# Основы алгоритмизации
# Типы алгоритмов

Лекция №1 по курсу
«Комбинаторные алгоритмы»

.

# Понятие и свойства алгоритма

**Алгоритм** – это набор точных предписаний, последовательное выполнение которых однозначно приводит задачу к решению за конечное число шагов.

Алгоритм обладает следующими свойствами:

- Детерминированность(определенность,точность) – четкость и ясность всех предписаний: исполнителю алгоритма должно быть точно известно, какая команда алгоритма выполняется следующей («Уходя, гасите свет»)

- Результативность – способность алгоритма приводить к решению задачи за конечное число шагов

- дискретность – предписание представляет собой последовательность четко выраженных отдельных команд, причем, выполнив одну команду, исполнитель выполняет другую команду, промежуточных состояний нет
- массовость (универсальность) – применимость алгоритма к решению задач определенного класса, чем шире этот класс, тем ценнее алгоритм

Существуют следующие способы записи алгоритмов:

- словесно-формульная запись
- графическая запись (схема алгоритма, иначе, графическая схема алгоритма, блок-схема)
- запись на конкретном языке программирования

- ***Словесный способ*** **записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.**

***Пример***.

Записать алгоритм нахождения **наибольшего общего делителя (НОД)** двух натуральных чисел (алгоритм Евклида).

Алгоритм может быть следующим:

1. задать два числа
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2.

**Графическая схема алгоритма** состоит из отдельных блоков, связанных линиями потоков

Каждый блок описывает конкретный шаг алгоритма

Схемы алгоритмов должны соответствовать действующим стандартам на оформление схем алгоритмов, программ, данных и систем

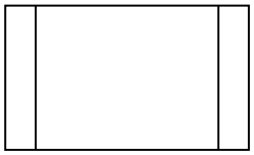
**[ГОСТ 19.701-90].**

Ниже приводятся некоторые символы, определенные в стандарте и рекомендуемые к использованию в графических схемах алгоритмов.

**1.Процесс**

Символ отображает функцию обработки данных любого вида.

**2.Предопределенный процесс**

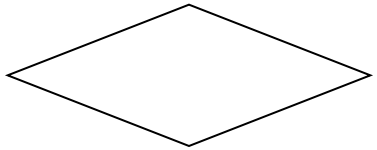


Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле).

3. **Данные**

Символ отображает данные, носитель данных не определен.

4. **Решение**



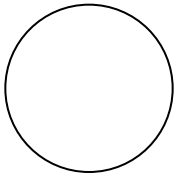
Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и  из которых может быть активизирован после вычисления условий, определенных внутри этого символа.

5. **Линия**



Символ отображает поток данных или управления

**6.Соединитель**

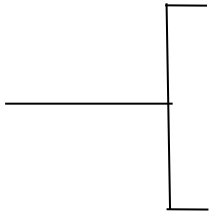


Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы соединители должны содержать одно и то же уникальное обозначение.

## 7. Терминатор

Символ отображает начало или конец схемы программы, внешнее использование и источник или пункт назначения данных.

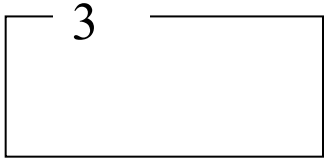
## 8. Комментарий



Текст, описывающий функцию символа, следует располагать внутри данного символа.

