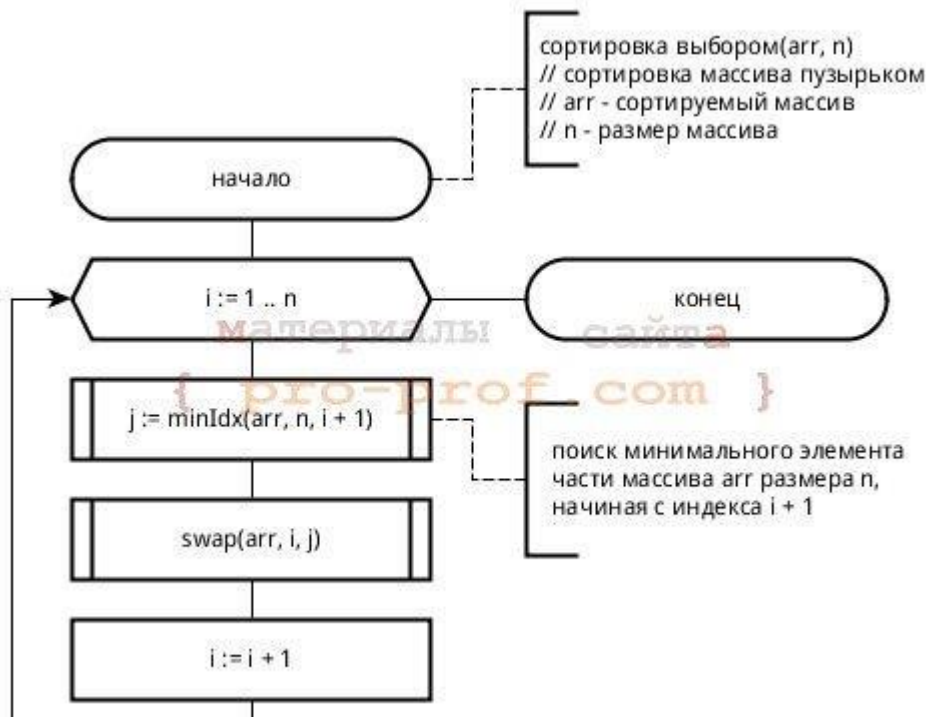
схема алгоритма сортировки пузырьком

Блок-

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.

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).



Блок-схема сортировки выбором

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,

На блоге можно найти другие примеры блок-схем:

- блок-схема проверки правильности расстановки скобок арифметического выражения ;
- блок-схемы алгоритмов быстрой сортировки и сортировки слиянием .

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 [5], 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.

Нужны ли блок-схемы? Альтернативы

Частные конторы никакие блок-схемы не используют, в книжках по алгоритмам вместо них применяют словесное описание (псевдокод) как более краткую форму. Возможно блок-схемы применяют на государственных предприятиях, которые должны оформлять документацию согласно требованиям ЕСПД, но есть

сомнения — даже для регистрации программы в Государственном реестре программ для ЭВМ никаких блок-схем не требуется.

Тем не менее, рисовать блок-схемы заставляют школьников (примеры из учебников ГОСТ не соответствуют) — выносят вопросы на государственные экзамены (ГИА и ЕГЭ), студентов — перед защитой диплом сдается на нормоконтроль, где проверяется соответствие схем стандартам.

Разработка блок-схем выполняется на этапах проектирования и документирования, согласно каскадной модели разработки ПО, которая сейчас почти не применяется, т.к. сопровождается большими рисками, связанными с ошибками на этапах проектирования.

Появляются подозрения, что система образования прогнила и отстала лет на 20, однако аналогичная проблема наблюдается и за рубежом. Международный стандарт ISO 5807:1985 мало чем отличается от ГОСТ 19.701-90, более нового стандарта за рубежом нет. Там же производится множество программ для выполнения этих самых схем — Dia, MS Visio, yEd, ..., а значит списывать их не собираются. Вместо блок-схем иногда применяют диаграммы деятельности UML, однако удобнее они оказываются, разве что при изображении параллельных алгоритмов.

