

Пояснительная записка к презентации «Основы алгоритмизации».

- Данная презентация может быть использована в качестве наглядного пособия при изучении темы «Алгоритмизация» с учащимися 5-9 классов.
- Презентация «Основы алгоритмизации» знакомит учащихся с понятием алгоритма, его свойствами, формами представления, с исполнителями алгоритмов, их средой, основными алгоритмическими конструкциями(на примерах задач), с программной записью алгоритмов. Также с помощью данной презентации учащиеся научатся составлять блок-схемы, переводить алгоритмы на языки программирования.
- Эти знания и умения они смогут применить для решения различных задач при дальнейшем изучении информатики и ИКТ.

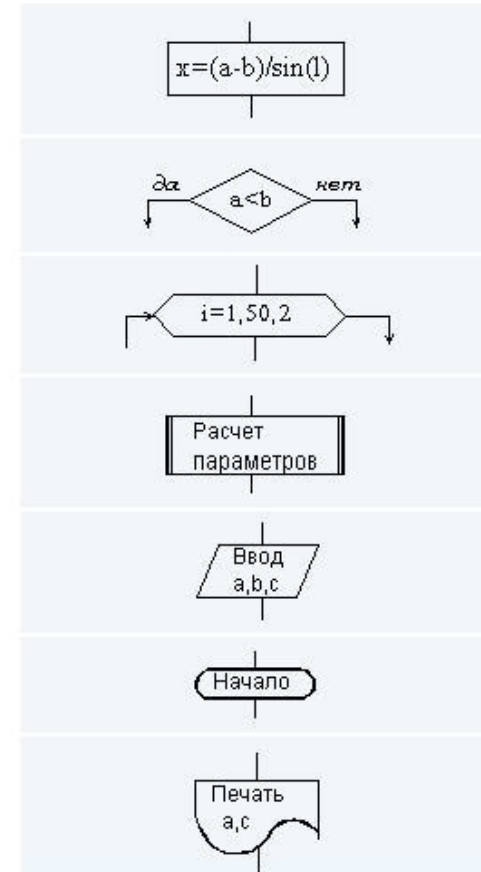
ОСНОВЫ АЛГОРИТМИЗАЦИИ.

Краткие теоретические сведения
в помощь учителю.



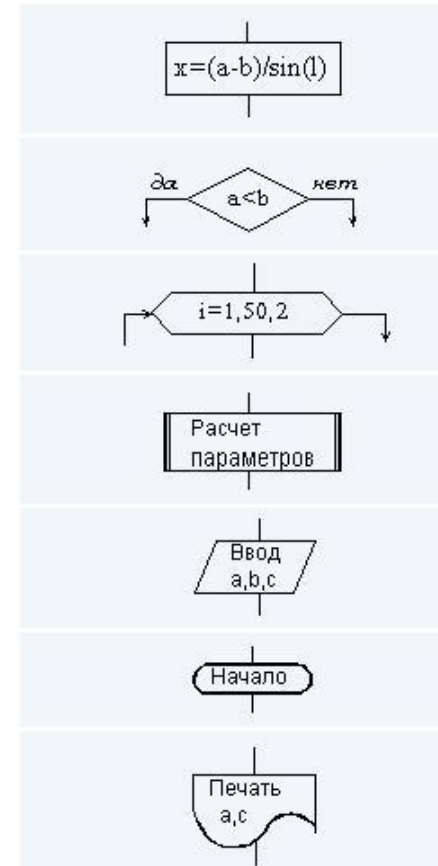
Алгоритм.

- ✓ Определение.
- ✓ Свойства.
- ✓ Исполнители алгоритмов.
- ✓ Формы представления.
- ✓ Основные алгоритмические конструкции.



Алгоритм-это

- Последовательность действий, которые следует выполнить для получения результата поставленной задачи.



