

Понятие алгоритма и исполнителя

Алгоритм задаёт порядок действий, а исполнитель превращает команды в результат.



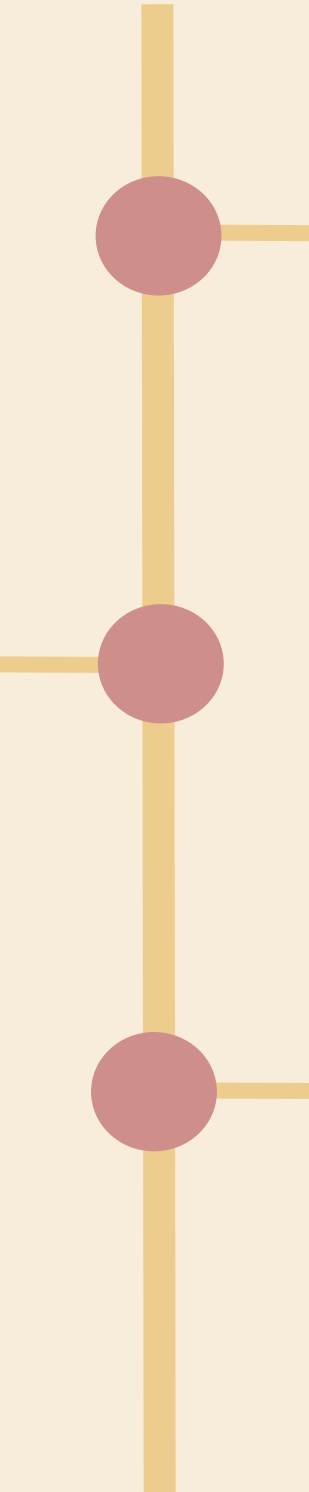
Где школьник встречает алгоритмы каждый день

Алгоритмы легко узнать по шагам: рецепт, правила игры, инструкция к конструктору и маршрут до школы. В информатике это связывают с именем Мухаммеда аль-Хорезми (IX век). Для 6 класса важно: любое последовательное действие можно представить как алгоритм.



Что такое алгоритм

Каждый шаг выполняется в определённом порядке. Если пропустить действие или поменять шаги местами, итог может стать другим: получится не то, что планировали, даже при правильных командах.



Алгоритм — конечная и понятная последовательность команд для решения задачи. У него есть цель: что получить в итоге, и старт: с какого шага всё начинается, затем идут конкретные действия.

После выполнения всех шагов должен быть результат. Он показывает, что алгоритм действительно завершился и привёл к нужному ответу или готовому действию.

Основные свойства алгоритма

1

Дискретность

Дискретность означает, что алгоритм разбит на отдельные шаги. Исполнитель выполняет их по очереди, а не «сразу всё». Благодаря этому легко проверить каждый этап.



Определённость

Определённость делает каждую команду точной: что именно нужно сделать и как. Если формулировка неоднозначна, исполнитель может понять её по-разному и получить неверный результат.



Понятность

Понятность предполагает, что команды записаны так, чтобы исполнитель их воспринимал. Для школьника это «язык действий», а для устройства — команды из его набора.



