Основы алгоритмизации и программирования

Лабораторная работа №1.

Цель работы - изучить построение блок-схем.
Содержание отчёта:

1. Титульный лист
2. Формулировка цели и задач работы
3. Описание результатов выполнения
4. Выводы, согласованные с целью

Алгоритм — конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающих порядок действий исполнителя для решения некоторой задачи. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Независимые инструкции могут выполняться в произвольном порядке, параллельно, если это позволяют используемые исполнители.

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.

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

Любая команда алгоритма записывается в блок-схеме в виде графического элемента – блока, и дополняется словесным описанием. Блоки в блок-схемах соединяются линиями потока информации. Направление потока информации указывается стрелкой. В случае потока информации сверху вниз и слева направо стрелку ставить не обязательно. Блоки в блок-схеме имеют только один вход и один выход (за исключением логического блока – блока с условием).

Блок-схемы могут описывать уже существующих код, либо наоборот, описывать необходимую последовательность действий для последующего кодирования. Блок-схемы помогают яснее видеть логику алгоритма, тем самым и избавлять от возможных ошибок при разработке алгоритма.

Правильным способом проектирования блок-схем является абстрагирование от входных параметров. Иными словами, как в математике заменять числа именами переменных. Таким образом, алгоритм будет описывать логику действий и никак не будет зациклен на узкоспециализированную задачу со входными параметрами по-умолчанию.

Всего можно выделить восемь свойств алгоритма (независимо от его вида):

1. Присутствует функция ввода изначальных данных.

2. Есть вывод некоего результата после завершения алгоритма. Нужно помнить, что алгоритм нужен для того, чтобы достичь определенной цели, а именно – получить результат, который имеет прямое отношение к исходным данным.

3. У алгоритма должна быть структура дискретного типа. Он должен представляться последовательными шагами. Причем каждый следующий шаг может начаться только после завершения предыдущего.

4. Алгоритм должен быть однозначным. Каждый шаг четко определяется и не допускает произвольной трактовки.

5. Алгоритм должен быть конечным – необходимо, чтобы он выполнялся за строго определенное количество шагов.

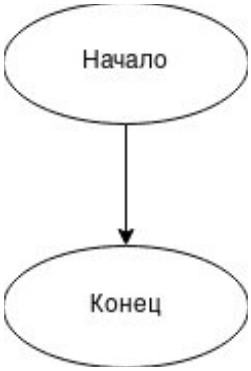
6. Алгоритм должен быть корректным – задавать исключительно верное решение поставленной задачи.

7. Общность (или массовость) – он должен работать с различными исходными данными.

8. Время, которое дается на решение алгоритма, должно быть минимальным. Это определяет эффективность решения поставленной задачи.

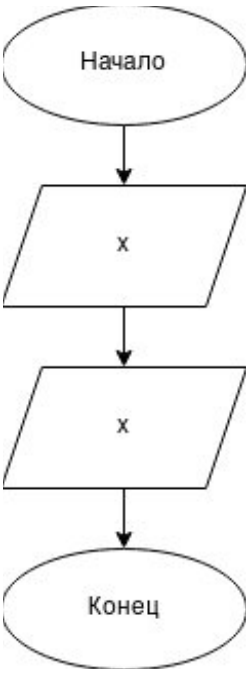
1. Блок - Начало/Конец

Построение блок-схем начинается и заканчивается всегда двумя блоками - начало и конец, их именуют терминаторами начала и конца. Алгоритмом нельзя назвать действия не имеющие данных блоков. Тип возвращаемого значения и аргументов функции обычно указывается в комментариях к блоку терминатора.