Свойства алгоритма:

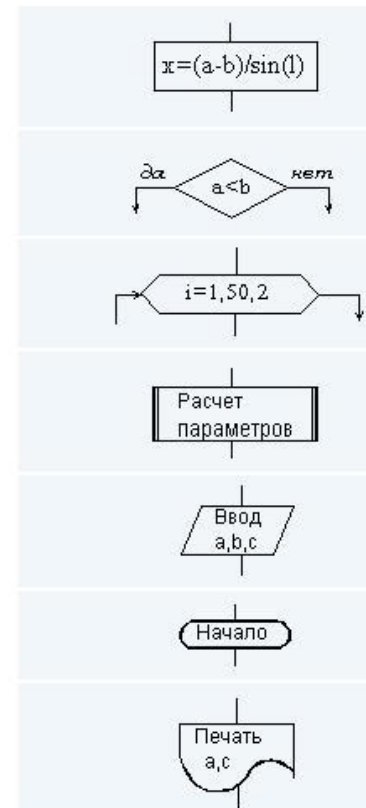
Дискретность
(Разбиение алгоритма
на шаги)

Понятность(каждый шаг
алгоритма должен быть
понятен исполнителю)

Точность(указание
последовательности шагов)

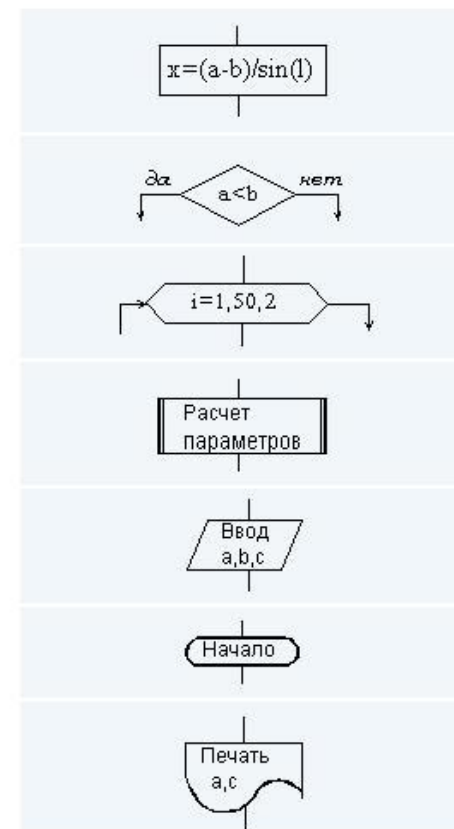
Результативность(получен
результата за конечное число
шагов)

Массовость(исп
алгоритма для р
однотипных зад



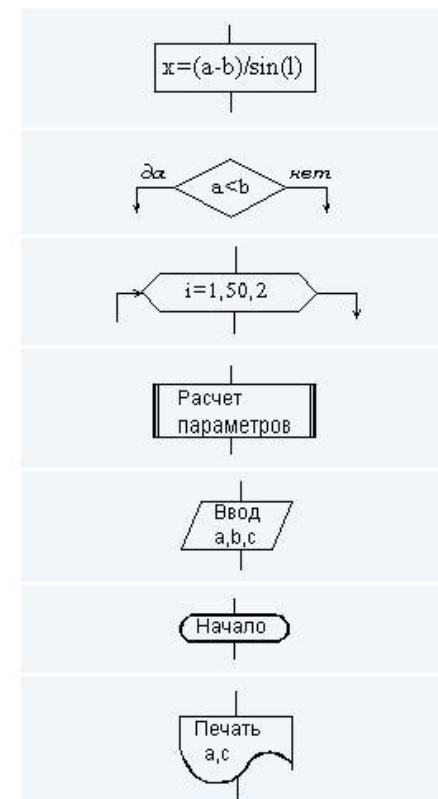
Исполнитель алгоритма

- Некоторая (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.
- Например: человек, компьютер и т.д.



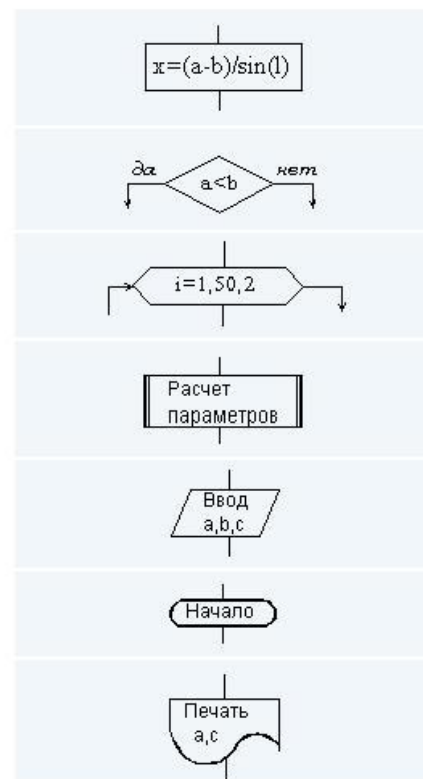
Исполнителя характеристик

- Среда.
- Система команд.
- Элементарные действия.
- Отказы.



