

УПРАВЛЕНИЕ И АЛГОРИТМЫ

9 КЛАСС



ПРОИСХОЖДЕНИЕ ПОНЯТИЯ «КИБЕРНЕТИКА»



Платон



Ампер А.М.

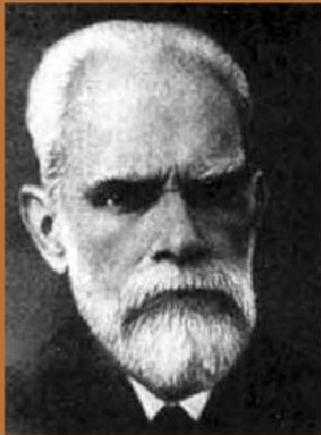
Слово **«кибернетика»** происходит от греческого слова «кюбернетес», что первоначально означало «рулевой», «кормчий», а затем стало обозначать «правитель над людьми». Древнегреческий философ **Платон** в своих трудах в одних случаях называет кибернетикой искусство управления кораблем или колесницей, а в других — искусство управления людьми.

Римляне слово «кюбернетес» преобразовали в «губернатор».

Известный французский ученый-физик **А. М. Ампер** (1775-1836 гг.) в своей работе «Опыт о философии наук, или Аналитическое изложение естественной классификации всех человеческих знаний», первая часть которой вышла в 1834 г., назвал кибернетикой науку о текущем управлении государством (народом), которая помогает правительству решать встающие перед ним конкретные задачи с учетом разнообразных обстоятельств в свете общей задачи принести стране мир и процветание.



**Вышнеградский
И.А.**



Ляпунов А.М.

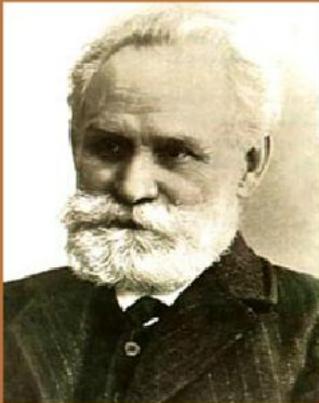
Развитие кибернетики как науки было подготовлено многочисленными работами ученых в области математики, механики, автоматического управления, вычислительной техники, физиологии высшей нервной деятельности.

Основы теории автоматического регулирования и теории устойчивости систем регулирования содержались в трудах выдающегося русского математика и механика **Ивана Алексеевича Вышнеградского** (1831—1895 гг.), который разработал теорию и методы расчета автоматических регуляторов паровых машин.

Общие задачи устойчивости движения, которые являются фундаментом современной теории автоматического управления, были решены одним из крупнейших математиков своего времени **Александром Михайловичем Ляпуновым** (1857—1918 гг.). Его труды сыграли огромную роль в разработке теоретических вопросов технической кибернетики.



Сеченов И.М.



Павлов И.П.

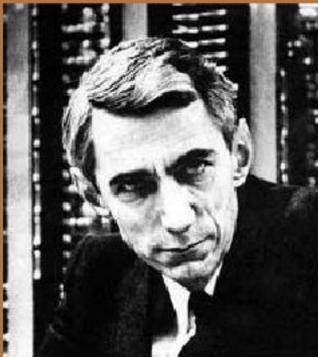
Исследование процессов управления в живых организмах связывается прежде всего с именами великих русских физиологов - **Ивана Михайловича Сеченова** (1829—1905 гг.) и **Ивана Петровича Павлова** (1849—1936 гг.).

И. М. Сеченов еще во второй половине XIX столетия показал, что в основе психологических явлений лежат физиологические процессы, что коренным образом противоречило господствовавшей тогда доктрине о духовном начале человеческого мышления и психики.

Блестящие работы **И. П. Павлова** обогатили физиологию высшей нервной деятельности учением об условных рефлексах и формулировкой принципа обратной афферентации, являющегося аналогом принципа обратной связи в теории автоматического регулирования. Труды И. П. Павлова стали основой и отправным пунктом для ряда исследований в области кибернетики, и биологической кибернетики в частности.



Дж. фон Нейман



Клод Шеннон

Важнейшее значение для развития кибернетики имели работы американского ученого **Джона фон Неймана** (1903—1957 гг.). Он внес фундаментальный вклад в область теории множеств, функционального анализа, квантовой механики, статистической физики, математической логики теории автоматов, вычислительной техники. Благодаря ему получили развитие новые идеи в области этих научных направлений. Джон фон Нейман в середине 40-х годов разработал первую цифровую ЭВМ в США. Он — создатель новой математической науки — теории игр, непосредственно связанной с теоретической кибернетикой.

Важнейшие для кибернетики проблемы измерения количества информации разработаны американским инженером и математиком **Клодом Шенноном**, опубликовавшим в 1948 г. классический труд «Теория передачи электрических сигналов при наличии помех» в котором заложены основные идеи существенного раздела кибернетики — теории информации.

НОРБЕРТ ВИННЕР



1894-1964

Появление кибернетики как науки неразрывно связано с именем американского математика **Норберта Винера**.

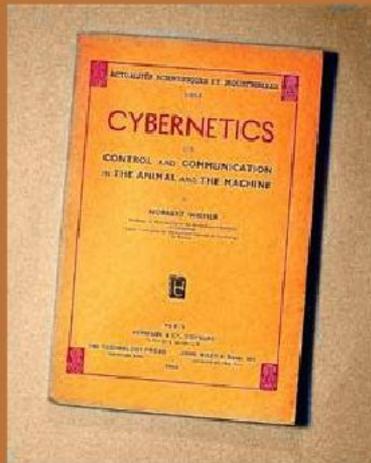
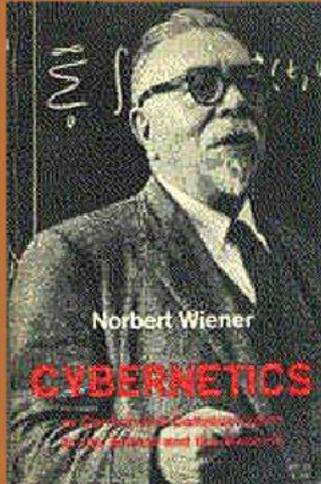
Норберт Винер родился **26 ноября 1894** г. в городе Колумбия, штат Миссури. Его родители были иммигрантами — выходцами из России.

В семь лет Норберт читал Дарвина и Данте, в одиннадцать закончил среднюю школу, в четырнадцать — колледж и получил ученую степень — бакалавра искусств. В восемнадцать лет он становится доктором философии (по специальности «математическая логика») Гарвардского университета.

Во время **второй мировой войны** Винер разработал новую вероятностную модель управления силами ПВО.

Винер столкнулся с тем, что машина должна выполнять сложные действия по предсказанию поведения цели, заменяя собой наводчика.