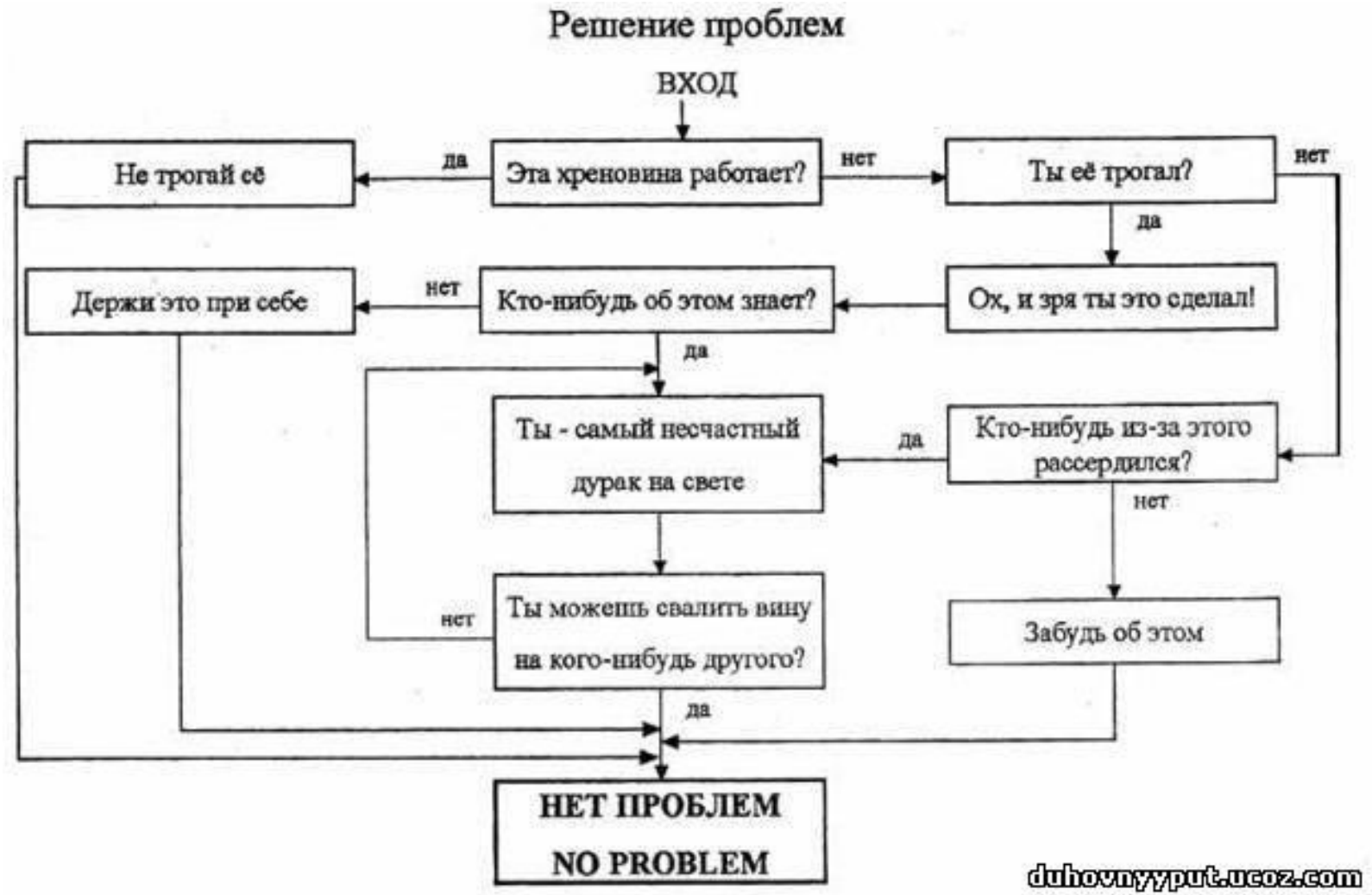
Если текст не помещается внутри символа, следует использовать символ комментария.

При необходимости блоки в схеме можно нумеровать (например, чтобы иметь возможность ссылаться на тот или иной символ) слева вверху в разъеме символа. Например,



***Правила выполнения соединений:***

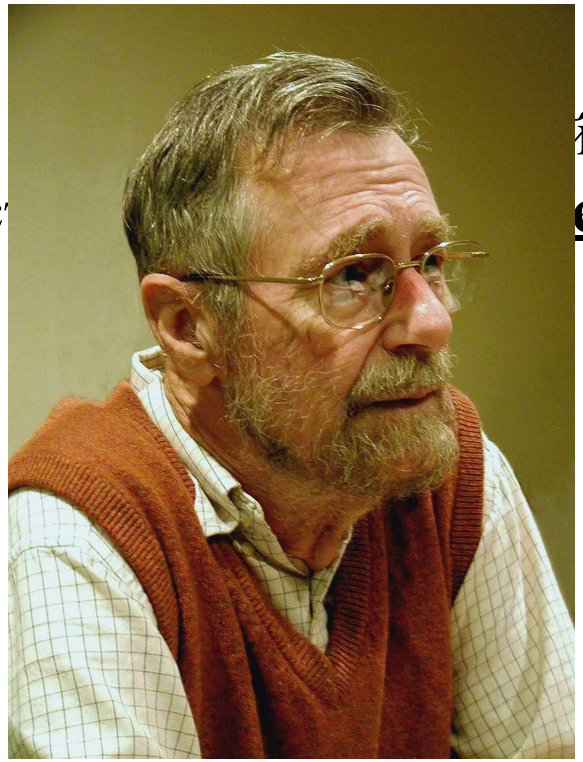
- Стандартное направление линий потока – слева направо и сверху вниз
- Если направление потока отличается от стандартного, это направление указывается стрелками
- В схемах следует избегать пересечения линий
- Линии в схемах должны подходить к символу либо слева, либо сверху, а выходить либо справа, либо снизу.
- Вход в блок и выход из блока следует размещать по центру символа


