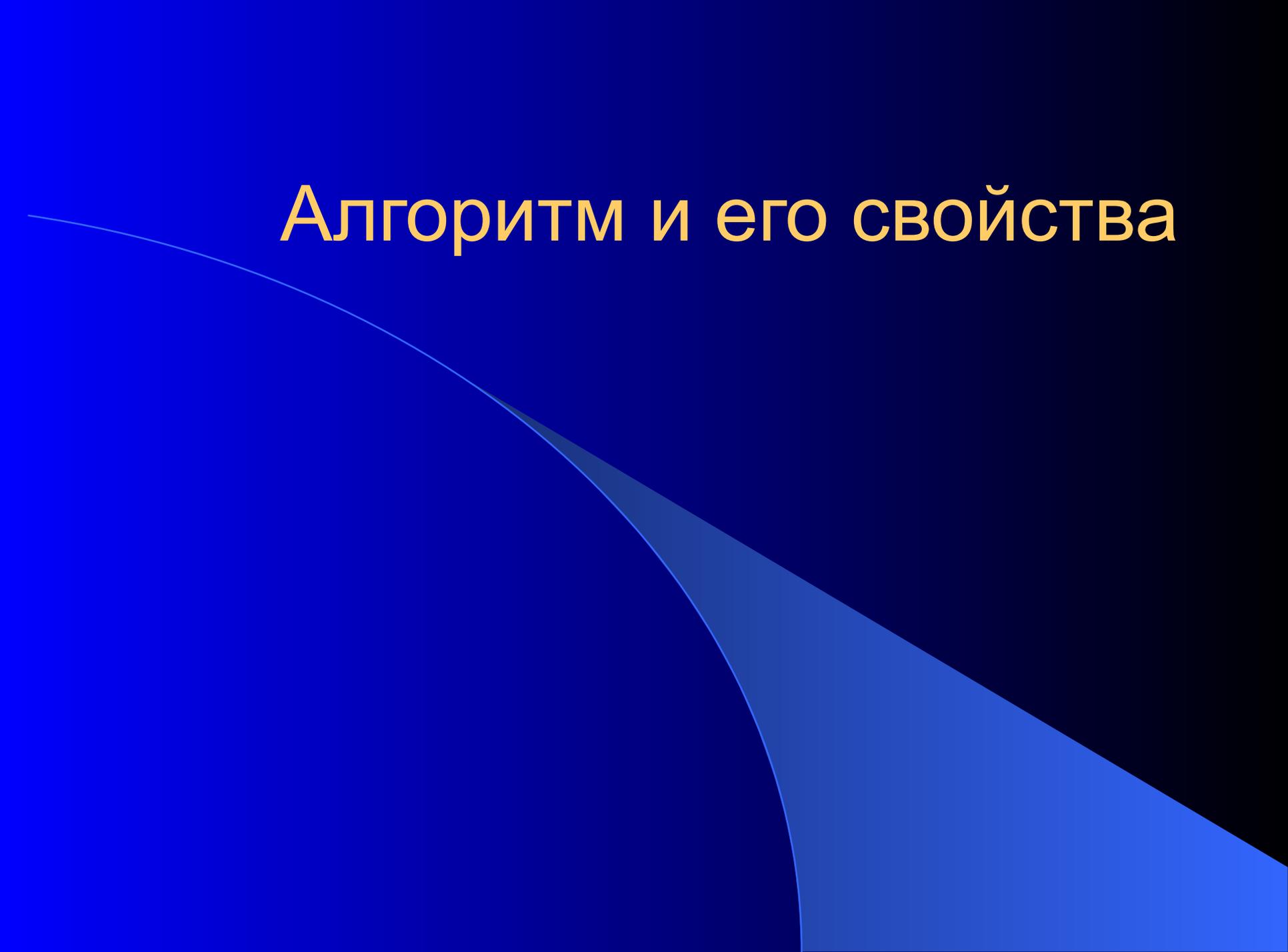


# Алгоритм и его свойства

A decorative graphic element consisting of a blue gradient shape that starts as a thin line on the left and curves downwards and to the right, ending as a solid blue area in the bottom right corner.

*Алгоритм — это конечное  
упорядоченное множество  
правил, однозначно определяющих  
последовательность действий  
для решения задачи*

## Выделяют три крупных класса алгоритмов:

- **вычислительные** алгоритмы, работающие со сравнительно простыми видами данных, такими как числа и матрицы, хотя сам процесс вычисления может быть долгим и сложным;
- **информационные** алгоритмы, представляющие собой набор сравнительно простых процедур, работающих с большими объемами информации (алгоритмы баз данных);
- **управляющие** алгоритмы, генерирующие различные управляющие воздействия на основе данных, полученных от внешних процессов, которыми алгоритмы управляют.

# Свойства алгоритмов

- Дискретность состоит в том, что процесс решения задачи по алгоритму может быть разбит на отдельные действия связанные между собой. (Этот процесс можно свести к последовательности элементарных операций, как бы не был сложен алгоритм)

# Свойства алгоритмов

- Результативность состоит в том, что правильное применение алгоритма должно привести к получению точного ответа или выдачи сообщения о невозможности решения задачи по заданным исходным данным.