КИБЕРНЕТИКА КАК НАУКА



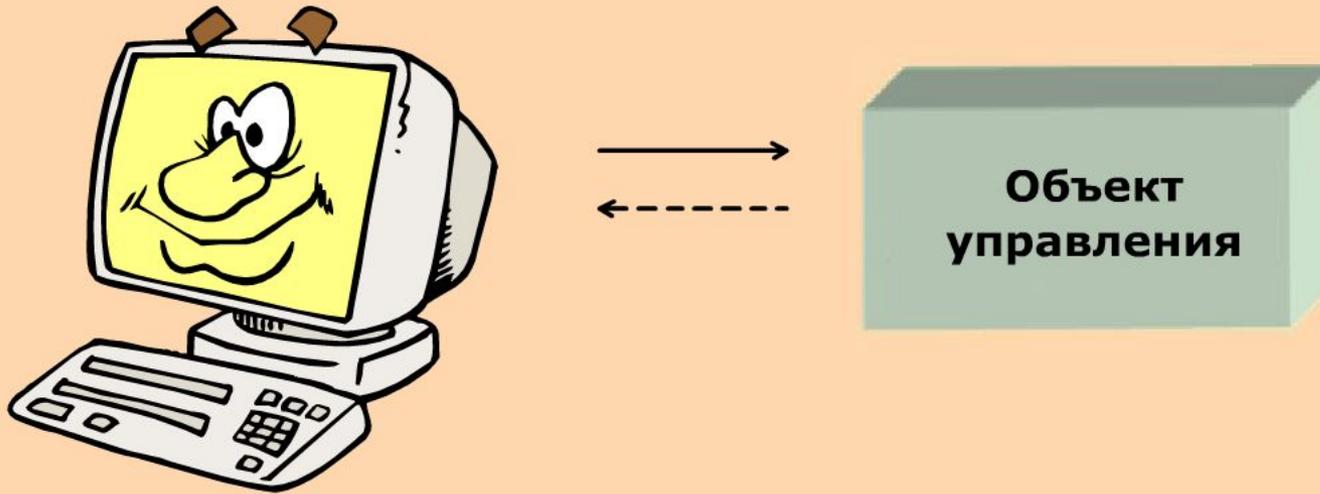
В **1948** году вышла книга "Кибернетика", которая и заложила основы современного компьютерного мира. Полное название главной книги Винера - **"Кибернетика, или управление и связь в животном и машине."**

Следующая работа вышла под названием **"Человеческое использование человеческих существ, или кибернетика и общество"**.

В своей книге Винер дал следующее определение кибернетики:

Кибернетика - это наука об управлении, связях и обработке информации.

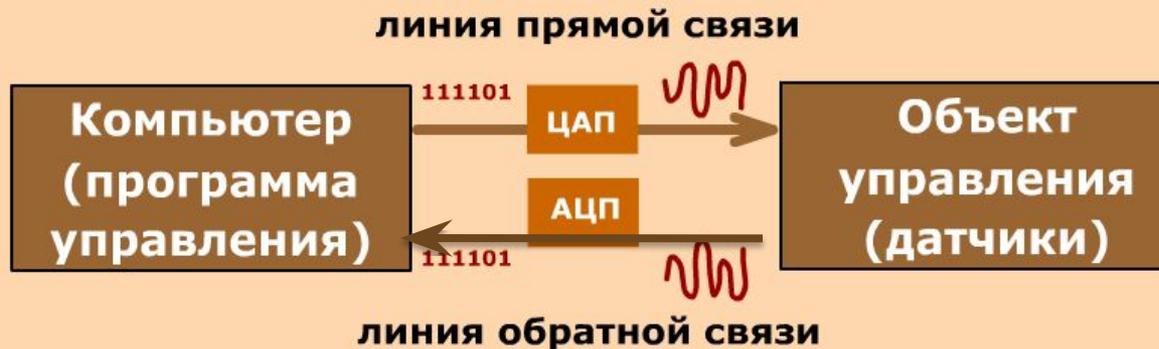
СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ (САУ)



Системы, в которых роль управляющего объекта поручается компьютеру, называются **системами автоматического управления (САУ)**. Человек вмешивается в работу САУ только в случае аварийных ситуаций.

Для функционирования такой системы в память компьютера должна быть заложена **программа** управления, поэтому такой способ управления называется **программным**.

ЦАП и АЦП



Управляющие команды в компьютере имеют форму двоичного кода. Для преобразования двоичного кода в аналоговый сигнал используются **цифро-аналоговые преобразователи (ЦАП)**.

Приборы, которые выдают информацию о состоянии объекта управления, называются **датчиками**.

Если показания датчиков имеют аналоговую форму, то их преобразование в двоичный код выполняется **аналого-цифровыми преобразователями (АЦП)**.

СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ (САУ)

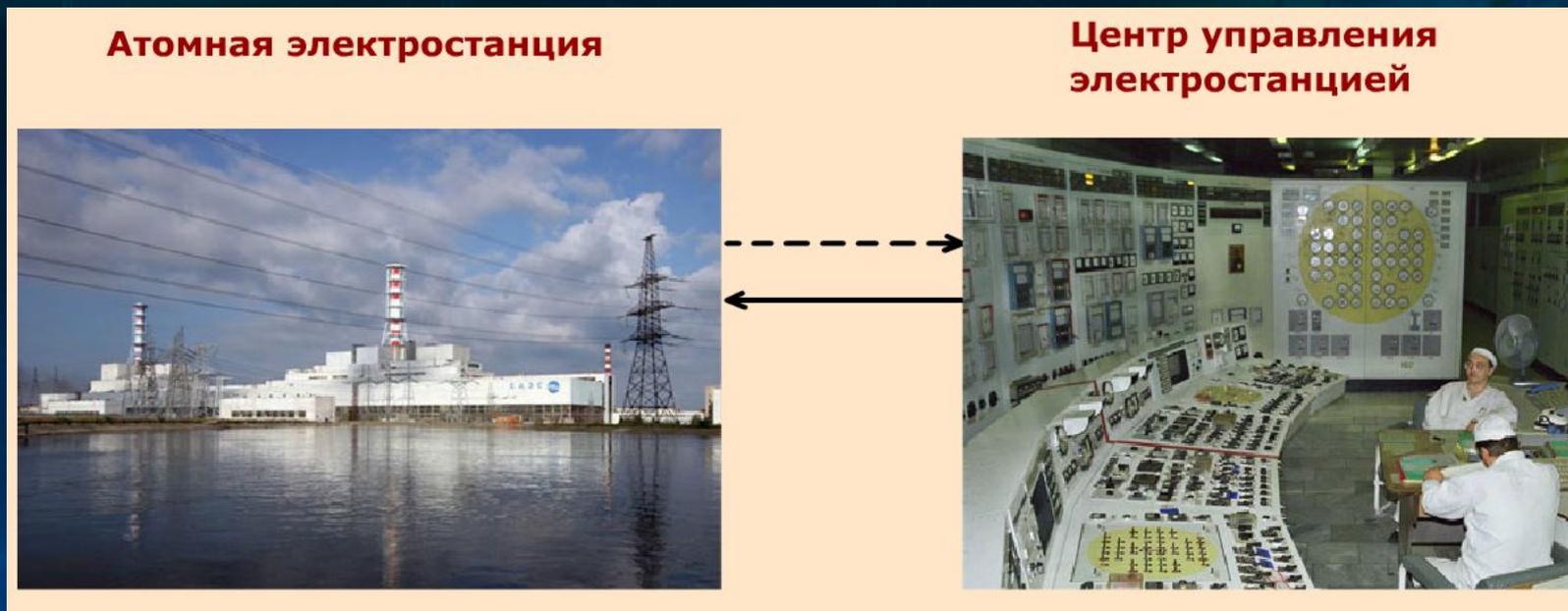


В системах автоматического управления не всегда используется компьютер с полным комплектом всех устройств. В простейших случаях для автоматического управления используются **микрпроцессоры**, встроенные в управляемое устройство.