2. Блок - Ввод/Вывод

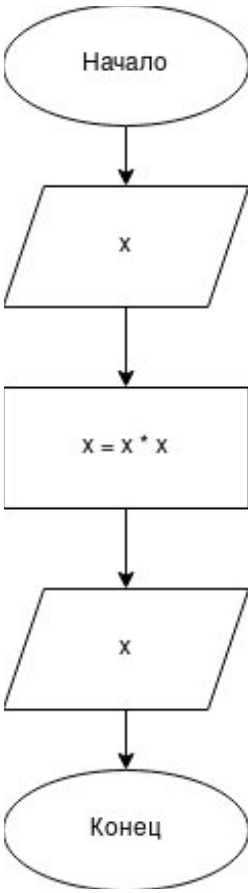
Нет смысла от алгоритмов, которые только начинаются и сразу же заканчиваются. Все алгоритмы оперируют какими-либо данными, которые подаются на вход и выдают получившиеся данные, которые подаются на выход. В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно используется символ параллелограмма. Подробности ввода/вывода могут быть указаны в комментариях. Входные параметры могут быть числами, массивами, строками и т.д. Массивы указываются следующим образом: [x[N]], где N - это количество элементов в массиве. Также входные параметры могут иметь значения по-умолчанию, например [x := 5], [x[N] := [1, 2, 3]].



3. Блок - Действие

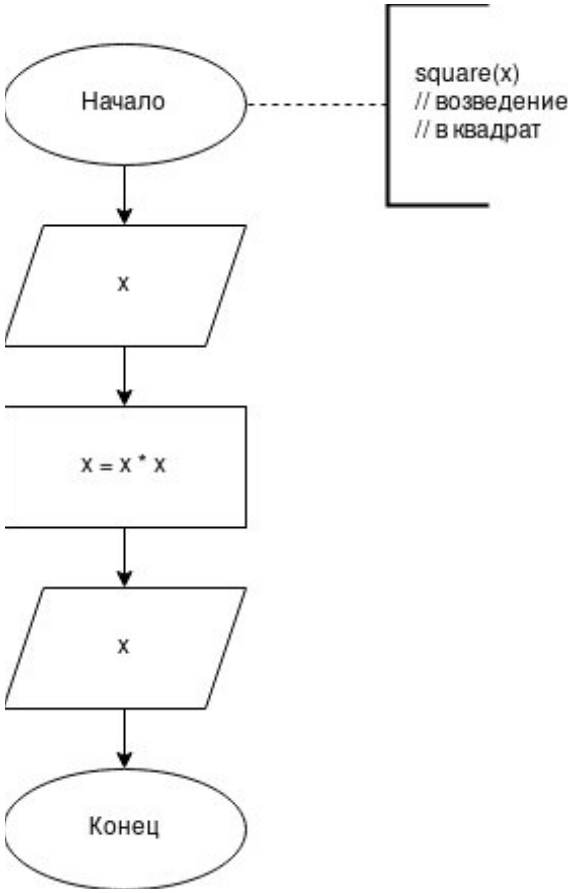
Смыслом любого алгоритма является выполнение списка действий над входными данными. В блоке действий обычно размещают одно или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций.

В примере указан алгоритм возведения числа x в квадрат. Стоит заметить, что необходимым является присваивание результата к переменной x (знак равно).



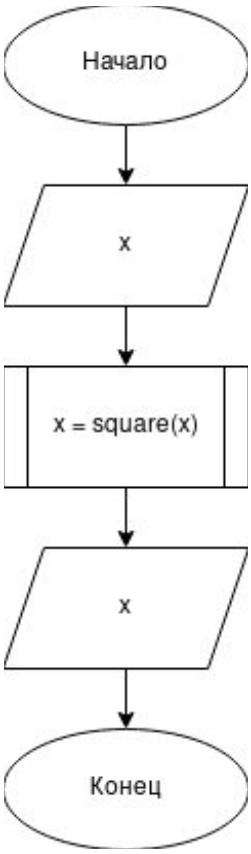
4. Блок - Комментарий

Комментарии необходимы при описании алгоритма и его имени. Так в примере, я назвал алгоритм возведение в квадрат как square. В скобках указано, сколько аргументов он принимает. Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией.