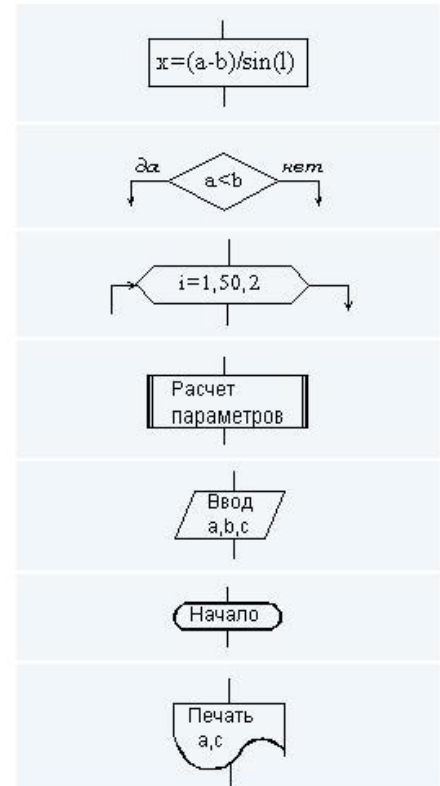
Среда (или обстановка) –

- "место обитания" исполнителя. Например, для исполнителя Робота из школьного учебника среда — это бесконечное клеточное поле. Стены и закрашенные клетки тоже части среды. А их расположение и положение самого Робота задают конкретное состояние среды.



Система команд.

- Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя. Для каждой команды должны быть заданы условия применимости и описаны результаты выполнения команд

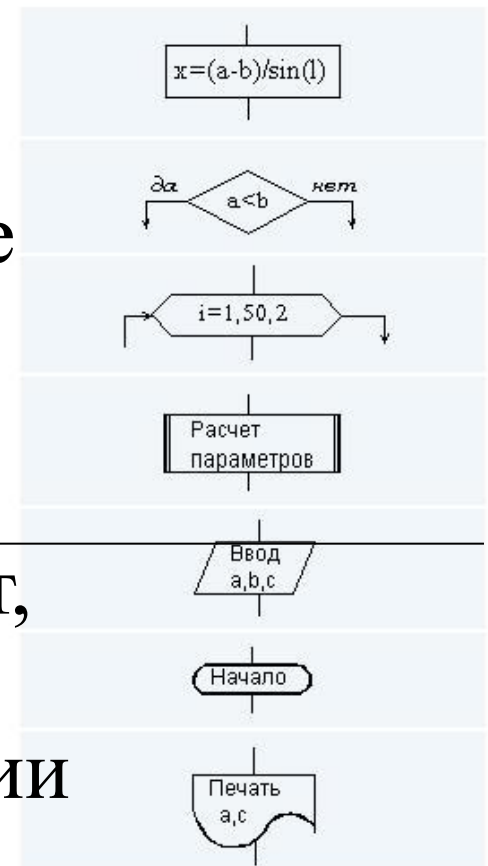


Элементарные действия.

- ✓ После вызова команды исполнитель совершает соответствующее элементарное действие.

Отказы.

- ✓ Отказы исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.



Формы представления.

Словесный
(письменно или
устно)

Графический
(стрелками,
блок-схемами)

Программный

Пример словесного алгоритма:

Алгоритм приготовления настоя шиповника:

- ❑ 1. Столовую ложку сушеных плодов шиповника измельчить.
- ❑ 2. Залить стаканом кипящей воды.
- ❑ 3. Кипятить 10 минут на слабом огне.
- ❑ 4. Охладить.
- ❑ 5. Процедить.

Основные алгоритмические структуры.

