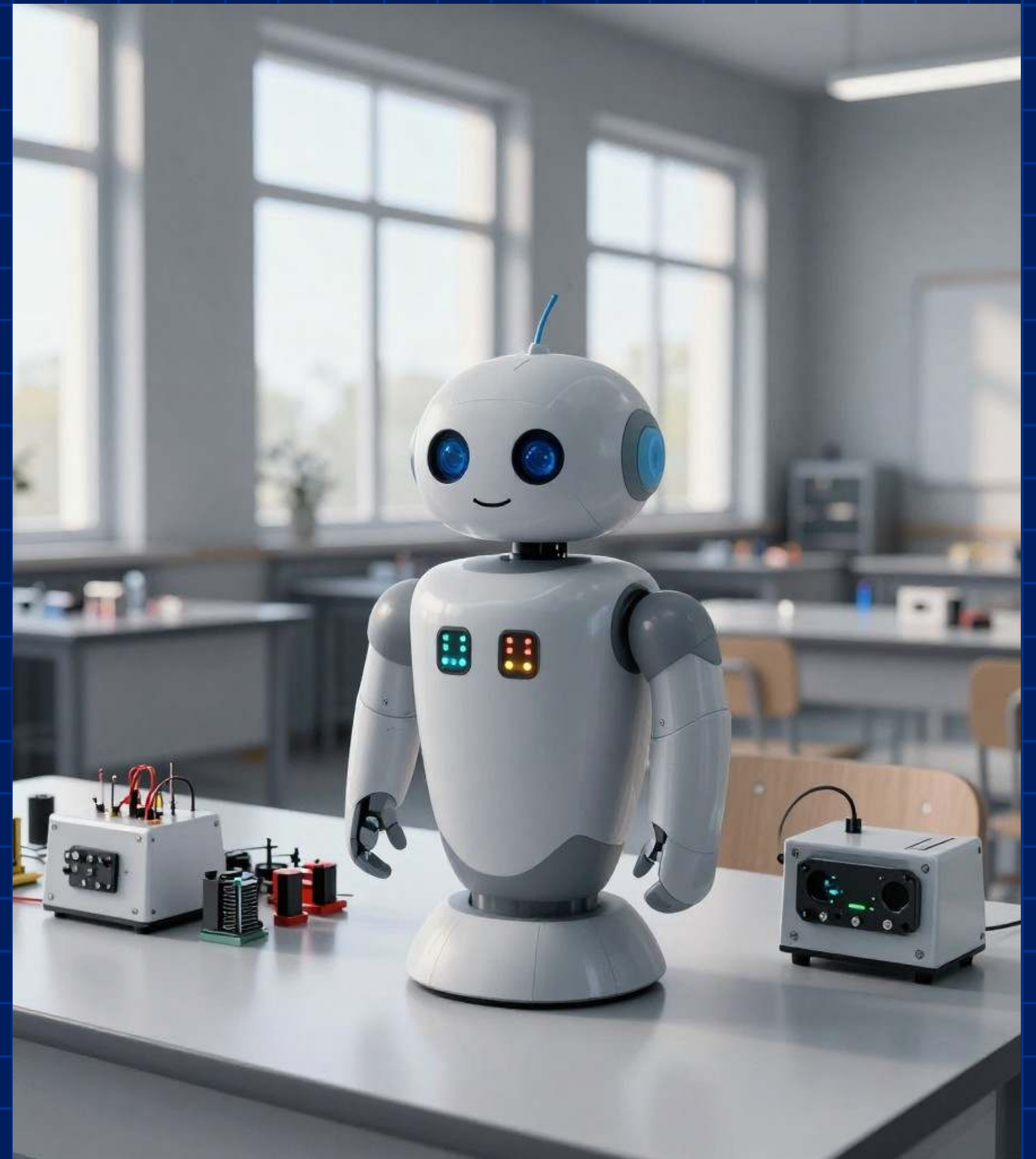


Команды-запросы робота и условие

Робот спрашивает датчики, сравнивает ответы и выбирает действие по правилу «если-то».

