

Алгоритмы и исполнители

Мышаева В.Д

учитель информатики
МОУ СОШ №7



С. Марьины Колодцы
2007 год

Содержание

✓ Алгоритмы и исполнители

- Основные понятия
- Свойства алгоритмов
- Способы записи алгоритмов

✓ Конструирование алгоритмов

- Основные алгоритмические конструкции
- Соединение основных алгоритмических конструкций
- Простейшие примеры
- Задачи на соединение основных алгоритмических конструкций



Основные понятия

Алгоритм:

понятное и точное предписание исполнителю выполнить порядок действий, направленных на решение конкретной задачи или достижение поставленной цели

Алгоритм состоит из законченных действий, называемых командами

Команды выполняются одна за другой

Исполнитель :

живое существо или технический объект, выполняющий команды алгоритма



Основные понятия

Исполнитель

Формальный

не вносит никаких изменений в алгоритм



Не формальный

Может вносить изменения в алгоритм



Основные понятия

Основные характеристики исполнителя

СКИ (система команд исполнителя):

набор команд, которые исполнитель понимает и может выполнить

Среда:

условия, в которых исполнитель может выполнять команды

Отказы:

- 1) «Не понимаю» - команда не входит в СКИ
- 2) «Не могу» - нарушение среды



[к содержанию](#)

Свойства алгоритмов

- Дискретность – каждая команда должна быть выполнена прежде, чем исполнитель перейдет к выполнению следующей
- Понятность – каждая команда должна входить в СИ
- Точность (определенность) – команда должна пониматься исполнителем однозначно
- Результативность – выполнение всех команд алгоритма должно привести к решению конкретной задачи за конечное число шагов
- Массовость – по одному и тому же алгоритму можно решать однотипные задачи



[к содержанию](#)

Способы записи алгоритмов

1. Словесный – для записи используются специальные формальные языки с ограниченным набором слов и строгими правилами записи
2. Формульный
3. Словесно-формульный
4. Графический – в виде блок-схемы

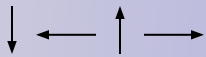


Способы записи алгоритмов

Блок-схема:

Каждая команда записывается с использованием графических символов

Условные обозначения:



указывают порядок действий



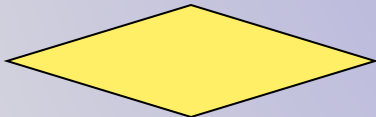
начало, конец алгоритма



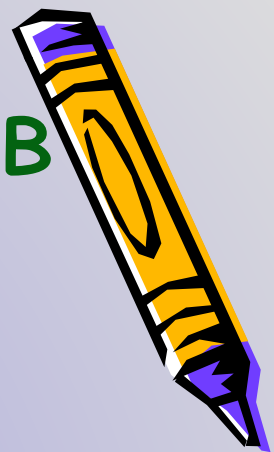
простое действие, вычисление



задание исходных данных, вывод результата

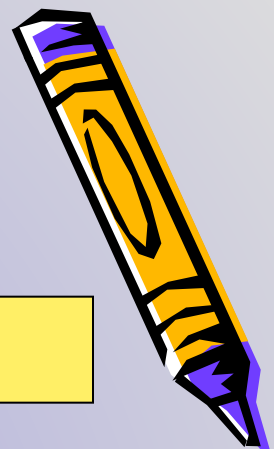


проверка условия



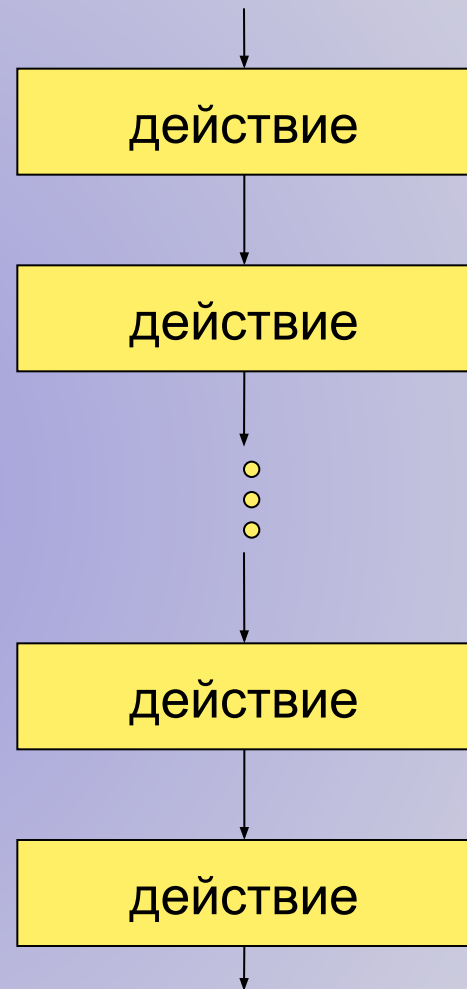
[к содержанию](#)

