

Лекция №1: Основы алгоритмизации

Преподаватель: Чишиев Эльдар
Рафаэлевич

Понятие алгоритма. Свойства алгоритма.

Алгоритм - это последовательность действий, приводящих к требуемому результату.

Таким образом, при разработке алгоритма решения задачи математическая формулировка преобразуется в процедуру решения, представляющую собой последовательность арифметических действий и логических связей между ними. При этом алгоритм обладает следующими **свойствами**:

1. **Дискретность** - процесс преобразования данных, т.е. на каждом шаге алгоритма выполняется очередная одна операция;
2. **Результативность** - алгоритм должен давать некоторый результат;
3. **Конечность** - алгоритм должен давать результат за конечное число шагов;
4. **Определенность** - все предписания алгоритма должны быть однозначны, понятны пользователю;
5. **Массовость** - алгоритм должен давать решения для целой группы задач из некоторого класса, отличающихся исходными

Понятие алгоритма. Свойства алгоритма.

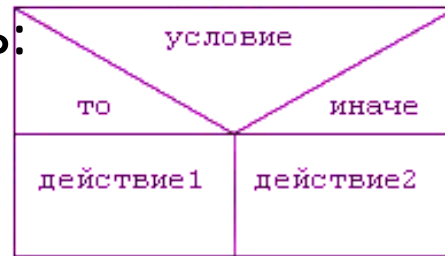
Действия в алгоритме выполняются в порядке их записи. Нельзя менять местами никакие два действия алгоритма, а так же нельзя не закончив одного действия переходить к следующему.

Для записи алгоритмов используются специальные языки:

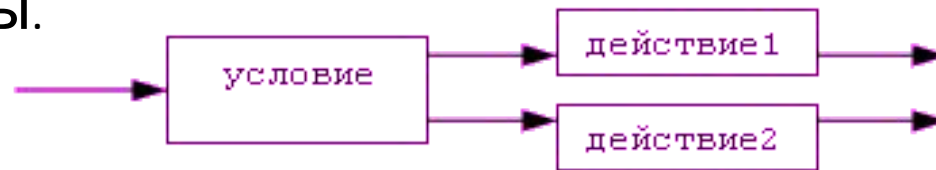
1. Естественный язык (словесная запись). Запись алгоритма происходит с помощью словесных слов:
2. **если** условие **то** действие1 **иначе** действие2
3. Формулы.
4. Псевдокод.

Понятие алгоритма. Свойства алгоритма.

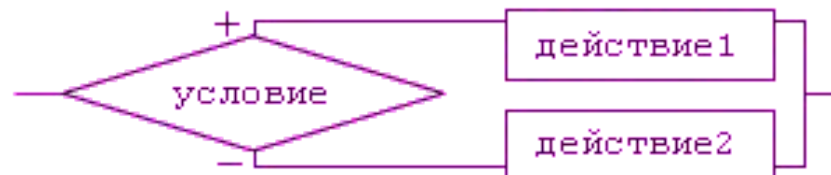
4. Структурограммы. Используется структурированная словесная запись:



5. Синтаксические диаграммы.

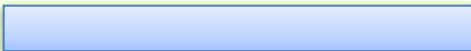
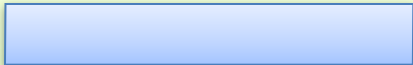
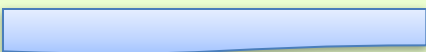
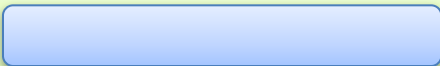
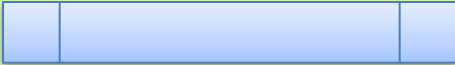
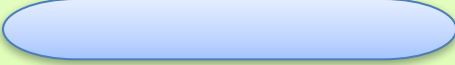


6. Графический (язык блок-схем).



