

Алгебра логики в информатике: основы, операции, таблицы истинности

Из простых условий строим логические выражения и проверяем, когда они истинны.

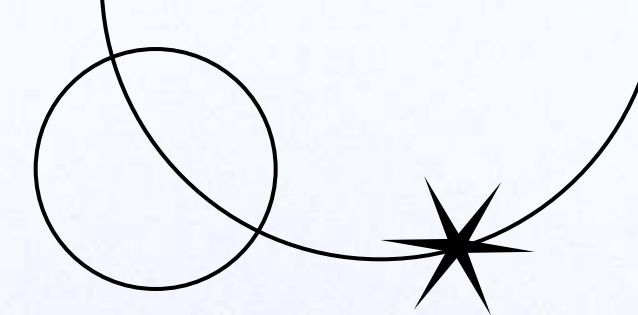


Почему эту тему проходят в 9 классе

Логика учит компьютер принимать решения по условиям: «если», «и», «или», «не». Поэтому в информатике она нужна в проверках программ, поиске и фильтрах данных. Историческая основа — идеи Джорджа Буля (XIX век).



Высказывание как основное понятие логики



1

Высказывание — это предложение, для которого можно однозначно определить, истинно оно или ложно. Например: «7 — простое число» или «Москва — столица России» имеют ясный ответ.

2

Не являются высказываниями вопросы, просьбы, команды и неопределённые фразы вроде «Скоро будет тепло» или «Открой тетрадь». Их истинность заранее не задана.

3

В логике важно, чтобы у утверждения было только два значения: истина или ложь. Такая строгая проверяемость делает высказывания удобными для вычислений и таблиц.

Логические величины: истина и ложь

1

Что означают 1 и 0

В алгебре логики истина обозначается числом 1, а ложь — числом 0. Так информация легко переводится в язык цифровых устройств: у них устойчивые состояния фиксируются как «есть сигнал» и «нет сигнала».

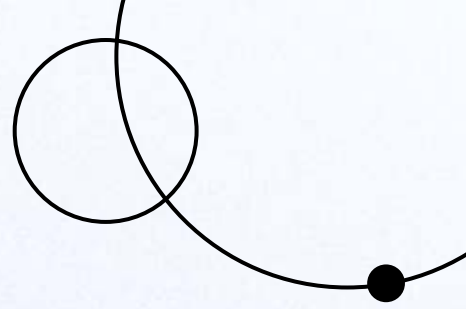
2

Зачем компьютеру две величины

Логические величины применяются для описания простых условий: включено/выключено, да/нет, верно/неверно. На схеме обычно показывают два варианта с подписями «1 = истина» и «0 = ложь».



Основные логические операции и их смысл



Операции И, ИЛИ и НЕ задают правила получения результата из входных значений 0 и 1.

Операция	Когда результат 1	Когда результат 0
И	оба входа истинны	хотя бы один ложный
ИЛИ	истинен хотя бы один	оба входа ложны
НЕ	вход ложно	вход истинно

Каждая операция по-своему «собирает» условия: И требует совместности, ИЛИ — хотя бы одного условия, НЕ — меняет смысл на противоположный.

Таблица истинности операции И

Операция И означает «одновременно выполняются оба условия». Поэтому достаточно одного ложного входа, чтобы результат стал 0.

График показывает строгую зависимость: единственный способ получить 1 — комбинация $A=1$ и $B=1$.

