**Добрый день, 12 группа!**

Продолжаем общаться дистанционно.

Сегодня мы рассмотрим алгоритмические конструкции

Задать вопросы, а также прислать ответы вы можете

1. на адрес электронной почты: [ddrmx@ya.ru](mailto:ddrmx@ya.ru)
2. через соцсеть <https://vk.com/ddrmx>

С уважением, Максим Андреевич.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО ТЕМЕ:

Примеры алгоритмов. (1 ЧАС)

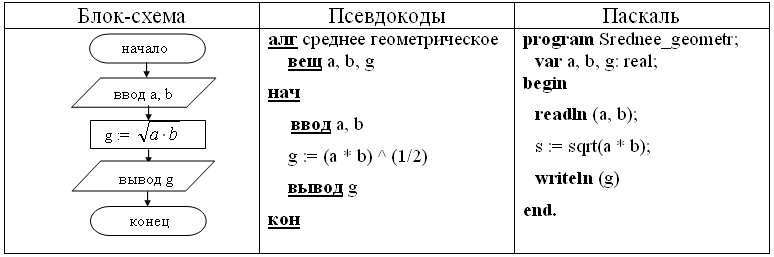
Наиболее понятно структуру алгоритма можно представить с помощью блок-схемы, в которой используются геометрические фигуры (блоки), соединенные между собой стрелками, указывающими последовательность выполнения действий. Приняты определенные стандарты графических изображений блоков. Например, команду обработки информации помещают в блок, имеющий вид прямоугольника, проверку условий - в ромб, команды ввода или вывода - в параллелограмм, а овалом обозначают начало и конец алгоритма.

Структурной элементарной единицей алгоритма является простая команда, обозначающая один элементарный шаг переработки или отображения информации. Простая команда на языке схем изображается в виде функционального блока.

|  |  |
| --- | --- |
|  | Данный блок имеет *один вход* и *один выход*. Из простых команд и проверки условий образуются составные команды, имеющие более сложную структуру и тоже *один вход и один выход*.  Структурный подход к разработке алгоритмов определяет использование только базовых алгоритмических структур (конструкций): следование, ветвление, повторение, которые должны быть оформлены стандартным образом. |
|  | Рассмотрим основные структуры алгоритма.  Команда *следования* состоит только из простых команд. На рисунке простые команды имеют условное обозначение *S1* и *S2*. Из команд следования образуются линейные алгоритмы. Примером линейного алгоритма будет нахождение суммы двух чисел, введенных с клавиатуры. |
|  | Команда *ветвления* - это составная команда алгоритма, в которой в зависимости от условия Р выполняется или одно *S1*, или другое *S2* действие. Из команд следования и команд ветвления составляются разветвляющиеся алгоритмы (алгоритмы ветвления). Примером разветвляющегося алгоритма будет нахождение большего из двух чисел, введенных с клавиатуры. |
|  | Команда ветвления может быть полной и неполной формы. Неполная форма команды ветвления используется тогда, когда необходимо выполнять действие *S* только в случае соблюдения условия *P*. Если условие *P* не соблюдается, то команда ветвления завершает свою работу без выполнения действия. Примером команды ветвления неполной формы будет уменьшение в два раза только четного числа. |
|  | Команда *повторения* - это составная команда алгоритма, в которой в зависимости от условия *Р* возможно многократное выполнение действия *S*. Из команд следования и команд повторения составляются циклические алгоритмы (алгоритмы повторения). На рисунке представлена команда повторения с предусловием. Называется она так потому, что вначале проверяется условие, а уже затем выполняется действие. Причем действие выполняется, пока условие соблюдается. Пример циклического алгоритма может быть следующий. Пока с клавиатуры вводятся положительные числа, алгоритм выполняет нахождение их суммы.  Команда повторения с предусловием не является единственно возможной. Разновидностью команды повторения с предусловием является команда повторения с параметром. Она используется тогда, когда известно количество повторений действия. В блок-схеме команды повторения с параметром условие записывается не в ромбе, а в шестиугольнике. Примером циклического алгоритма с параметром будет нахождение суммы первых 20 натуральных чисел. |
|  | В команде повторения с постусловием вначале выполняется действие *S* и лишь затем, проверяется условие *P*. Причем действие повторяется до тех пор, пока условие не соблюдается. Примером команды повторения с постусловием будет уменьшение положительного числа до тех пор, пока оно неотрицательное. Как только число становится отрицательным, команда повторения заканчивает свою работу.  С помощью соединения только этих элементарных конструкций (последовательно или вложением) можно "собрать" алгоритм любой степени сложности. |

***Линейный алгоритм***

Приведем пример записи алгоритма в виде блок-схемы, псевдокодов и на языке Паскаль. Ручное тестирование и подбор системы тестов выполняются аналогично предыдущему заданию.



Домашнее задание: Напишите вывод (в какой форме записываются алгоритмы?)

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО ТЕМЕ:

Алгоритмические конструкции. (1 ЧАС)

Решение любой задачи на ЭВМ можно разбить на следующие этапы: разработка алгоритма решения задачи, составление программы решения задачи на алгоритмическом языке, ввод программы в ЭВМ, отладка программы (исправление ошибок), выполнение программы на ПК, анализ полученных результатов.

Первый этап решения задачи состоит в разработке алгоритма.

Алгоритм – это точная конечная система правил, определяющая содержание и порядок действий исполнителя над некоторыми объектами (исходными и промежуточными данными) для получения после конечного числа шагов искомого результата.

Алгоритм может быть описан одним из трех способов:

* словесным (пример в начале раздела);
* графическим (виде специальной блок-схемы);
* с помощью специальных языков программирования.

**Блок-схема** – распространенный тип схем, описывающий алгоритмы или процессы, изображая шаги в виде блоков различной формы, соединенных между собой стрелками.

1. **Линейный алгоритм** – это такой алгоритм, в котором все операции выполняются последовательно одна за другой.
2. **Алгоритмы разветвленной структуры** применяются, когда в зависимости от некоторого условия необходимо выполнить либо одно, либо другое действие.
3. **Алгоритмы циклической структуры**.

**Циклом** называют повторение одних и тех же действий (шагов). Последовательность действий, которые повторяются в цикле, называют **телом цикла**.

Циклические алгоритмы подразделяют на алгоритмы с предусловием, постусловием и алгоритмы с конечным числом повторов. В алгоритмах с предусловием сначала выполняется проверка условия окончания цикла и затем, в зависимости от результата проверки, выполняется (или не выполняется) так называемое тело цикла.

**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       с:=(a+b)/2

        иначе с:=a/b

    все

    вывод с

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**

**Технология выполнения работы**

В рамках выполнения работы необходимо составить алгоритм решения задачи в виде блок-схемы и с помощью языка псевдокода.

Домашнее задание:

### Содержание отчета

1. Цель работы и задание.
2. Условие задачи.
3. Алгоритм, написанный с помощью псевдокода и блок- схемы.

### Вопросы для защиты работы

1. Что такое алгоритм?
2. Свойства алгоритма.
3. Способы записи алгоритма.
4. Основные элементы блок-схемы.
5. Виды алгоритмов.
6. Отличительные особенности алгоритмов с предусловием и постусловием.