

**Учитель информатики
МКОУ «Каменная средняя
общеобразовательная школа»
Н.С. Чернышова**

АЛГОРИТМЫ





Цель урока:



Создание условий для формирования первичного представления об алгоритме, о его исполнении, о вспомогательных алгоритмах.





Задачи урока:

- 
- Учебная: познакомиться с понятием алгоритма, вспомогательного алгоритма, исполнение алгоритма, переменной.
 - Развивающая: развитие алгоритмического мышления, памяти, внимательности.
 - Воспитательная: развитие трудолюбия, навыков самостоятельной работы.
- 
- 



Подготовка к уроку:

- 
- Каждому ученику на стол выдается лист опорного конспекта;
 - Составить разноуровневые задания по теме.
- 
- 



Ход урока:

- 
1. Организационный момент.
 2. Постановка целей урока.
 3. Объяснение темы (презентация).
 4. Обобщение, закрепление изученного (работа в тетради, выполнение упражнений).
 5. Домашнее задание.
- 
- 

*Изучив эту
тему, вы
узнаете:*

- В чем состоит назначение алгоритма и каковы его основные свойства;
- Какие типовые конструкции алгоритма существуют;
- Как представить алгоритм в виде блок-схемы;
- Каковы стадии разработки алгоритма.

Понятие алгоритма



- Появление алгоритма связывают с зарождением математики. Более 1000 лет назад (в 825 году) ученый из города Хорезма Абдулла (или Абу Джафар) Мухаммед бен Муса аль-Хорезми создал книгу по математике, в которой описал способы выполнения арифметических действий над многозначными числами. Эти способы и сейчас изучают в школе.
- Само слово «алгоритм» возникло в Европе после перевода на латынь книги этого среднеазиатского математика, в которой его имя писалось как «Алгоритми».
- Научное определение понятия алгоритма дал А.Черч в 1930 году. Позже и другие математики вносили свои уточнения в это определение.
- В школьном курсе информатики мы будем пользоваться следующими определениями:





Алгоритм -

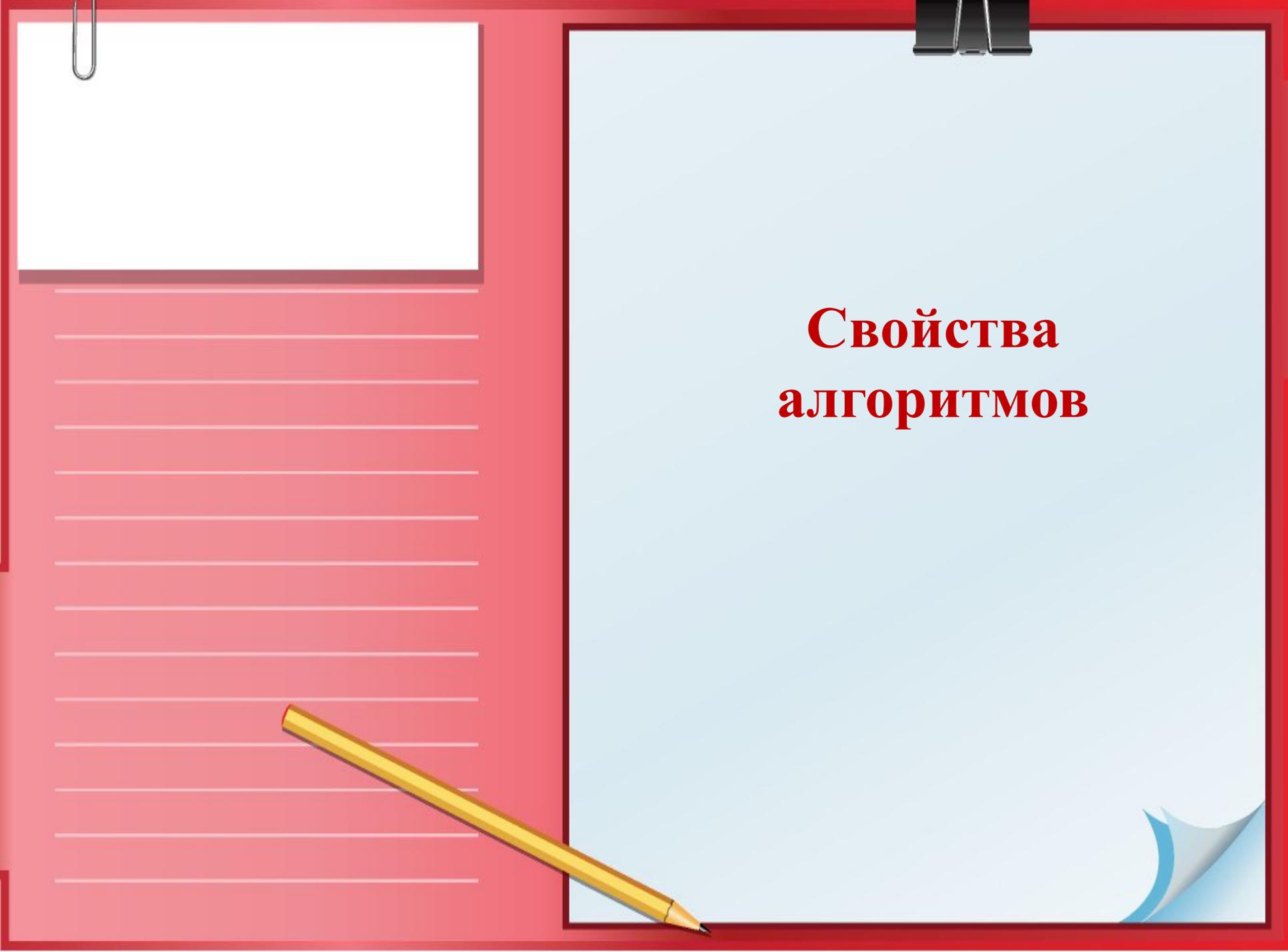


Описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов

Алгоритмизация -

Процесс разработки алгоритма для решения задачи





Свойства алгоритмов

Дискретность

От лат. Diskretus –
разделенный,
прерывистый.

это свойство указывает,
что любой алгоритм
должен состоять из
конкретных действий,
следующих в
определенном порядке.

Детерминированность

От лат. Determinante – определенность, точность.

Это свойство указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.



Массовость



Это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными.

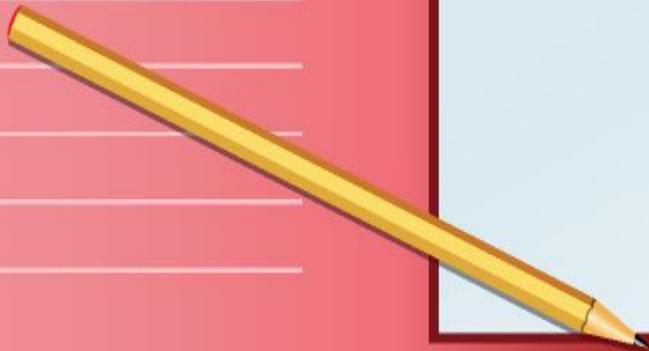




Результативность



Это свойство требует,
чтобы в алгоритме не
было ошибок.





Конечность



Это свойство определяет,
что каждое действие в
отдельности и алгоритм
в целом должны иметь
возможность
завершения



Типовые конструкции алгоритма

Предположим, требуется составить алгоритм вычисления результата выражения: $100 + 15 - 40 + 20$

1. Сложить числа 100 и 15.
2. Из полученной суммы вычесть 40.
3. К результату прибавить 20.

В этом примере действия выполняются в том порядке, в котором записаны. Подобные алгоритмы получили название линейных, или последовательных.



Линейный
(последовательный)
алгоритм



– описание действий,
которые выполняются
однократно в заданном
порядке.





Многие процессы основаны на многократном повторении одной и той же последовательности действий. Допустим, Робот обучен красить забор. Он последовательно закрашивает доску за доской. Для Робота составлен следующий алгоритм:

1. Покрасить доску.
2. Переместиться к следующей доске.
3. Перейти к действию 1.

Робот, закрасив одну доску, перейдет ко второй, затем к третьей и т.д. Робот не сможет закончить работу, т.к. алгоритм не предусматривает окончания работы. В приведенном примере необходимо добавить в алгоритм действие по анализу результата:

1. Покрасить доску.
2. Если есть еще доска, переместиться к следующей; перейти к действию 1.
3. Если доски закончились, завершить работу.

-описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Перечень повторяющихся действий называется телом цикла.



**Циклический
алгоритм**



Вспомните сюжет из русской сказки.

Царевич останавливается у развилки дороги и видит камень с надписью: «Направо пойдешь- коня потеряешь, налево пойдешь- сам пропадешь...». Здесь видна ситуация , заставляющая принимать решение в зависимости от некоторого условия.