Решение проблем
ВХОД
Эта хреновина работает?
да
Не трогай её
нет
Ты её трогал?
нет
да
Ох, и зря ты это сделал!
Кто-нибудь об этом знает?
нет
Держи это при себе
да
Кто-нибудь из-за этого рассердился?
да
Ты - самый несчастный дурак на свете
нет
Забудь об этом
Ты можешь свалить вину на кого-нибудь другого?
нет
да
НЕТ ПРОБЛЕМ
NO PROBLEM
duhovnyyput.ucoz.com

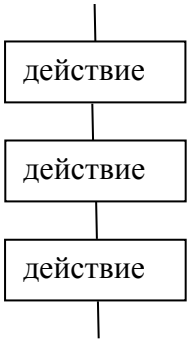
# Типы алгоритмов

**Теорема Дейкстра**. Алгоритм любой сложности можно реализовать, используя только три конструкции: следования (линейные), выбора (ветвления) и повторения (циклические).

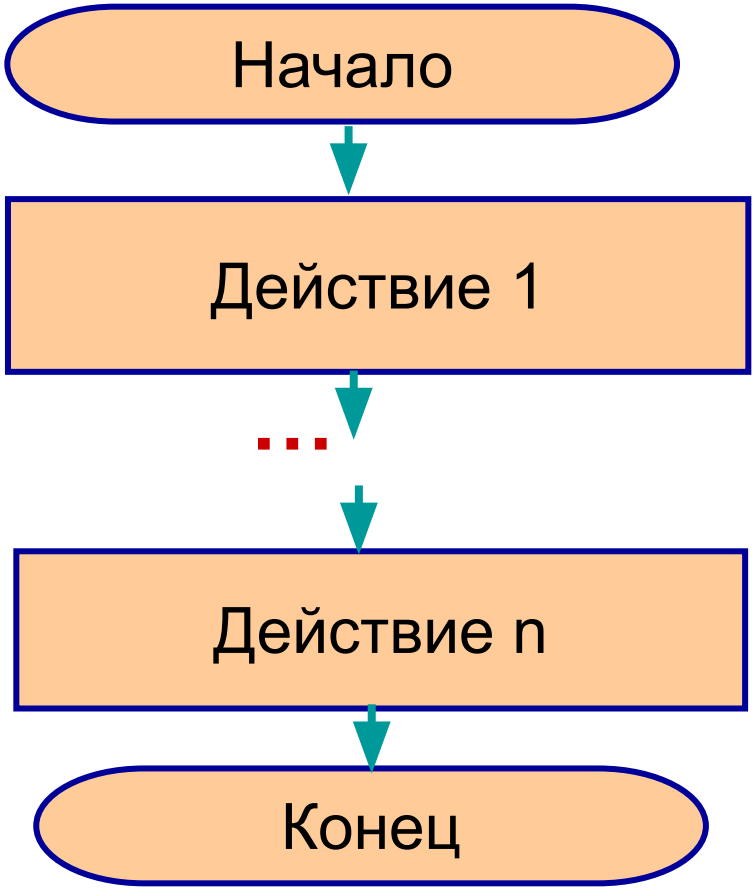
**Линейный** - алгоритм, в котором все указанные действия выполняются один раз в том порядке, в котором они записаны.

