**Практическая работа №4\_1: Программный принцип работы компьютера.**

**Цель:** изучить способы представления алгоритмов в виде блок - схем.

**Краткие теоретические сведения:**

**Алгоритм** — точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

**Исполнитель алгоритма** — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Исполнителя характеризуют: сpеда; элементаpные действия; cистема команд; отказы.

Основные свойства алгоритмов следующие:

**Понятность для исполнителя** — т.е. исполнитель алгоритма должен знать, как его выполнять.

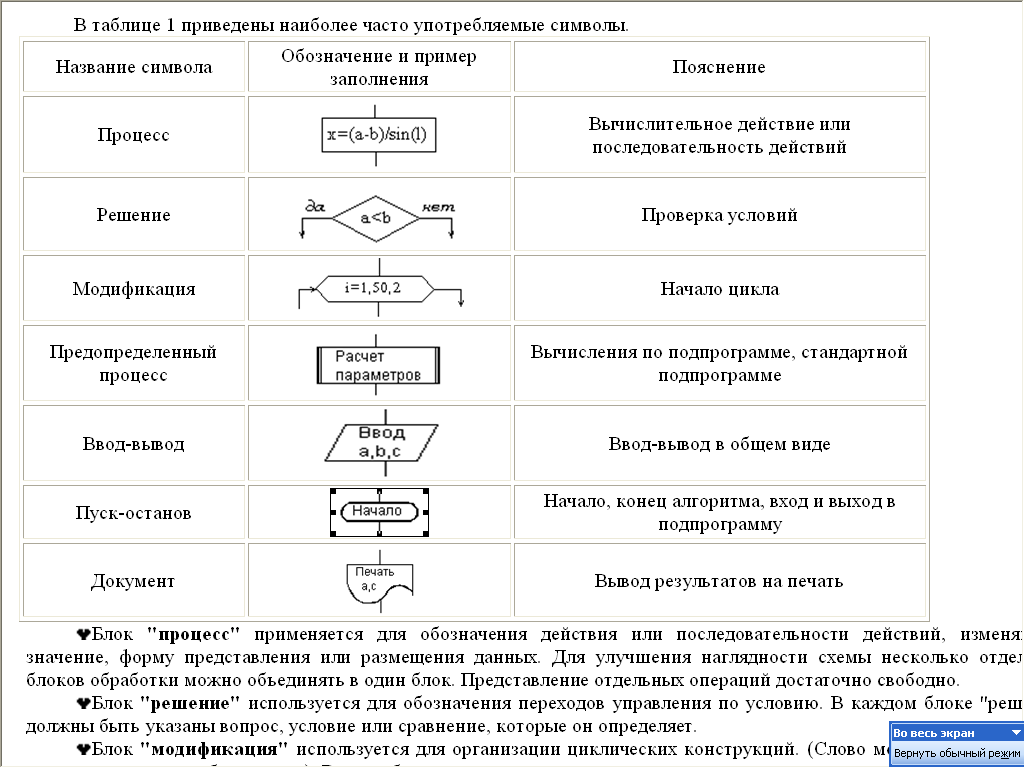
**Дискpетность** (прерывность, раздельность) — т.е. алгоpитм должен пpедставлять пpоцесс pешения задачи как последовательное выполнение пpостых (или pанее опpеделенных) шагов (этапов).

**Опpеделенность** — т.е. каждое пpавило алгоpитма должно быть четким, однозначным и не оставлять места для пpоизвола. Благодаpя этому свойству выполнение алгоpитма носит механический хаpактеp и не тpебует никаких дополнительных указаний или сведений о pешаемой задаче.

**Pезультативность (или конечность).** Это свойство состоит в том, что алгоpитм должен пpиводить к pешению задачи за конечное число шагов.

**Массовость.** Это означает, что алгоpитм pешения задачи pазpабатывается в общем виде, т.е. он должен быть пpименим для некотоpого класса задач, pазличающихся лишь исходными данными. Пpи этом исходные данные могут выбиpаться из некотоpой области, котоpая называется областью пpименимости алгоpитма.

На практике наиболее распространены следующие формы представления алгоритмов:

* [словесная](http://college.biysk.secna.ru/inform/1_7_5.html) (записи на естественном языке);
* [графическая](http://college.biysk.secna.ru/inform/1_7_6.html) (изображения из графических символов);
* [псевдокоды](http://college.biysk.secna.ru/inform/1_7_7.html) (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
* [программная](http://college.biysk.secna.ru/inform/1_7_12.html) (тексты на языках программирования).