Составление блок-схем

Составление блок-схем демонстрирует следующая таблица:

НАЗВАНИЕ	РИСУНОК	ВЫПОЛНЯЕМАЯ ФУНКЦИЯ
1. Блок вычислений		Выполняет вычислительное действие или группу действий
2. Логический блок		Выбор направления выполнения алгоритма в зависимости от условия
3. Блоки ввода/вывода		Ввод или вывод данных вне зависимости от физического носителя
		Вывод данных на печатающее устройство
4. Начало/конец		Начало или конец программы, вход или выход в подпрограмму
5. Предопределенный процесс		Вычисления по стандартной или пользовательской подпрограмме
6. Блок модификации		Выполнение действий, изменяющих пункты алгоритма
7. Соединитель		Указание связи между прерванными линиями в пределах одной страницы
8. Межстраничный соединитель		Указание связи между частями схемы, расположенной на разных страницах

Составление блок-схем

Блок-схема выстраивается в одном направлении: либо сверху вниз, либо слева направо. Все повороты соединительных линий выполняются под углом 90 градусов.

Общими правилами при проектировании визуальных алгоритмов (блок-схем) являются следующие:

- 1) В начале алгоритма должны быть блоки ввода значений входных данных.
- 2) После ввода значений входных данных могут следовать блоки обработки и блоки условия.
- 3) В конце алгоритма должны располагаться блоки вывода значений выходных данных.
- 4) В алгоритме должен быть только один блок начала и один блок окончания.

Связи между блоками указываются направленными или ненаправленными линиями.

Составление блок-схем

Описание на языке блок-схем, в общем случае, применимо к любому целенаправленному действию (не обязательно вычислению). Зачастую оно не является полностью формализованным (и поэтому не может непосредственно использоваться компьютером), но оно очень хорошо читаемо, его легко модифицировать и, главное, оно естественно отражает сущность процесса алгоритмизации задачи.

Этапы решения задач на ЭВМ

Решение задач на ЭВМ разбивается на следующие этапы:

- 1) Постановка задачи
- 2) Формализация (математическая постановка)
- 3) Выбор (или разработка) метода решения
- 4) Разработка алгоритма
- 5) Составление программы
- 6) Отладка программы.
- 7) Вычисление и обработка результатов

1. Постановка задачи

Прежде чем понять задачу, следует уточнить ее основные характеристики, сформулировать цель решения задачи, подробно описать ее содержание, провести анализ характера и сущности всех известных и неизвестных данных, определить условия, при которых задача может быть решена.

При постановке задачи выясняется конечная цель и вырабатывается общий подход к решению задачи. Выясняется, сколько решений имеет задача и имеет ли их вообще. Изучаются общие свойства рассматриваемого физического явления или объекта, анализируются возможности данной системы программирования.

1. Постановка задачи

Исходные данные должны быть полными, т.е. содержать данные, необходимые и достаточные для решения задачи. Если данные неполные, необходимо приложить дополнительные усилия для сбора дополнительных сведений; эта ситуация может также возникнуть на последующих этапах при выборе метода решения.

Различают исходные данные трех видов: постоянные, условно-постоянные и переменные.

Постоянные исходные данные - это данные, которые сохраняют свои значения в процессе решения задачи (математические константы, координаты неподвижных объектов) и не зависят от внешних факторов.

1. Постановка задачи

Условно-постоянные данные - это данные, которые могут иногда изменять свои значения; причем эти изменения не зависят от процесса решения задачи, а определяются внешними факторами (величина подоходного налога, курс валют, количество дней в году).

Переменные данные - это данные, которые изменяют свои значения в процессе решения задачи.

На этом этапе важно не только классифицировать данные по отношению к процессу решения, но определить их наименование, тип, структуру и ограничения, накладываемые на значения. Желательно также определить допустимые и недопустимые операции по отношению к различным типам исходных данных

2. Формализация

После проведения анализа постановки задачи, выявления данных, их структуры и отношений между ними можно приступить к построению формальной модели. Это наиболее важный этап в процессе решения задачи.