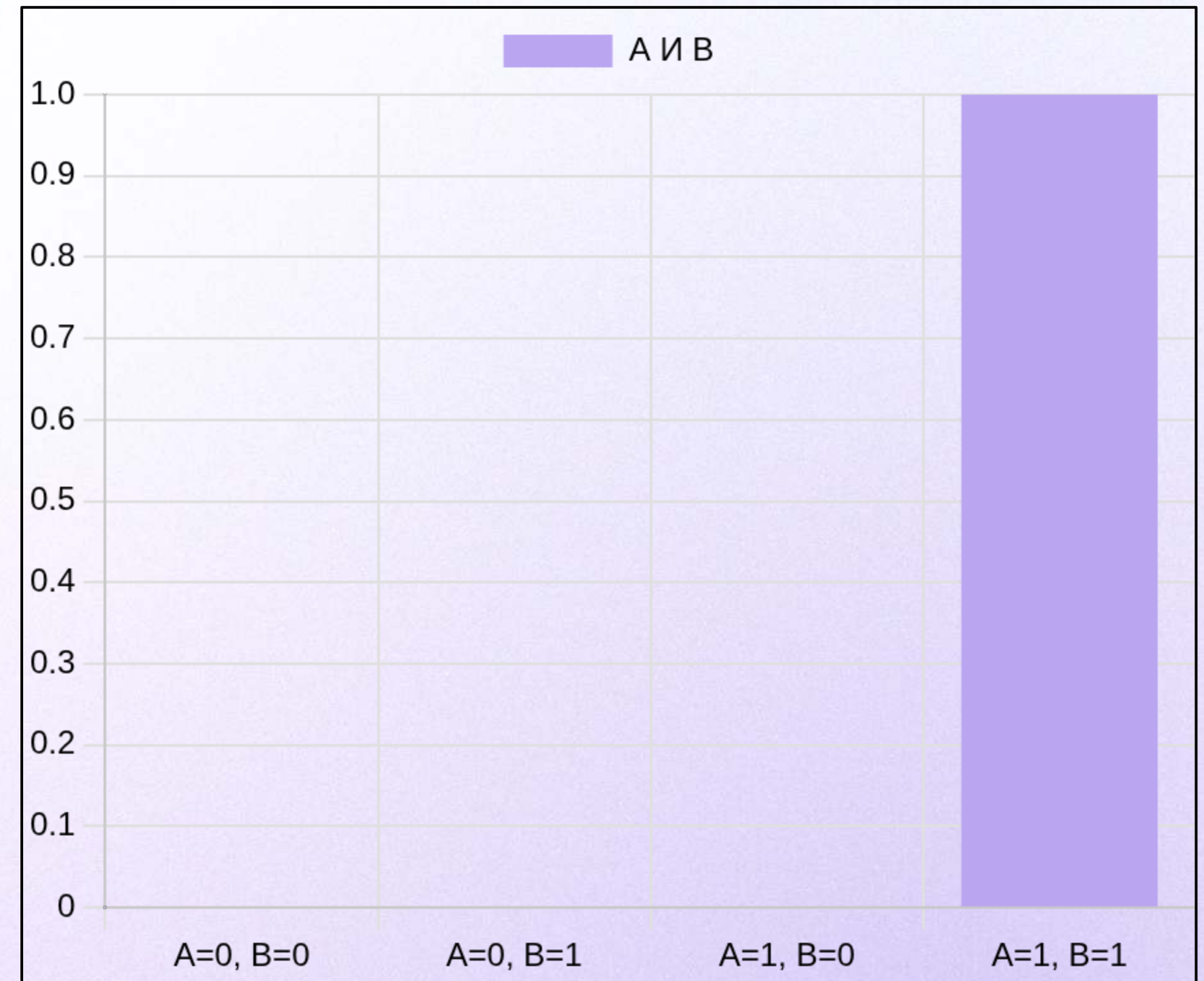


Таблица истинности операции ИЛИ

Операция ИЛИ соответствует выбору: подходит условие A или подходит условие B . Поэтому результат становится 1 уже при одном истинном входе.

По данным видно: 0 получается только в ситуации, когда оба входа равны 0; в остальных случаях выражение истинно.