Периодически поднимается вопрос о том, что ни блок-схемы, ни UML не нужны, да и документация тоже не нужна. Об этом твердят программисты, придерживающиеся методологии экстремального программирования (XP), хотя даже в их кругу нет единого мнения.

В ряде случаев, программирование невозможно без рисования блок-схем, т.к. это один процесс — существуют визуальные языки программирования, такие как ДРАКОН, кроме того, блок-схемы используются для верификации алгоритмов (формального доказательства их корректности) методом индуктивных утверждений Флойда.

В общем, единого мнения нет. Очевидно, есть области, в которых без чего-то типа блок-схем обойтись нельзя, но более гибкой альтернативы нет. Для формальной верификации необходимо рисовать подробные блок-схемы, но для

проектирования и документирования такие схемы не нужны — я считаю разумным утверждение экстремальных программистов о том, что нужно рисовать лишь те схемы, которые помогают в работе и не требуют больших усилий для поддержания в актуальном состоянии.

Задачи на составление блок-схем алгоритмов

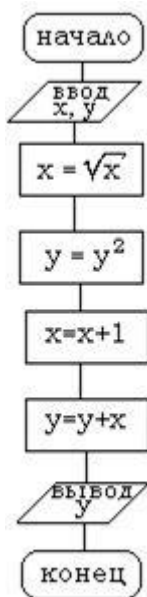
Основные блоки



Задачи определения результата выполнения алгоритма по заданной блок-схеме.

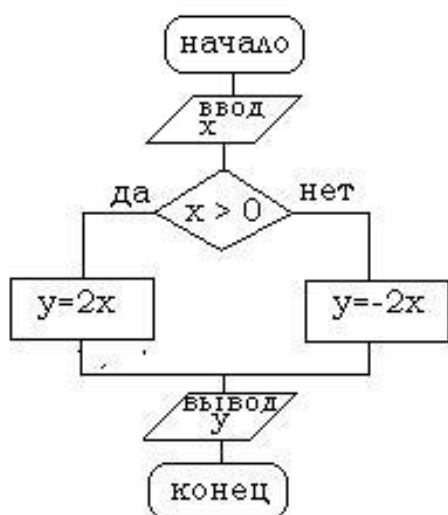
Задача 1.

Дана блок-схема алгоритма



Определить результат выполнения алгоритма при определённых значениях исходных данных
Например, при $x=16$ и $y=2$

Задача 2.



Дана блок-схема алгоритма

Определить результат выполнения алгоритма при определённых значениях исходных данных
Например, при $x=-6$ или $x=0$ или $x=7$

Задача 3.