Модель - это упрощенное представление о реальном объекте, процессе или явлении. **Моделирование** - построение моделей для исследования и изучения моделируемого объекта, процесса, явления с целью получения новой информации при решении конкретных задач.

Для описания модели предметной области решаемой задачи необходимо выбрать некоторую формальную систему. Обычно, исходя из постановки задачи, можно сразу определить один или несколько видов моделей, подходящих для описания и моделирования решения вашей задачи: математические, геометрические, структурные, логические и др.

2. Формализация

Наиболее распространенными и хорошо изученными являются математические модели, описывающие зависимости между данными числового типа. Например, в качестве математической модели звезды можно использовать систему уравнений, описывающих процессы, происходящие в недрах звезды. Математической моделью другого рода являются математические соотношения, позволяющие рассчитать оптимальный план работы предприятия. К основным достоинствам математических моделей безусловно относятся хорошо изученные и широко применяемые математические методы решения большого класса задач, что значительно облегчает формирование основной идеи и выбор методов решения задачи

2. Формализация

Приступая к разработке модели, следует попытаться решить задачу для конкретных входных данных, затем обобщить полученное решение на основе его анализа для любых значений входных данных.

На этом этапе все объекты задачи описываются на языке математики, выбирается форма хранения данных, составляются все необходимые формулы.

Если задача четко поставлена, то для нее несложно разработать математическую модель. Выбор модели существенно влияет на остальные этапы в процессе решения. Математическая формулировка и последующий выбор метода решения являются основой для определения последовательности действий, приводящих к получению искомого результата. В зависимости от содержания задачи построение ее модели может быть областью исследований таких дисциплин, как исследование операций, методы оптимизации, математическая статистика, численный анализ, теория информации и др.

3. Выбор метода решения

4. Разработка алгоритма

Выбор существующего или разработка нового метода решения (очень важен и, в то же время, личностный этап).

На этом этапе метод решения записывается применительно к данной задаче на одном из алгоритмических языков.

Наиболее распространенными методами разработки алгоритмов являются: метод частных целей, метод подъема и эвристический алгоритм.

4. Разработка алгоритма

Для разработки алгоритма методом частных целей необходимо определить варианты возможностей решения задачи:

1. Можно ли решить хотя бы часть задачи, игнорируя некоторые условия?
2. Можно ли решить задачу для частных случаев?
3. Есть ли что-то, что недостаточно понятно?
4. Встречалась ли похожая задача? Можно ли видоизменить ее для решения данной задачи?

4. Разработка алгоритма

При разработке эвристического алгоритма необходимо помнить, что такой алгоритм обычно помогает найти хорошее, но не обязательно оптимальное решение. Общий подход заключается в перечислении всех требований к точному решению с указанием, для каких из них возможен компромисс, а какие непременно должны быть выполнены.

5. Составление программы.

*Решение задачи переводится на язык,
понятный машине.*

6. Отладка программы.

7. Вычисление и обработка результатов.

Алгоритмические конструкции

Различают:

- алгоритм линейной
- разветвляющейся
- циклической структуры
- алгоритмы со структурой вложенных циклов

Алгоритмы решения сложных задач могут включать все перечисленные структуры, которые используются для реализации определенных участков общего алгоритма.

Алгоритм линейной структуры

Алгоритм линейной структуры - алгоритм, в котором блоки выполняются последовательно друг за другом, в порядке, заданном схемой. Такой порядок выполнения называется естественным.