Очень часто микропроцессоры применяются в транспортных средствах: автомобилях, самолётах, поездах. Многие бытовые приборы также содержат встроенные микропроцессоры.



ПРИМЕРЫ САУ



АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ (АСУ)

Для решения задач управления в крупных масштабах (производственным предприятием, отраслью экономики) создаются компьютерные системы, называемые автоматизированными системами управления (АСУ).

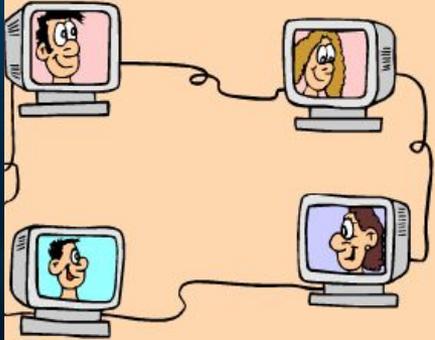
Автоматизированные системы управления (АСУ) помогают человеку в сборе информации и принятии управляющих решений. Окончательный выбор решения остаётся за человеком.



Примеры автоматизированных систем управления:

- **Интегрированные системы делопроизводства** (координация деятельности подразделений, оптимизация административно-хозяйственной деятельности и т.д.)
- **Финансовые аналитические системы** (для банковских и биржевых структур)
- **ГАС "Выборы"** (государственная автоматизированная система для проведения выборов)

КОМПЬЮТЕРНЫЕ СЕТИ В УПРАВЛЕНИИ



С распространением персональных компьютеров технической основой автоматизированных систем управления стали **компьютерные сети**.

Автоматизированные системы управления, работающие в масштабах отдельного предприятия, используют **локальные компьютерные сети**.



Автоматизированные системы управления, работающие в масштабах отрасли, в государственных масштабах, используют **глобальные компьютерные сети**.

УПРАВЛЕНИЕ И АЛГОРИТМЫ

Кибернетическая модель управления

Управляющий объект

Объект (субъект), осуществляющий управление

Объект управления

Объект (субъект), выполняющий команды управления

Прямая связь

Канал передачи команд управления

Обратная связь

Канал передачи данных о состоянии объекта управления

Алгоритм управления

Последовательность команд управления

Автоматические системы с программным управлением

Технические системы, в которых функции управляющего объекта выполняет компьютер

Алгоритмизация

Исполнитель алгоритмов

СКИ – система команд исполнителя

Свойства алгоритма

Дискретность

Понятность

Точность

Конечность

Алгоритмические структуры

Следование

Ветвление

Цикл

Структурная методика алгоритмизации

Построение алгоритма из базовых алгоритмических структур

- следование
- ветвление
- цикл

Последовательная детализация

Основной алгоритм

Вспомогательные алгоритмы

Проектирование сверху вниз

УПРАВЛЕНИЕ



Управление - это целенаправленное воздействие одних объектов, которые являются управляющими, на другие объекты - управляемые.

С точки зрения кибернетики взаимодействие между управляющим и управляемым объектами рассматривается как **информационный процесс**.

Управляющий объект - это объект в системе управления, который производит управляющее воздействие на управляемый объект (объект управления).

Управляемый объект (объект управления) - это объект, который принимает команды от управляющего объекта и выполняет их.

Объект управления является **исполнителем алгоритма управления**.



Процесс управления
уличным движением

Регулировщик - управляющий объект,
Водители - объекты управления.



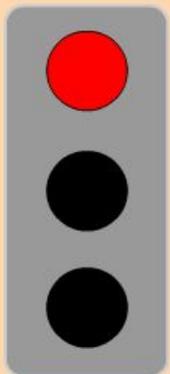
Процесс
обучения

Учитель - управляющий объект,
Ученики - объекты управления

АЛГОРИТМ УПРАВЛЕНИЯ



Алгоритм управления - последовательность команд управления, выполнение которой приводит к заранее поставленной цели.



Красный

Управляющий объект - светофор.

Алгоритм управления:

красный-жёлтый-зелёный-жёлтый-красный-жёлтый-зелёный-жёлтый-красный...

Исполнители алгоритма: пешеходы и водители.

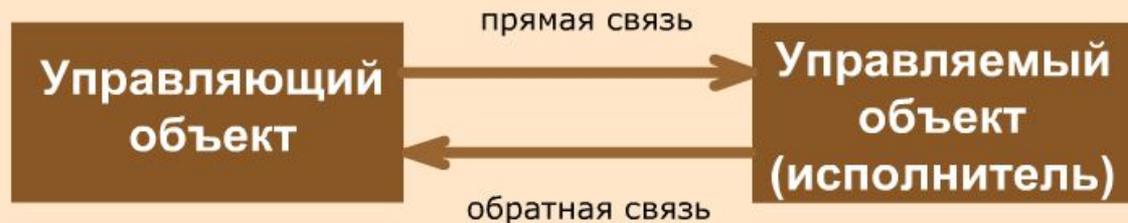
Прямая связь



Прямая связь - это процесс передачи **команд** управления от управляющего объекта к управляемому.

С кибернетической точки зрения все варианты управляющих воздействий следует рассматривать как управляющую информацию, передаваемую в виде команд.

Управление с обратной связью



Прямая связь - это процесс передачи **команд** управления от управляющего объекта к управляемому.

Обратная связь - это процесс передачи информации о состоянии исполнителя.

Примером системы управления с обратной связью может служить любое предприятие, где по линии **прямой связи** передаются распоряжения управленческого аппарата, а по линии **обратной связи** - информация об исполнении этих распоряжений подчинёнными сотрудниками, которую в данном случае можно рассматривать как информацию о состоянии исполнителей.

Управление с обратной связью



Алгоритм с обратной связью может содержать проверку условий, альтернативные и повторяющиеся команды.



1. Дать команду "Приготовиться"
2. Дать команду "Внимание!"
3. Дать команду "Старт!!"
4. Если был фальстарт то
 - 4.1 Остановить забег
 - 4.2 Вынести предупреждение нарушителю
 - 4.3 Вернуться к команде 1

задание

Тема: *Управление и кибернетика. Управление с обратной связью*

1. В приведённом ниже списке найдите соответствие между управляющим и управляемым объектами и заполните таблицу: оркестр, лошадь, тренер, наездник, актёр, дирижёр, водитель, режиссёр, спортсмен, автобус.

Управляющий объект	Управляемый объект

2. Внесите изменения в алгоритм, приведённый в задаче 2, таким образом, чтобы мама предусмотрела следующие ситуации, когда: а) в холодильнике две жёлтых кастрюли, б) дома нет спичек и нечем зажечь газ, в) газ вообще отключен из-за аварии, г) часы остановились.