1 $\div$ 2 $\log$

8



Зачем изучают условия на уроке информатики

Условие – проверка, которая помогает программе выбирать между вариантами. В играх, приложениях и роботах это основа алгоритмов и логики. В 6 классе обычно рассматривают простые случаи: нажата ли кнопка, близко ли препятствие, темно ли вокруг и корректно ли сравнение чисел.



Что делает команда-запрос

1

log



Запрос у датчика

Команда-запрос получает ответ от датчика или устройства ввода. Этот ответ можно представить как «истину» или «ложь», чтобы дальше принять решение в программе робота.



Проверка ситуации

Запрос формулирует вопрос в понятной форме: «есть ли перед роботом препятствие?», «нажата ли клавиша?», «касается ли датчик поверхности?». Так робот переводит реальность в данные.



Сравнение с правилом

Команда-запрос часто связана со сравнением: меньше ли расстояние заданного значения, сработал ли датчик контакта, совпал ли сигнал с нужным признаком. Это запускает нужную ветку.



Отличие от обычной команды

Обычная команда выполняет действие. Команда-запрос не только подсказывает «что делать», но и проверяет «что именно происходит сейчас», чтобы действие было обоснованным.

2

x

Как работает логика: датчик → условие → действие



Датчик сообщает, что происходит

Сначала робот считывает данные с датчика. Например, он «видит» расстояние, ощущает касание или различает цвет. Эти сведения нужны для дальнейшей проверки — без данных условие не заработает.



Условие сравнивает с порогом

Дальше программа сравнивает полученное значение с порогом или признаком. Если условие выполнено, это означает, что ситуация соответствует правилу. При другом результате выбирается другой вариант действий.



Действие включается после проверки

После проверки робот выполняет команду, соответствующую выбранной ветке. Так при расстоянии меньше 10 см он поворачивает, при нажатой кнопке начинает движение, а при обнаружении линии меняет направление.

Условный переход: как выбирается ветка

Датчик → сравнение → ветвление: первая ветка при истинном условии, вторая – при ложном.



Основные виды условий

Вид условия	Пример проверки	Результат
Равно	расст. = 10 см	действие 1
Не равно	цвет ≠ черный	действие 2
Больше	скорость > 50	действие 3
Меньше	расст. < 15 см	действие 4
Датчик-ответ	кнопка нажата	старт

Школьные примеры по робототехнике (условные проверки)

Условия бывают разными: они проверяют равенство, различие, а также «больше» или «меньше», опираясь на ответ датчика.

Один и тот же робот может реагировать по-разному: нужный выбор определяется видом условия и тем, как датчик описывает реальность.



Порог — это число, с которым сравнивают значение датчика. Если датчик меняется и пересекает порог, программа переключается на другую ветку.