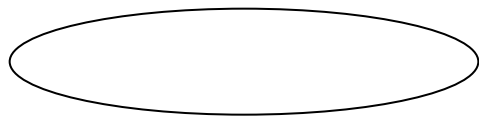
```
graph TD; A[Основные алгоритмические структуры.] --> B[Линейный алгоритм<br/>(следование)]; A --> C[Разветвляющийся алгоритм<br/>(ветвление)]; A --> D[Алгоритм с повторением<br/>(циклический)];
```

Линейный алгоритм
(следование)

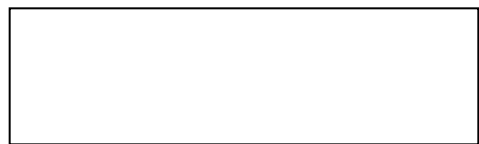
Разветвляющийся алгоритм
(ветвление)

Алгоритм с повторением
(циклический)

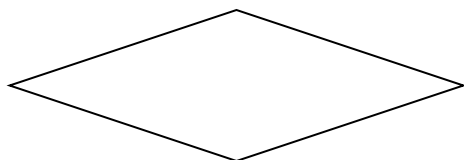
Стандартные фигуры, используемые при составлении алгоритмических структур :



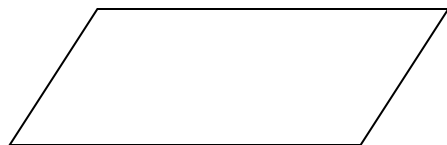
- начало(конец) алгоритма $x = (a-b)/\sin(l)$



- действие



- проверка условия

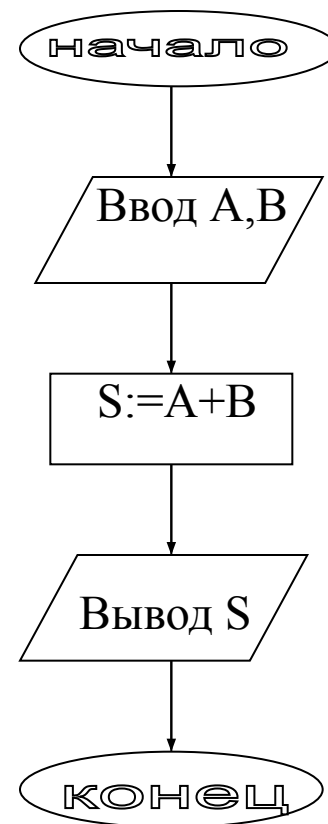


- ВВОД ИЛИ ВЫВОД ДАННЫХ

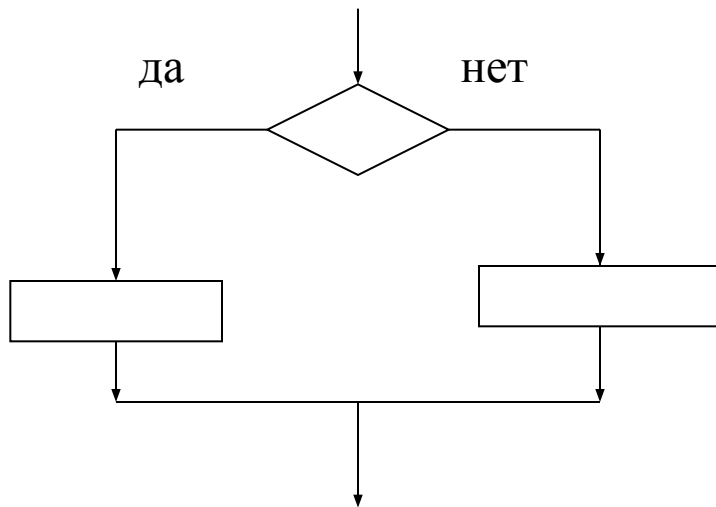


Линейный алгоритм(следование, когда команды выполняются строго одна за одной).

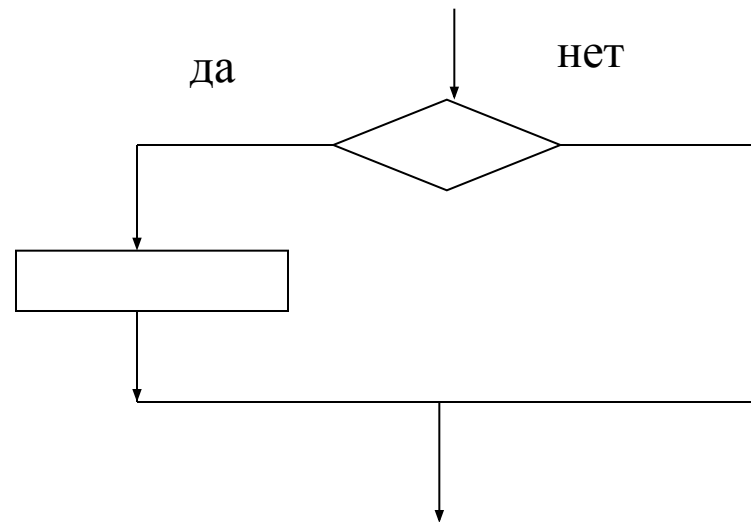
Пример: Даны два числа.
Вычислить их сумму.



