ОБРАЗОВАТЕЛЬНЫЙ МИНИМУМ. ПАМЯТКА.	Предмет	**Информатика**
	Класс	**7**
Дата проведения		**3 четверть**

1. Алгоритм и Исполнитель

Исполнитель **-** некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Формальный исполнитель одну и ту же команду всегда выполняет одинаково. Для каждого формального исполнителя можно указать: круг решаемых задач, среду, систему команд и режим работы.

Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека.

Алгоритм — это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами ***дискретности, понятности, определённости, результативности и массовости.***

Дискретность (от лат. *discretus* – разделенный, прерывистый) означает, что путь решения задачи разделён на отдельные шаги (действия). Каждому действию соответствует предписание (команда). Только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.

Понятность означает, что алгоритм состоит только из команд, входящих в систему команд исполнителя, т. е. из таких команд, которые исполнитель может воспринять и по которым может выполнить требуемые действия.

Определённость означает, что в алгоритме нет команд, смысл которых может быть истолкован исполнителем неоднозначно; недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какую команду выполнять на следующем шаге.

Результативность означает, что алгоритм должен обеспечивать возможность получения результата после конечного, возможно, очень большого, числа шагов. При этом результатом считается не только обусловленный постановкой задачи ответ, но и вывод о невозможности продолжения по какой-либо причине решения данной задачи.

Массовость означает, что алгоритм должен обеспечивать возможность его применения для решения любой задачи из некоторого класса задач с различными исходными данными.

Величина в информатике – это отдельный информационный объект (число, символ, строка, таблица и др.).

Величины делятся на:

постоянные - значения указываются в тексте алгоритма и не меняются в процессе его исполнения

переменные - значения меняются в процессе исполнения алгоритма.

Тип величины: целый, вещественный, логический, символьный и литерный.

Для ссылок на величины используют их ***имена*** (идентификаторы). Имя величины может состоять из одной или нескольких латинских букв, из латинских букв и цифр.

Таблица (массив) - набор некоторого числа однотипных элементов, которым присвоено одно имя. Положение элемента в таблице однозначно определяется его индексами.

Виды алгоритмов:

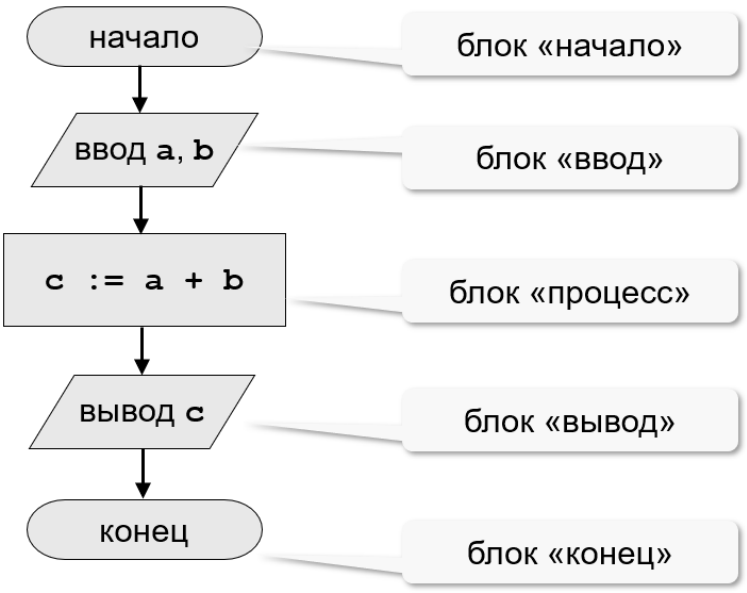
1. Линейный алгоритм (описание действий, которые выполняются однократно в заданном порядке, последовательно одна за другой);
2. Циклический алгоритм (описание действий, которые могут повторятся указанное число раз или пока не выполнено заданное условие);
3. Разветвляющийся алгоритм (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий);
4. Вспомогательный алгоритм (алгоритм, который можно использовать в других алгоритмах, указав только его имя).

Существуют различные способы записи алгоритмов:

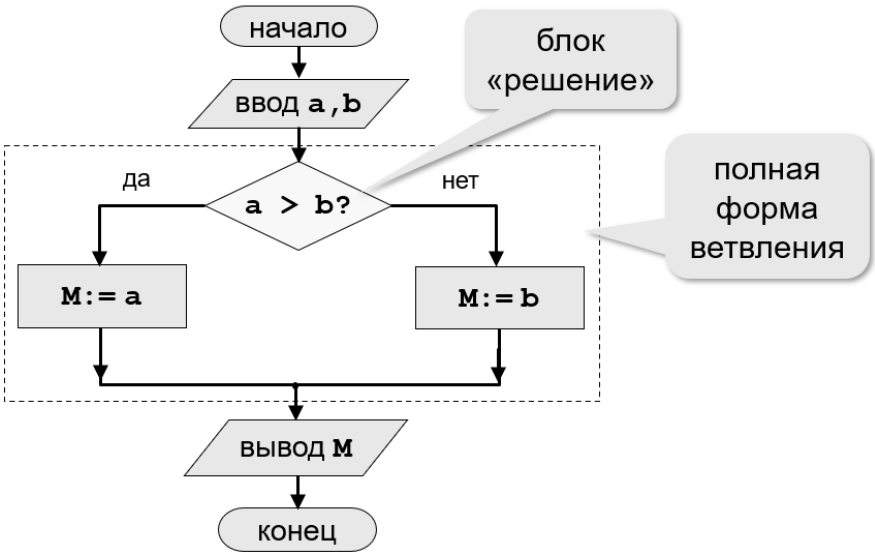
- ***словесное описание***
- ***построчная запись***
- ***блок-схема (графическая форма)***
- ***школьный алгоритмический язык*** **и другие.**

В блок-схеме каждому типу действий соответствует геометрическая фигура.