Основные алгоритмические конструкции

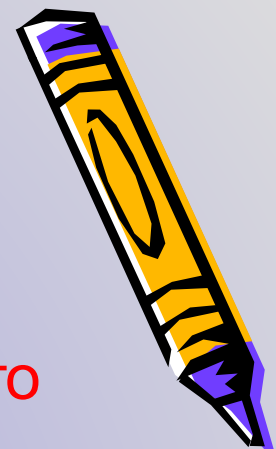


Следование (линейный тип алгоритма):

Все команды алгоритма следуют последовательно друг за другом



Основные алгоритмические конструкции

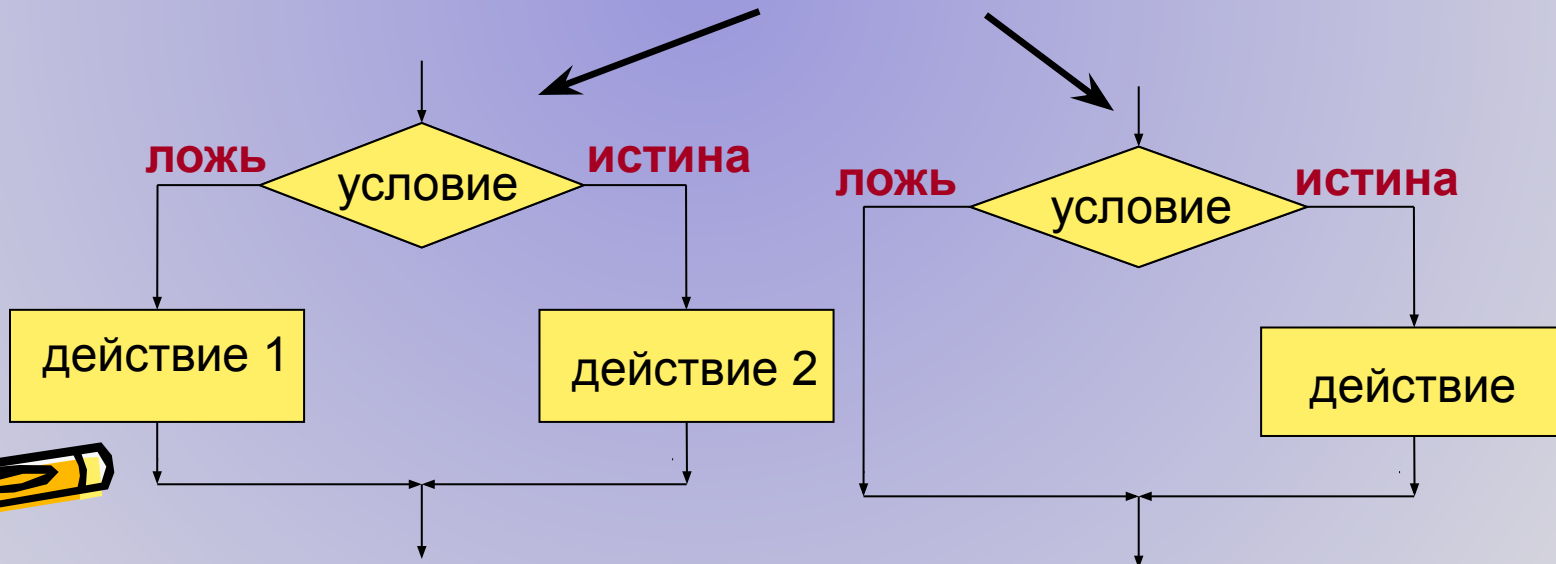


Ветвление (условный тип алгоритма):

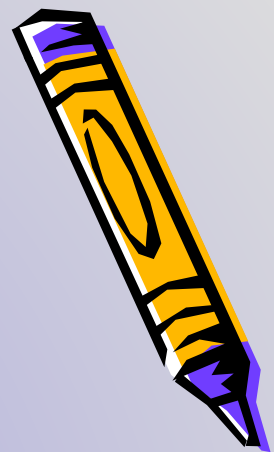
Выбор действия зависит от выполнения некоторого условия.

Условие – выражение, которое может принимать значение либо истина, либо ложь.

Ветвления бывают **полные** и **неполные**



Основные алгоритмические конструкции



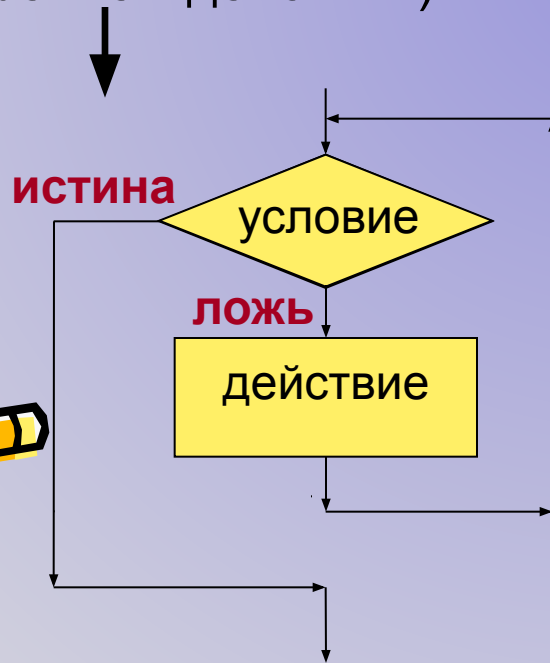
Повторение (циклический тип алгоритма)

В алгоритме есть повторяющиеся действия.