Разветвляющийся алгоритм (ветвление, когда в алгоритме содержится какое-либо условие и приходится делать выбор действий в зависимости от этого условия).



(полная форма)

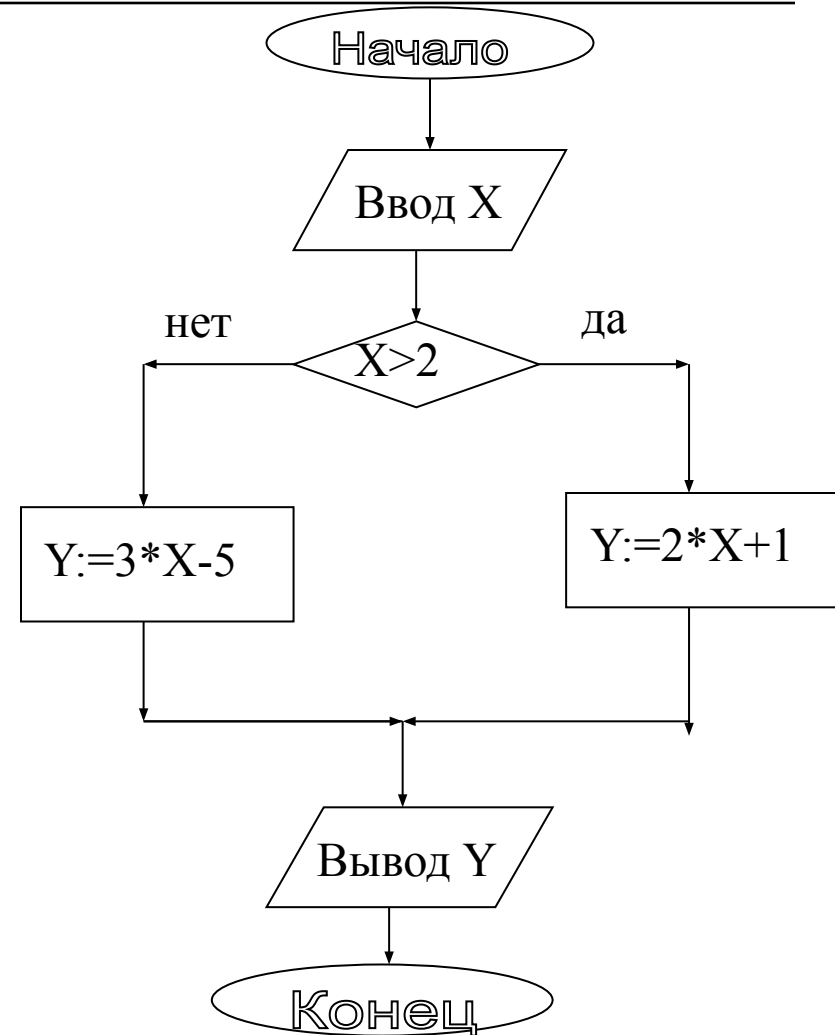


(сокращенная форма)

Полная форма разветвляющегося алгоритма:

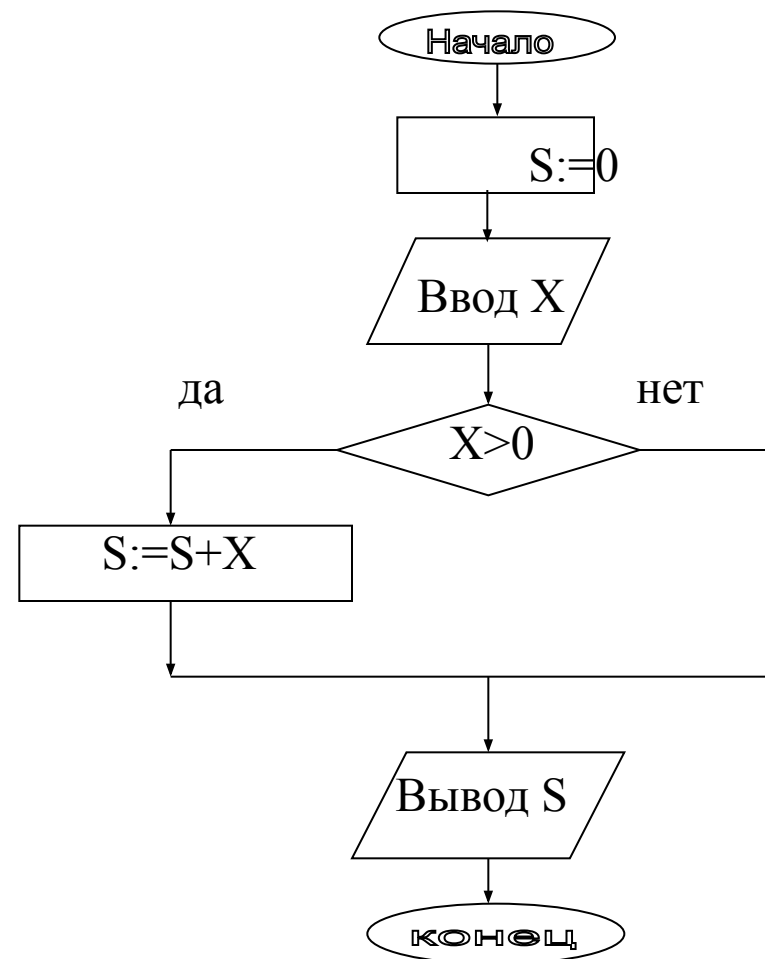
Пример: Вычислить по формулам значения Y , если известен X .

$$Y = \begin{cases} 2X+1, & \text{если } X > 2 \\ 3X-5, & \text{если } X \leq 2 \end{cases}$$



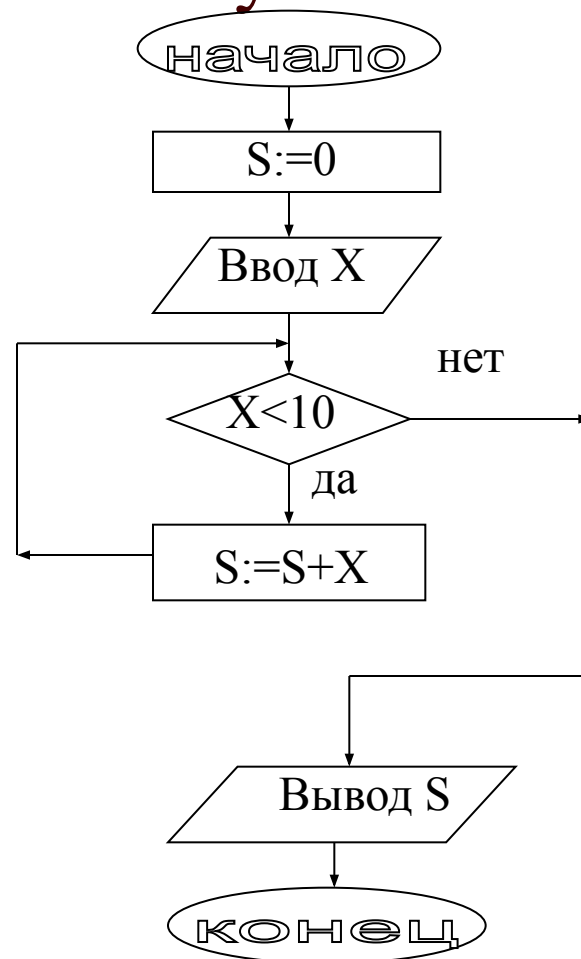
Сокращенная форма разветвляющегося алгоритма:

- Пример: Вычислить сумму положительных чисел.



Алгоритм с повторением(циклический), когда
одни и те же действия выполняются
несколько раз при определенном условии

Пример: Найти сумму
чисел, меньших 10.



Пример алгоритма на языке программирования QB 4.5.

Даны длина и
ширина прямоугольника.
Вычислить площадь
и периметр фигуры.

```
CLS
INPUT "Введи 2
      величины: ",a,b
P=(a+b)*2
S=(a*b)
PRINT "Периметр =";P
PRINT "Площадь =";S
```

Заключение:

- Познакомившись с основами алгоритмизации учащиеся смогут применить полученные знания для решения различных задач и уроках ИКТ и информатик