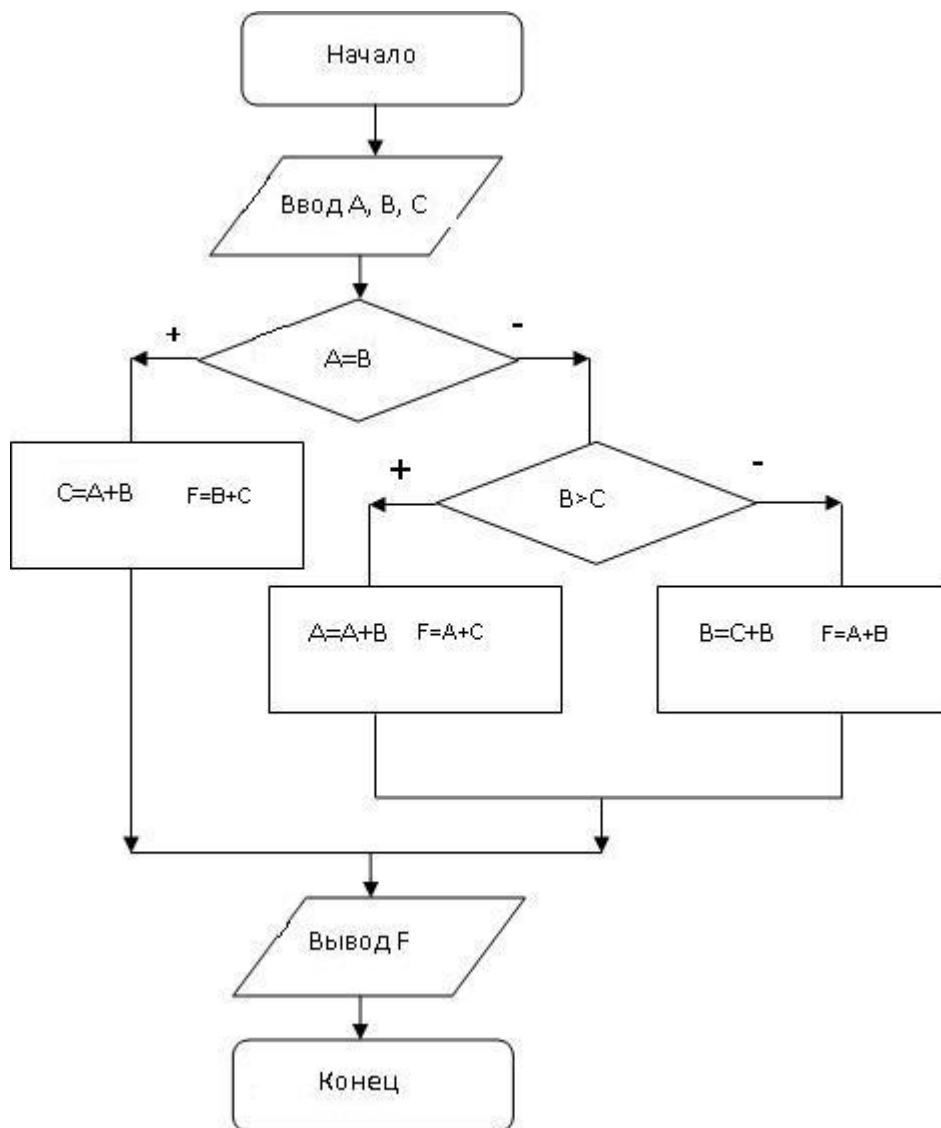
Дана блок-схема алгоритма



Определить результат выполнения алгоритма при определённых значениях исходных данных
Например, при $n=15$ или $n=0$ или $n=-7$

Задача4.

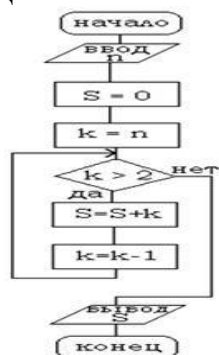
Дана блок-схема алгоритма



Определить результат выполнения алгоритма при определённых значениях исходных данных
 Например, при $A=7; B=8; C=9$ или $A=6; B=6; C=-10$ или $A=6; B=10; C=-10$

Задача5.

Дана блок-схема алгоритма



Определить результат выполнения алгоритма при определённых значениях исходных данных
 Например, при $n=4$ или $n=1$

задание 1. Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h).

Запись решения задачи на алгоритмическом языке:

алг трапеция

вещ a,b,h,s

нач

ввод f,b,h

$s:=((a+b)/2)*h$

ВЫВОД s

кОН

Запись алгоритма в виде блок-схемы (рис. 1):

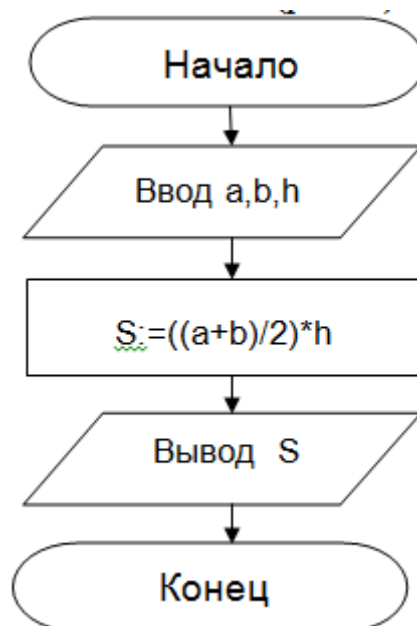


Рисунок 1. Блок-схема линейного алгоритма

Задание 2. Определить среднее арифметическое двух чисел, если a положительное и частное (a/b) в противном случае.