задание

Тема: *Управление и кибернетика. Управление с обратной связью*

3. Первоклассник пришёл домой и увидел, что мама оставила ему записку с информацией о том, как разогреть обед:

- *открой холодильник,*
- *достань из холодильника жёлтую кастрюлю,*
- *поставь кастрюлю на газовую плиту,*
- *зажги газ,*
- *подожди 5 минут,*
- *выключи газ,*
- *налей из кастрюли суп в тарелку.*

К какому типу относится данный алгоритм (с обратной связью или без обратной связи?) Поясните свой ответ.

4. Приведите примеры использования встроенных в бытовые приборы микропроцессоров (3-4 примера).

задание

Тема: *Управление и кибернетика. Управление с обратной связью*

5. Какие из приведённых ниже систем относятся к САУ, а какие - к АСУ:

- система противопожарной сигнализации,
- компьютеризированная система «Метеоролог»,
- компьютеризированная система управления предприятием,
- станок с числовым программным управлением,
- «автопилот» в самолёте,
- компьютеризированная система управления электроснабжением.

АСУ:

САУ:

ВОЗНИКНОВЕНИЕ ПОНЯТИЯ «АЛГОРИТМ»



Слово "**алгоритм**" происходит от имени выдающегося математика средневекового Востока **Мухаммеда аль-Хорезми** (787-850). Около 825 года он написал книгу, в которой им были предложены приёмы выполнения арифметических вычислений с многозначными числами.

В первой половине XII века книга аль-Хорезми в латинском переводе проникла в Европу. Переводчик (имя его неизвестно) дал ей название **Algoritmi de numero Indorum** («Алгоритми о счёте индийском»). Слово **algorism** (или **algorismus**) обрело значение способа выполнения арифметических действий посредством арабских цифр, то есть на бумаге, без использования абака. Именно в таком значении оно вошло во многие европейские языки

Таким образом, сочинения по искусству счёта стали называть алгоритмами.

РАЗВИТИЕ ПОНЯТИЯ «АЛГОРИТМ»



Готфрид Лейбниц
(1646-1716)

В 1684 году **Готфрид Лейбниц** в сочинении «Nova Methodus pro maximis et minimis, itemque tangentibus...» впервые использовал слово «алгоритм» (Algorithmus) в ещё более широком смысле: как систематический способ решения проблем дифференциального исчисления.



Леонард Эйлер
(1707-1783)

Пользовался словом "алгоритм" и ещё один выдающийся математик - **Леонард Эйлер**, одна из работ которого так и называется — «Использование нового алгоритма для решения проблемы Пелля». Здесь видно, что Эйлер уже понимает алгоритм в ещё более широком смысле, а именно: как синоним способа решения задачи.

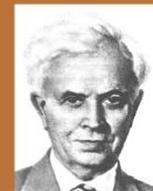
В 30-ые годы XX века возникает научное направление "**Теория алгоритмов**", предметом исследования которого стала разработка универсальной алгоритмической модели. Наибольший вклад в теорию алгоритмов внесли английский математик **Алан Тьюринг** и русский математик **Андрей Марков**.

Алан Тьюринг в 1935-1936 годах создаёт теорию "логических вычисляющих машин". Разработанная им "машина Тьюринга" стала обязательной частью обучения будущих математиков и компьютерщиков. На одной из лондонских гостиниц мемориальная доска гласит: "**Здесь родился Алан Тьюринг (1912 — 1954), взломщик кодов и пионер информатики**".

Андрей Марков в 1947 ввёл понятие "нормального алгоритма" и впервые систематически и строго построил общую теорию алгоритмов. Современные языки символьной обработки информации (Пролог) берут своё начало от нормальных алгоритмов Маркова.



Алан Тьюринг
(1912-1954)



Андрей Марков
(1903-1979)

Единого «истинного» определения понятия «алгоритм» не существует. Вот лишь некоторые из предлагаемых определений:

«**Алгоритм** — это всякая система вычислений, выполняемых по строго определённым правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи.» (А. Колмогоров)

«**Алгоритм** — это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату.» (А. Марков)

«**Алгоритм** — строго детерминированная последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд.» (Угринович Н.)



СВОЙСТВА АЛГОРИТМА

Любой алгоритм должен удовлетворять четырём основным свойствам:



Кроме того, для выполнения любого алгоритма должен иметься определённый набор **исходных данных**.

Конечность алгоритма означает, что за конечное число шагов должен быть получен результат. Поэтому иногда это свойство называют **результативностью**.

Пример:

Пусть имеется последовательность команд:



1. Взять книгу,
2. Открыть первую страницу.
3. Пока не конец книги выполнять следующие действия:
 - 3.1 Прочитать текст
 - 3.2 Перелистнуть книгу на следующую страницу
 - 3.3 Прочитать текст
 - 3.4 Открыть первую страницу

Легко догадаться, что данная последовательность команд будет выполняться бесконечно и поэтому алгоритмом не является.

Что надо изменить в алгоритме, что бы он стал конечным?

ПОДСКАЗКА

Надо исключить из алгоритма пункты 3.3 и 3.4

Следующее свойство алгоритма - дискретность.

Дискретность означает, что алгоритм должен быть разбит на последовательность отдельно выполняемых шагов.

Пусть необходимо решить следующий пример: $(80+10)-5*(3+5)=$

Запишем алгоритм решения примера, разбив его на шаги:

1. Вычислить $80+10$
2. Вычислить $3+5$
3. Умножить 5 на результат предыдущего действия
4. Вычесть из результата 1-го действия результат 3-го действия

В результате выполнения алгоритма получим 50.

Если в данном алгоритме начать, например, выполнять четвёртое действие, не дожидаясь окончания выполнения третьего, то результат не может быть получен.

Понятность алгоритма означает, что алгоритм должен содержать только те команды, которые входят в СИИ.



Рассмотрим алгоритм:

1. Пойти на кухню
2. Вскипятить чайник
3. Насыпать в чашку 1 чайную ложку кофе
4. Положить в чашку 3 чайных ложки сахара
5. Налить полную чашку кипячёной воды

Очевидно, что он легко может быть выполнен 10-летней девочкой, которая понимает все команды, входящие в данный алгоритм. Однако, для 10-месячного малыша данный алгоритм будет непонятен.

Точность алгоритма означает, что любая его команда должна определять однозначное действие исполнителя. Иными словами, алгоритм не должен быть рассчитан на принятие каких-либо самостоятельных решений исполнителем.

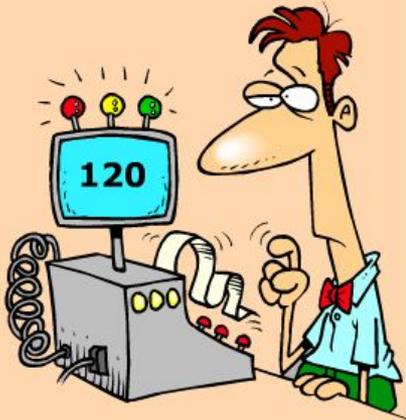


Рассмотрим следующий алгоритм, описывающий, как добраться до стадиона :

1. Идти прямо
2. Повернуть
3. Идти прямо
4. Сесть на автобус
5. Доехать до остановки "Стадион"

Данный алгоритм не уточняет, какое расстояние нужно пройти прямо, в какую сторону повернуть, на какой автобус сесть, поэтому разные исполнители будут выполнять его по-разному и цель вряд ли будет достигнута.