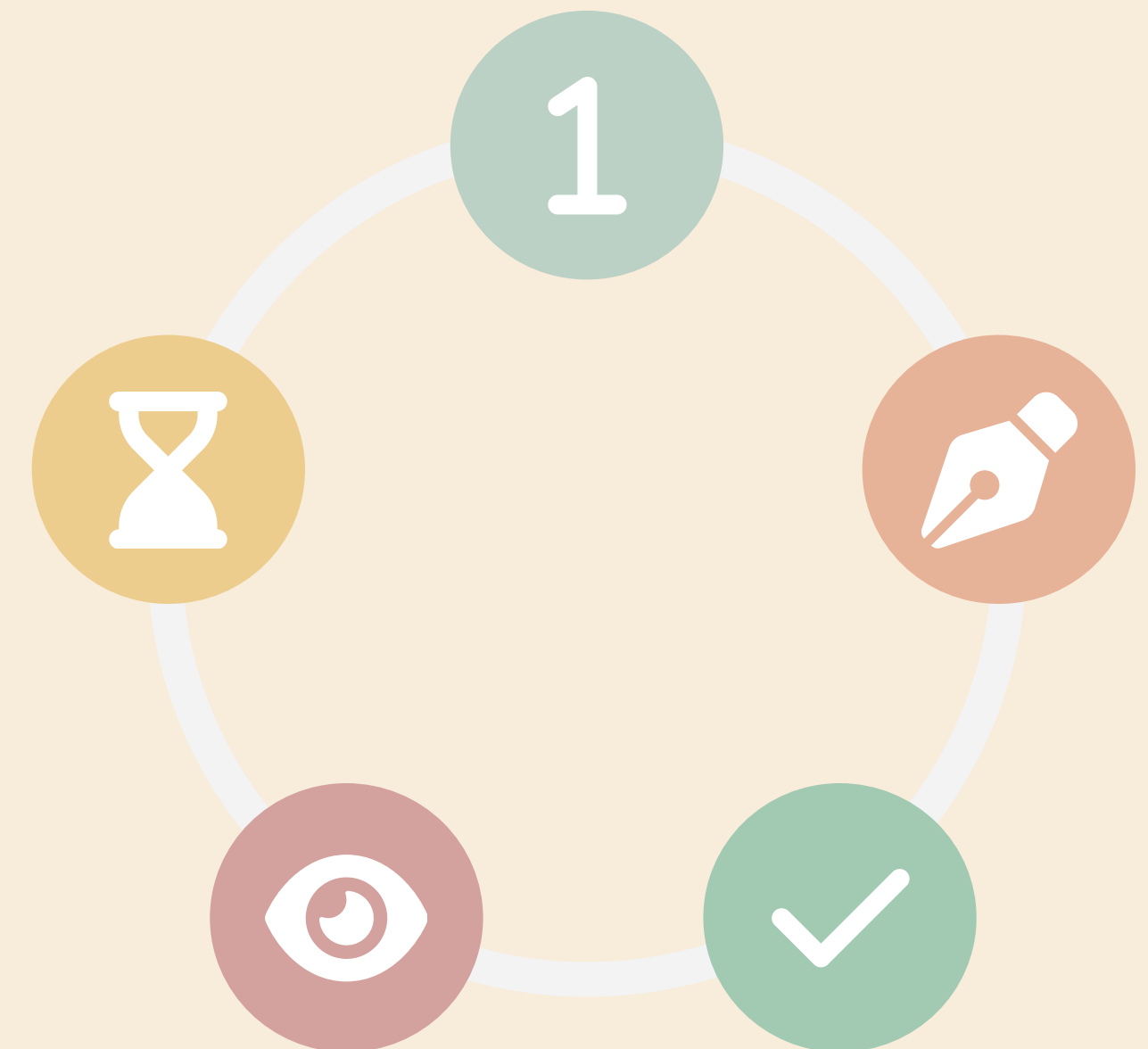
Конечность

Конечность означает, что алгоритм обязательно заканчивается. Он не должен работать бесконечно без результата: после нужного набора шагов наступает завершение.



Результативность

Результативность показывает, что в конце получается ответ или выполненное действие. То есть алгоритм не просто описывает процесс, а приводит к конкретному результату.





Алгоритм в привычных ситуациях

Повседневные шаги

Чистка зубов, сбор портфеля и приготовление бутерброда — это пример алгоритма: есть последовательность шагов и цель. Если убрать один пункт или нарушить порядок, результат ухудшится. Поэтому удобнее действовать по инструкции.

Цифровые действия

Запуск приложения на телефоне тоже можно описать алгоритмом: разблокировать экран, открыть нужное приложение, дождаться загрузки. Для удобства иногда используют нумерованный список, чтобы ничего не забыть и выполнить всё по порядку.