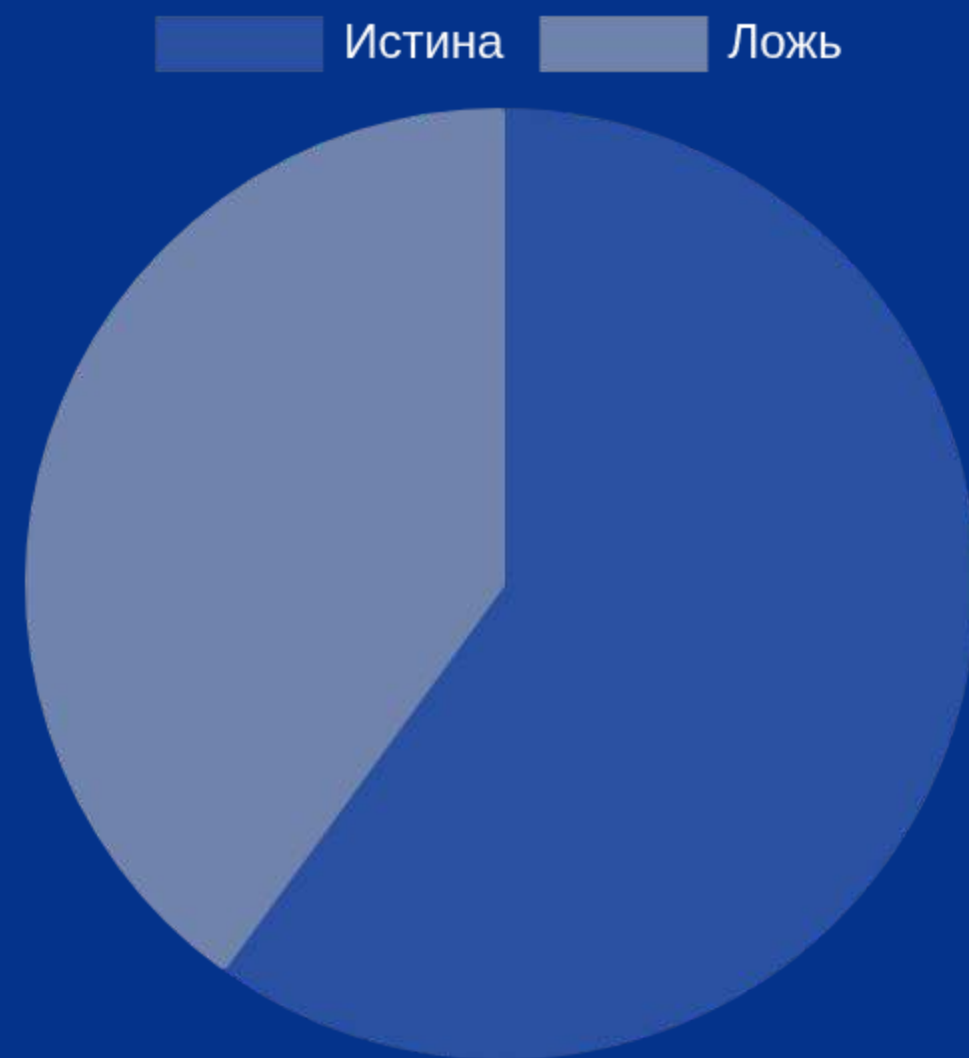
10

**Когда расстояние
оказывается меньше
порога, робот выбирает
действие немедленно.**

Обобщение примеров из школьной робототехники (условие расстояния $<$ порога).



Результаты проверки условия



Учебный пример: 20 проверок условия, 12 истинных и 8 ложных.

Если условие истинно чаще, одна ветка выполняется чаще. Но точные доли зависят от датчика, выбранного порога и обстановки вокруг робота.

Диаграмма показывает соотношение ветвей в учебном примере: робот реагирует не «на всё одинаково», а по реальному результату проверки.

Логические слова: И, ИЛИ, НЕ



«И» — требуется оба условия

Связка «И» означает, что верными должны быть сразу две проверки. Например, робот может ехать дальше только при наличии линии и отсутствии препятствия одновременно.



«ИЛИ» — достаточно одного условия

Связка «ИЛИ» срабатывает, если выполнена хотя бы одна проверка. Например, старт может начаться при нажатой кнопке или при получении сигнала датчика.



«НЕ» — меняет смысл проверки

«НЕ» делает ответ противоположным: было «да» — становится «нет», и наоборот. Например, можно задать движение только когда не темно в помещении.



Какие датчики особенно подходят для условий

Датчик касания: реагирует на столкновение

Датчик касания «отвечает», произошло ли прикосновение или контакт с препятствием. Поэтому его удобно использовать в условиях вида «если датчик сработал, то остановись» или «поверни».



Датчик цвета: помогает следовать линии

Датчик цвета определяет, какой цвет он «видит» — например, черный или белый. Условие позволяет роботу удерживать траекторию: при черной линии продолжать движение, при другой — менять направление.



Датчик расстояния: определяет близость объекта

Датчик расстояния показывает, насколько далеко находится препятствие. Условие сравнивает показание с порогом, например 10 см: если слишком близко, робот заранее выбирает безопасное действие.



