

# БЛОК-СХЕМЫ



Школа::Кода  
ОСНОВЫ КОМПЬЮТЕРНОЙ  
ГРАФИКИ И  
РАЗРАБОТКИ ИГР  
2022-2023 Таганрог

# Определение

- Блок-схема — распространённый тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности.

# Почему важно создавать блок-схемы?

- 1> Чтобы быстро ввести в курс дела новых людей.
- 2> Чтобы увидеть потенциальные ошибки дизайна на ранней фазе разработки продукта.
- 3> Блок-схемы это, как говорит старинная американская народная поговорка, "helicopter view". Глядя на схему можно увидеть архитектуру проекта с высоты птичьего полета.
- 4> Чтобы упростить навигацию по сложной программе

# Основные блоки



Начало и конец алгоритма



Ввод исходных данных, вывод результатов



Какое-либо действие, например вычисление

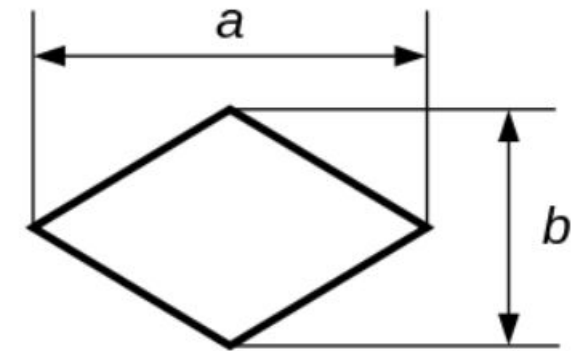


Проверка условия

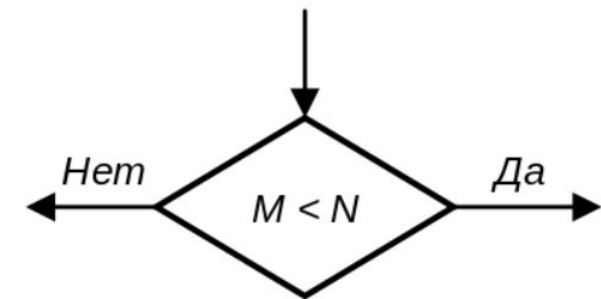
# Вопрос (условие или решение)

- Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определённых внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.

Начертание



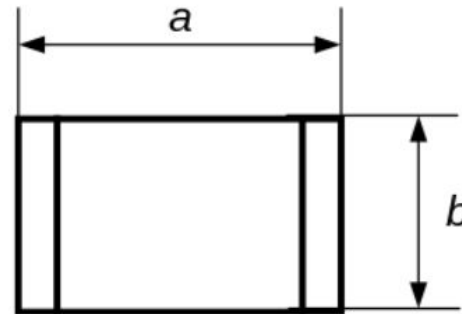
Пример #1



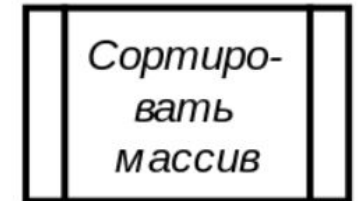
# Предопределённый процесс (функция)

- Символ отображает предопределённый процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле). Например, в программировании – вызов процедуры или функции.

*Начертание*

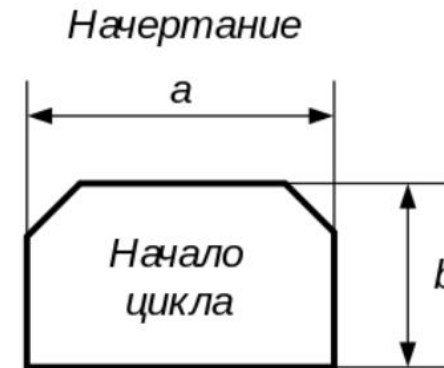


*Пример*

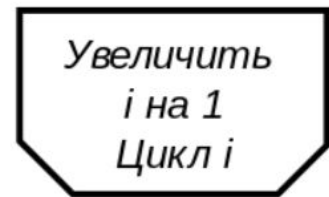
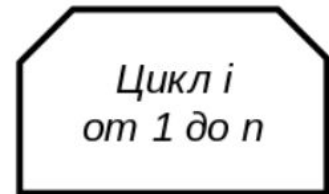


# Цикл

- Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие

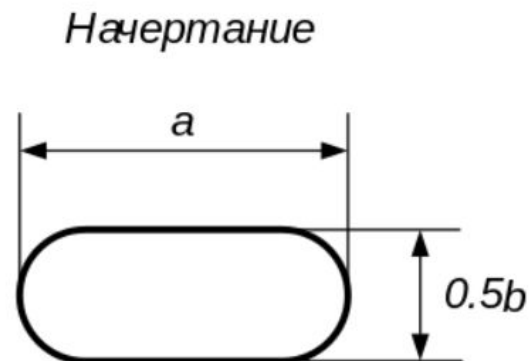


Пример

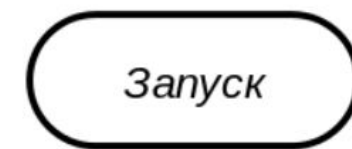


# Ограничитель\*

- Символ отображает вход из внешней среды и выход во внешнюю среду (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).