Как выполняется алгоритм

Поток действий: шаги выполняются по порядку, а при условии выбирается ветка.

Кто такой исполнитель

01

Исполнитель — это объект, который умеет понимать команды алгоритма и выполнять их. При этом он действует только в рамках своих возможностей и собственного набора команд, а не «как захочет».

02

Человек может выполнять словесные инструкции, если они понятны и выполнимы. Компьютер работает по программам и системным командам, а бытовое устройство — по фиксированным операциям и режимам.

03

Если команда выходит за возможности исполнителя, он не сможет её сделать. Поэтому при составлении алгоритма важно учитывать, кто именно будет его выполнять.

Сравнение исполнителей

Алгоритм исполняется только тогда, когда команды входят в систему команд конкретного исполнителя.

Главный вывод: исполнитель выполняет лишь те команды, которые соответствуют его системе команд.

Исполнитель	Какие команды понимает	Что умеет делать
Человек	Словесные инструкции	Выполняет действия по инструкциям
Компьютер	Программы, команды системы	Открывает файлы, считает, печатает
Робот	Программные команды	Двигается, поворачивает, останавливается, захватывает
Бытовое устройство	Кнопки, режимы	Работает по заранее заданным операциям

Система команд исполнителя

Система команд — это перечень действий, которые исполнитель способен выполнить. Команды задают границы его «умений», поэтому алгоритм строят именно под этот набор.

01

У компьютера система команд — это команды и операции, которые поддерживает операционная система и программы: открыть файл, сохранить, выполнить печать, провести вычисления.

03

02

У робота система команд может включать движение, поворот, остановку и захват предмета. Он реагирует на команды, но выполняет их строго в пределах доступных действий и параметров.

04

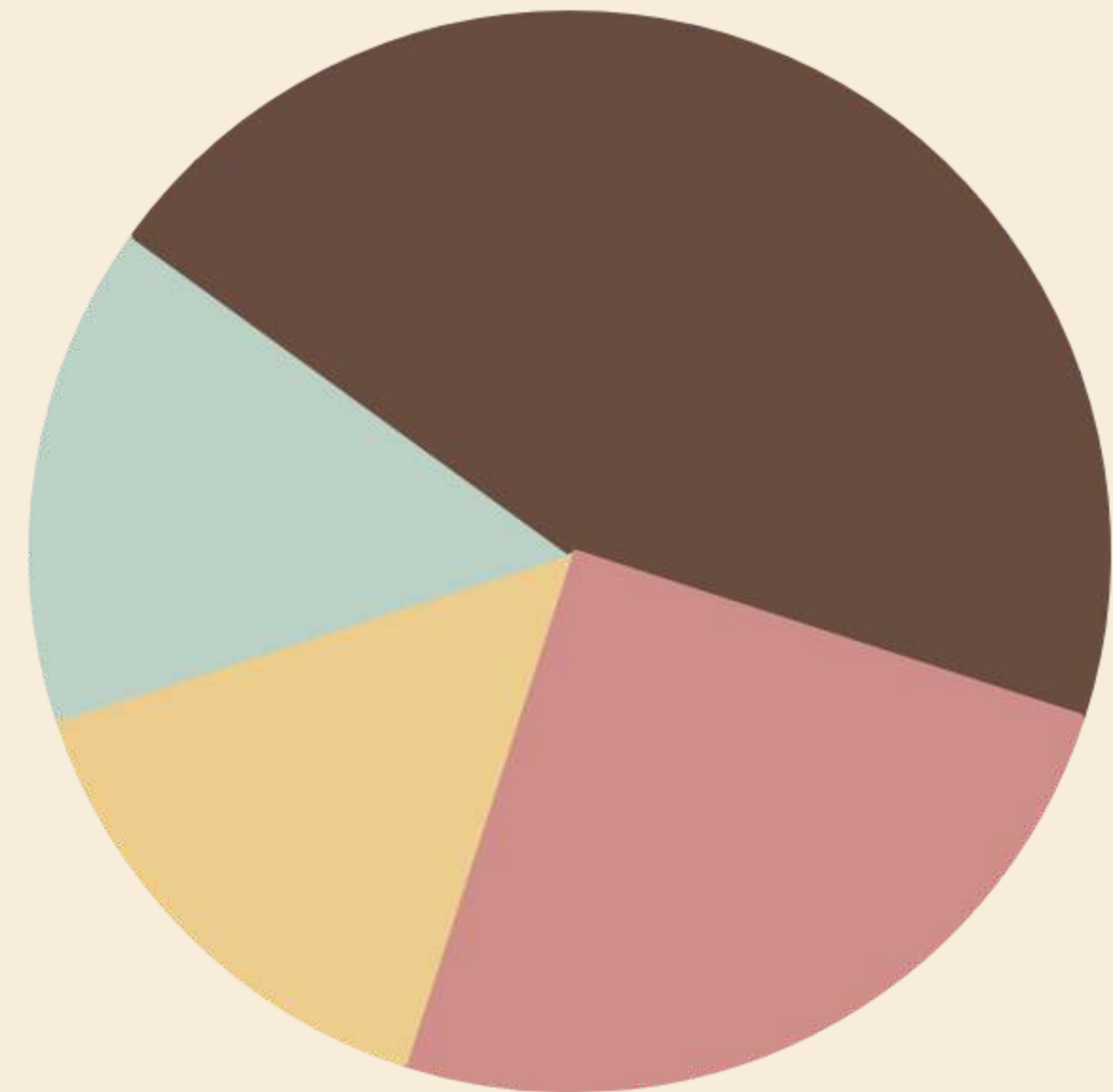
Если в алгоритме встречается команда, которой нет в системе исполнителя, результат не получится: исполнитель не сможет выполнить этот шаг, а значит алгоритм окажется непригодным.