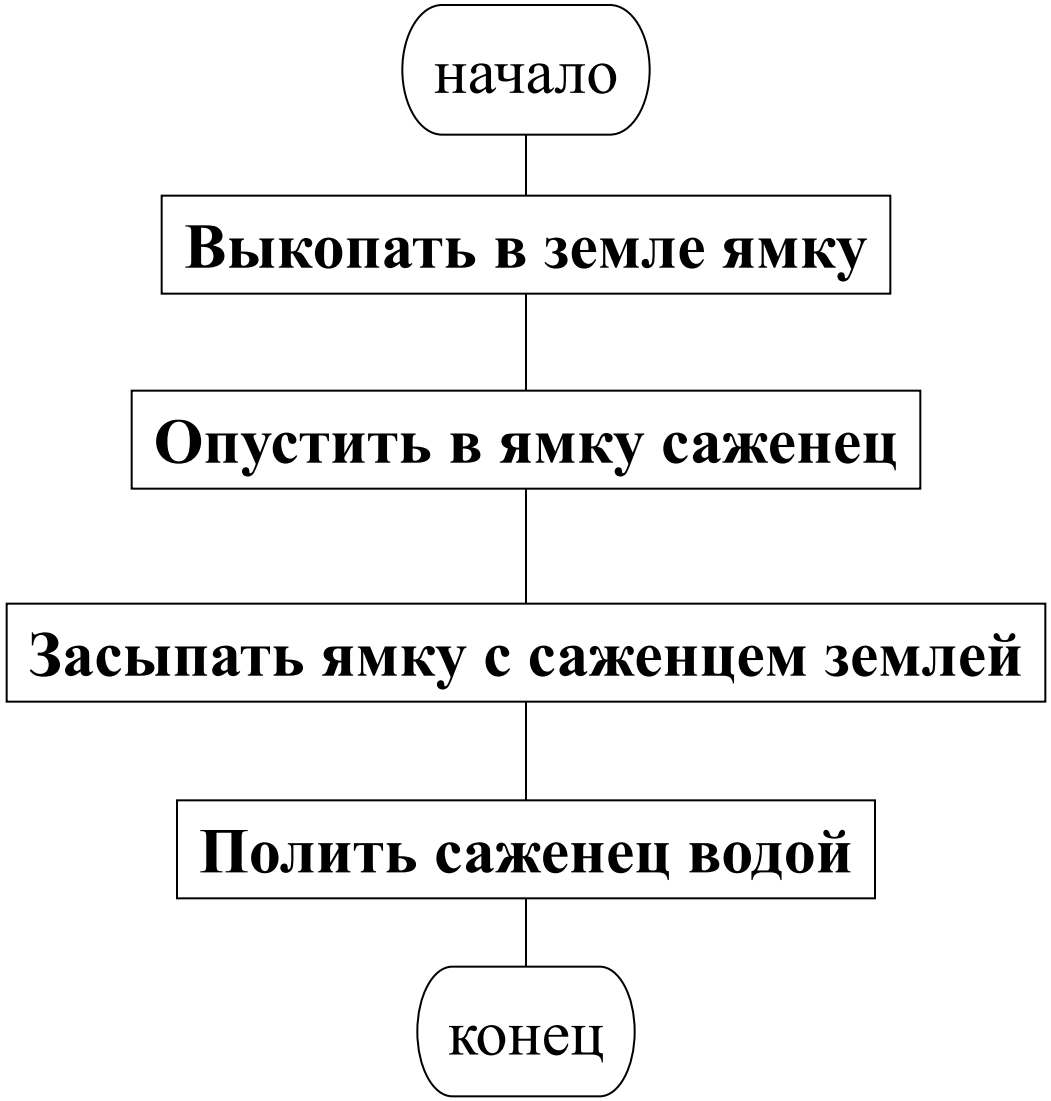


й алгоритм представляется в виде типовой с **е:**




**Эдсгер Вибе Дейкстра**


Начало
Действие 1
...
Действие n
Конец


начало
Выкопать в земле ямку
Опустить в ямку саженец
Засыпать ямку с саженцем землей
Полить саженец водой
конец

- **Разветвляющийся -** алгоритм, в котором некоторые действия выполняются один раз или не выполняются в зависимости от заданного условия.