*Пример*

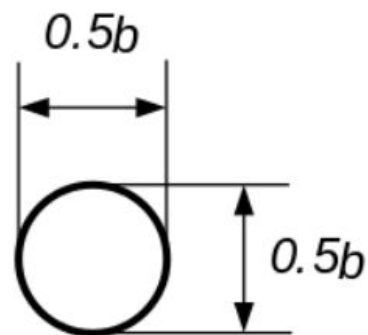




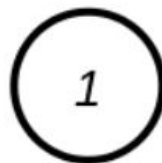
# Соединитель\*

- Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения её в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.

*Начертание*



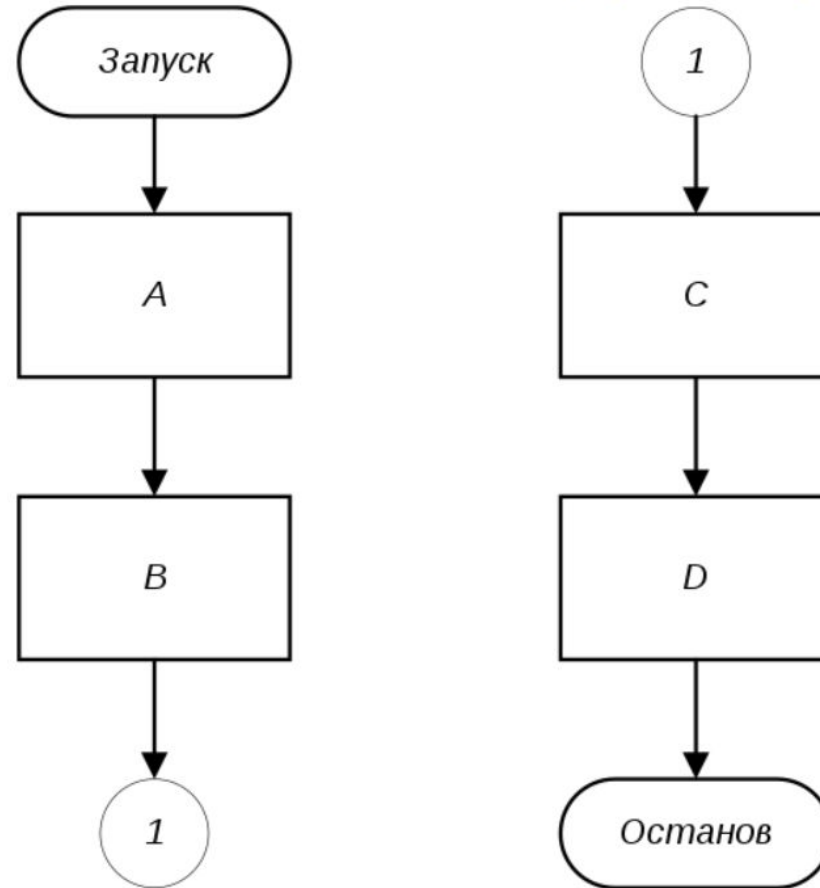
*Пример #1*



*Пример #2*

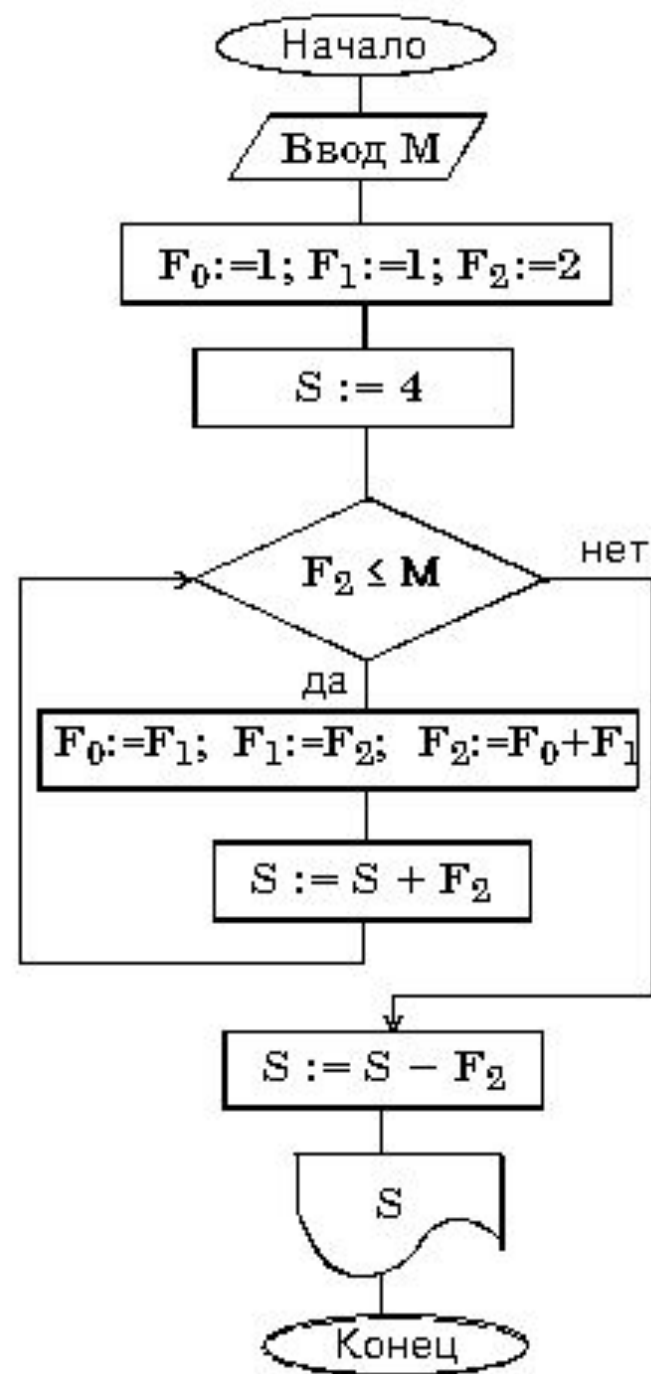