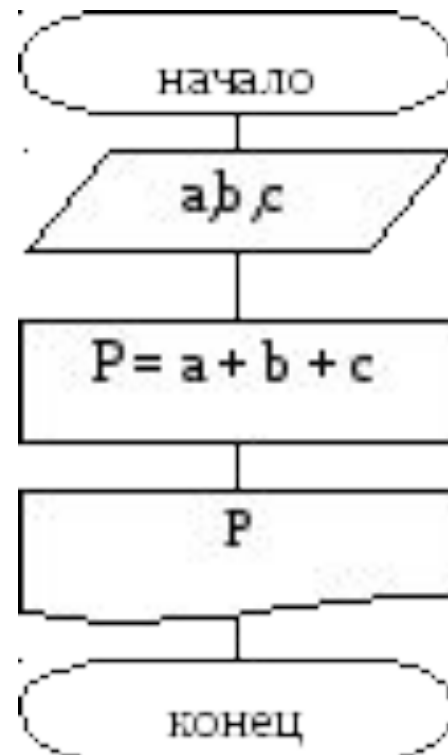
Пример. Вычислить периметр треугольника со сторонами a, b, c .

Алгоритм разветвляющейся структуры

На практике редко удается представить решение задачи в виде алгоритма линейной структуры. Часто в зависимости от каких-либо промежуточных результатов вычисления осуществляются либо по одним, либо по другим формулам, т.е. в зависимости от выполнения некоторого логического условия вычислительный процесс осуществляется по одной или другой формуле.

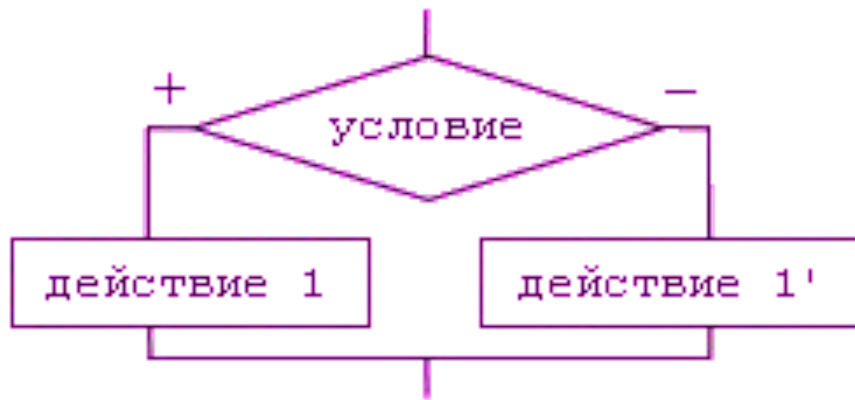
Алгоритм такого вычислительного процесса называется алгоритмом разветвляющейся структуры.

Ветвление - управляющая структура, организующая выполнение лишь одного из двух указанных действий в зависимости от справедливости некоторого условия.

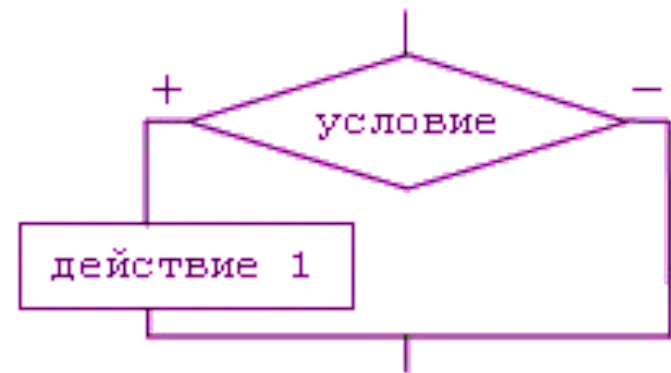


Алгоритм разветвляющейся структуры

Условие - вопрос, имеющий два варианта ответа: да или нет. Запись ветвления выполняется в двух формах: полной и неполной.