Признаки хорошего алгоритма

Крупные части диаграммы подсказывают: чем точнее и понятнее команды, тем меньше вероятность ошибок при выполнении шагов.

Все признаки связаны между собой: вместе они обеспечивают рабочий алгоритм, который завершается и приводит к нужному результату.

Точность Понятность Конечность
Пошаговость Результативность



Сводное учебное представление о свойствах алгоритмов (модель для наглядности)

Линейные алгоритмы

Сначала исполнитель выполняет самое простое действие — подготовку. Например, умывание начинается с того, чтобы открыть воду и намочить руки, затем берётся нужное средство.

⋮

Шаг 1: подготовка

· ·

Шаг 2: основные действия

· ·

Шаг 3: завершение

⋮

В конце выполняется финальный шаг и получается результат: руки чистые, бутерброд готов, а на компьютере запускается нужная презентация. Алгоритм завершается без повторов и выборов.

⋮

Далее идут действия по порядку. При приготовлении бутерброда режут ингредиенты, выкладывают их на хлеб и аккуратно собирают заготовку, не пропуская этапы.

Разветвляющиеся алгоритмы

Разветвляющийся алгоритм всегда начинается с проверки условия: «выполняется или нет?». От этого зависит следующий шаг, поэтому решение можно описать как выбор между двумя путями.

Например, если идёт дождь, берут зонт; если дождя нет — не тратят время на поиск. Так же поступают при проверке задачи: нужно перейти к следующей или вернуться к исправлениям.

Иногда условие связано с результатом: если ответ неверный, выполняют корректировку. Такой подход помогает исполнителю не ошибаться и действовать по правилам, а не наугад.

10

Если одно действие требуется много раз, удобнее оформлять его как цикл и записывать правило один раз.