Циклы бывают

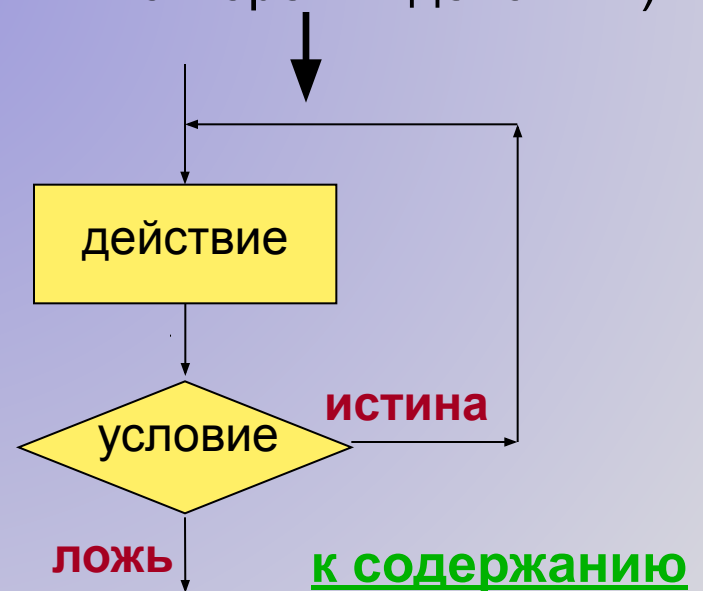
с предусловием

(условие стоит перед повторением действий)



с постусловием

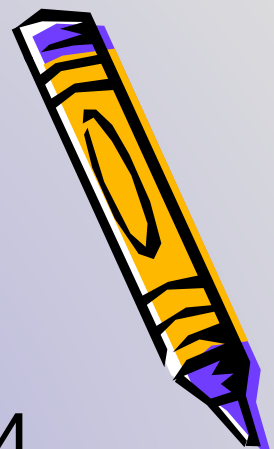
(условие стоит после повторения действий)



Соединение основных алгоритмических конструкций

В основной алгоритмической конструкции
каждое **простое действие** может быть
заменено на **любую** алгоритмическую
конструкцию.

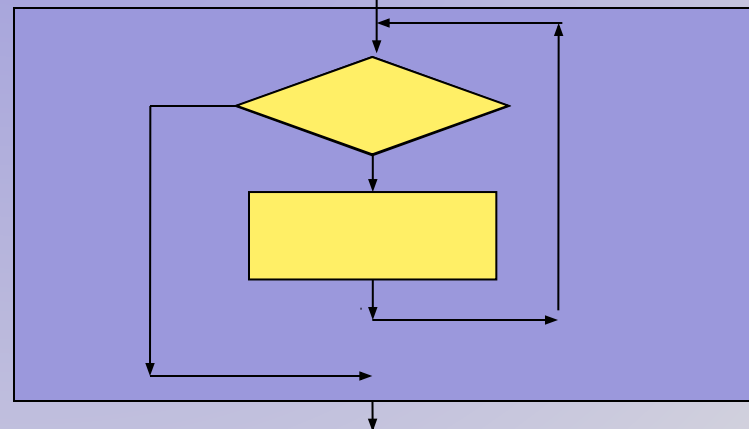
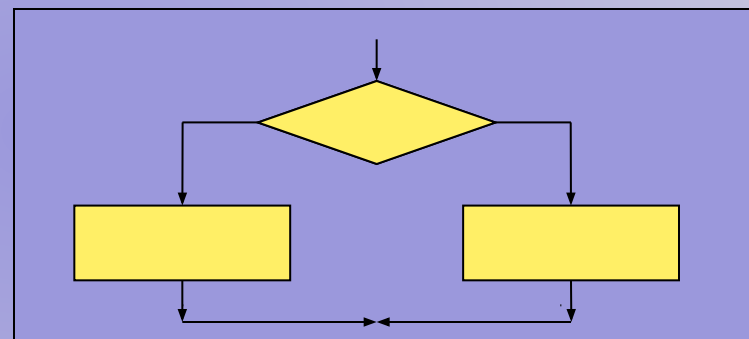
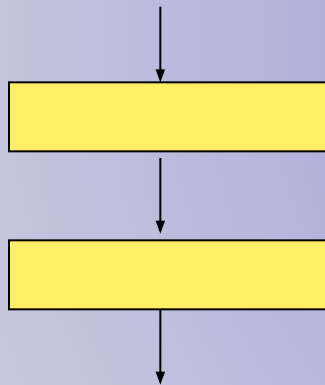
Таким образом получается соединение
алгоритмических конструкций конструкций.