Запись решения задачи на алгоритмическом языке:

алг числа

вещ a,b,c

нач

ввод a,b

если a>0

то c:=(a+b)/2

иначе c:=a/b

все

вывод c

кон

Запись алгоритма в виде блок-схемы (рис. 2):

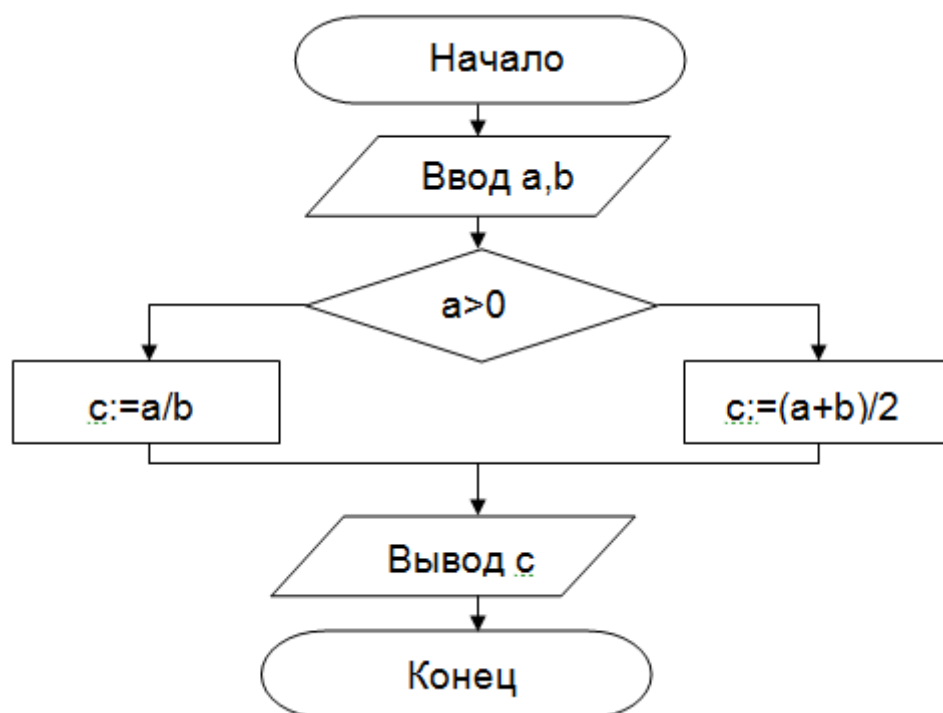


Рисунок 2. Блок-схема алгоритма с ветвлением

Задание 3. Составить алгоритм нахождения суммы целых чисел в диапазоне от 1 до 10.

Запись решения задачи на алгоритмическом языке:

алг сумма

вещ a,s

нач

 S:=0;

 A:=1;

нц

пока a<=10

 S:=S+a;

 A:=a+1;

кц

вывод S

кон

Запись алгоритма в виде блок-схемы (рис. 3):