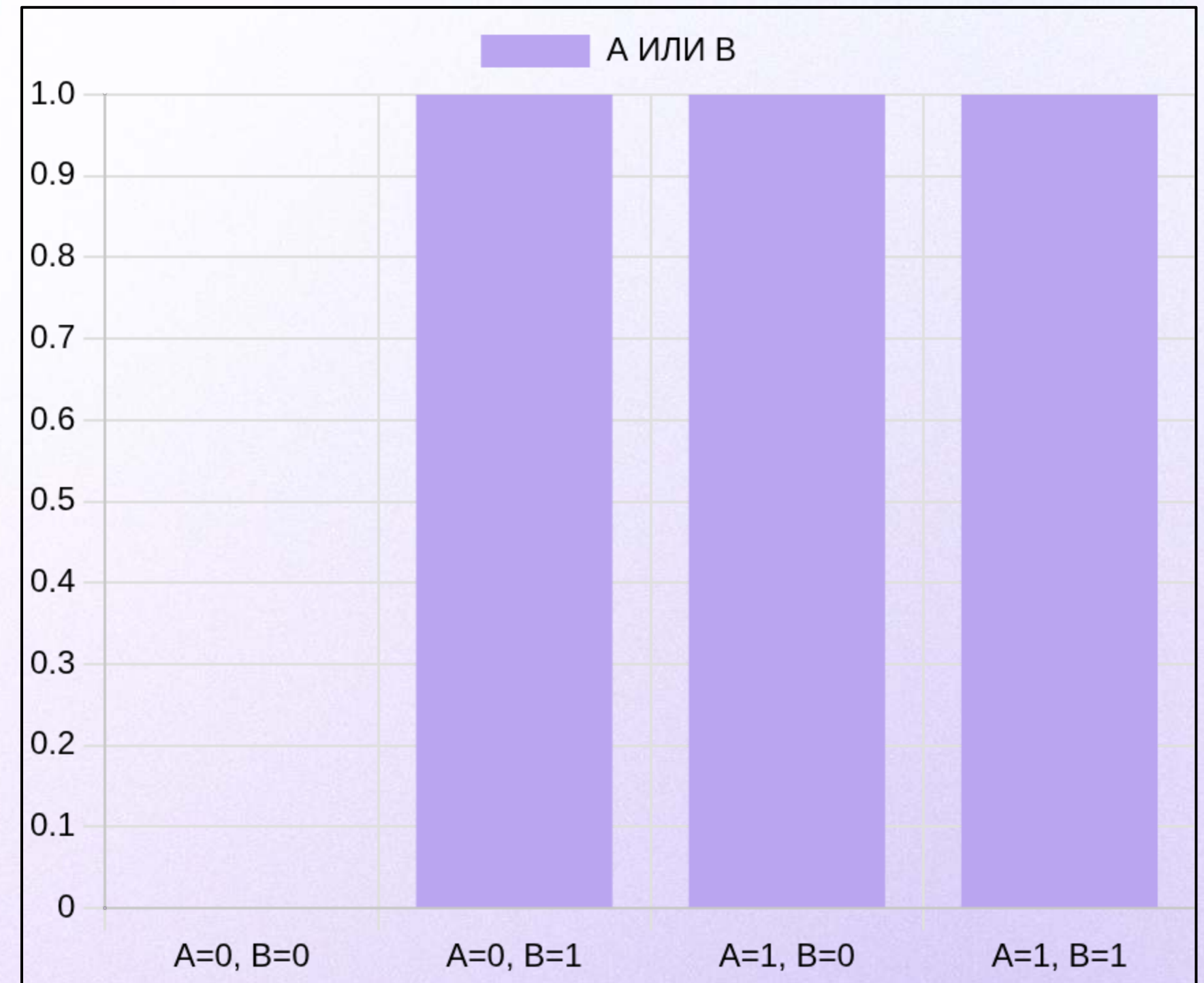
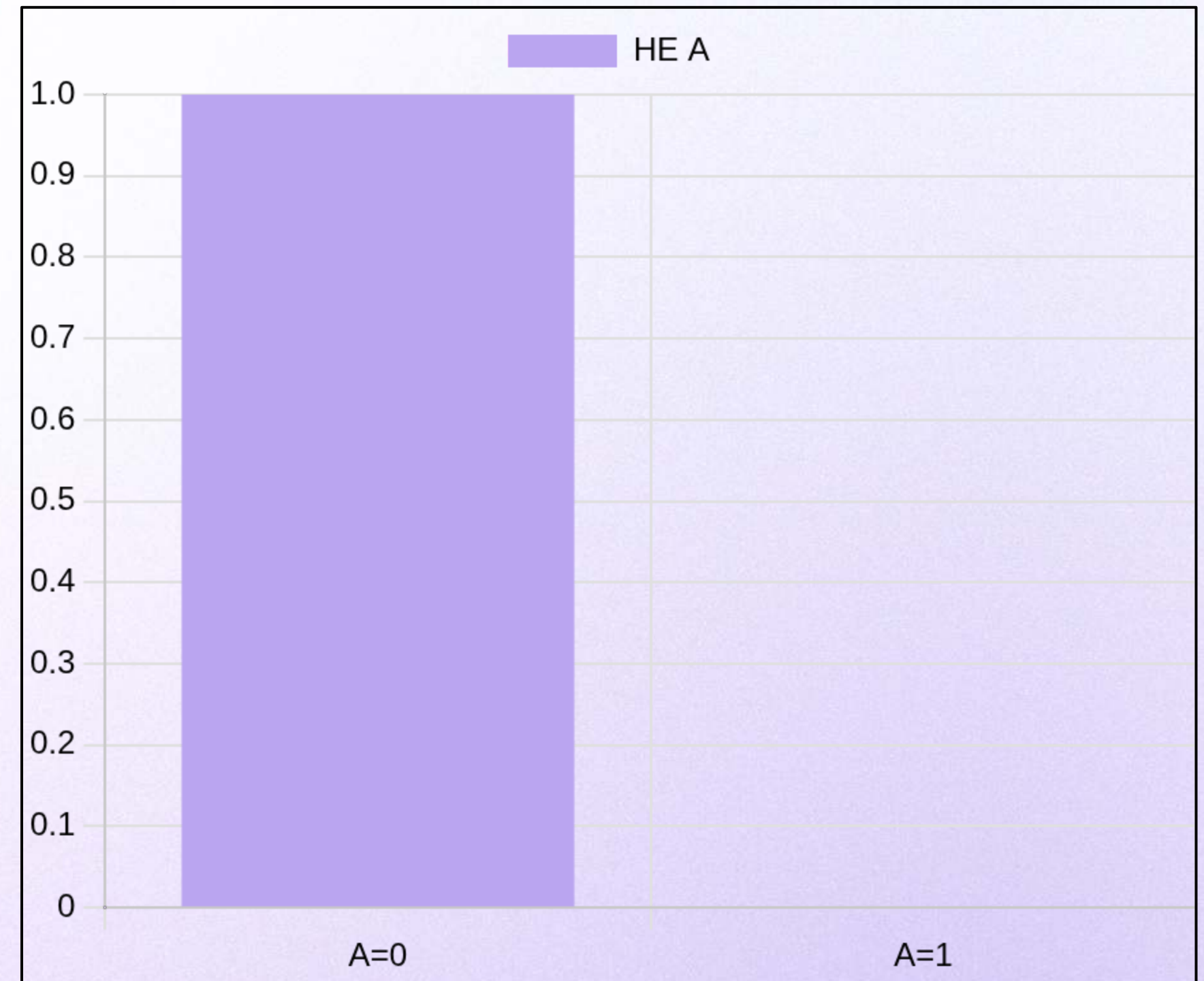