Соединение основных алгоритмических конструкций

Пример 1: в линейной конструкции заменим одно простое действие **полным ветвлением**, а другое **циклом**.

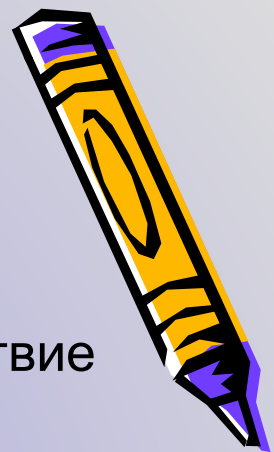
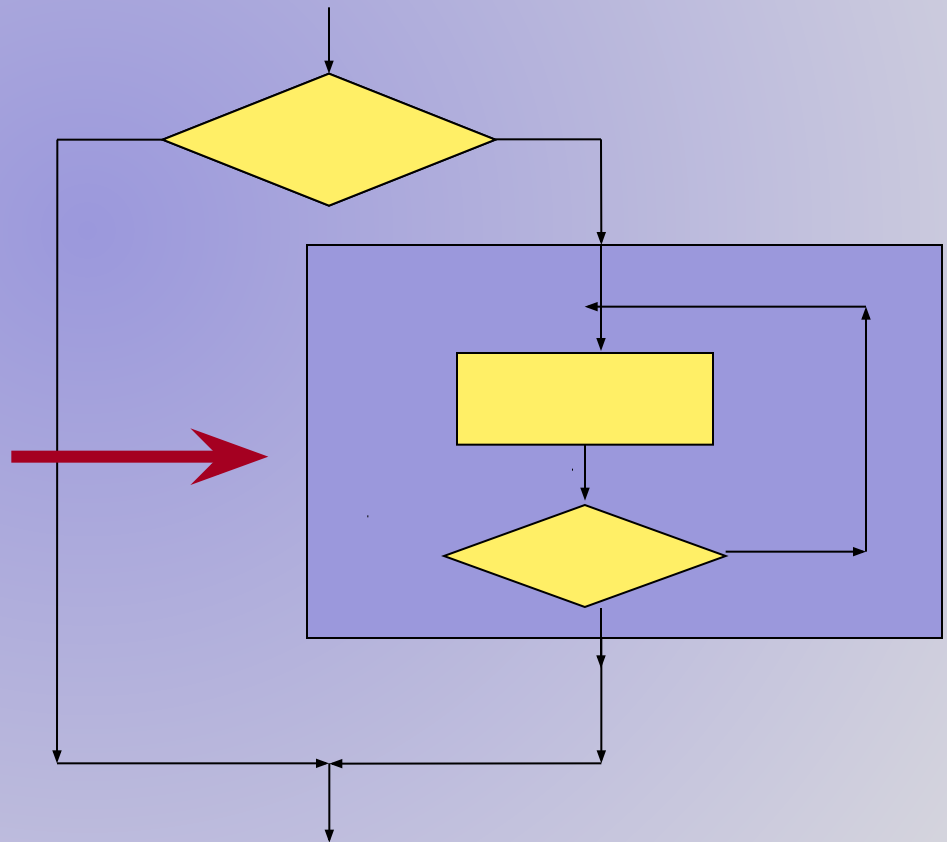
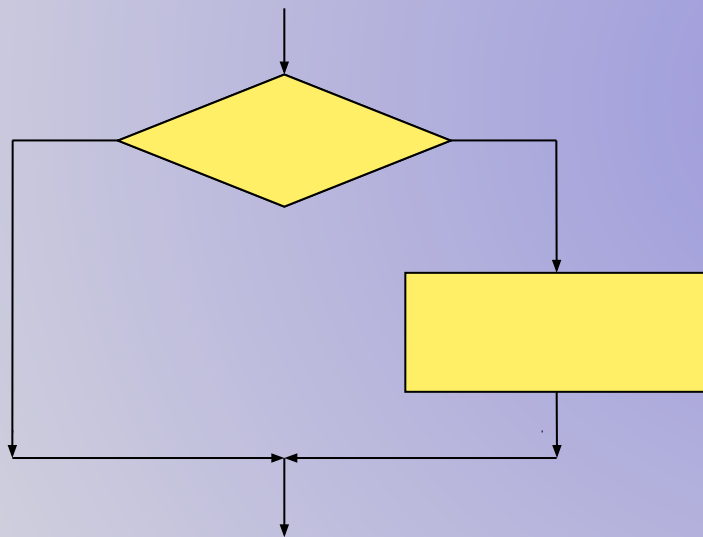
Получим **последовательное соединение ветвления и цикла**.