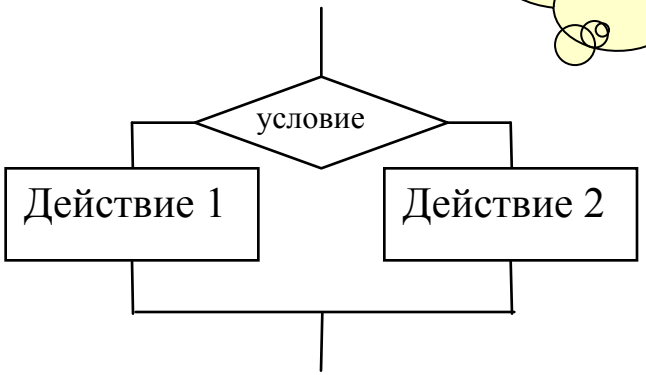

В схеме разветвляющийся алгоритм представляется в виде типовых структур

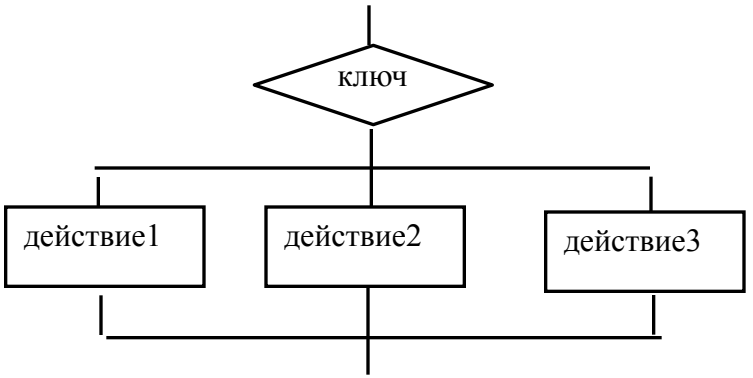
**Ветвление** и **выбор**

# Ветвление

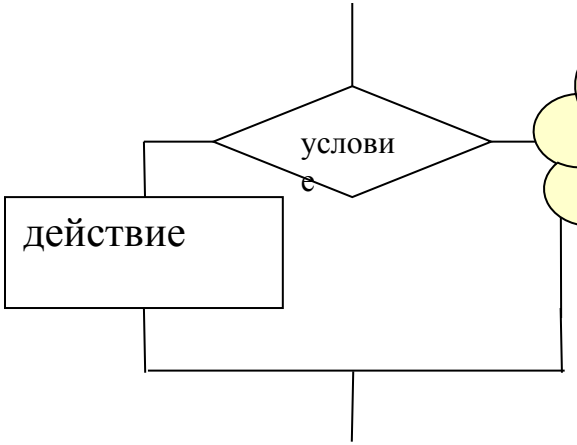
# выбор

Полная форма