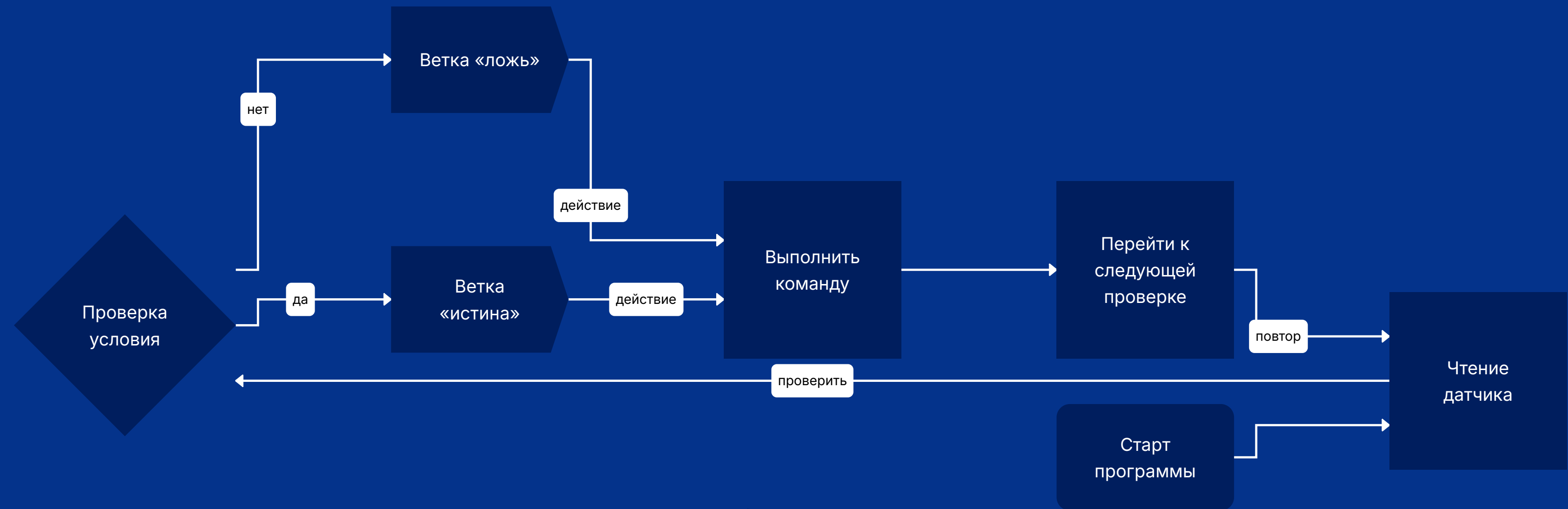
Датчик освещенности: различает свет и темноту

Датчик освещенности измеряет яркость вокруг. Условие позволяет реагировать: при низком уровне света включать подсветку или подавать сигнал, а при достаточной яркости — продолжать маршрут.



Ветвление программы: как робот выбирает действие

Датчик → проверка условия → выбор ветки → команда → возврат к следующему шагу



Типичные ошибки при составлении условий

Нередко берут порог «на глаз»: он слишком большой или слишком маленький. Тогда робот либо реагирует поздно, либо срабатывает постоянно, даже когда обстановка не требует действия.

Вторая частая проблема – перепутать знак сравнения: вместо «меньше» написать «больше» или заменить «равно» на «не равно». Из-за этого ветки меняются местами и поведение становится противоположным ожиданиям.

Если датчик настроен чувствительно неправильно, робот может «терять» нужный сигнал. Тогда условие срабатывает не в тот момент: он может проехать мимо линии или не заметить препятствие.

Ещё одна ошибка – забыть про вторую ветку. В результате программа выполняет одинаковые действия при истине и лжи, и условие перестаёт выбирать поведение по ситуации.

Где находится порог срабатывания



Учебная схема: пример линейного графика (точные данные датчика не заданы).

Когда значение датчика пересекает порог, условие меняет результат: программа выбирает другую ветку. В этом примере переключение происходит между третьей и четвертой точками.

Линия показывает, что одно и то же устройство может вести себя по-разному только из-за момента пересечения порога. Значит, корректная настройка порога критична.

Примеры команд-запросов в задачах

Если расстояние до стены меньше 10 см, робот поворачивает, чтобы избежать столкновения. Здесь команда-запрос получает данные датчика расстояния и сравнивает их с заранее заданным порогом.

Если кнопка нажата, робот начинает движение. Запрос проверяет не число, а состояние входа, поэтому вместо сравнения используется проверка признака «нажата/не нажата».

Если датчик цвета видит черную линию, робот едет по маршруту. Когда света слишком мало, по условию включается подсветка или подается сигнал, чтобы скорректировать дальнейшее поведение.