Соединение основных алгоритмических конструкций

Пример 2: в неполном ветвлении заменим простое действие циклом.

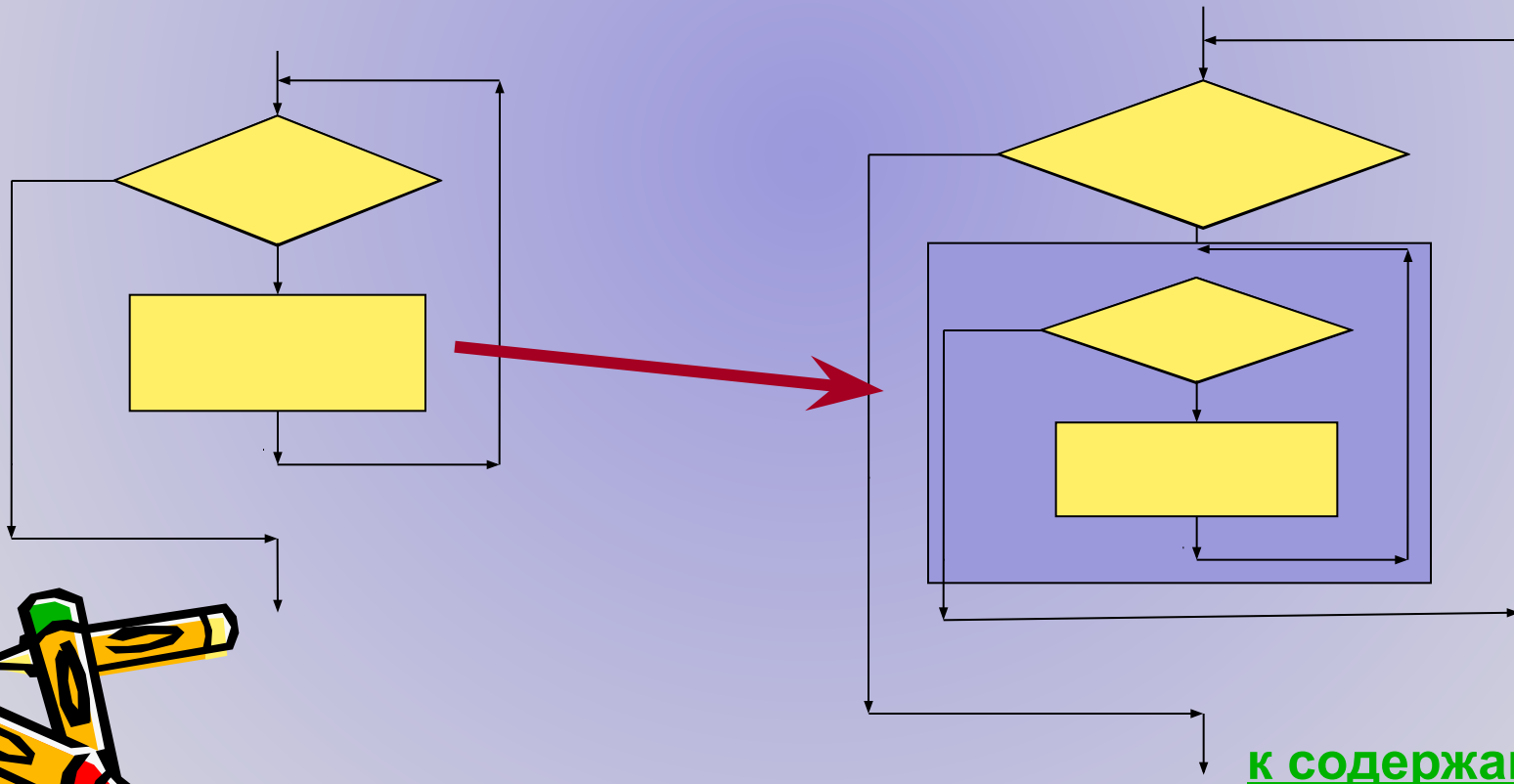
Получим **цикл В** ветвлении.



Соединение основных алгоритмических конструкций

Пример 3: в цикле заменим простое действие **циклом**.