а) полная форма

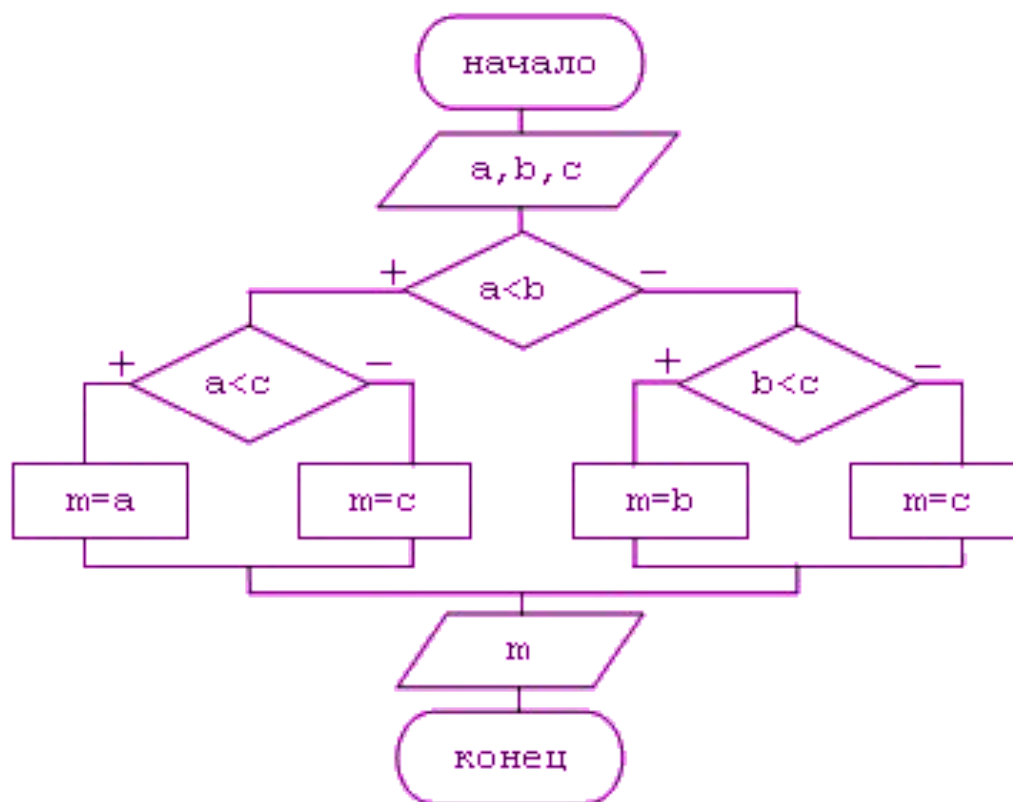


б) неполная

Алгоритм разветвляющейся структуры

Полная и неполная форма ветвлений

Пример. *Найти наименьшее из трех чисел.*



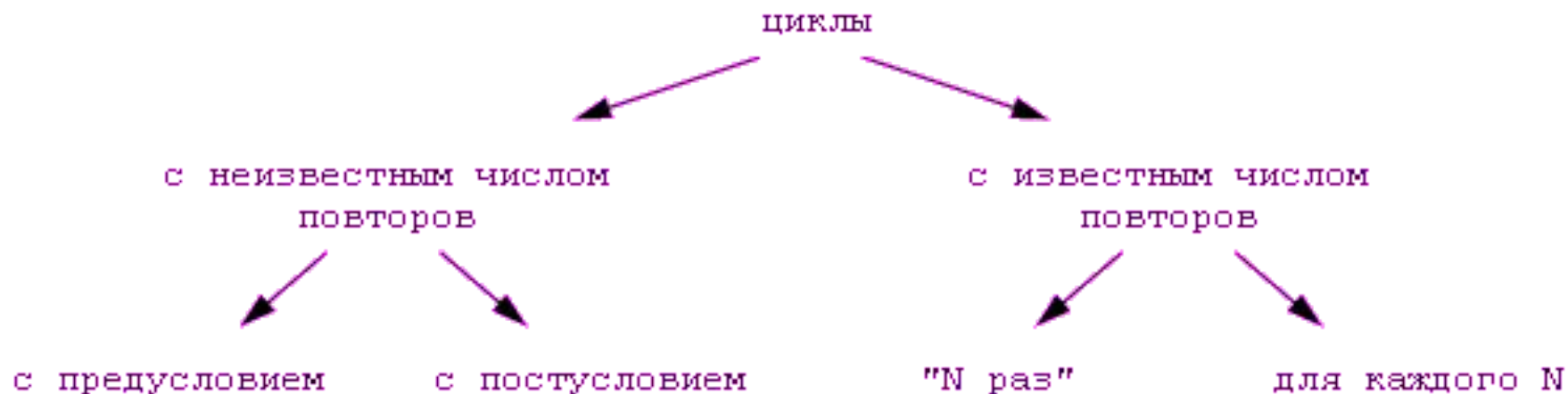
Алгоритм циклической структуры

Часто при решении задач приходится многократно вычислять значения по одним и тем же зависимостям для различных значений входящих в их величины.

Такие многократно повторяемые участки вычислительного процесса называются циклами.

Использование циклов позволяет существенно сократить объем схемы алгоритма и длину соответствующей ей программы. Различают циклы с заданным и неизвестным числом повторений. С заданным числом повторений - цикл со счетчиком. С неизвестным числом повторений - цикл с предусловием, цикл с постусловием.

Алгоритм циклической структуры



Алгоритм циклической структуры

Для организации цикла необходимо выполнить следующие действия:

1. *Задать перед циклом начальное значение переменной, изменяющейся в цикле.*
2. *Изменять переменную перед каждым новым повторением цикла.*