Проверка условий в разных задачах

Задача	Датчик	Условие	Действие
Обход препятствия	расстояния	меньше 10 см	поворот
Старт по кнопке	касания	кнопка нажата	движение
Поиск линии	цвета	цвет чёрный	движение прямо
Остановка при касании	касания	кнопка нажата	стоп

Примеры из школьной робототехники (учебные формулировки задач).

Во всех задачах применяется один принцип: датчик формирует запрос, условие сравнивает результат с требованием, а затем выполняется соответствующая команда.

Один и тот же подход работает с разными датчиками: меняются запрос и условие, но логика «если – то» остаётся одинаковой, поэтому алгоритм легко переносить между заданиями.

Как выглядит блок условия в программе

1

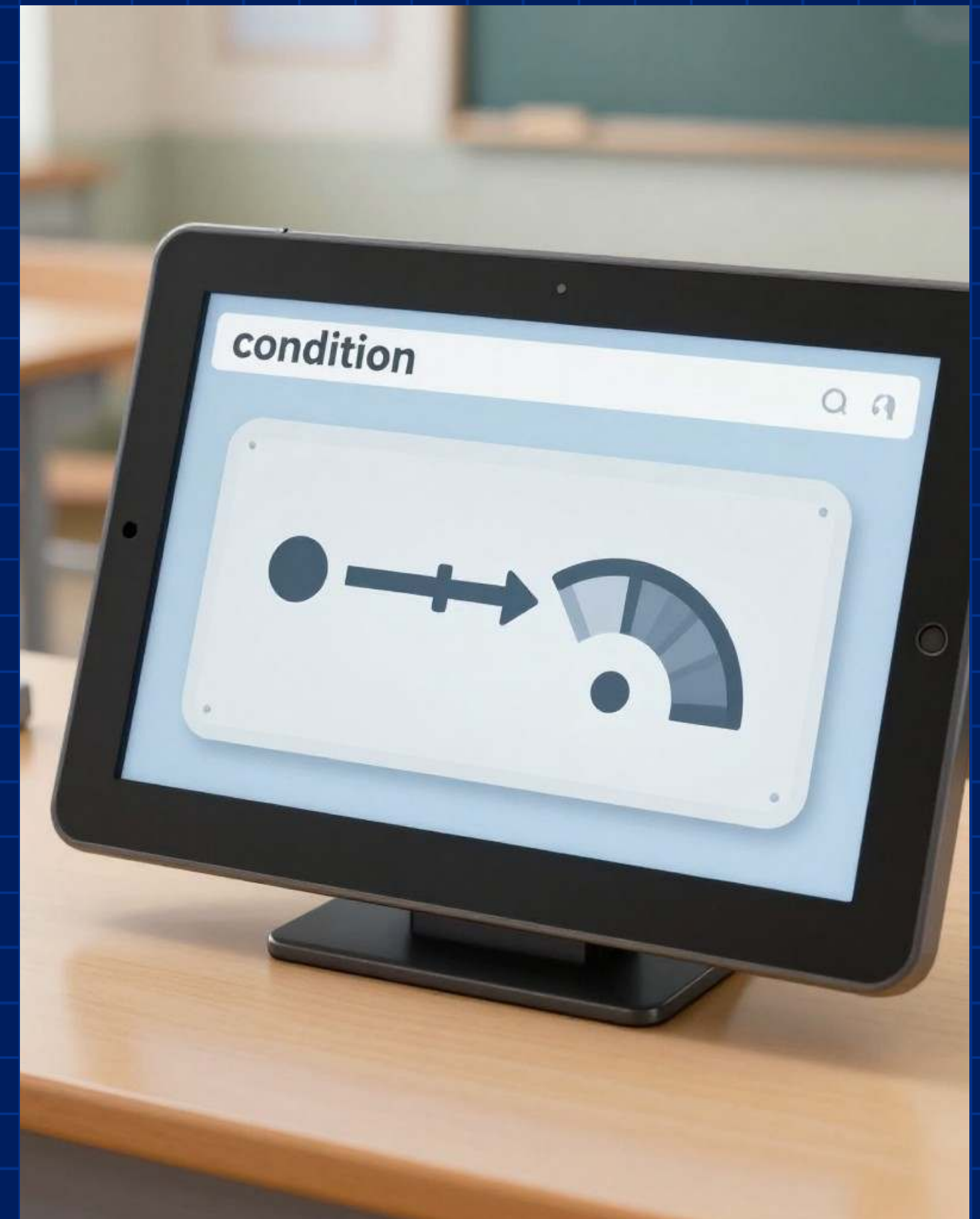
2

Внутри блока — сравнение и выбор ветки

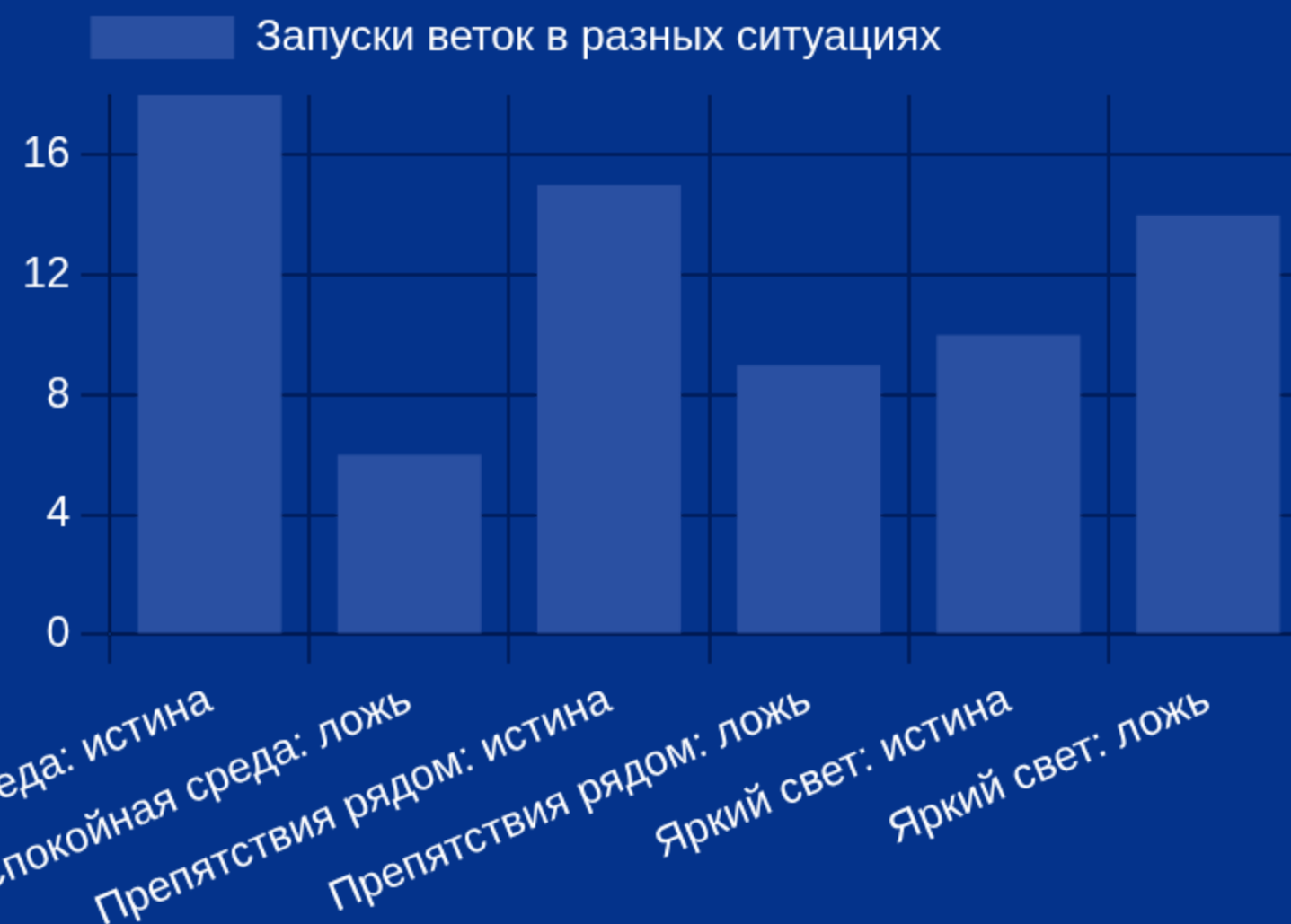
В условном блоке программа сравнивает значение датчика с числом или признаком. Например, это может быть проверка: расстояние меньше порога или кнопка нажата — то есть ответ «да» или «нет».