Для успешной работы алгоритма необходимо также, чтобы имелся **полный набор исходных данных**, необходимый для его выполнения. Если исходные данные неполные, то либо задачу вообще нельзя решить, либо она будет иметь неоднозначное решение.



Пусть вы пришли в магазин самообслуживания и решили подсчитать стоимость предполагаемых покупок, чтобы узнать, хватит ли вам денег. Вам нужно купить 2 кг сахарного песка, 3 кг муки и 2 батона хлеба. Тогда для вычисления общей стоимости вам надо:

1. Умножить стоимость 1 кг сахарного песка на 2
2. Умножить стоимость 1 кг муки на 3
3. Умножить стоимость 1 батона на 2
4. Сложить все полученные результаты

Если по какой-либо причине не будет вывешен ценник хотя бы для одного из продуктов, то данная задача не сможет быть решена.

Обобщая всё сказанное, можно дать следующее определение алгоритма:



Алгоритм - это понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящих от исходных данных к искомому результату

Поскольку у алгоритма такие свойства, то работа по нему будет производиться исполнителем формально. От исполнителя не требуется понимания сущности алгоритма, он должен лишь точно выполнять команды, не нарушая их последовательности.

задание

Тема: *Определение и свойства алгоритма*

1. Запишите исполнителей для приведённых ниже видов работ:

- Уборка мусора во дворе –
- Перевозка пассажиров в поезде –
- Выдача заработной платы –
- Приём экзаменов в школе –
- Сдача экзамена в университете –
- Набор текста на компьютере –
- Приготовление еды в ресторане -

2. Есть исполнитель «Перевозчик», который перевозит через реку волка, козу и капусту. Напишите алгоритм перевоза через реку волка, козы и капусты, если СКИ «Перевозчика» содержит 5 команд: **ВЗЯТЬ КОЗУ, ВЗЯТЬ ВОЛКА, ВЗЯТЬ КАПУСТУ, ВЫСАДИТЬ, ПЕРЕПЛЫТЬ**. В лодку может поместиться только один предмет или животное. Нельзя оставлять на берегу одних волка с козой и козу с капустой.

задание

Тема: *Определение и свойства алгоритма*

3. Определите полный набор данных для решения следующих задач обработки информации:

- а) вычисление стоимости покупок в магазине
- б) вычисление суммы сдачи от данных Вами продавцу денег
- в) определение времени показа по телевизору интересующего Вас фильма
- г) вычисление площади треугольника
- д) определение времени падения кирпича с крыши дома
- е) определение месячной платы за расход электроэнергии
- ж) перевод русского текста на итальянский язык
- з) перевод итальянского текста на русский язык

4. Напишите алгоритм приготовления какого-либо блюда (алгоритм должен иметь линейную структуру).

задание

Тема: *Определение и свойства алгоритма*

5. Есть исполнитель «Арифмометр», который понимает следующие команды:

- взять число N (занести в память число N),
- умножить (перемножаются занесённые в память последние два числа),
- сложить (складываются занесённые в память последние два числа),
- вычесть (вычисляется разность занесённых в память последних двух чисел),
- результат (вывести результат)

Например, в результате выполнения алгоритма:

- взять число 5,
- взять число 10,
- взять число 2,
- вычесть,
- умножить,
- результат

получим ответ 40, так как $5 \cdot (10 - 2) = 40$.

Какой результат будет получен при выполнении приведённого ниже алгоритма?

- взять число 4,
- взять число 8,
- взять число 2,
- вычесть,
- взять число 10,
- умножить,
- взять число 56,
- вычесть,
- вычесть,
- результат.

Дайте объяснение своему ответу (приведите формулу для вычисления).

задание

Тема: *Определение и свойства алгоритма*

6. Почему приведённые ниже алгоритмы для исполнителя «Арифмометр» не могут быть выполнены (какие свойства алгоритма нарушены)?

А) – взять число 4,

- взять число 5,
- умножить,
- вычесть,
- результат.

Б) – взять число 6,

- взять число 3,
 - разделить,
 - результат
- В) – взять число,
- взять число,
 - сложить,
 - результат

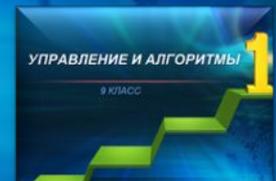
А)

Б)

В)

Способы описания

1. Словесно-формульный
2. Графический (с помощью блок-схем)
3. На алгоритмическом языке



При словесно-формульном способе алгоритм записывается в виде текста с формулами по пунктам, определяющим последовательность действий.

Пусть, например, необходимо найти значение следующего выражения:

$$y = 2a - (x+6)$$

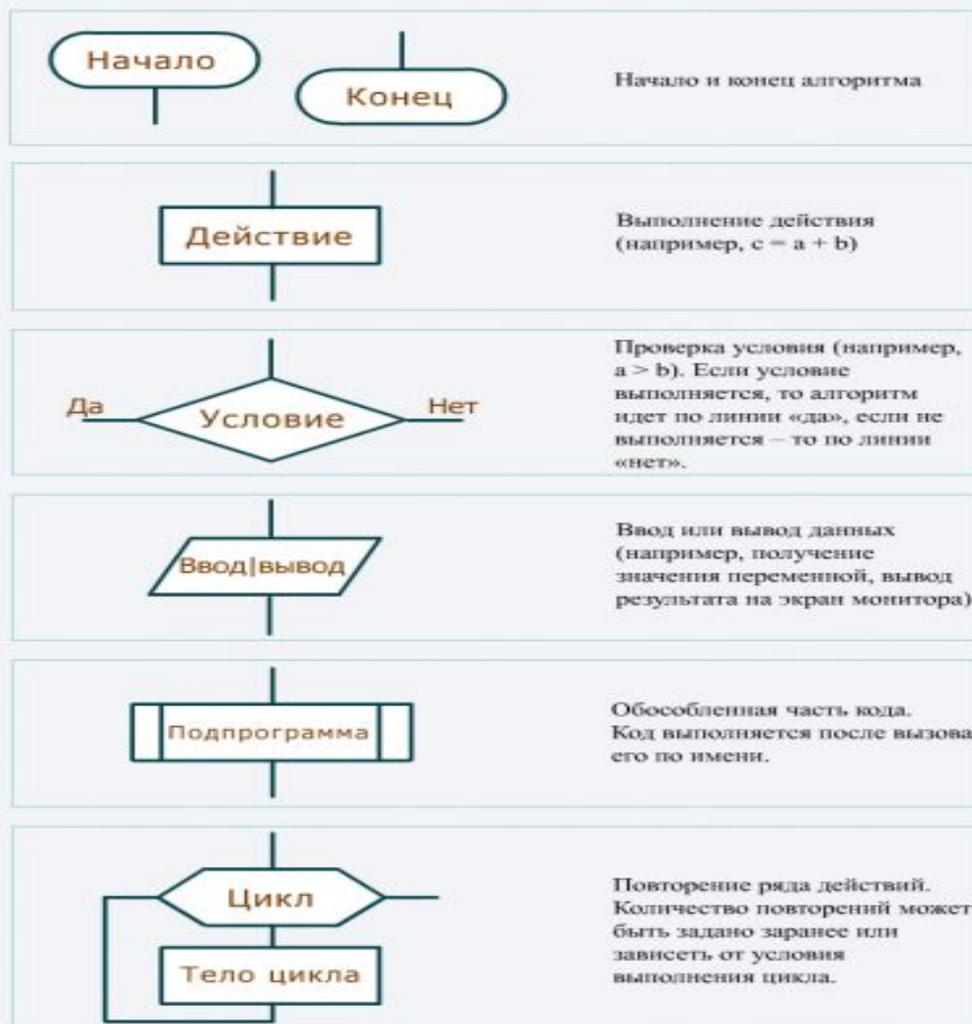
Словесно-формульным способом алгоритм решения этой задачи может быть записан в следующем виде:

1. Ввести значения a и x .
2. Сложить x и 6 .
3. Умножить a на 2 .
4. Вычесть из $2a$ сумму $(x+6)$.
5. Вывести y как результат вычисления выражения.