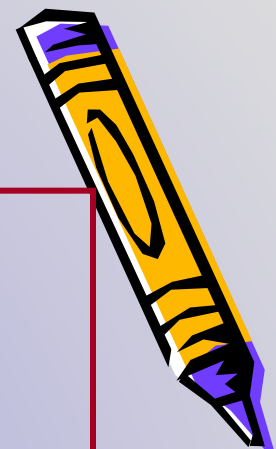
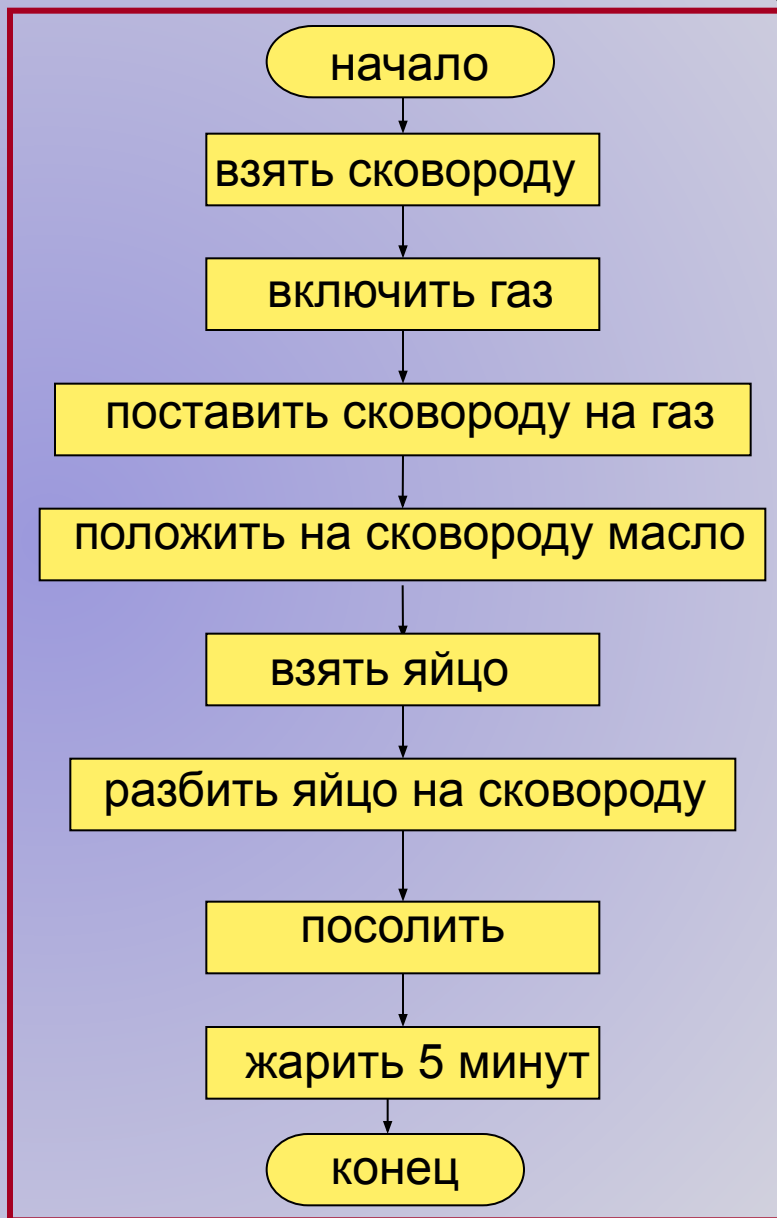
Получим **цикл В цикле**.



Простейшие примеры

Задача 1: приготовить яичницу.

Это линейный тип
алгоритма
(следование)