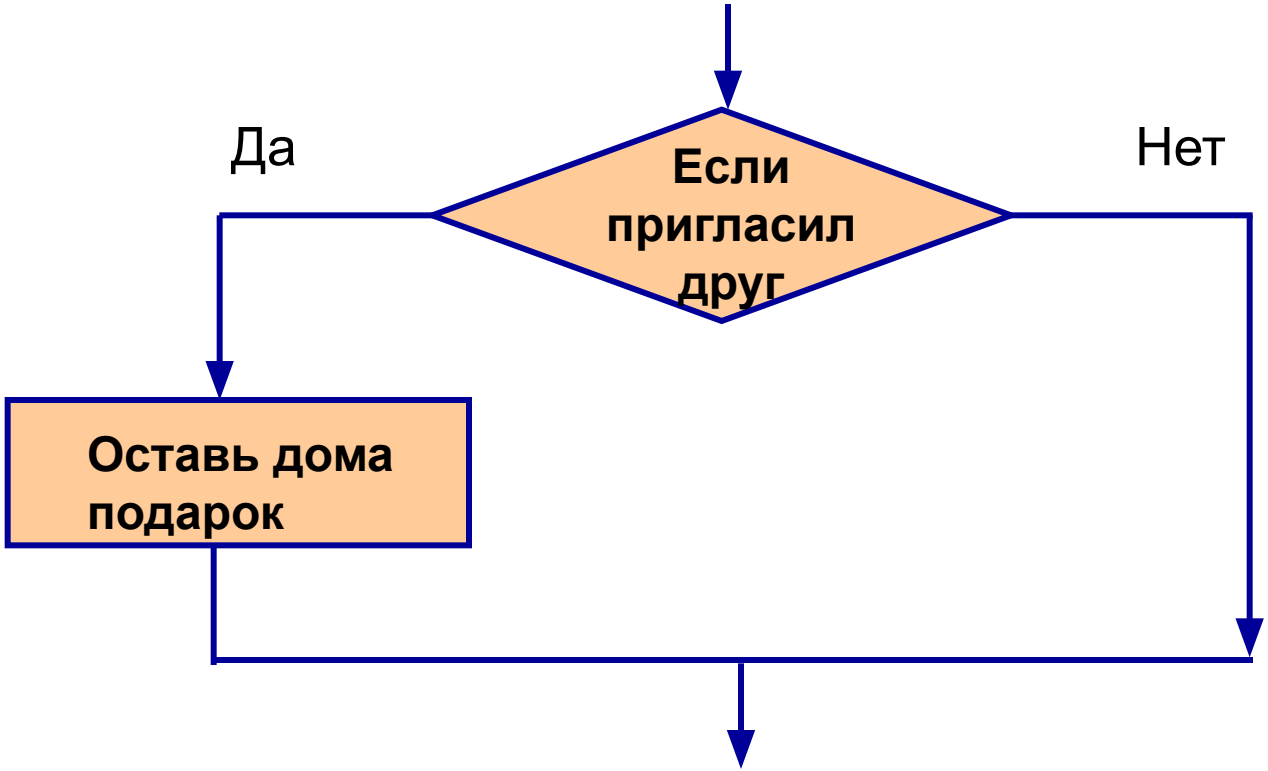


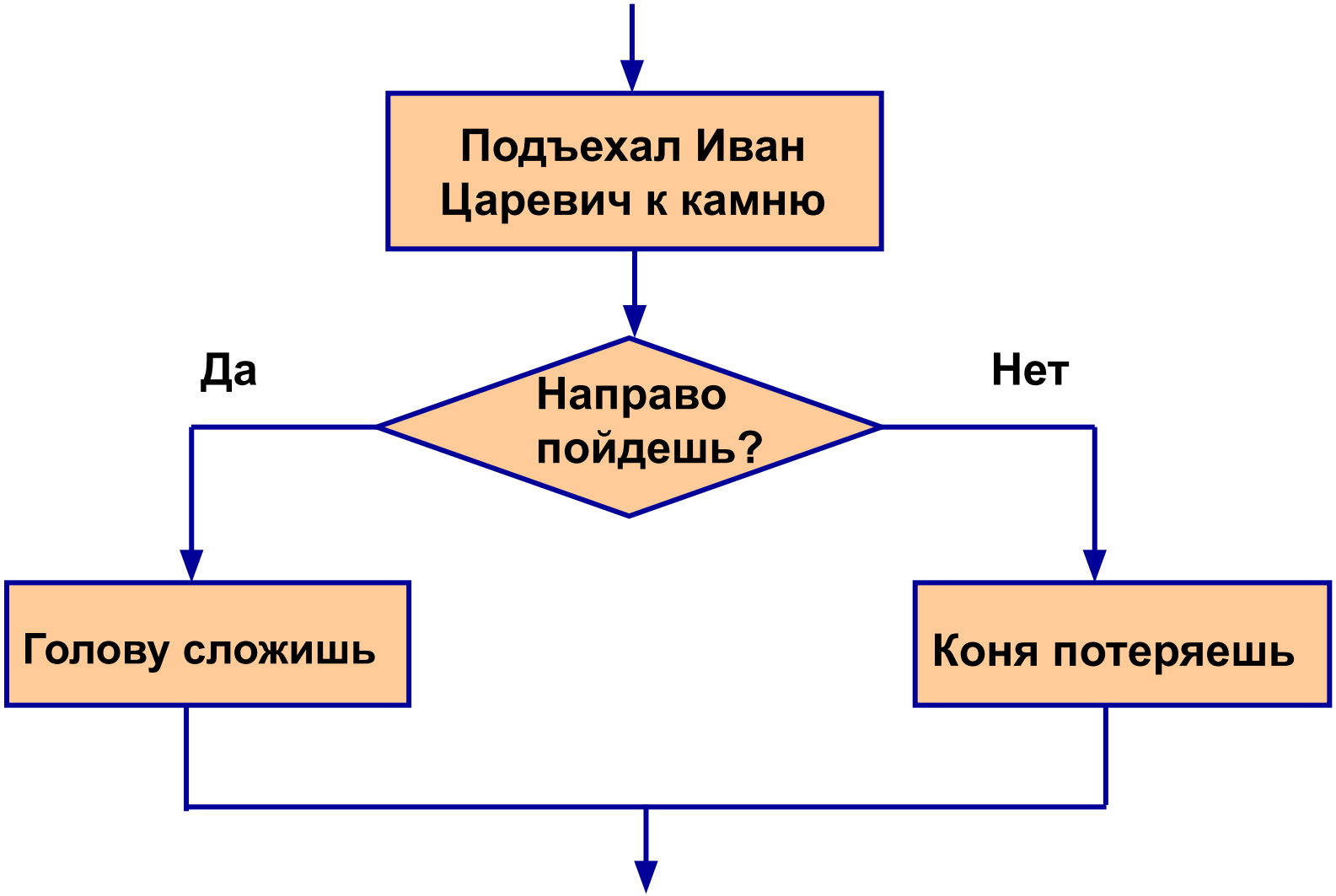

Неполная форма



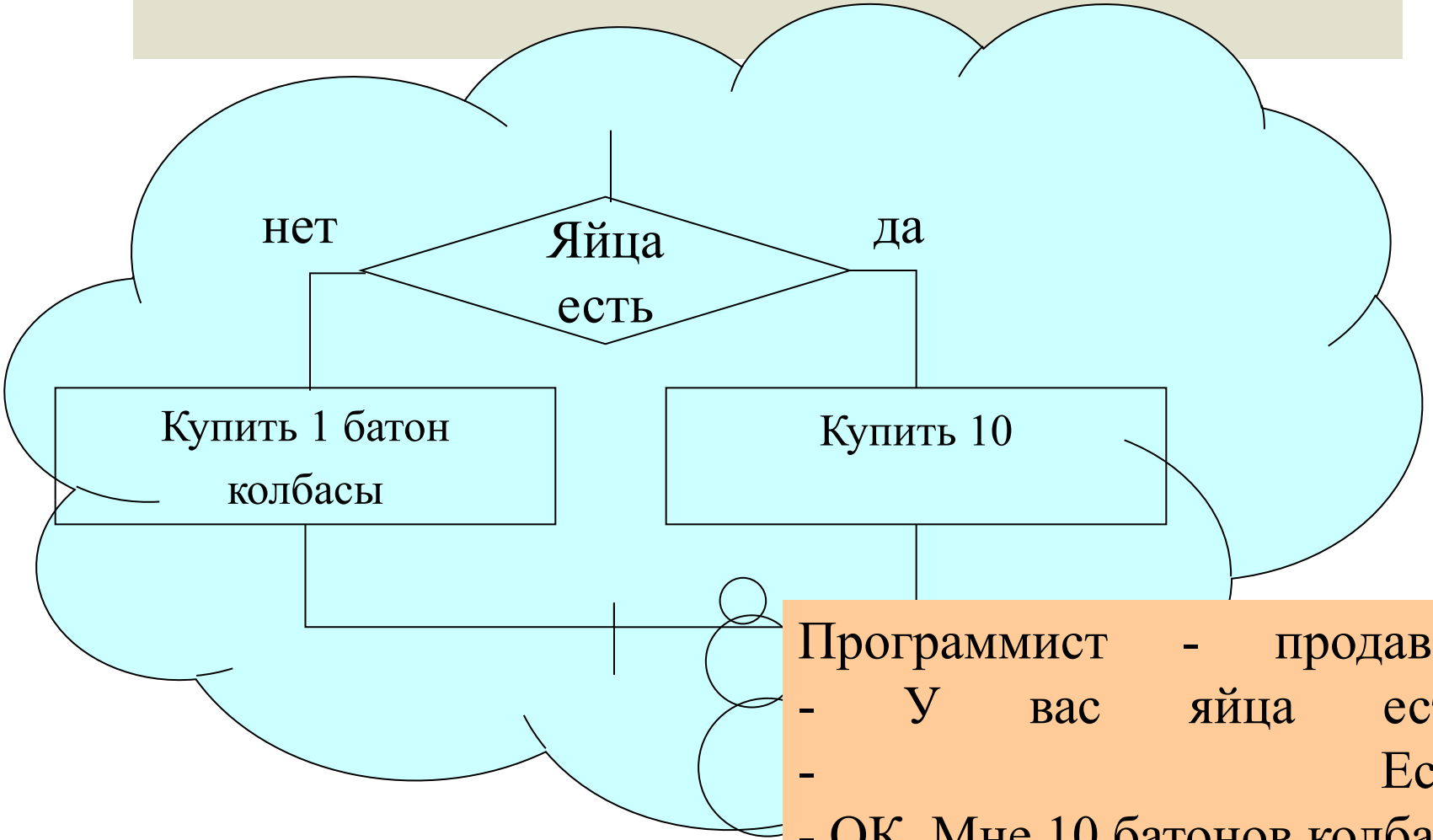
Если друг на день рожденья
Пригласил тебя к себе,
То оставь подарок дома –
Пригодится самому…




.


Подъехал Иван Царевич к камню
Да
Направо пойдешь?
Нет
Голову сложишь
Коня потеряешь

Жена отправляет программиста в магазин.
Купи батон колбасы и если будут яйца купи десяток.




Программист - продавцу.
- У вас яйца есть?
- Есть!
- ОК. Мне 10 батонов колбасы.