Графический (с помощью блок-схем)

<http://inf1.info>

Основные элементы блок-схем



При блок-схемном описании алгоритм изображается геометрическими фигурами (блоками), связанными по управлению линиями (направлениями потока) со стрелками. В блоках записывается последовательность действий.



На алгоритмическом языке

Это запись алгоритма на специальном языке (в том числе и на языке программирования).

Она осуществляется, строго следуя правилам того или иного алгоритмического языка.

Заголовок включает в себя название алгоритма, имена исходных данных (это величины, без которых выполнить алгоритм невозможно) и имена результатов (это величины, значения которых вычисляются в алгоритме).

Для указания начала и конца алгоритма используются служебные слова нач и кон.

Между ними записывают одну или несколько команд алгоритма, их называют тело алгоритма.

Например:

Алгоритм вычисления значения выражения $Y=z-a+2b$.

алг ВЗВ $Y=z-a+2b$

арг z, a, b

рез Y

нач

$Y := z - a + 2 * b$

кон

<- название алгоритма

<- исходные данные (аргументы)

<- результат

<- начало алгоритма

<- тело алгоритма

<- конец алгоритма



Стадии создания алгоритма

1. Алгоритм должен быть представлен в форме, понятной человеку, который его разрабатывает.
2. Алгоритм должен быть представлен в форме, понятной тому объекту (в том числе и человеку), который будет выполнять описанные в алгоритме действия.

Исполнители алгоритмов

Объект, который будет выполнять алгоритм, обычно называют исполнителем.

Исполнитель - объект, который выполняет алгоритм.

Идеальными исполнителями являются машины, роботы, компьютеры...

Компьютер – автоматический исполнитель алгоритмов.

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется программой.

Исполнители алгоритмов

Каждый исполнитель имеет следующее:

Среда - это «место обитания» исполнителя.

Система команд. Каждый исполнитель может выполнять команды только из некоторого строго заданного соответствующее элементарное действие.

Знать систему команд исполнителя это значит:

- знать название или обозначение каждой команды исполнителя;
- знать, каким образом она передается исполнителю;
- знать, как выполняется каждая команда.

Формальное исполнение. Исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные команды, не задавая вопросов «почему?» и «зачем?».

Управление исполнителями заключается в последовательном вызове команд. Человек дает команду исполнителю, анализирует результат, отдает следующую команду и т.д.

Графический учебный исполнитель

Система команд:

шаг – перемещение ГРИС на 1 шаг вперед с рисованием линии;

поворот – поворот на 90 градусов против часовой стрелки;

прыжок – перемещение на 1 шаг вперед без рисования линии

задание

Тема: *Определение и свойства алгоритма*

1. Имеются цинк, 96%-ная серная кислота, вода, а также колба и пробирка. Исправьте ошибки в алгоритме получения водорода:

Поставить колбу на стол

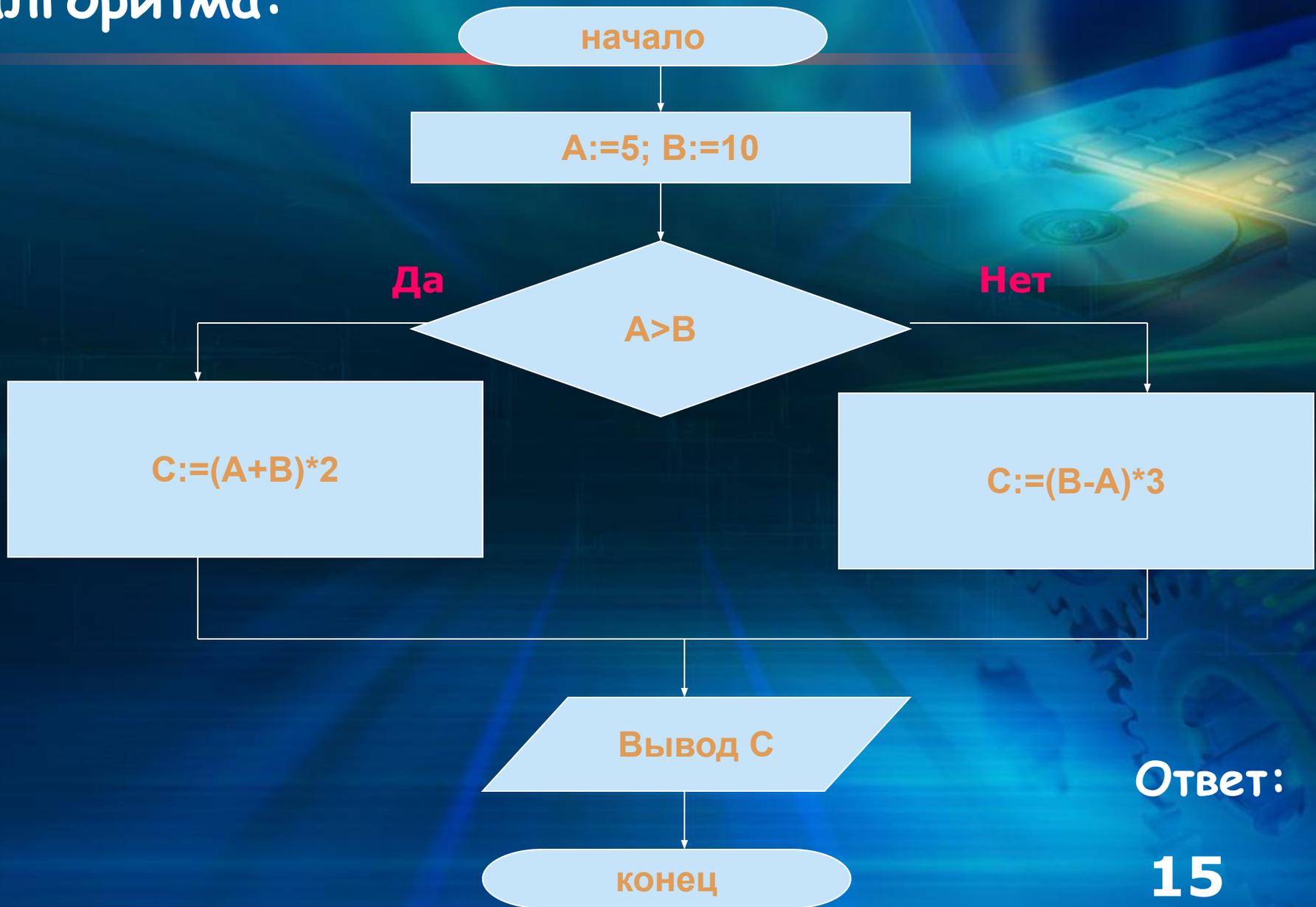
Налить в колбу кислоту

Налить в колбу воду

Собрать выделяющийся газ в пробирку

Бросить в колбу цинк

2. Определите результат выполнения алгоритма:



Ответ:

15

Определить значение переменной X при заданном значении переменной A :



A	2	3
X	2	5

Первый тип — **линейный алгоритм**; такой, в котором все действия выполняются в строгом порядке, последовательно, одно за другим.