3. *Проверять условие окончания или повторения цикла.*
4. *Управлять циклом, т.е. переходить к его началу, если он не закончен, или выходить из него по окончанию.*

Последние три функции выполняются многократно. Переменная, изменяющаяся в цикле, называется параметром цикла. В одном цикле может быть несколько параметров.

Алгоритм цикла с предусловием

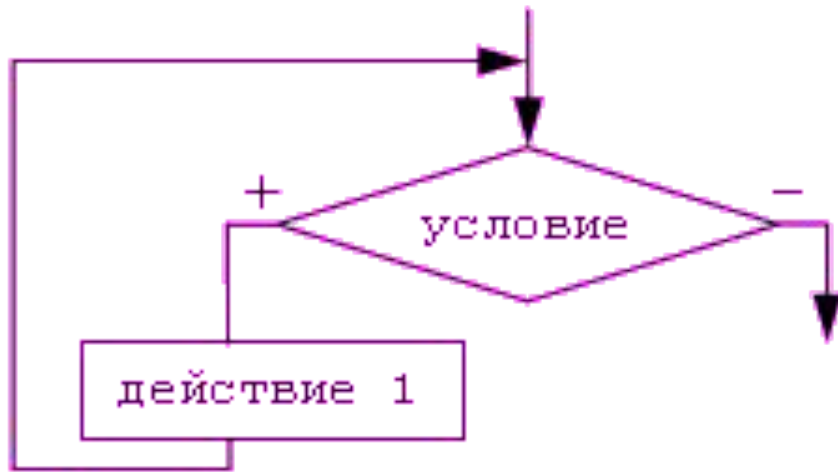
Выполнение цикла "пока" начинается с проверки условия, поэтому такую разновидность циклов называют циклы с предусловием. Переход к выполнению действия осуществляется только в том случае, если условие выполняется, в противном случае происходит выход из цикла (Рис. 3 а). Можно сказать что условие цикла "пока" - это условие входа в цикл. В частном случае может оказаться, что действие не выполнялось ни разу. Условие цикла необходимо подобрать так, чтобы действия, выполняемые в цикле, привели к нарушению его истинности, иначе произойдет зацикливание.

Зацикливание - бесконечное повторение выполняемых действий.