Таблица истинности операции НЕ

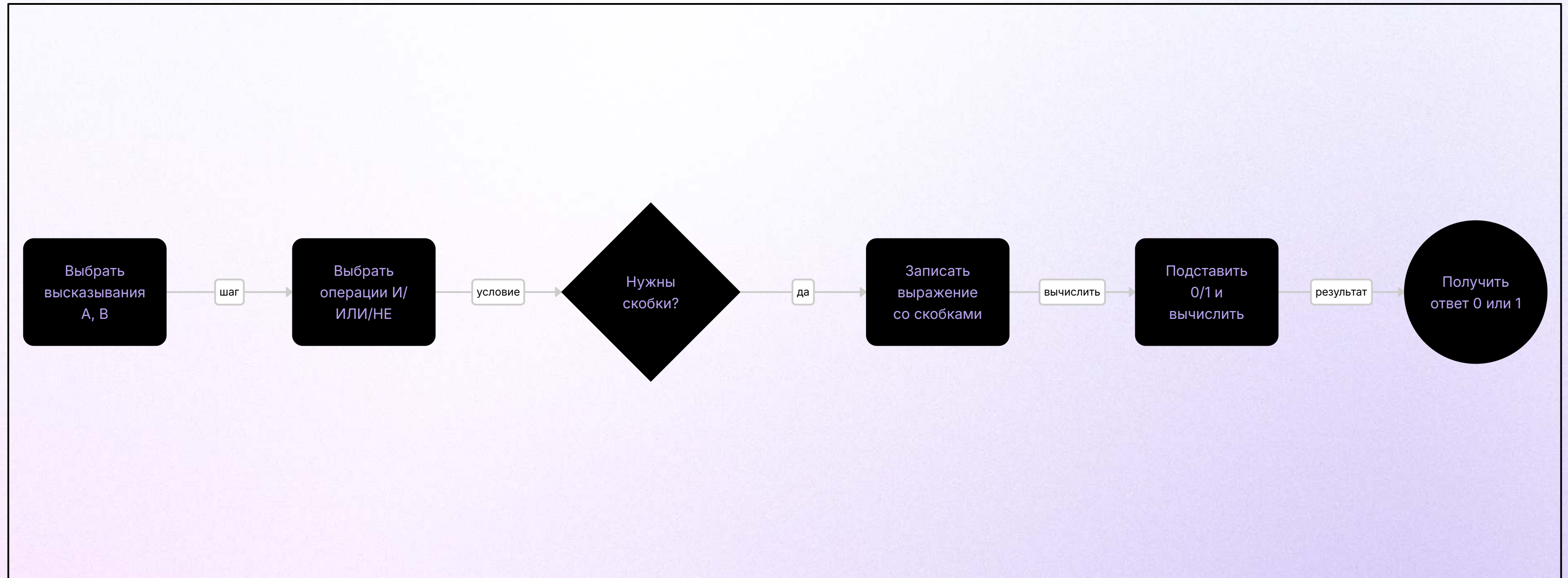
НЕ не «склеивает» условия, а выполняет инверсию: меняет смысл высказывания на противоположный.

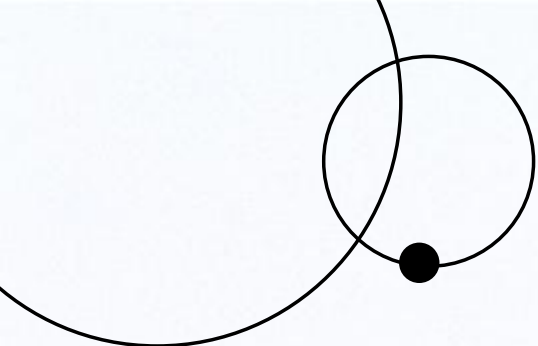
Таблица подчёркивает простую логику отрицания: если вход истинный — на выходе ложь, и наоборот.



Как строится логическое выражение

Пошаговый алгоритм записи и вычисления выражения по значениям А и В





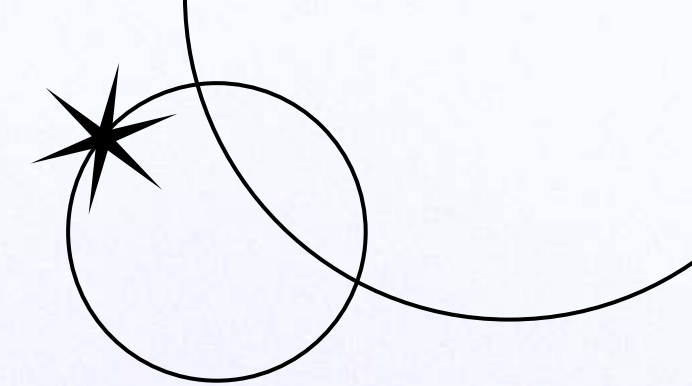
3

Сначала
выполняется
самый «сильный»
тип, затем
действие И, а в
конце — ИЛИ

В выражении три
ключевых операции: НЕ,
И и ИЛИ. Их выполняют
по приоритету (а скобки
задают порядок), чтобы
таблица истинности
была верной.