Типичный жизненный *пример* такого алгоритма — рецепт пирога.

Второй тип — **разветвляющийся алгоритм**; такой, в котором выполняются те или иные действия в зависимости от выполнения или невыполнения некоего условия.

Пример из жизни — правило перехода улицы по светофору. Если горит красный — стоим, если горит зеленый — идем.

Третий тип — **циклический алгоритм**; такой, в котором присутствуют повторяющиеся действия с какой-либо изменяющейся величиной, так называемым параметром.

Пример — колка дров. Берем полено — колем топором, берем второе полено и т. д., пока поленья не закончатся, и эта работа нам не надоест.

Виды алгоритмов

1. **Линейный алгоритм – следование**
(описание действий, которые выполняются однократно в заданном порядке)



Виды алгоритмов

**2. Разветвляющийся алгоритм–
ветвление** (алгоритм, в котором в
зависимости от условия выполняется
либо одна, либо другая
последовательность действий)



Ветвление - это алгоритмическая структурная команда, которая определяет выбор того или иного действия в зависимости от истинности проверяемого условия.

Полное ветвление используется, если при истинном условии необходимо выполнить одну серию команд, а при ложном условии - другую.

Полное ветвление на языке блок-схем



Полное ветвление на алгоритмическом языке

Если <условие>

то <1 серия команд>

иначе <2 серия команд>

Конец ветвления

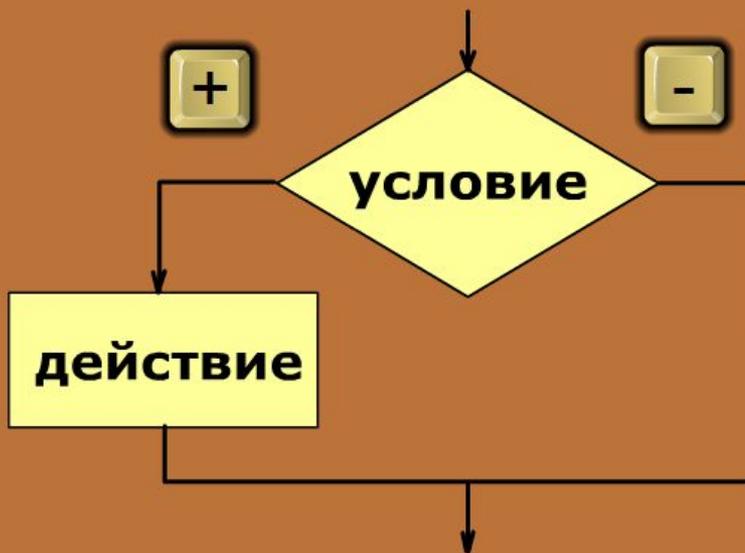
Пример полного ветвления



Неполное ветвление

Неполное ветвление используется, если при истинном условии необходимо выполнить какие-либо действия, а при ложном условии - ничего не делать.

Неполное ветвление на языке блок-схем



Неполное ветвление на алгоритмическом языке

Если <условие> то

<команды>

Конец ветвления

Примеры неполного ветвления



посмотреть в окно
если на улице идёт дождь
то взять зонтик
конец ветвления
выйти на улицу



посмотреть на календарь
если сегодня 15 апреля
то поздравить друга с Днём
рождения
конец ветвления

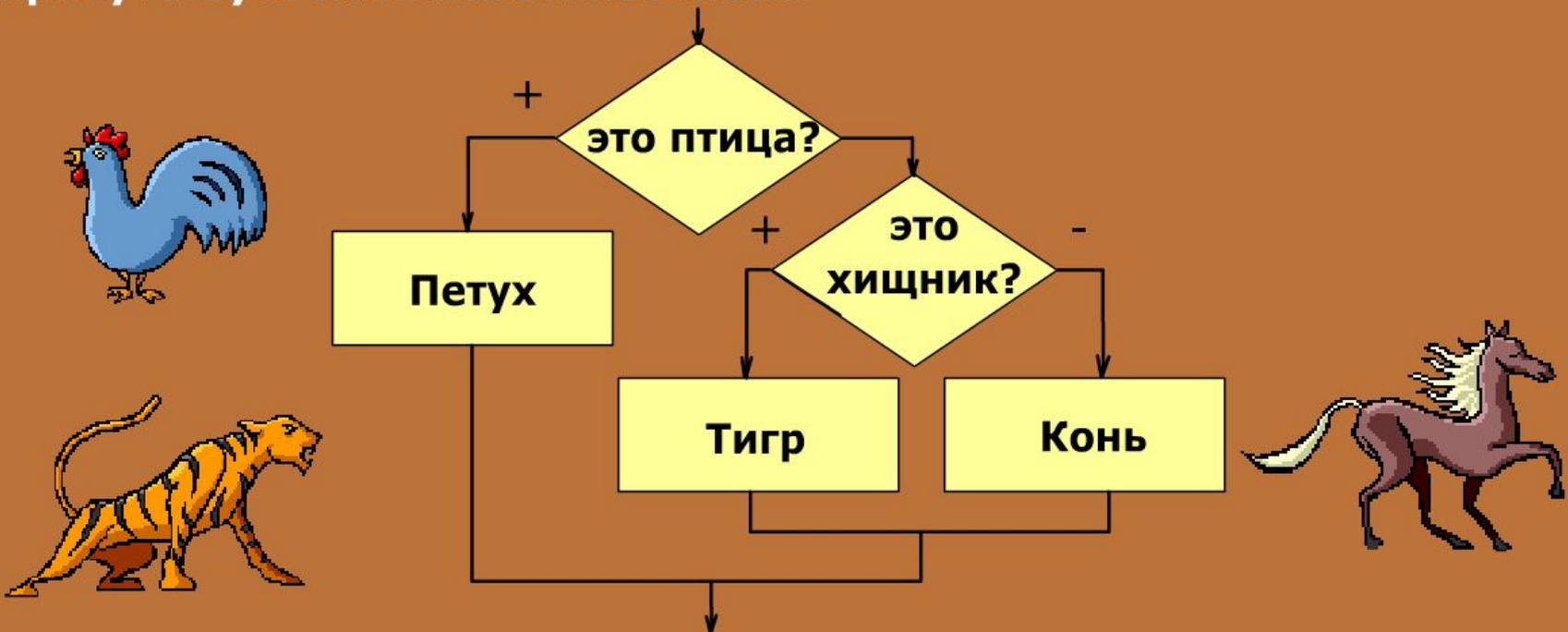


бросить монету в игровой автомат
нажать ручку
если выпала выигрышная комбинация
то забрать выигрыш
конец ветвления

Вложенные ветвления

Ветвления могут быть вложенными друг в друга.