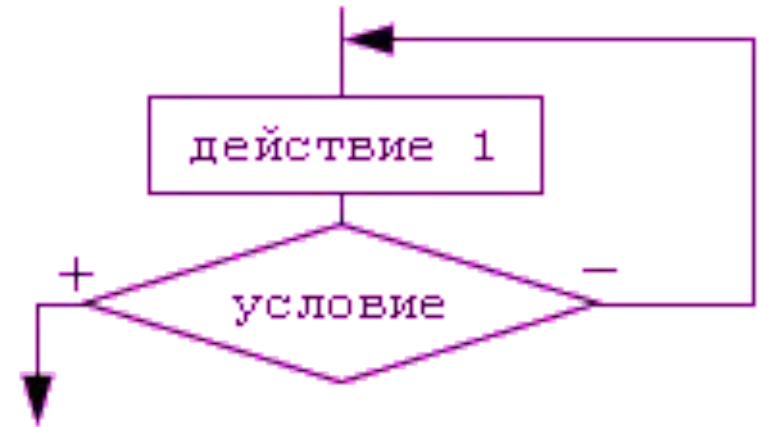
Алгоритм цикла с постусловием

Исполнение цикла начинается с выполнения действия. Таким образом, тело цикла будет реализовано хотя бы один раз. После этого происходит проверка условия. Поэтому цикл "до" называют циклом с постусловием (Рис. 3 б). Если условие не выполняется, то происходит возврат к выполнению действий. Если условие истинно, то осуществляется выход из цикла. Таким образом, условие цикла "до" - это условие выхода. Для предотвращения заикливания необходимо предусмотреть действия, приводящие к истинности условия

Алгоритм цикла с постусловием



а) цикл с предусловием
постусловием



б) цикл с

Алгоритм цикла со счетчиком

Цикл со счетчиком или цикл с параметром является частным случаем цикла с предусловием. Отличие состоит в том, что в цикле со счетчиком задаются границы диапазона, по которым определяется количество повторений тела цикла.

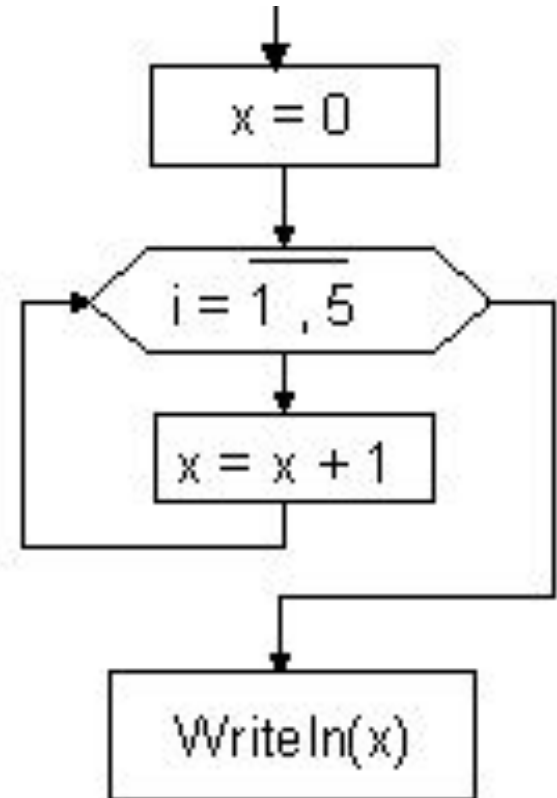
Примеры цикла со счетчиком:

for i := 1 to 5 do

for i := 5 downto 1 do

{счетчик в обратную сторону}

for i := a to z do



Алгоритм цикла с параметром для обработки массива

Массив - упорядоченная структура, предназначенная для хранения однотипных данных.

Упорядочение элементов в массиве происходит по их индексам.

Индекс - порядковый номер элемента.

Массив задается именем (заглавные латинские буквы), типом данных и размерностью.