# Свойства алгоритмов

- Детерминированность состоит в том, что применение алгоритма к одним и тем же исходным данным должно приводить к одним и тем же исходным данным.

# Свойства алгоритмов

- *Массовость* проявляется в том, что алгоритм применим к различным наборам исходных данных, а не к одному какому-нибудь варианту.

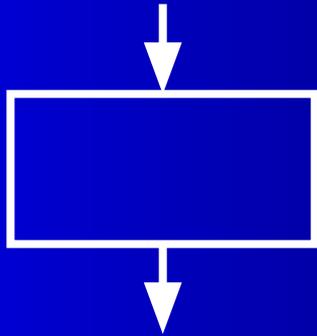
# Способы представления алгоритмов

- *Словесный (тривиальный) способ* – запись алгоритма на естественном языке
- *Графический способ* – запись алгоритма в виде схемы, состоящей из геометрических фигур, называемых *блоками*, и соединительных линий. Такие схемы часто называют *блок-схемами*.
- *Программный способ* – запись алгоритма на каком-либо языке программирования.

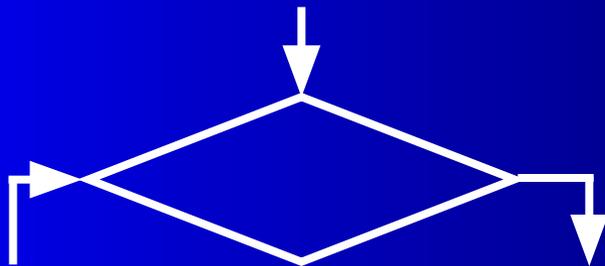
# Тривиальный способ

- 1) Задать три значения  $X, Y, Z$ . Перейти к пункту 2.
- 2) Сравнить  $X$  и  $Y$ . Если  $X$  больше  $Y$ , то  $X$  примем за наибольшее. Иначе за наибольшее примем  $Y$ . Перейти к пункту 3.
- 3) Сравнить  $Z$  с наибольшим среди первых двух значений. Если  $Z$  больше, чем максимальное среди предыдущих, его принимаем за максимум. Перейти к пункту 4.
- 4) Вывести максимальное значение.

# Блоки, используемые при составлении блок-схем

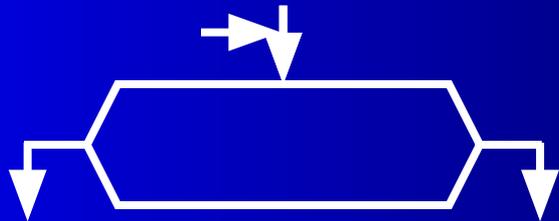


- Элементарное действие

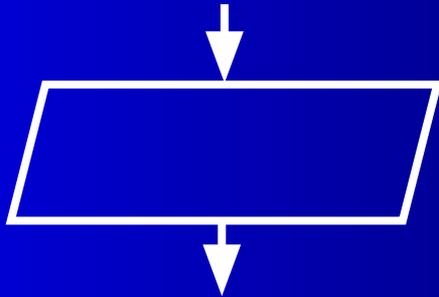


- Условие

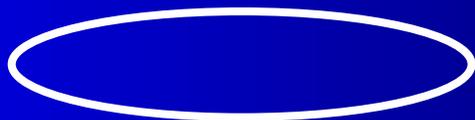
# Блоки, используемые при составлении блок-схем



- Цикл



- Ввод-вывод



- Начало, конец алгоритма

# СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

- уменьшение трудностей при тестировании;
- повышение производительности труда программистов;
- улучшение читабельности и ясности программ.

# СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

- Логическая структура любой программы может быть выражена комбинацией трех базовых структур: следование, развилка и цикл.
- Избегать не оправдано сложных арифметических выражений
- Использовать комментарии для документирования программ.
- Не использовать оператор GO TO

# Базовые структуры

- Алгоритм любой сложности можно реализовать с помощью 3 базовых структур: следование разветвление и цикл.
- Все базовые структуры имеют один вход и один выход