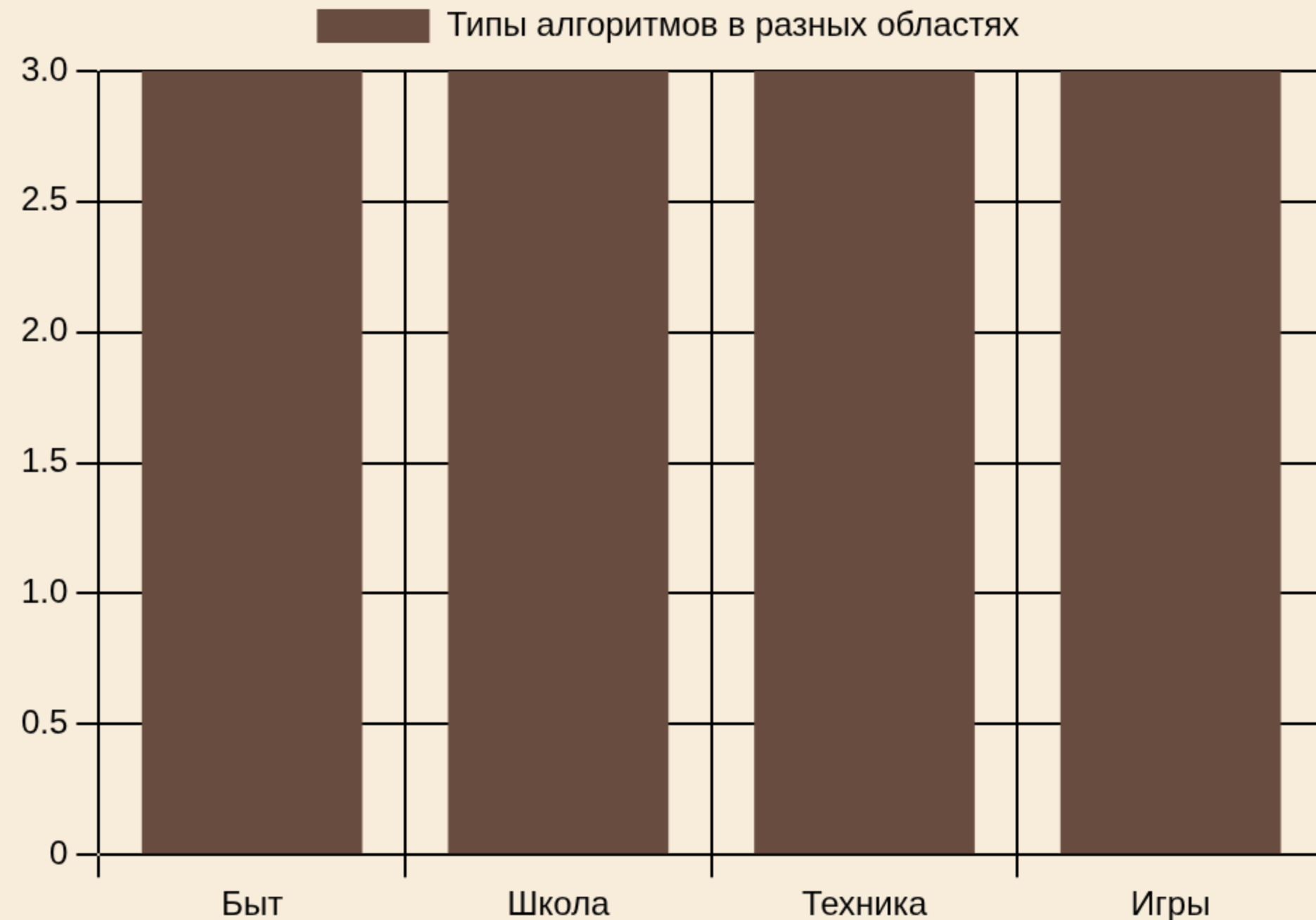
Учебные примеры по теме циклов (школьный курс информатики, 6 класс)

Повторение в алгоритмах экономит запись: вместо длинного списка шагов появляется понятный цикл. Это особенно заметно, когда нужно выполнить набор действий одинаковым образом.