Блок **"процесс"** применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок **"решение"** используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок **"модификация"** используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок **"предопределенный процесс"** используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

|  |  |
| --- | --- |
| **Линейный алгоритм** – это алгоритм, в котором блоки выполняются последовательно сверху вниз от начала до конца.  На **рисунке** приведен пример блок-схемы алгоритма вычисления периметра Р и площади S квадрата со стороной длины A. |  |
| **Разветвляющийся** **алгоритм** это **алгоритм**, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий. |  |
| ****Циклические алгоритмы**** Часто при решении задач приходится повторять выполнение операций по одним и тем же зависимостям при различных значениях входящих в них переменных и производить многократный проход по одним и тем же участкам алгоритма. Такие участки называются циклами. Алгоритмы, содержащие циклы, называется циклическими. Использование циклов существенно сокращает объем алгоритма. Различают циклы с наперед известным и наперед неизвестным количеством проходов. | |

|  |  |  |
| --- | --- | --- |
| ../Downloads/Массивы/В_А_Коднянко_%20Алгоритмы%20и%20алгоритмизация.files/RazvAlg2.gif | ../Downloads/Массивы/В_А_Коднянко_%20Алгоритмы%20и%20алгоритмизация.files/CircleAlg.gif | |
| Сначала производится вход в цикл. После этого начинается его выполнение | | |
| ../Downloads/Массивы/В_А_Коднянко_%20Алгоритмы%20и%20алгоритмизация.files/CircleZag.gif |  | ../Downloads/Массивы/В_А_Коднянко_%20Алгоритмы%20и%20алгоритмизация.files/Circle.gif |
| ****Структура цикла**** |  | ****Структура заголовка цикла**** |

Внутри заголовка счетчику первоначально присваивается значение i = j. Затем выполняется блоки, образующие тело цикла. Обработка блоков внутри цикла производится по часовой стрелке. В результате после первого выполнения тела цикла управление вновь передается заголовку. Здесь к текущему значению счетчика добавится шаг. Теперь, если новое значение счетчика не вышло за свои пределы (т. е. не стало больше своего конечного значения при положительном шаге или меньше конечного значения – при отрицательном шаге), то снова выполняется тело цикла, вновь после возврата к заголовку к счетчику добавляется шаг. Так цикл будет выполняться до тех пор, пока значение счетчика однажды не выйдет за предписанный предел. Как только такой предел будет преодолен, произойдет выход из цикла и управление будет передано блоку, который следует сразу за циклом.

Вернемся к блок-схеме рис. Заголовок ее цикла представлен блоком 4. Роль счетчика цикла играет переменная i, которая должна в цикле изменяться от 1 до N. Поскольку шаг явно не указан, то по умолчанию он подразумевается равным 1. Тело цикла образуют блоки 5 и 6.

Сразу после входа в цикл переменная i примет начальное значение  i = 1. Далее в блоке 5 выполняется проверка положительности первого элемента массива Z (т. к. i = 1). Если этот элемент действительно положителен, то в блоке б он будет добавлен к переменной S, после чего выполняется возврат к заголовку цикла. Если этот элемент не положителен (т. е. нуль или отрицательный), то будет выполнен переход сразу к заголовку цикла, минуя блок суммирования 6.

На втором круге цикла счетчик i в заголовке увеличится на 1 и станет равным 2. Теперь, при новом выполнении тела цикла, в блоке 5 проверяется на положительность второй элемент массива Z и, если он положителен, то добавляется в сумму и т. д. Последний раз тело цикла выполнится при i = N. При этом значении счетчика проверяется последний элемент массива. Наконец, в заголовке цикла i примет значение N+1. Это значение выходит за предписанный предел, следовательно, произойдет выход из цикла и управление перейдет блоку 7. В этом блоке выводится накопленная сумма и алгоритм закончит работу.

***Практические задания*:**

**Задание 1.** Вычислить сумму элементов числового массива   A = (a1 , a2 , ... , aN ).

Составить блок-схему алгоритма

|  |  |  |
| --- | --- | --- |
| Данные | | Результат |
| N=5 | A=(3, 5, -2, 6, 3) | S= |

Записать **исполнение алгоритма** виде таблицы:

|  |  |
| --- | --- |
| i | S |
|  |  |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |

**Задание 2** Найти максимальный элемент числового массива A = (a1 , a2 , ... , aN ) и определить его номер.

Составить блок-схему алгоритма.

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | | Результат | |
| N=5 | A=(3, -5, -2, 6, 3) | Аmax= | k= |

Записать **исполнение алгоритма** виде таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **A[i] > Amax**  **(да/нет)** | **Amax** | **k** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |

**Задание 3** Вычислить сумму положительных элементов числового массива  A = (a1 , a2 , ... , aN ).Составить блок-схему алгоритма.

|  |  |  |
| --- | --- | --- |
| Данные | | Результат |
| N=5 | A=(3, 5, -2, 6, -3) | S= |

Записать **исполнение алгоритма** виде таблицы:

|  |  |
| --- | --- |
| i | S |
|  |  |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |

**Задание 4.** Элементы заданного одномерного массива A = (a1 , a2 , ... , aN ) разделить на его первый элемент.

Составить блок-схему алгоритма

|  |  |  |
| --- | --- | --- |
| Данные | | Результат |
| N=5 | A=(2, 5, -4, 6, -3) | B=( , , , , ) |

**Задание 5.** Найти число и произведение отрицательных элементов заданного массива A = (a1 , a2 , ... , aN ) Составить блок-схему алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | | Результат | |
| N=5 | A=(3, -5, -2, 6, 3) | Р= | m= |

Записать **исполнение алгоритма** виде таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **A[i] < 0**  **(да/нет)** | **P** | **m** |
| **1**  **2**  **3**  **4**  **5** |  |  |  |