Две ветки: «если истина» и «если ложь»

Дальше блок имеет два выхода: на истинную ветку и на ложную. Визуальные среды показывают подписи «если — то — иначе», чтобы было понятно, какая команда выполняется при каждом результате проверки.



Как часто выбирается каждая ветка



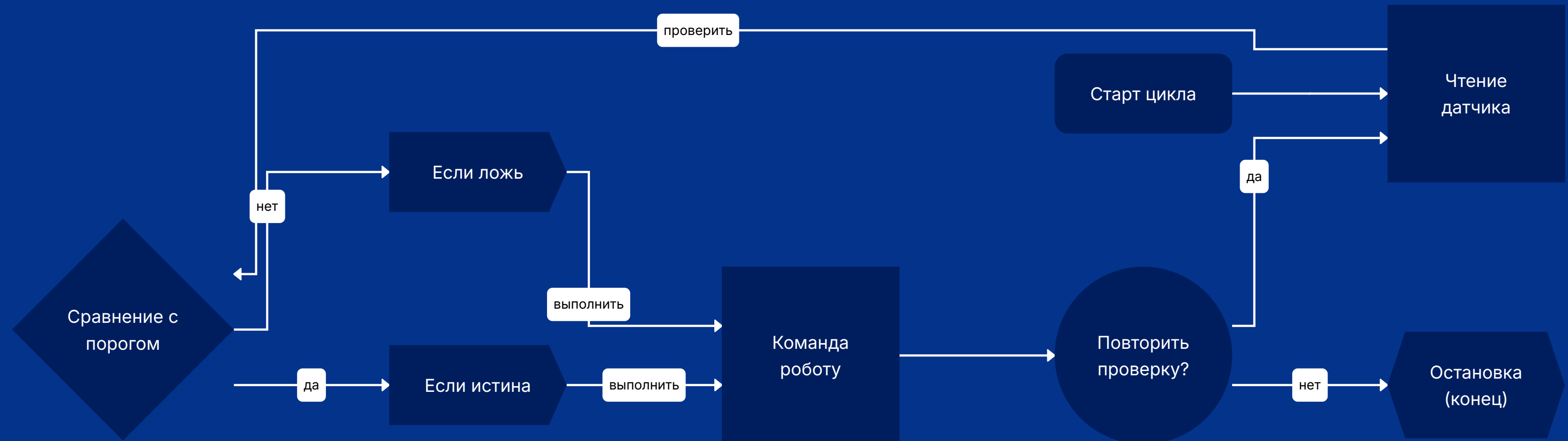
Учебная модель распределения запусков веток (точные числа приведены как пример).

В спокойной обстановке чаще выполняется одна ветка, потому что датчик устойчиво даёт значения по одну сторону порога. При изменении условий картина сдвигается.

Диаграмма показывает, что распределение «истина/ложь» зависит от внешней среды и настроек порога: робот фактически подстраивает поведение под ситуацию.

Алгоритм с повторением и условием

Непрерывный контроль датчика: проверка → команда → возврат в цикл



Зачем условия нужны в роботах

1

Условия делают поведение робота безопаснее. Например, при приближении к краю он может остановиться, а перед стеной – повернуть, чтобы не продолжать движение вслепую.



2

Благодаря условиям робот становится полезнее в реальной обстановке: он реагирует на кнопку и меняет действия при изменении света или других показателей вокруг.



3

Без условий программа состояла бы только из одинаковых команд и не учитывала бы ситуацию. Условие помогает автоматизировать простые задачи так, чтобы робот принимал решения по данным.

4

Робот действует разумно, когда умеет выбирать

Команды-запросы превращают данные датчиков в решение программы. Условие проверяет значение, сравнивает его с порогом и направляет выполнение по нужной ветке. Логические слова «И», «ИЛИ» и «НЕ» помогают строить более точные проверки.



cos

sin

1

tg

ctg