

Урок 48. Блок-схемы алгоритмов

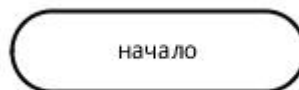
Блок-схема — распространённый тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности.

Сплошная линия отражает зависимости по управлению между символами и может снабжаться стрелкой. *Пунктирная линия* используется для соединения символа с комментарием.

1. Основные элементы блок-схем алгоритмов

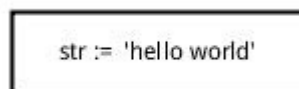
1.1. Ограничитель (терминатор)

Символ отображает вход из внешней среды и выход во внешнюю среду. На практике имеют смысл следующие описания ограничителей: *начало/конец*, *запуск/останов*, *перезапуск* (подразумевает перезапуск данной блок-схемы), *ошибка* (подразумевает завершение алгоритма с ошибкой), *исключение* (подразумевает исполнение программного исключения).



1.2. Действие (обработка данных, процесс, блок операций, функциональный блок)

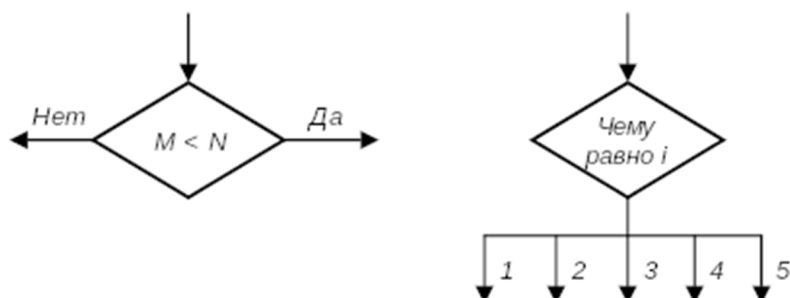
Символ отображает функцию обработки данных любого вида. В блоке операций обычно размещают одну или несколько операций присваивания, не требующих вызова внешних функций. (Впрочем, можно для упрощения схемы все операции размещать в таких прямоугольных блоках)



1.3. Вопрос (условие, решение, альтернативный блок)

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.

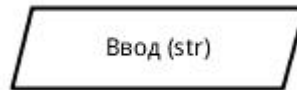
В случае, если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения — «да/нет». Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах — значения этой переменной.



2. Дополнительные элементы (для более сложных схем)

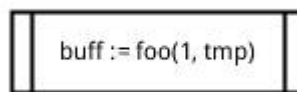
2.1. Данные (ввод/вывод)

Символ отображает преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).



2.2. Предопределённый процесс (функция, процедура, подпрограмма)

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.



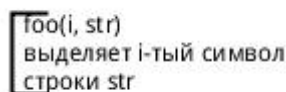
2.3. Соединитель

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.

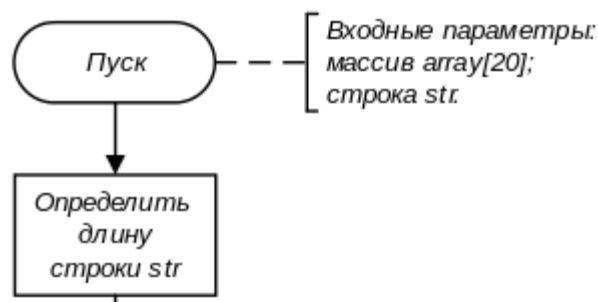


2.4. Комментарий

Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обходить группу символов. Текст комментариев или примечаний должен быть помещён около ограничивающей фигуры.

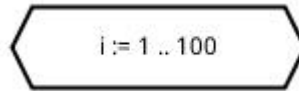


Комментарии используют совместно с терминаторами для описания входных аргументов алгоритма при описании функций.



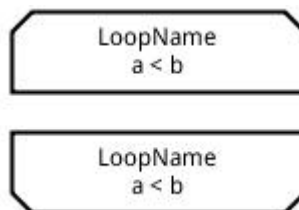
2.5. Подготовка данных

Символ «подготовка данных» в произвольной форме задает входные значения. Используется обычно для задания циклов со счетчиком (for).