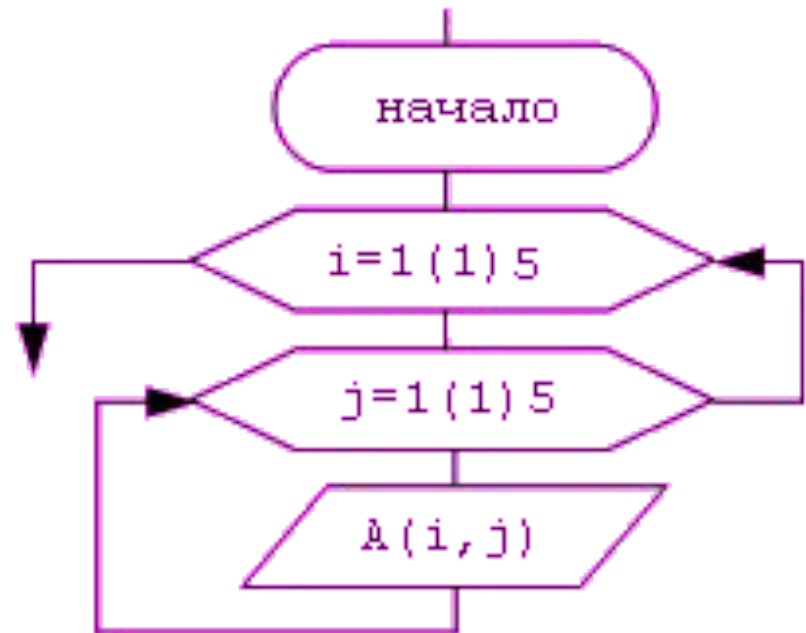
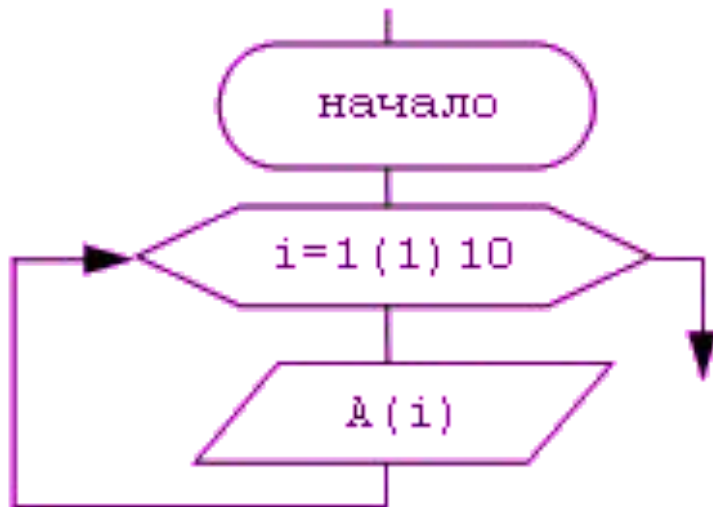
Размерность - максимально возможное количество элементов в массиве. В один момент времени можно обратиться только к одному элементу массива. Для этого указывается имя массива и в скобках индекс элемента.

Алгоритм цикла с параметром для обработки массива

Массивы делятся на одномерные (линейные) и двумерные. Прообразом в математике для одномерного массива является вектор. Для двумерного – матрица.

Пример. Ввести элементы массива

а) одномерного, размерности 10; б) двумерного, размерности 5x5.



Алгоритмы со структурой вложенных циклов

В цикл, называемый внешним, могут входить один или несколько вложенных циклов, называемых внутренними.

Организация как внешнего, так и внутреннего цикла осуществляется по тем же правилам, что и простого цикла. Параметры внешнего и внутреннего циклов разные и изменяются не одновременно, т.е. при одном значении параметра внешнего цикла параметр внутреннего цикла принимает поочередно все значения.

Алгоритмы со структурой вложенных циклов

Пример: Вычислить сумму положительных элементов каждой строки матрицы. $A(m,n)$;