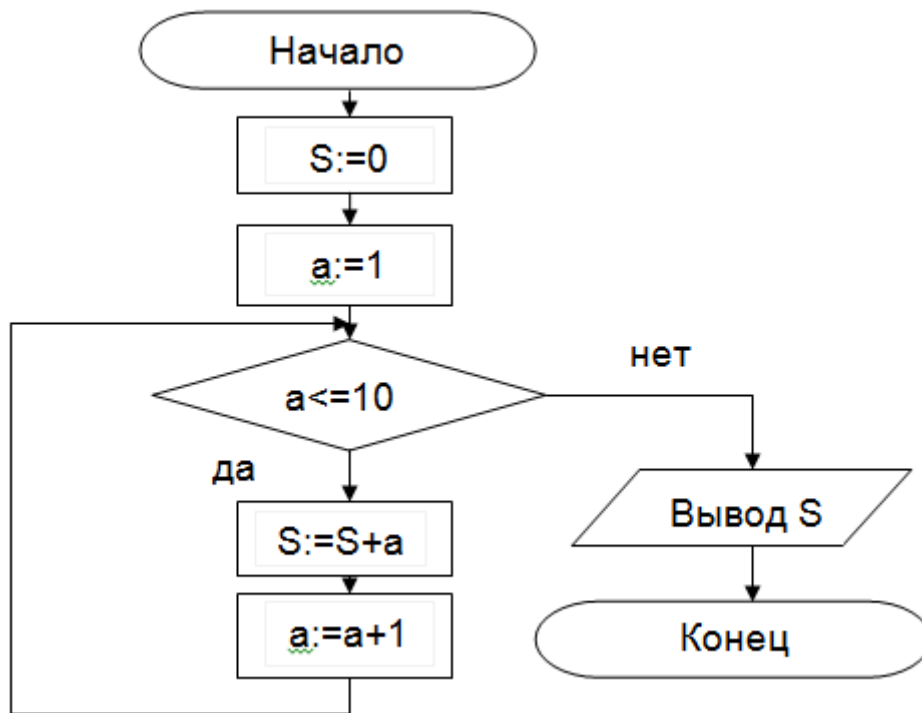


Рисунок 3. Циклический алгоритм с предусловием

В алгоритме с постусловием сначала выполняется тело цикла, а затем проверяется условие окончания цикла. Решение задачи нахождения суммы первых десяти целых чисел в данном случае будет выглядеть следующим образом:

алг сумма

вещ a,s

нач

S:=0;

A:=1;

нц

S:=S+a;

A:=a+1;

пока a<=10

кц

Вывод S

КОН

Запись алгоритма в виде блок-схемы (рис. 4):

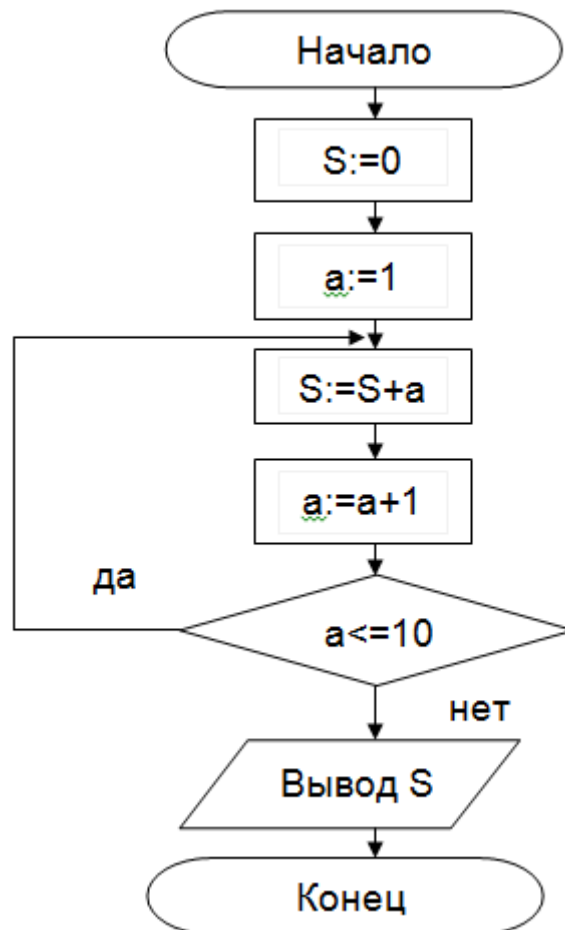


Рисунок 4. Циклический алгоритм с постусловием