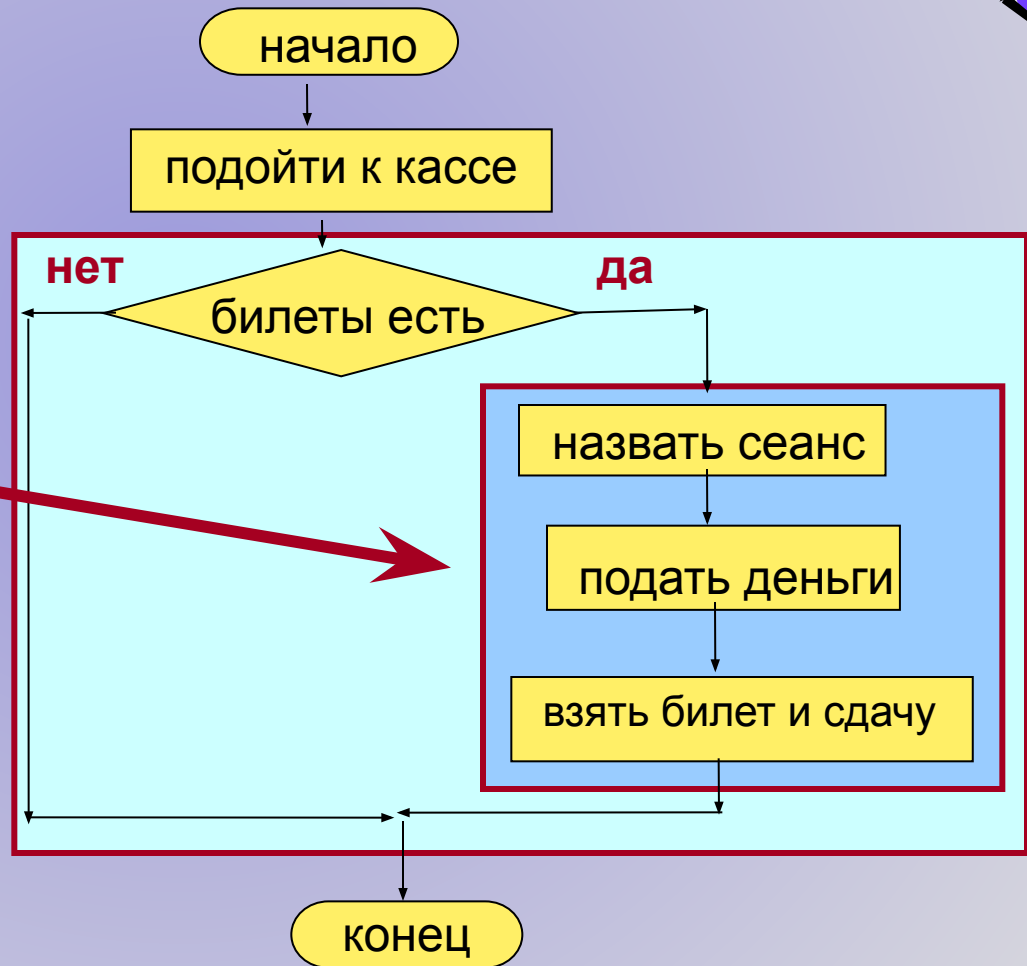
Простейшие примеры

Задача 2: покупка билетов в кино.

Это условный тип алгоритма (ветвление)

Это линейный тип алгоритма (следование)

Получилось сочетание условного и линейного типов алгоритмов



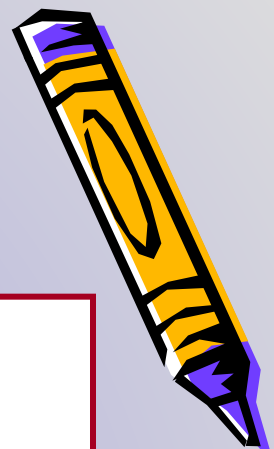
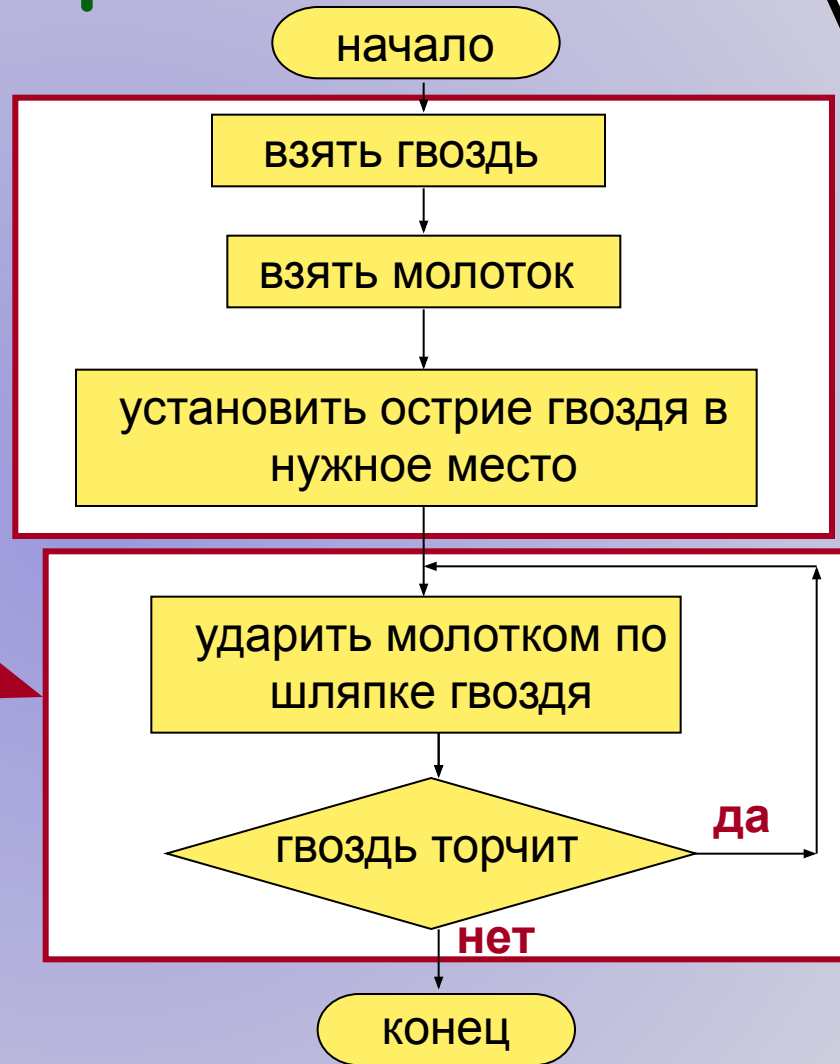
Простейшие примеры

Задача 3: **забить гвоздь.**

Это линейный тип алгоритма (следование)

Это циклический тип алгоритма (повторение)

Получилось сочетание линейного и циклического типов алгоритмов



Простейшие примеры

Задача 4: собрать гербарий.