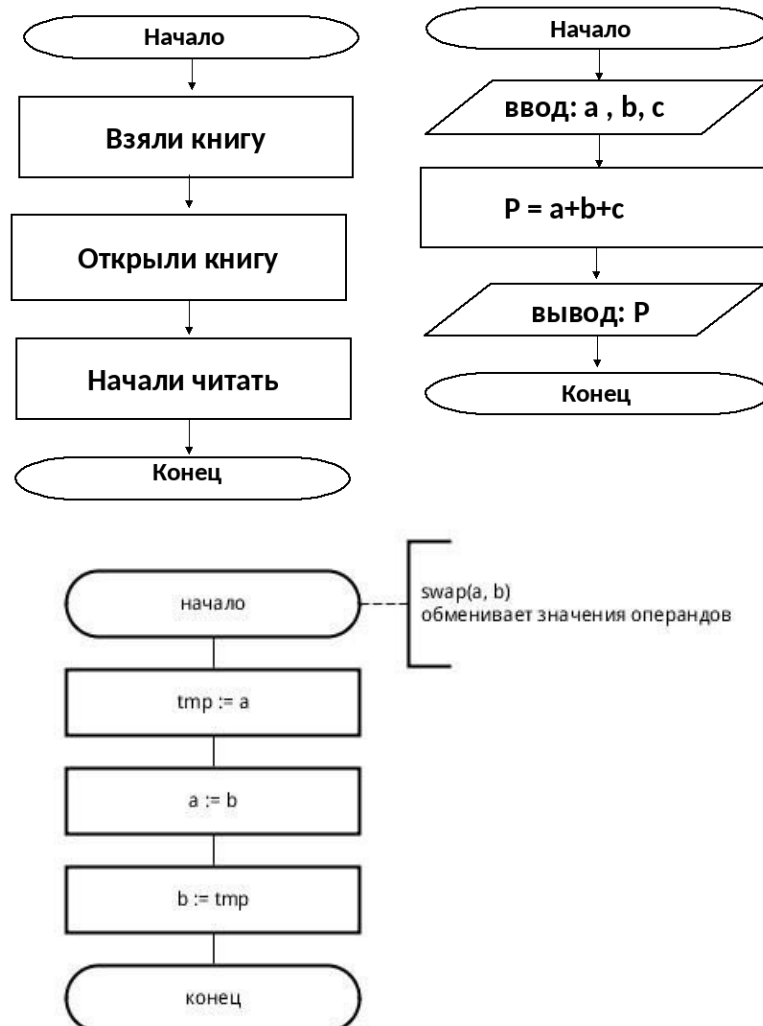


2.6. Цикл

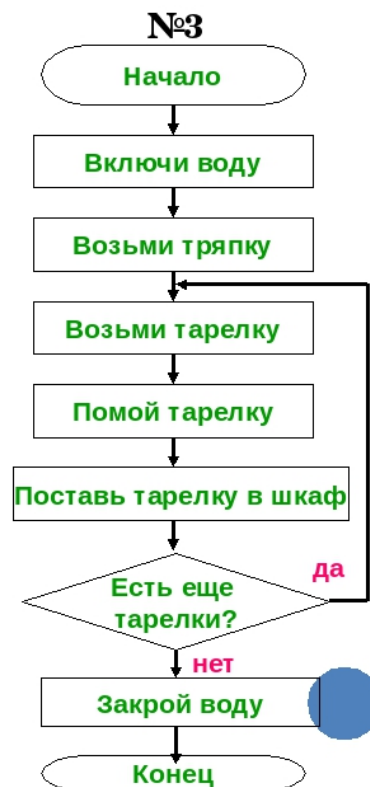
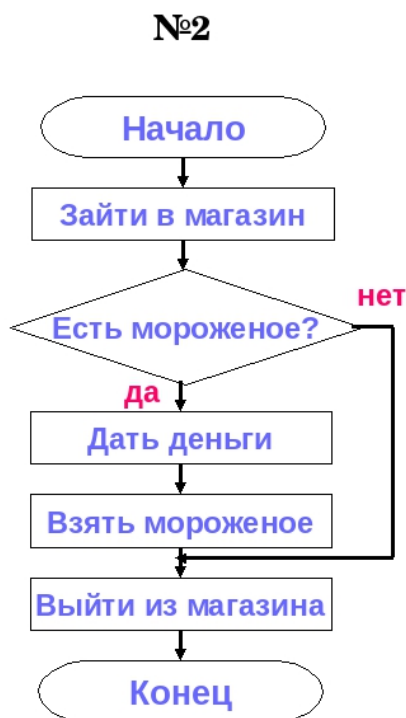
Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие. Расположение условия определяет тип оператора, соответствующего символам на языке высокого уровня — оператор с предусловием (while) или постусловием (repeat ... until).



3. Примеры блок-схем алгоритмов



Задание: определите тип и дайте название каждому алгоритму.

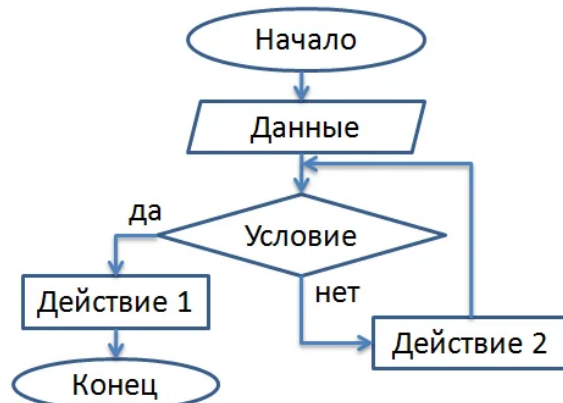


4. Виды алгоритмов

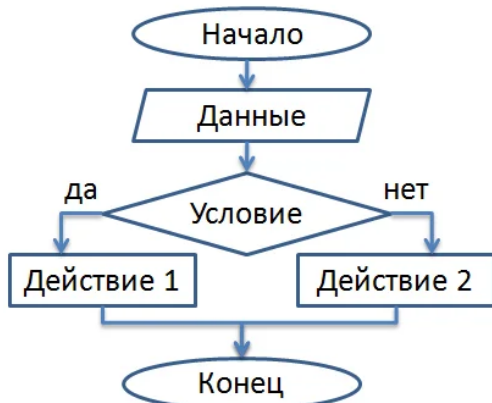
Линейный



Циклический



Разветвляющийся



Распараллеливающийся