- *это выражение, находящееся между словом «если» и словом «то» и принимающее значение «истина» или «ложь».*

— алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Условие

**Разветвляющийся
алгоритм**

В общем случае схема разветвляющегося алгоритма будет выглядеть так: «если условие, то....., иначе.....»

В разветвляющемся алгоритме при невыполнении условия действия могут не предусматриваться.

Такое представление алгоритма получило название

Тогда это будет

ПОЛНОЙ ФОРМЫ

НЕПОЛНАЯ ФОРМА

Например, «...идет направо – песнь заводит, налево – сказку говорит...»

Например, водитель едет по шоссе, если бензин на исходе, то он заезжает на ближайшую автозаправочную станцию



Вспомогательный алгоритм

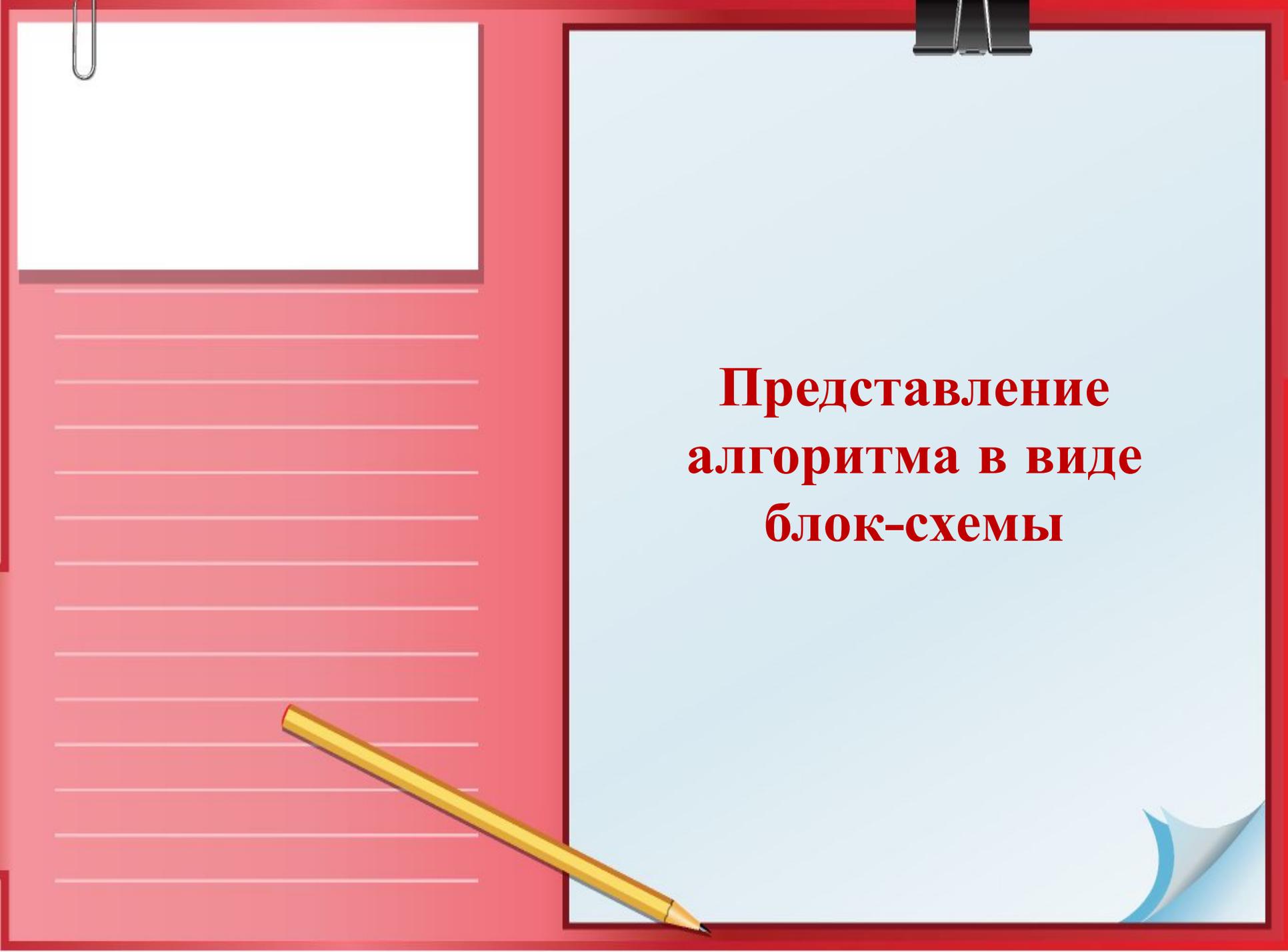


Если в процессе алгоритмизации удастся выделить более простые этапы и для каждого из них установить промежуточные цели, то для их достижения рекомендуется разрабатывать вспомогательные алгоритмы, которым можно дать уникальные имена. Итоговый алгоритм выглядит как связанные между собой вспомогательные алгоритмы, представленные только своими именами, причем описания самих вспомогательных алгоритмов хранятся отдельно.