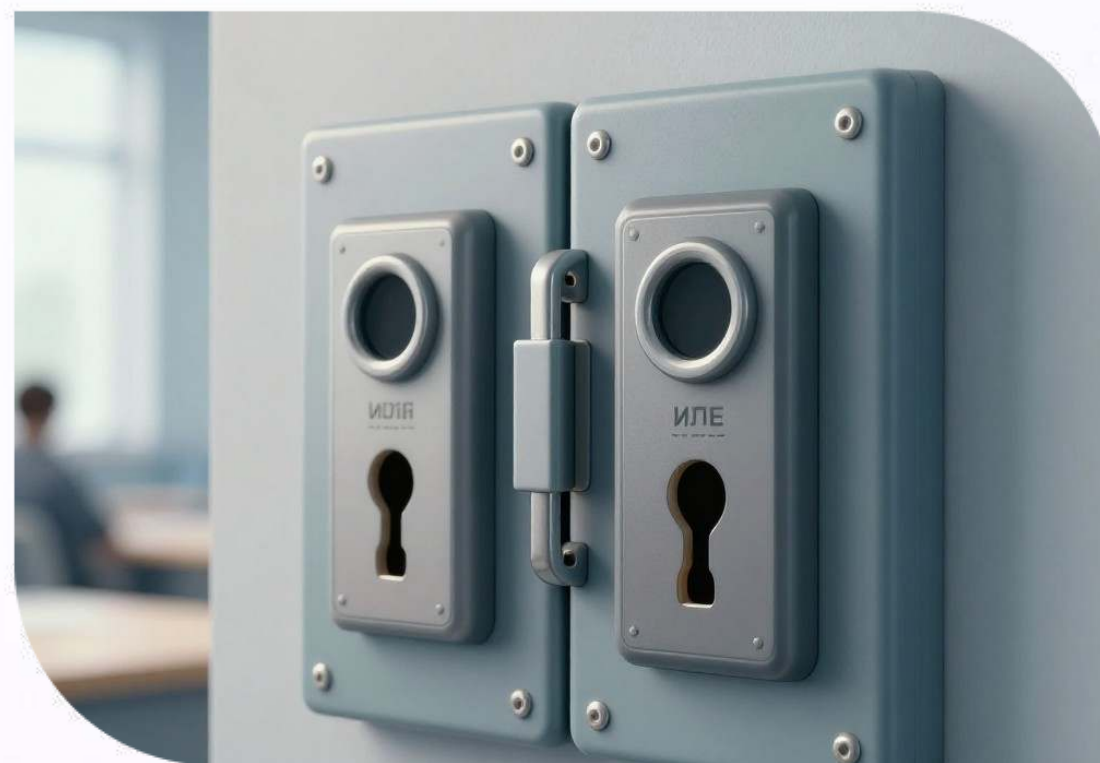
✦ Школьный курс информатики, 9 класс

Как логика становится «железом»: И, ИЛИ, НЕ



Кнопка и лампа: что решает элемент И

Внутри электронных устройств логические правила превращаются в схемы. Элемент «И» передаёт сигнал дальше, только когда оба входа активны. Поэтому его изображают как блок с двумя линиями, которые должны «совпасть» по условиям.



Выбор хотя бы одного: работа элемента ИЛИ

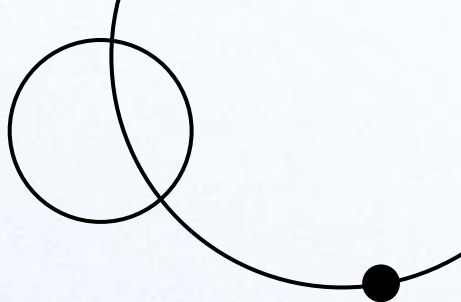
Элемент «ИЛИ» устроен так, что выход включается, если активен хотя бы один вход. Это похоже на правило доступа: нужен билет или паспорт. На схеме хорошо видны два входа, а выход реагирует на любое их «срабатывание».



Отрицание: элемент НЕ как переключатель смысла

Один вход и один выход — так выглядит элемент «НЕ». Он меняет смысл условия на противоположный: было «да» — становится «нет», было «1» — превращается в «0». Именно поэтому его удобно использовать для условий с отрицанием в программах и схемах.

Законы алгебры логики: упрощаем формулы без потери смысла



Закон	Формула	Смысл
Переместительный	$A \sqcap B = B \sqcap A$	Можно менять местами
Сочетательный	$A \sqcap (B \sqcap C) = (A \sqcap B) \sqcap C$	Скобки переставлять
Поглощение	$A \sqcap (A \sqcup B) = A$	Лишнее исчезает
де Моргана	$\neg(A \sqcup B) = (\neg A) \sqcap (\neg B)$	Отрицание переносится

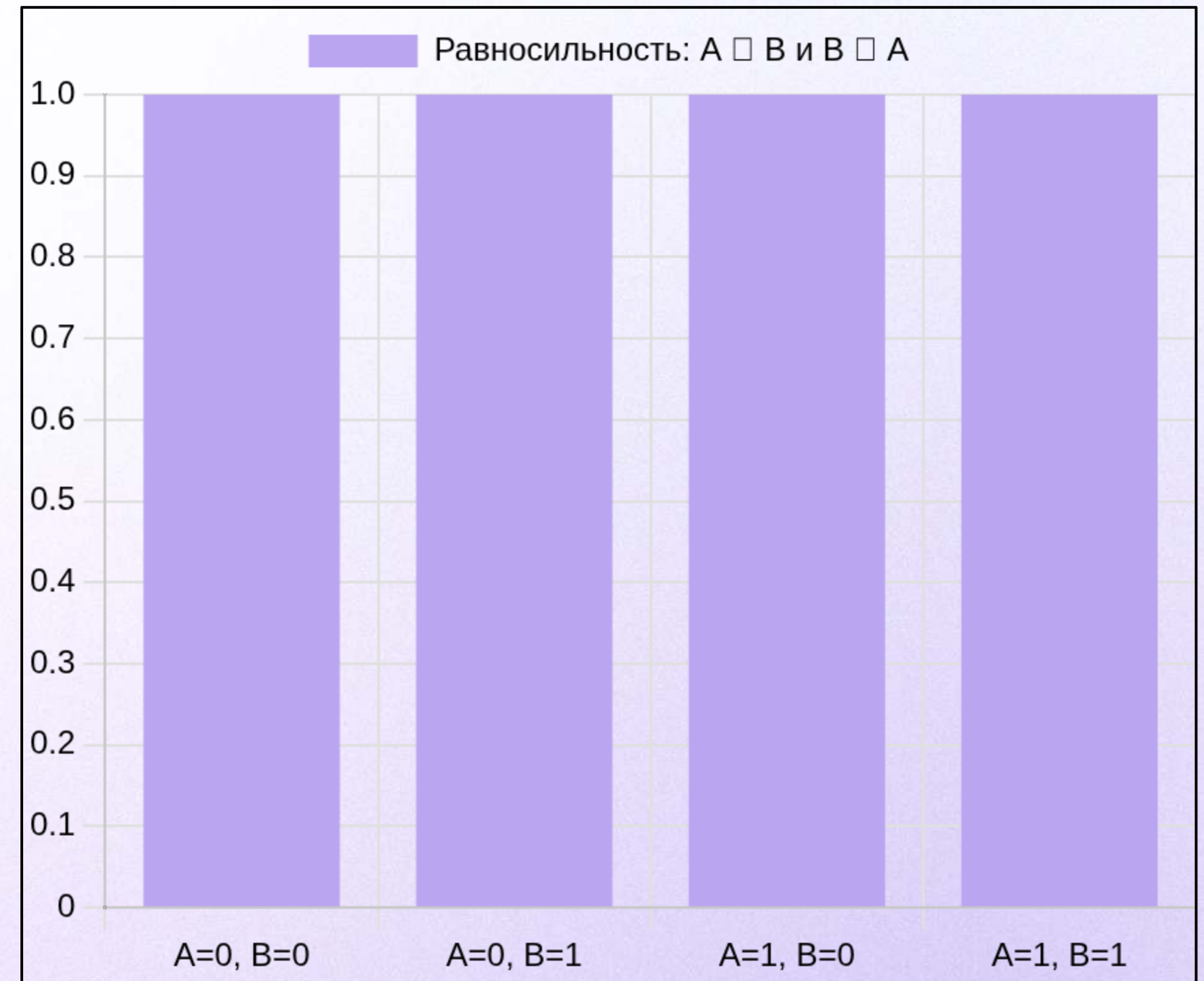
Законы задают, какие преобразования логических выражений можно делать, не меняя итогового значения. Это помогает сократить запись условий, проверить формулы и эффективнее понимать, какие части выражения действительно влияют на результат.