5. Блок - Процедура

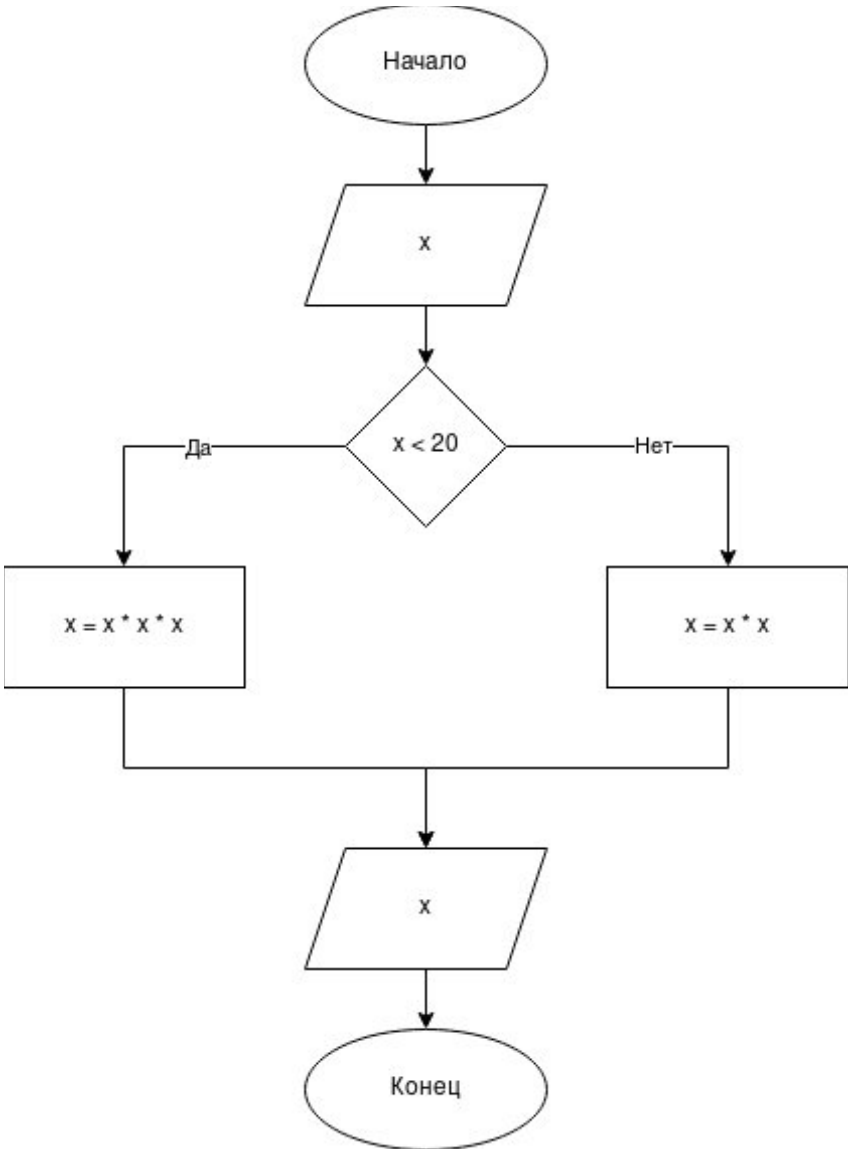
Если уже имеется на руках сделанный алгоритм - не нужно его снова реализовывать, достаточно лишь его указать в блоке процедуры. Пусть предполагается, что алгоритм возведения в квадрат уже имеется, тогда можно вызвать этот алгоритм по его имени (указанном в комментарии на блоке Начало).



6. Блок - Условие

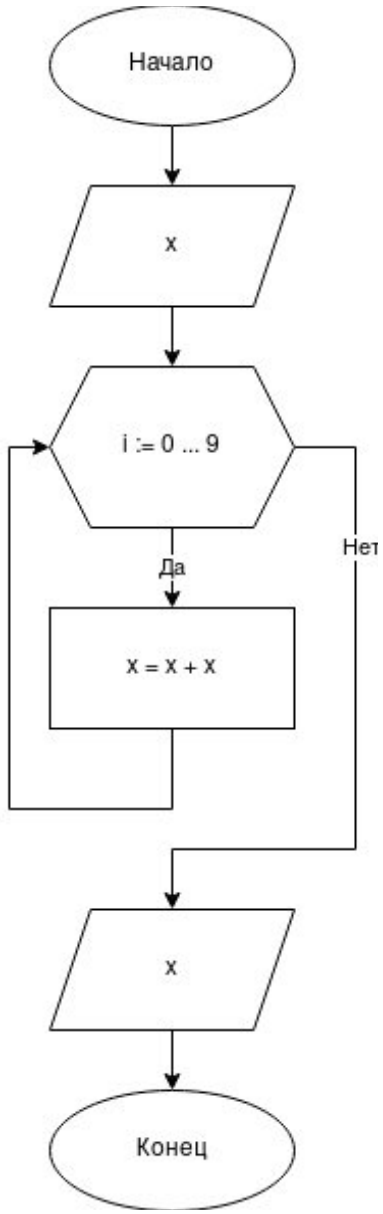
Блок в виде ромба имеет один вход и несколько подписанных выходов. В случае, если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения — «да/нет». Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах — значения этой переменной.

