**Алгоритмы и способы их описания**

**Алгоритм** - это система точных и понятных предписаний о содержании и последовательности выполнения конечного числа действий, необходимых для решения любой задачи данного типа.

Примеры: правила сложения, умножения, решения алгебраических уравнений и т.п.

**Свойства алгоритма:**

1.Универсальность (массовость) - применимость алгоритма к различным наборам исходных данных.

2.Дискретность - процесс решения задачи по алгоритму разбит на отдельные действия.

3.Конечность - каждое из действий и весь алгоритм в целом обязательно завершаются.

4.Результативность - по завершении выполнения алгоритма обязательно получается конечный результат.

5.Выполнимость (эффективность) - результата алгоритма достигается за конечное число шагов.

6.Детерминированность (определенность) - алгоритм не должен содержать предписаний, смысл которых может восприниматься неоднозначно. Т.е. одно и то же предписание после исполнения должно давать один и тот же результат.

7.Последовательность – порядок исполнения команд должен быть понятен исполнителю и не должен допускать неоднозначности.

**Классы алгоритмов.**

1.**вычислительные алгоритмы**, работающие со сравнительно простыми видами данных, такими как числа и матрицы, хотя сам процесс вычисления может быть долгим и сложным;

2.**информационные алгоритмы**, представляющие собой набор сравнительно простых процедур, работающих с большими объемами информации (алгоритмы баз данных);

3.**управляющие алгоритмы**, генерирующие различные управляющие воздействия на основе данных, полученных от внешних процессов, которыми алгоритмы управляют.

По типу передачи управления алгоритмы бывают: основные (главные выполняемые программы) и вспомогательные (подпрограммы).

Для задания алгоритма необходимо описать следующие его элементы:

1.набор объектов, составляющих совокупность возможных исходных данных, промежуточных и конечных результатов;

2.правило начала;

3.правило непосредственной переработки информации (описание последовательности действий);

4.правило окончания;

5.правило извлечения результатов.

**Способы описания алгоритмов.**

Символьный, когда алгоритм описывается с помощью специального набора символов (специального языка).

Словесная форма записи алгоритмов обычно используется для алгоритмов, ориентированных на исполнителя-человека. Команды такого алгоритма выполняются в естественной последовательности, если не оговорено противного.

Графическая запись с помощью блок-схем осуществляется рисованием последовательности геометрических фигур, каждая из которых подразумевает выполнение определенного действия алгоритма. Порядок выполнения действий указывается стрелками. Графическая запись алгоритма имеет ряд преимуществ: каждая операция вычислительного процесса изображается отдельной геометрической фигурой и графическое изображение алгоритма наглядно показывает разветвления путей решения задачи в зависимости от различных условий, повторение отдельных этапов вычислительного процесса и другие детали.

**Виды блоков.**



**Правила создания блок – схем:**

1.Линии, соединяющие блоки и указывающие последовательность связей между ними, должны проводится параллельно линиям рамки.

2.Стрелка в конце линии может не ставиться, если линия направлена слева направо или сверху вниз.

3.В блок может входить несколько линий, то есть блок может являться преемником любого числа блоков.

4.Из блока (кроме логического) может выходить только одна линия.

5.Логический блок может иметь в качестве продолжения один из двух блоков, и из него выходят две линии.

6.Если на схеме имеет место слияние линий, то место пересечения выделяется точкой. В случае, когда одна линия подходит к другой и слияние их явно выражено, точку можно не ставить.

7.Схему алгоритма следует выполнять как единое целое, однако в случае необходимости допускается обрывать линии, соединяющие блоки.

**Виды алгоритмов.**

В линейном алгоритме операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной, независимой от каких-либо условий. На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.

В алгоритме с ветвлением предусмотрено несколько направлений (ветвей). Каждое отдельное направление алгоритма обработки данных является отдельной ветвью вычислений. Направление ветвления выбирается логической проверкой, в результате которой возможны два ответа:

1.«да» — условие выполнено.

2.«нет» — условие не выполнено.

Циклические алгоритмы содержат цикл – это многократно повторяемый участок алгоритма. Различают циклы с предусловием и постусловием. Также циклы бывают детерминированные и итерационные. Цикл называется детерминированным, если число повторений тела цикла заранее известно или определено. Цикл называется итерационным, если число повторений тела цикла заранее неизвестно, а зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

**Задание подобрать тему для создания блок - схемы**