Логические тождества позволяют преобразовывать выражения, чтобы уменьшать сложность условия и делать формулу понятнее, не меняя наборы, на которых она истинна.

Проверка равносильности логических выражений

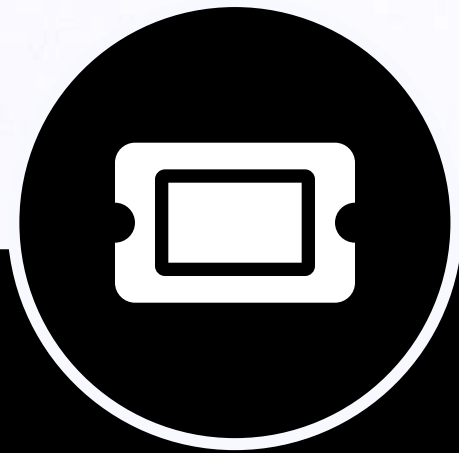
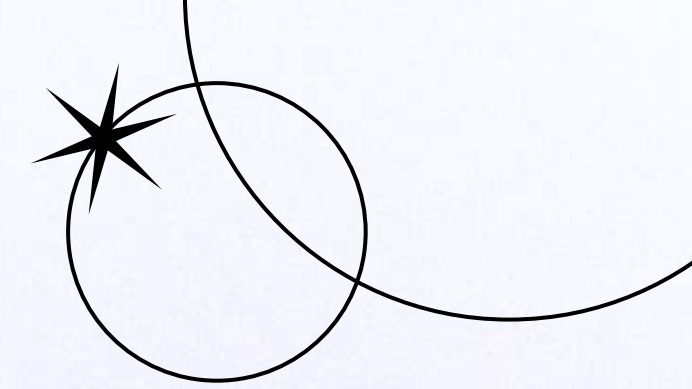
Для обоих выражений результат зависит только от того, есть ли среди A и B хотя бы одна единица, поэтому на всех наборах он совпадает.

Совпадение значений для 00, 01, 10 и 11 подтверждает, что запись отличается, а смысл — одинаковый; это проявление переместительного закона.



★ Сравнение по таблице истинности (примеры для A и B)

Логика вокруг нас: из правил — в условия



Проход по правилам «и/или»

Если есть билет и паспорт, проход разрешён. В логике это соответствует условию, где оба признака должны быть истинными: без одного из них результат меняется.



Отрицание условий в повседневных решениях

Если не идёт дождь, прогулка возможна. Здесь важна операция НЕ: она меняет смысл уже заданного признака, поэтому «дождя нет» превращается в условие для действия.



Выбор вариантов: «либо это, либо то»

Если выполнено домашнее задание или есть уважительная причина, результат проверяется отдельно. Такая запись удобно переводить в формулу ИЛИ и понимать, какой признак достаточно выполнить.



Фильтры и поиск как проверка нескольких признаков

В школьных заданиях и сервисах результаты часто зависят от нескольких условий одновременно: например, класс, тема, дедлайн. Логические операции позволяют объединять признаки в одно решение.