# Пример использования соединителей



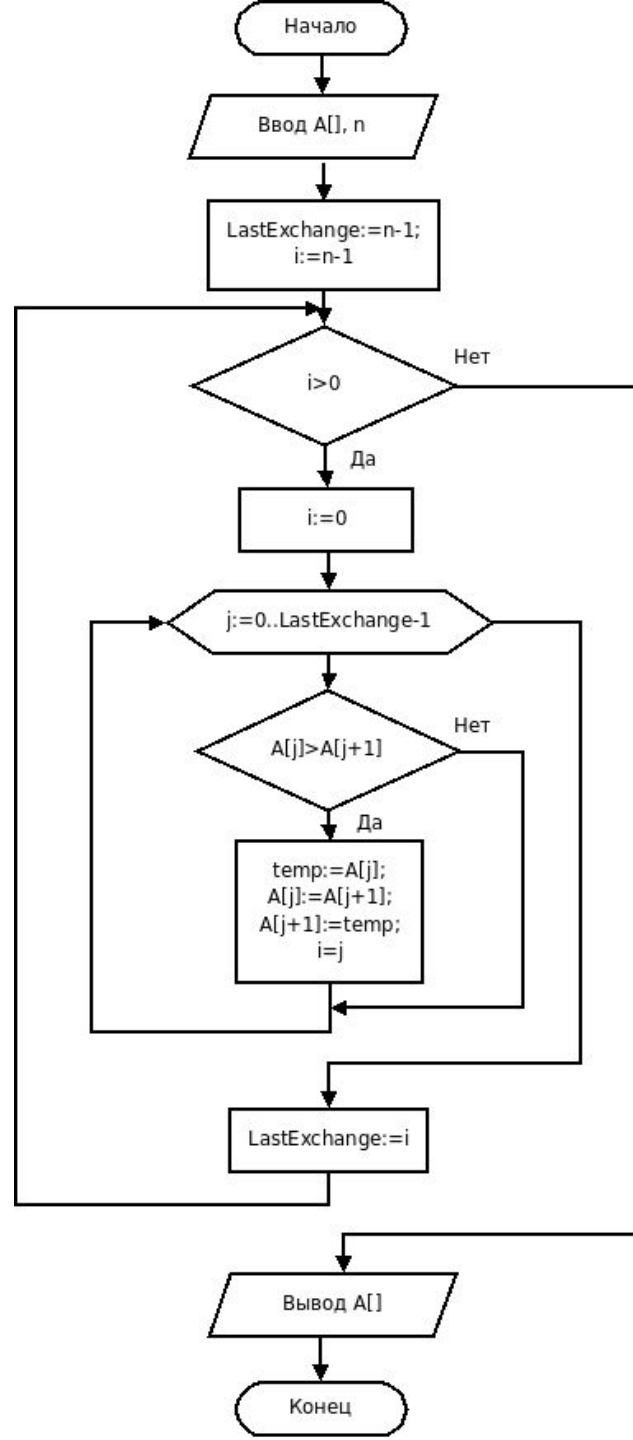
# Примеры

- Алгоритм чисел Фибоначчи



# Примеры

- Сортировка пузырьком

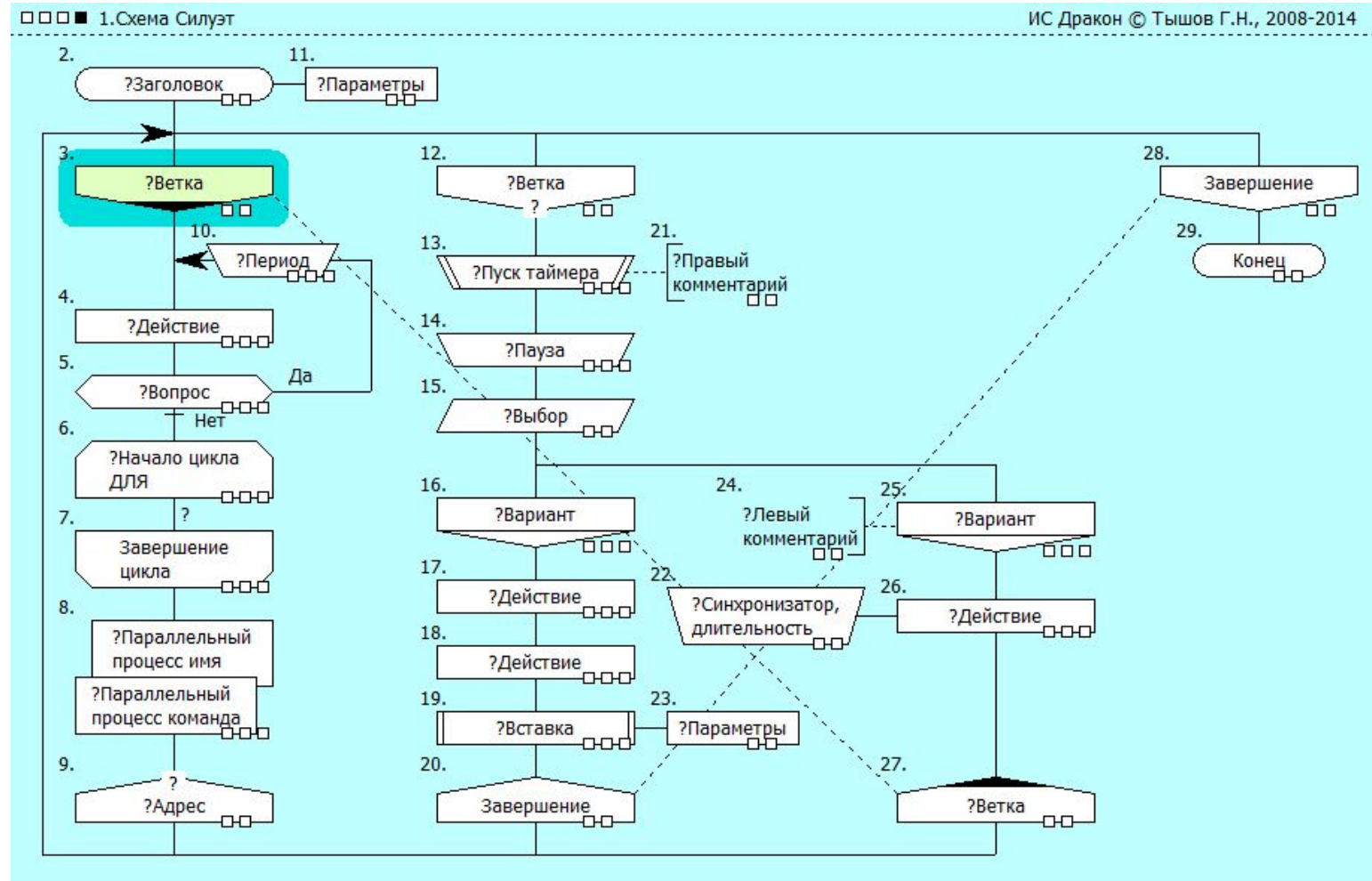


# Применение в промышленности

- Язык программирования дракон, использовался при запуске космического шаттла, так же программирование блок-схемами очень удобно на больших предприятия
- Подобные языки используются при контроле состояний станков, электростанций и при отладке программ.



# Пример программы на ДРАКОН



# Инструменты для создания

Draw.io

<https://app.diagrams.net/>

Program for you:

<https://programforyou.ru/block-diagram-redactor>

Microsoft Visio