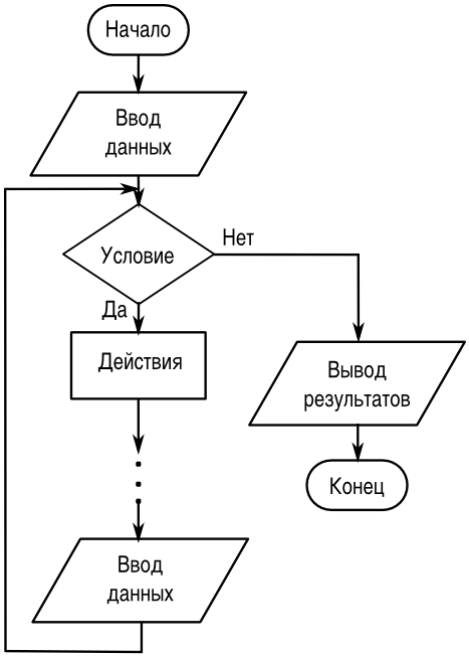
Блок-схема линейного алгоритма



Блок-схемы разветвляющегося алгоритма



Блок-схема повторяющегося алгоритма (цикла)



2. Исполнитель Чертежник

Исполнитель Чертежник предназначен для построения рисунков на координатной плоскости.

Алгоритм, решающий некоторую подзадачу основной задачи, называется **вспомогательным алгоритмом.**

Приказ на выполнение вспомогательного алгоритма записывается в основном алгоритме.

Для повторения n раз некоторой команды используют конструкцию повторения –

нц n раз

Система команд исполнителя **(СКИ) Чертёжника:**

- **поднять перо**
- **опустить перо**
- **сместиться в точку (а, в)**
- **сместиться на вектор (а, в)**

Команду СМЕСТИТЬСЯ В ТОЧКУ (а, в) называют командой ***абсолютного смещения.***

Команду СМЕСТИТЬСЯ НА ВЕКТОР (а, в) называют командой ***относительного смещения.***

Пример алгоритма Чертежника:

сместиться в точку (1, 1)

опустить перо

сместиться в точку (3, 5)

сместиться в точку (5, 2)

сместиться в точку (1, 1)

Построен треугольник, вершины которого находятся в точках с координатами (1, 1), (3, 5) и (5, 2).

3. Программирование на алгоритмическом языке

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для исполнителя

Команда – это описание действий, которые должен выполнить исполнитель.

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

Имена переменных:

МОЖНО использовать	НЕЛЬЗЯ использовать
• латинские буквы (A-Z), русские буквы (А-Я). Заглавные и строчные буквы **различаются**. • цифры (**имя не может начинаться с цифры**) • знак подчеркивания _	• скобки • знаки +, =, !, ? и др.

Оператор – это команда языка программирова-ния (инструкция).

Оператор присваивания – это команда для записи нового значения в переменную.
Например: a **:=** 5 *При записи нового значения старое стирается!*

Основные типы данных в алгоритмическом языке

- цел — целочисленный тип данных.
- вещ — тип данных с плавающей запятой.
- сим — символьный тип данных.
- лит — строковый тип данных.
- лог — логический тип данных.

Арифметические операции

+ сложение – вычитание * умножение / деление

Ввод значений двух переменных

```
ввод a, b
```

Ввод значений двух переменных.

Вывод данных

```
вывод a          |вывод значения
                 |переменной a
вывод a, нс      |вывод значения
                 |переменной a и переход
                 |на новую строчку
вывод 'Привет!'  |вывод текста
вывод 'Ответ: ', c
 |вывод текста и значения переменной c
вывод a, '+', b, '=', c
```

Условный оператор

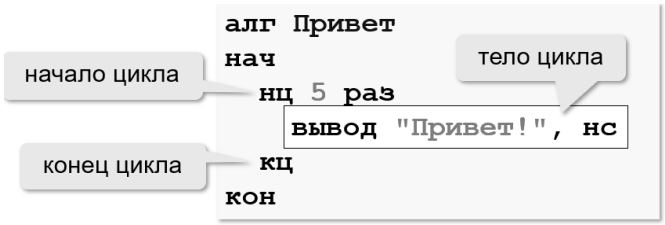
```
если условие то
  | что делать, если условие верно
 иначе
  | что делать, если условие неверно
все
```

(!) Вторая часть (иначе) может отсутствовать!

Циклы

Цикл – это многократное выполнение одинаковых действий.

- Цикл с **известным** числом шагов



- Цикл с **неизвестным** числом шагов (цикл с условием)

Особенности:

- можно использовать сложные условия:

```
нц пока a < 10 и b > 5
  a:= a + 5; b:= b - 2
кц
```

- можно записывать в одну строчку, разделяя команды точкой с запятой:

```
нц пока a < b; b:= b - 2 кц
```

- условие пересчитывается при **каждом** входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a := 4; b := 6
нц пока a > b; a:= a - b кц
```

- если условие никогда не станет ложным, программа **зацикливается**

```
a:= 4; b:= 6
нц пока a < b; d:= a + b кц
```

- **Цикл с переменной**

Особенности:

- переменная цикла может быть только целой (**цел**)
- начальное и конечное значения и шаг – целые
- можно записывать в одну строчку, разделяя команды точкой с запятой:
  нц для n от 1 до 4 ; вывод n кц
- если шаг > 0 и конечное значение < начального, цикл не выполняется ни разу (проверка условия в начале цикла, цикл с предусловием)
- если шаг < 0 и конечное значение > начального, цикл не выполняется ни разу