Алгоритм проверки логического условия

Как из признаков получить итог 0 или 1 (для доступа, выбора действия и условий в программах)

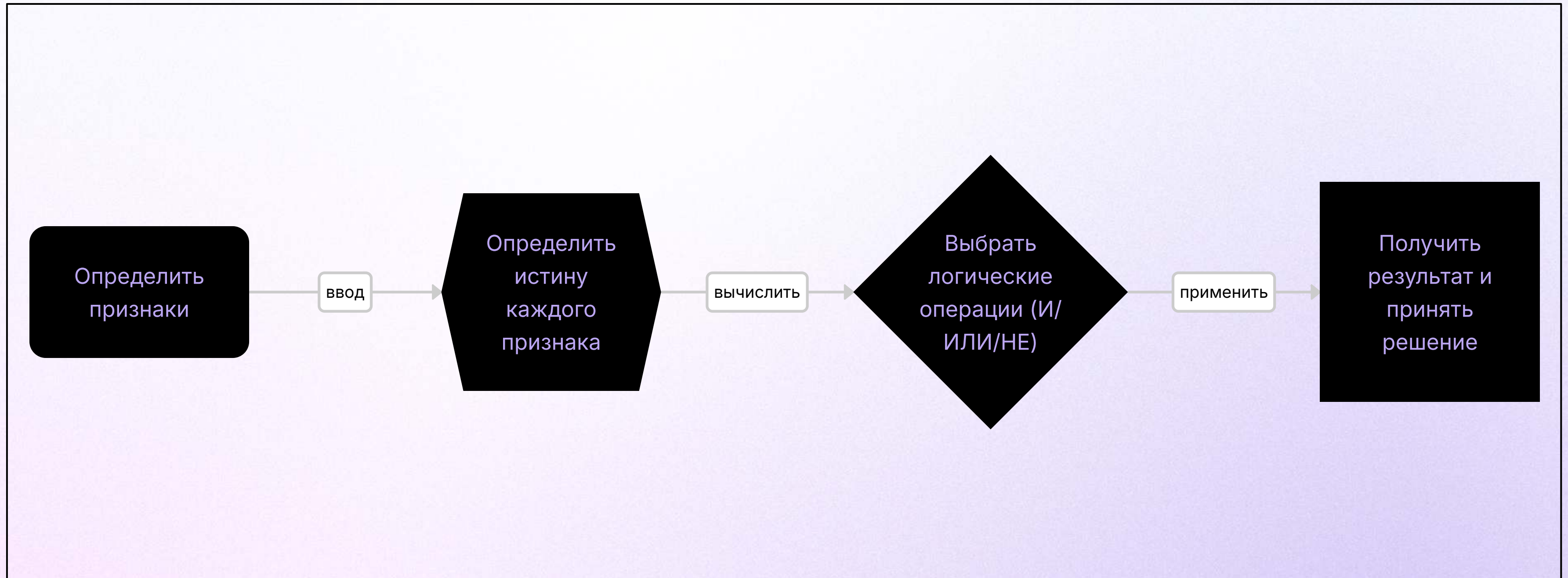
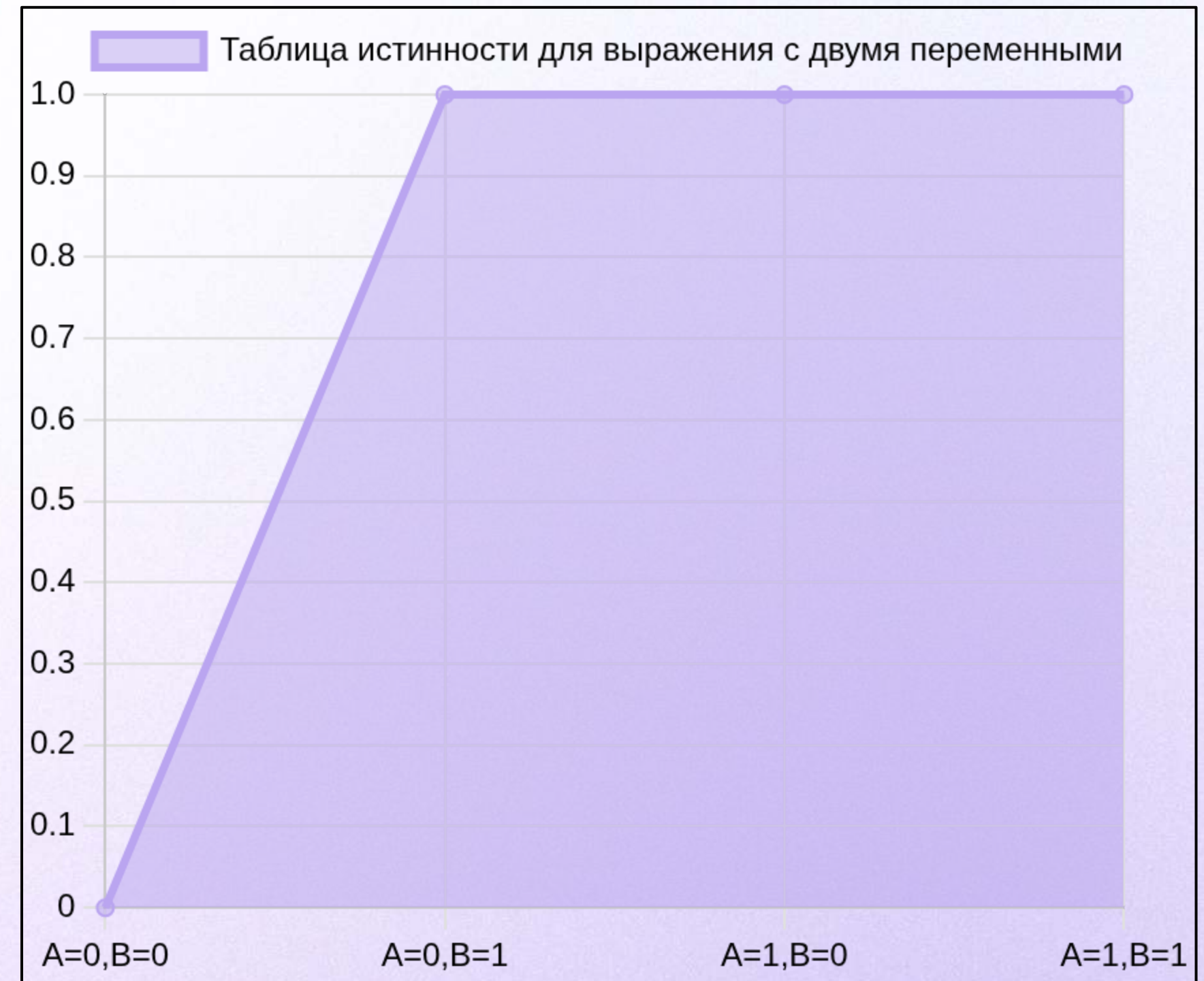