Где встречаются разные типы алгоритмов

В быту чаще всего заметны простые линейные шаги, а в играх и технике обычно чаще встречаются ситуации выбора и многократного повторения действий.

Хотя в каждой области «заметнее» определённый стиль, один алгоритм может совмещать линейные участки, условия и циклы. Поэтому типы не разделены жёстко.



Обобщённое учебное сравнение (без конкретных измерений)



Ошибки в алгоритмах

Ошибки возникают, когда пропускают шаг, меняют порядок действий или формулируют команду слишком неточно. Исполнитель выполняет инструкции буквально, поэтому даже мелкая неточность заметно влияет на итог.

Например, если в рецепте «перепутать» последовательность добавления ингредиентов, получится блюдо не по плану. Аналогично и с моделями: неверно собранный узел меняет работу всей конструкции.

Проверка алгоритма: от идеи до исправлений

Постановка задачи → составление шагов → проверка понятности команд → пробное выполнение → поиск ошибок → исправление → повторная проверка.

Алгоритм, инструкция и программа

Понятие	Кто использует	Пример
Алгоритм	человек и исполнитель	порядок сборки конструктора
Инструкция	человек или устройство	правила работы с прибором
Программа	компьютер	приложение, которое считает данные

Алгоритм задаёт последовательность действий для решения задачи. Инструкция — правила выполнения действия. Программа — запись алгоритма для компьютера.

Программа по смыслу — это алгоритм, записанный так, чтобы компьютер мог выполнять команды без «догадок».

Почему алгоритмы важны в информатике



Поиск и сортировка информации

Алгоритмы помогают находить нужные сведения, упорядочивать данные и быстро отвечать на запрос. Без чётких правил компьютер не смог бы корректно сравнивать и размещать элементы.



Навигация и маршруты

Навигаторы используют алгоритмы, чтобы строить маршрут и пересчитывать его при изменениях. Условные проверки и повторения позволяют выбрать более подходящий вариант пути.



Игры и логика персонажей

В играх алгоритмы задают поведение объектов: движение, реакции на события и правила победы. Там особенно важны условия и циклы, которые делают действия «живыми».



Мессенджеры и обработка сообщений

Алгоритмы поддерживают работу чатов: доставка сообщений, обработка уведомлений и ответы на команды пользователя. Точность алгоритма влияет на скорость и корректность работы сервиса.



Школьные программы и вычисления

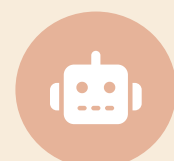
Когда компьютер считает, открывает файлы и выполняет команды, он работает по программам — записанным алгоритмам. Чем лучше алгоритм, тем меньше ошибок и сбоев.

Исполнители в цифровом мире



Смартфон

Смартфон выступает исполнителем: он открывает приложения, показывает уведомления и выполняет команды пользователя. Его возможности ограничены набором поддерживаемых программ и настроек.



Робот-пылесос

Робот-пылесос движется по заданной схеме и реагирует на препятствия. Он может делать только то, что предусмотрено программой: перемещаться, останавливаться и выполнять уборку по этапам.