В приведённом примере ниже показано следующее: ЕСЛИ x меньше 20, тогда выполнить возведение числа x в куб [x = x * x * x], ИНАЧЕ выполнить возведение числа x в квадрат [x = x * x].



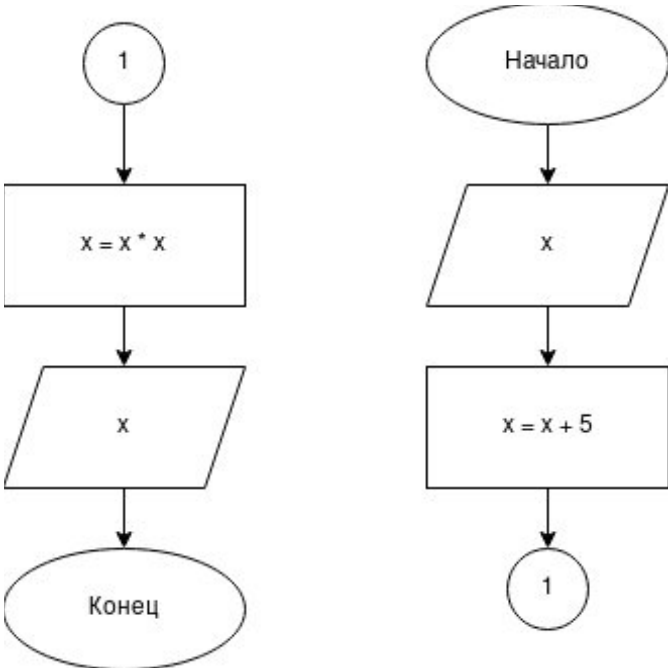
Если цикл представляет собой счётчик, иными словами цикл, который видоизменяет некую переменную на определённое количество шагов, то можно воспользоваться специальным блоком - шестиугольником.

Таким образом, мы убрали определение переменной i в блоке ввода, а также убрали блок действия [i = i + 1].



8. Блок - соединитель

В случае, если блок-схема не умещается 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.



Используемые источники:

1. Сайт для построения блок-схем: https://app.diagrams.net
2. Блок-схемы: https://pro-prof.com/archives/1462

Задание:

Постройте блок-схемы исходя из своего варианта.

Варианты:

1.
   1.1. Заданы два числа a,b. Если a число больше b, то результат равен 1, иначе 0.
   1.2. Задан массив из N чисел. Найти количество положительных чисел.
2.
   2.1. Заданы три числа a,b,c. Найти количество положительных чисел.
   2.2. Задан массив из N чисел. Найти сумму всех отрицательных элементов.
3.
   3.1. Заданы три числа a,b,c. Найти максимальное из них.
   3.2. Задан массив из N чисел. Найти среднее арифметическое.
4.
   4.1. Заданы три числа a,b,c. Найти сумму отрицательных чисел.
   4.2. Возвести число a в степень b, используя операцию умножения.
5.
   5.1. Заданы два числа a,b. Поменять местами их значения.
   5.2. Умножить число a на b, используя операцию сложения.
6.
   6.1. Заданы числа a,b,c. Найти дискриминант.
   6.2. Задан массив из N чисел. Обнулить все отрицательные числа.
7.
   7.1. Заданы стороны a,b прямоугольного треугольника. Найти гипотенузу.
   7.2. Задан массив из N чисел. Инвертировать элементы.