**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:

Алгоритмические конструкции. (1 ЧАС, ***повторение и продолжение***)

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.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО ТЕМЕ:

Операторы. (1 ЧАС)

**В записи логических выражений помимо арифметических операций** сложения, вычитания, умножения, деления и возведения в степень **используются операции отношения** < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), = (равно), <> (не равно), **а также логические операции и, или, не.**

*Запишите в тетрадь:*

### Примеры записи логических выражений, истинных при выполнении указанных условий.

|  |  |
| --- | --- |
| Условие | Запись алгоритмическом языке |
| Дробная часть вещественного числа a равна нулю | int(a) = 0 |
| Целое число a — четное | mod(a, 2) = 0 |
| Целое число a — нечетное | mod(a, 2) = 1 |
| Целое число k кратно семи | mod(a, 7) = 0 |
| Каждое из чисел a, b положительно | (a>0) **и** (b>0) |
| Только одно из чисел a, b положительно | ((a>0) **и** (b<=0)) **или**  ((a<=0) **и** (b>0)) |
| Хотя бы одно из чисел a, b, c является отрицательным | (a<0) **или** (b<0) **или** (c<0) |
| Число x удовлетворяет условию a < x < b | (x>a) **и** (x<b) |
| Число x имеет значение в промежутке [[1](http://book.kbsu.ru/theory/literature.html#1), [3](http://book.kbsu.ru/theory/literature.html#3)] | (x>=1) **и** (x<=3) |
| Целые числа a и b имеют одинаковую четность | ((mod(a, 2)=0) **и** (mod(b, 2)=0) **или** ((mod(a, 2)=1) **и** (mod(b, 2)=1)) |
| Точка с координатами (x, y) лежит в круге радиуса r  с центром в точке (a, b) | (x-a)\*\*2 + (y-b)\*\*2 < r\*r |
| Уравнение ax^2 + bx + c = 0 не имеет действительных корней | b\*b - 4\*a\*c < 0 |
| Точка (x, y) принадлежит первой или третьей   четверти | ((x>0) **и** (y>0)) **или**  ((x<0) **и** (y>0)) |
| Точка (x, y) принадлежит внешности единичного круга   с центром в начале координат или его второй четверти | (x\*x + y\*y > 1) **или**  ((x\*x + y\*y <= 1) **и** (x<0) **и** (y>0)) |
| Целые числа a и b являются взаимно-противоположными | a = -b |
| Целые числа a и b являются взаимообратными | a\*b = 1 |
| Число a больше среднего арифметического чисел b, c, d | a > (b+c+d) / 3 |
| Число a не меньше среднего геометрического чисел b, c, d | a >= (b+c+d) \*\* (1/3) |
| Хотя бы одна из логических переменных F1 и F2 имеет   значение **да** | F1 **или** F2 |
| Обе логические переменные F1 и F2 имеют значение **да** | F1 **и** F2 |
| Обе логические переменные F1 и F2 имеют значение **нет** | **не** F1 **и не** F2 |
| Логическая переменная F1 имеет значение **да**, а   логическая переменная F2 имеет значение **нет** | F1 **и не** F2 |
| Только одна из логических переменных F1 и F2   имеет значение **да** | (F1 **и не** F2) **или** (F2 **и не** F1) |

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО ТЕМЕ:

Условия, циклы. (1 ЧАС)

***Краткие теоретические сведения.***

Если в программе возникает необходимость неоднократно выполнить некоторые операторы, то используются *операторы повтора (цикла).* В языке Паскаль различают три вида операторов цикла: *while, for*. Они используются для организации циклов различных типов. Выражение, управляющее повторениями, должно иметь булевский тип.

Если число повторений оператора (составного оператора) заранее неизвестно, а задано лишь условие его повторения (или окончания), используются оператор *while*. Оператор *for* используется, если число повторений заранее известно.

**Оператор цикла for**

В случаях, когда число повторений может быть заранее известно, для организации циклической обработки информации применяется оператор повтора *for.* Часто этот оператор повтора называют *оператором цикла с параметром,* так как число повторений задается переменной, называемой *параметром цикла*, или управляющей переменной. Оператор повтора *for* состоит из *заголовка* и *тела цикла*.

Он может быть представлен в двух форматах:

**for <параметр цикла> := <S1> +1 <S2> do <оператор>;**

**for <параметр цикла> := <S1> -1 <S2> do <оператор>;**

где *Sl* и *S2* — выражения, определяющие соответственно начальное и конечное значения параметра цикла;

for ... do — заголовок цикла;

<оператор> — тело цикла.

Значение управляющей переменной изменяется на +1 или –1

Тело цикла может быть простым или составным оператором. Оператор *for* обеспечивает выполнение тела цикла до тех пор, пока не будут перебраны все значения параметра цикла от начального до конечного.

**Оператор цикла while**

Оператор *while (пока)* часто называют *оператором цикла с предусловием* за то, что проверка условия выполнения тела цикла производится в самом начале оператора.

Формат записи:

**while <условие продолжения повторений>**

**<тело цикла>;**

*Условие* - логическое выражение, *тело цикла* - простой или составной оператор.

Перед каждым выполнением тела цикла вычисляется значение выражения условия. Если результат равен *True,* тело цикла выполняется и снова вычисляется выражение условия. Если результат равен *False*, происходят выход из цикла и переход к первому после *while* оператору.

Цикл*while* может не выполниться ни разу, если при первой проверке условие оказалось ложным (*False).*

Домашнее задание:

### Содержание отчета

1. Операторы цикла. Общая характеристика.
2. Оператор цикла for. Форматы записи, описание работы цикла, ограничения использования параметра цикла.
3. Оператор цикла while. Формат записи, описание работы цикла.