# Следование

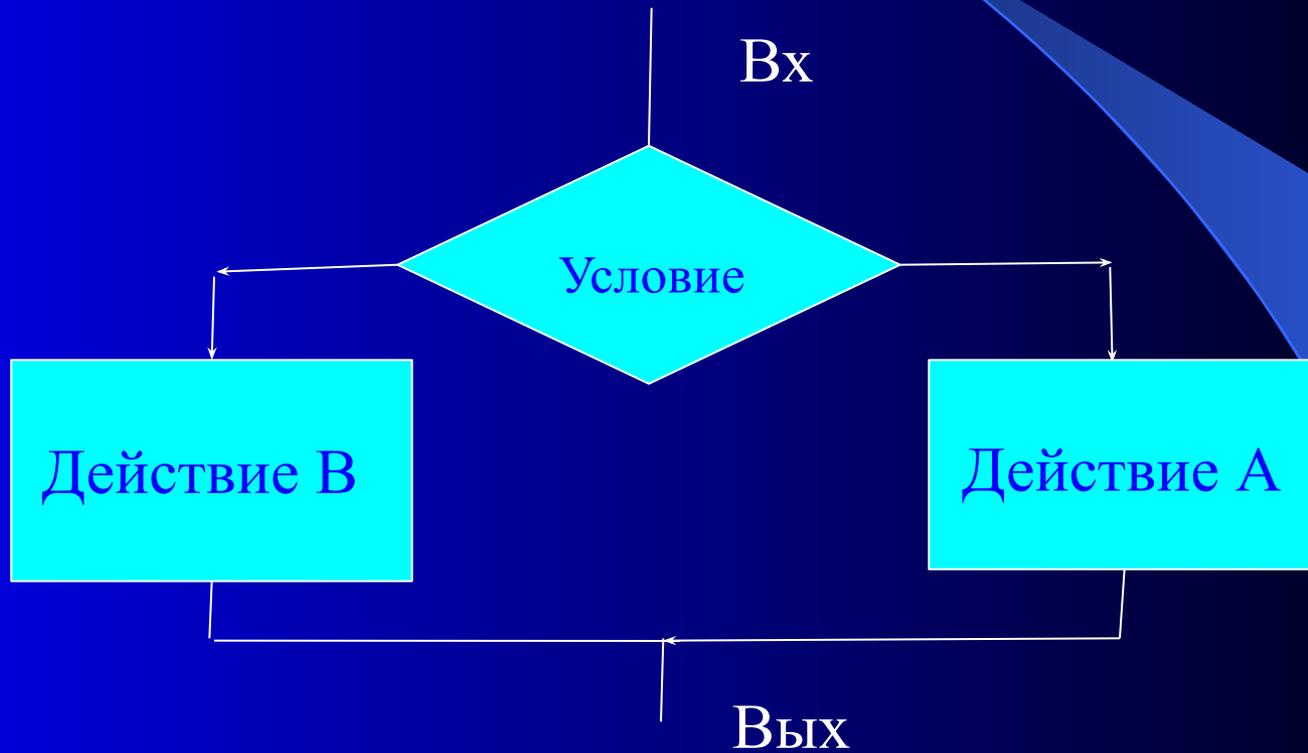
Вх

Действие А

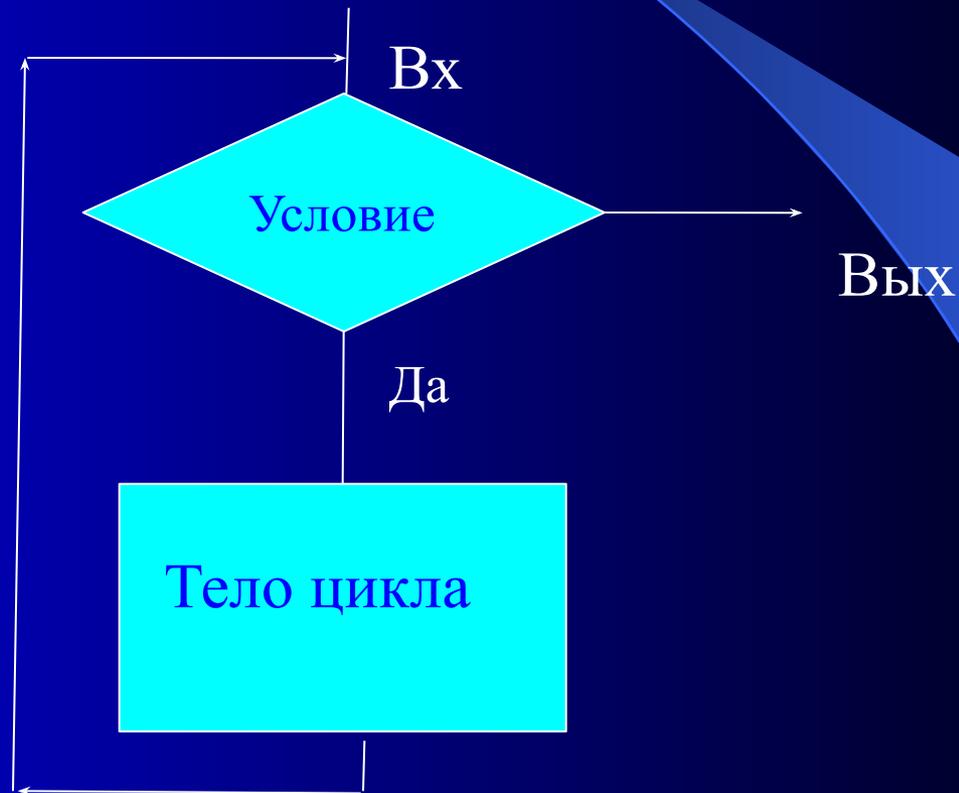
Действие В

Вых

# Разветвление



# Цикл в предусловии



# Цикл в постусловии