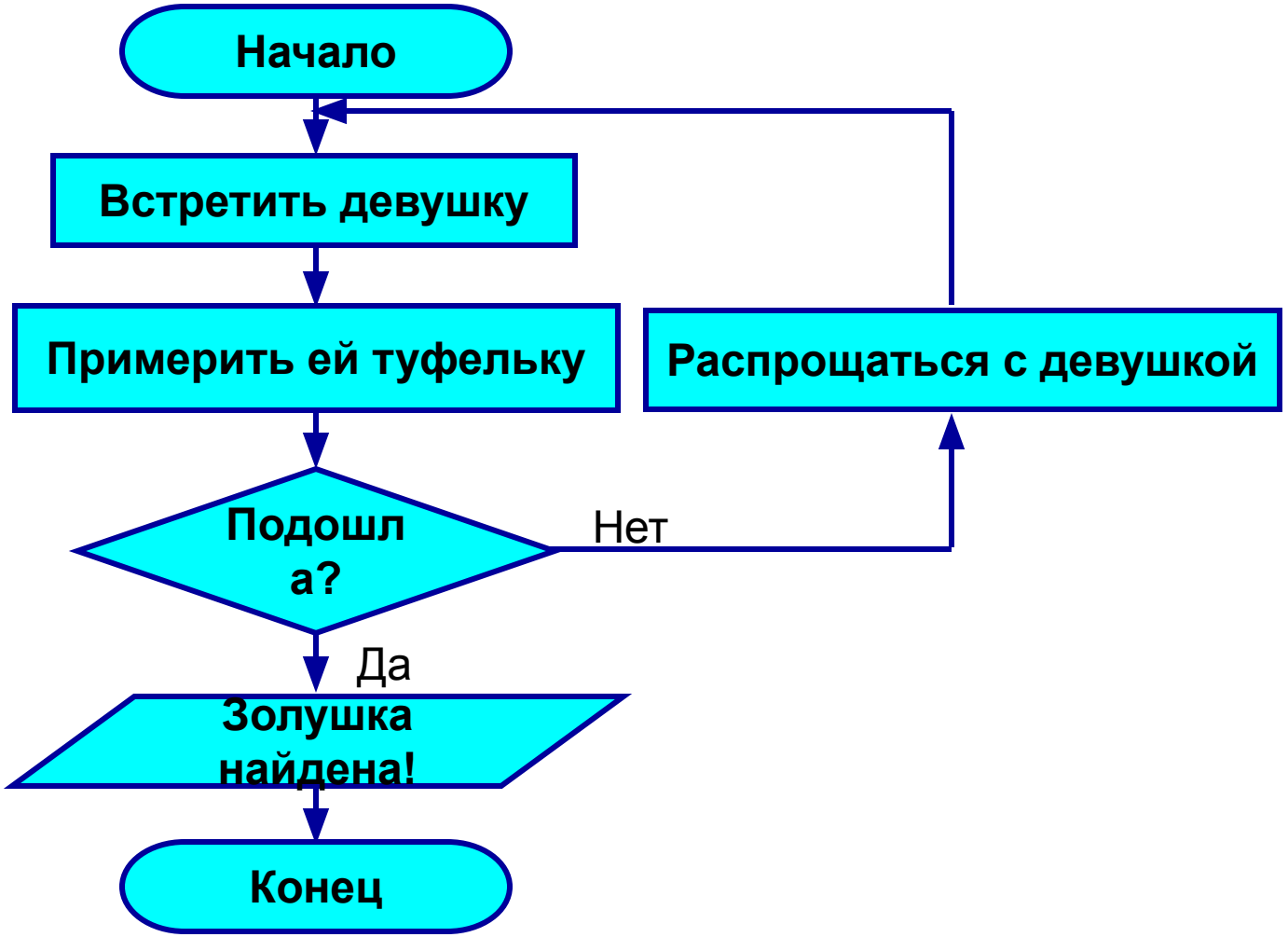
**Циклический -** алгоритм, в котором некоторая последовательность действий может выполняться несколько раз в зависимости от заданного условия.

В схеме циклический алгоритм представляется в виде типовой структуры **<u>цикл:</u>**

## Цикл "Пока"
## (цикл с предусловием)


## Цикл с постусловием

Алгоритм поиска Золушки:



Итак, алгоритмы делятся на

- линейные
- разветвляющиеся
- циклические

( можно также выделить в отдельный тип смешанные).

Алгоритмы могут классифицироваться и по другому направлению.

- Комбинаторные алгоритмы:
  - ❖ Общие комбинаторные алгоритмы (например, генерация случайных чисел )
  - ❖ Алгоритмы на графах
  - ❖ Алгоритмы поиска
  - ❖ Алгоритмы сортировки
  - ❖ Алгоритмы слияния
  - ❖ Алгоритмы работы со строками

- Алгоритмы сжатия данных
- Криптографические алгоритмы
- Теоретико-числовые алгоритмы
- Цифровая обработка сигналов

И т.д.