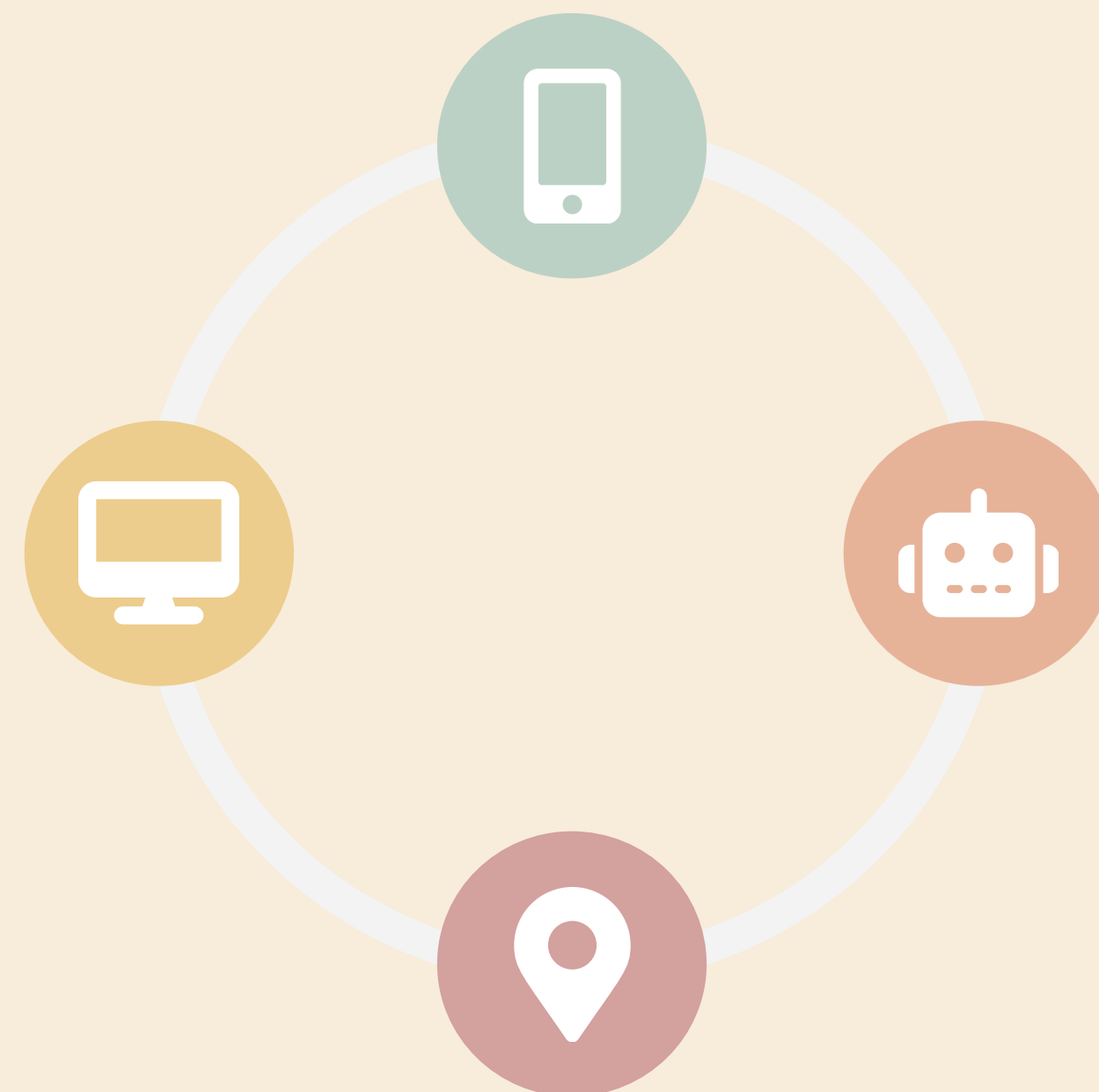
Навигатор

Навигатор строит маршрут и подсказывает направления во время поездки. Для него важны точные входные данные и команды: выбрать дорогу, перестроить путь и показать ориентиры.



Школьный компьютер

Компьютер выполняет программы: запускает задачи, обрабатывает данные и отображает результаты. Как исполнитель, он «понимает» только те команды, которые записаны в программе.



Алгоритм управляет действиями — исполнитель превращает их в результат

Алгоритм задаёт порядок действий, а исполнитель превращает команды в результат. Понятность, точность и конечность делают алгоритм рабочим. Один и тот же принцип помогает решать задачи и управлять устройствами.