– алгоритм, который можно использовать в других алгоритмах, указав только его имя.

Вспомогательному алгоритму должно быть присвоено имя.





**Представление
алгоритма в виде
блок-схемы**

**Линейная
алгоритмическая
конструкция**

начало

После школы иду гулять

Возвращаюсь домой

Делаю уроки

конец



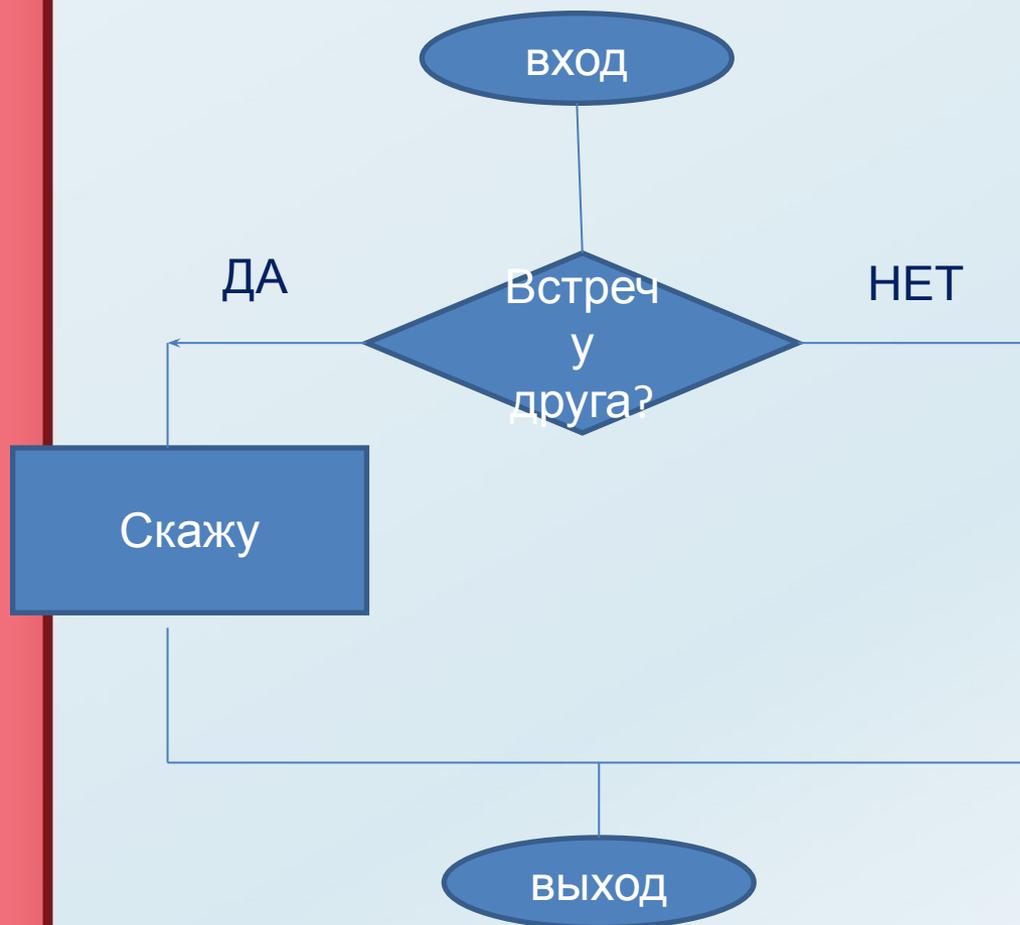
**Циклическая
алгоритмическая
структура, в которой
условие поставлено в
начале цикла**



**Циклическая
алгоритмическая
структура, в которой
условие поставлено в
конце цикла**



**Неполная форма
разветвляющегося
алгоритма**



**Полная форма
разветвляющегося
алгоритма**