Пример. Загадано одно из трёх живых существ: конь, петух, тигр. Необходимо составить оптимальный алгоритм для его угадывания, при условии, что можно задавать вопросы, требующие ответа только "да" или "нет". Нельзя задавать вопросы, в которых присутствует название животного.



Виды алгоритмов

3. Циклический алгоритм – повторение
(описание действий, которые должны повторяться указанное число раз или пока не выполнено задание)



Метод нисходящего проектирования

Метод программирования, при котором сначала составляется основная программа, в ней записываются обращения к пока ещё не составленным подпрограммам (процедурам), а затем описываются эти подпрограммы, называется **методом нисходящего программирования (программирования сверху вниз)** или **методом последовательной детализации**.



Пример. Спроектируем для графического исполнителя программу изображения слова БАОБАБ (высота букв 4 шага, ширина - 2 шага), используя метод последовательной детализации.

БАОБАБ



Рисование каждой отдельной буквы будет начинаться с правого нижнего угла (исполнитель смотрит вверх) и заканчиваться в левом нижнем (исполнитель смотрит влево).

Сделай Б
Сделай Переход
Сделай А
Сделай Переход
Сделай О
Сделай Переход
Сделай Б
Сделай Переход
Сделай А
Сделай Переход
Сделай Б

Процедура Б
Сделай Два_шаг
Сделай Два_прыжок

Поворот
Сделай Два_шаг
Поворот
Сделай Четыре_шаг
Поворот
Сделай Два_шаг
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_шаг
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_прыжок
Конец процедуры

Процедура А
Сделай Четыре_шаг
Поворот
Сделай Два_шаг
Поворот
Сделай Четыре_шаг
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_шаг
Поворот
Сделай Два_прыжок
Поворот
Сделай Два_прыжок
Конец процедуры

Процедура О
Сделай Четыре_шаг
Поворот
Сделай Два_шаг
Поворот
Сделай Четыре_шаг
Поворот
Сделай Два_шаг
Конец процедуры

Процедура Переход
Сделай Два_прыжок
Прыжок
Поворот
Конец процедуры

Процедура Четыре_шаг
Шаг
Шаг
Шаг
Шаг
Конец процедуры

Процедура Два_шаг
Шаг
Шаг
Конец процедуры

Процедура Два_прыжок
Прыжок
Прыжок
Конец процедуры

Метод восходящего проектирования

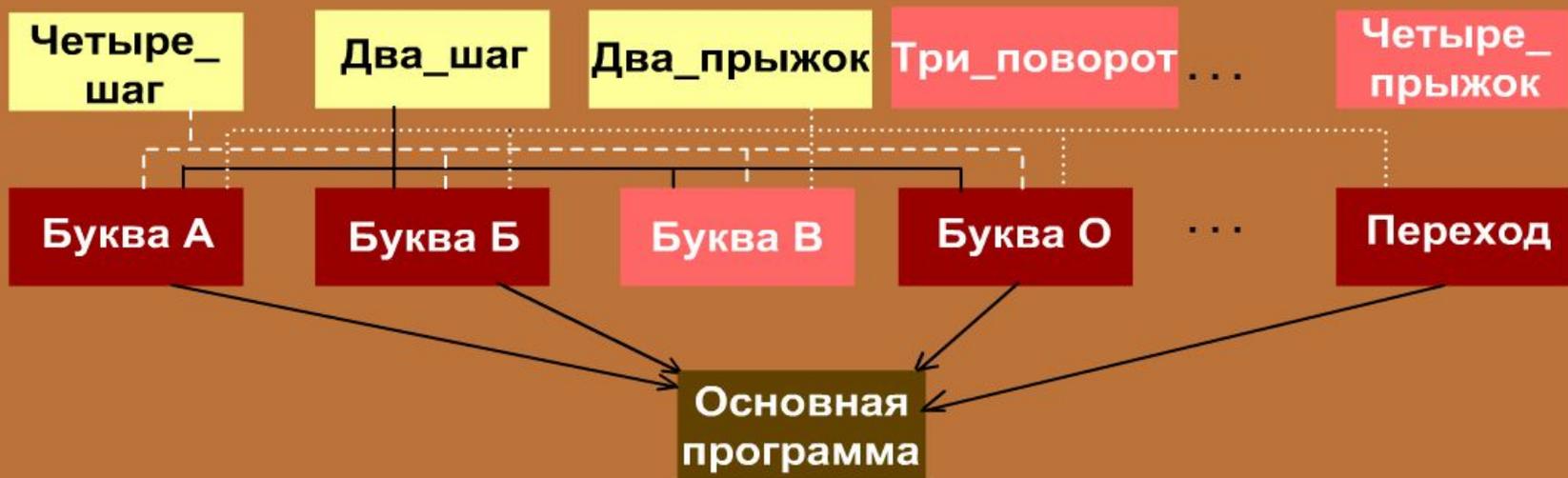
Метод программирования, при котором сначала составляется множество подпрограмм, которые могут понадобиться при решении задачи, а затем пишется основная программа, содержащая обращения к ним, называется **методом восходящего проектирования** алгоритма или **библиотечным методом**.

Созданные подпрограммы могут быть объединены в библиотеку, которая будет постепенно пополняться новыми подпрограммами. При использовании библиотечного метода некоторые созданные подпрограммы могут оказаться **незадействованы** в основном алгоритме.



Пример восходящего проектирования

Для рассмотренного примера с графическим исполнителем при использовании библиотечного метода проектирования сначала создаётся библиотека процедур, которые, возможно, будут использованы при написании того или иного слова, а потом пишется основная программа.



 - неиспользуемые текущей программой процедуры

Результат выполнения команд присваивания:

Команда	a	b
a:=1	1	-
b:=2*a	1	2
a:=b	2	2
b:=a+b	2	4

3 основных правила присваивания:

1. Пока переменной не присвоено значение, она остается не определенной;
2. Значение, присвоенное переменной, сохраняется в ней вплоть до выполнения следующего присваивания этой переменной нового значения;
3. Новое значение, присвоенное переменной, заменяет ее предыдущего значение.

В схематичном виде отразите изменение значений переменных А и В в ходе последовательного выполнения команд присваивания

A:=1

B:=2

A:=A+1

B:=2*A

A:=B+A

Команда	A	B
A:=1	1	-
B:=2	1	2
A:=A+1	2	2
B:=2*A	2	4
A:=B+A	6	4

Вопросы:

- Что такое алгоритм? Приведите примеры алгоритмов.
- Какие свойства алгоритмов вы знаете?
- Какие виды алгоритмов вы знаете?
- Какие способы записи алгоритмов вы знаете?
- Что такое исполнитель алгоритмов?
- Что такое программа?

www.themegallery.com

Thank You!

Your company slogan in here



L/O/G/O