Это линейный тип алгоритма (следование)

Это условный тип алгоритма (ветвление)

Это циклический тип алгоритма (повторение)

Получилось сочетание линейного алгоритма и условия в цикле



[к содержанию](#)

Примеры посложнее. Задача на соединение основных алгоритмических конструкций

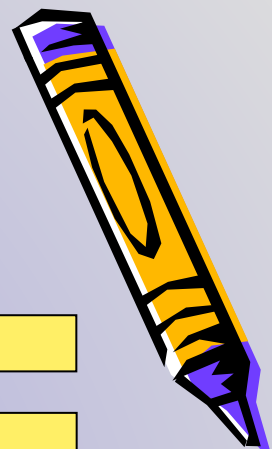
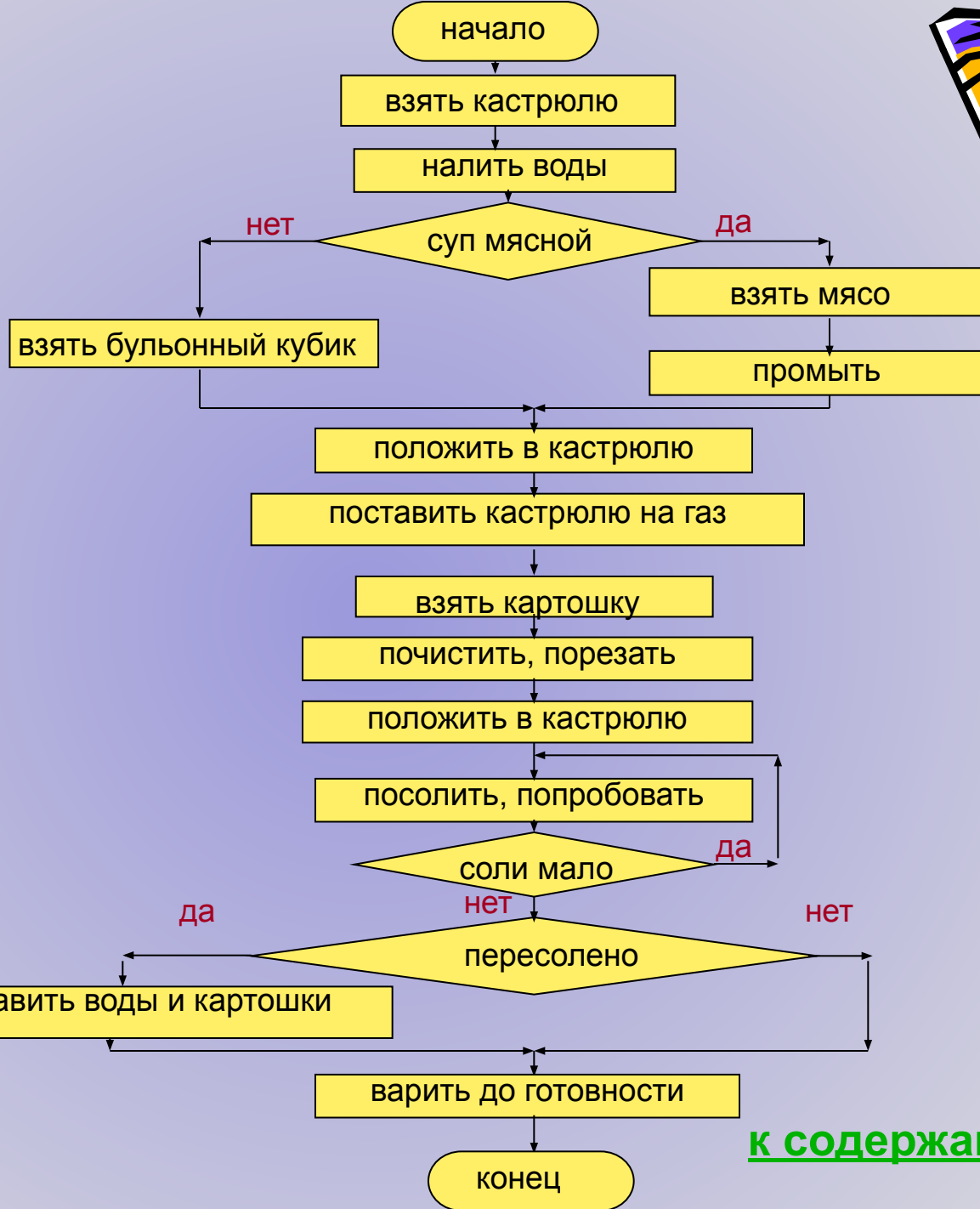


Алгоритм решения некоторых задач может содержать все типы алгоритмов и в самых различных сочетаниях.

Задача 5: сварить картофельный суп.

Алгоритм решения этой задачи может выглядеть так:





[к содержанию](#)



Хотите повторить?

ДА

НЕТ





Я думаю вы усвоили
пройденный
ДА материал. НЕТ
Проверим ?