Таблица истинности как способ решения задач

По каждой строке 00, 01, 10, 11 подставляют значения A и B и фиксируют результат выражения: 0 или 1, без догадок и «на глаз».

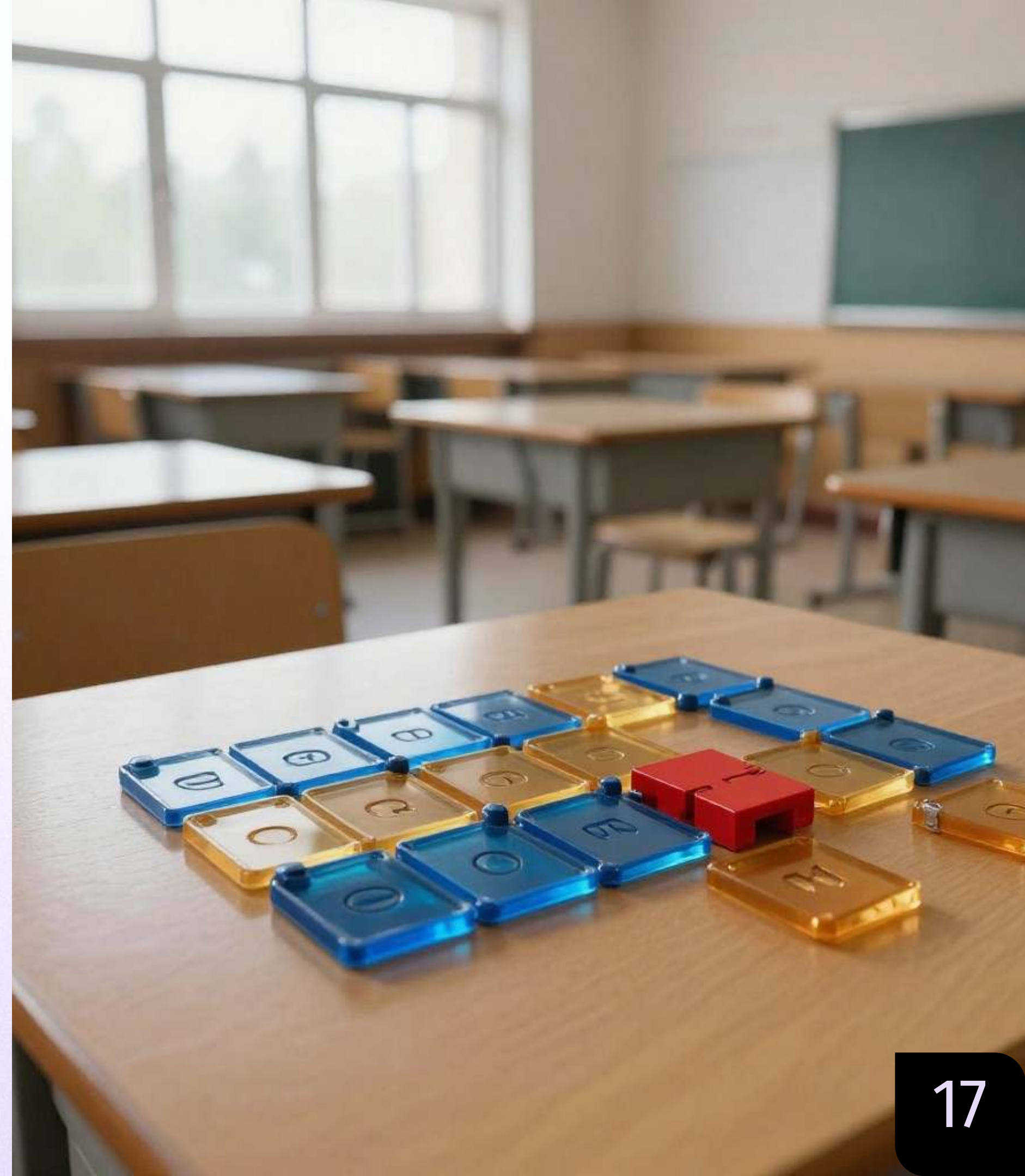
Если в одних строках значения совпадают, а в других нет, это помогает точно понять, при каких условиях формула истинна, а где ложна.



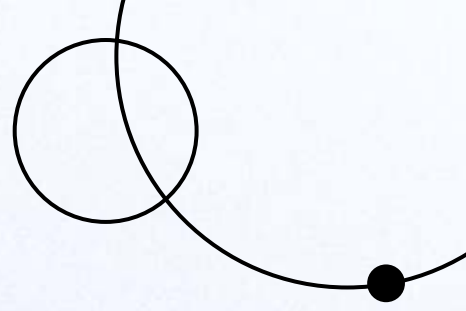
★ Методика из школьного курса: построение таблицы истинности для двух переменных

Типичные ошибки при работе с логическими выражениями

- Часто путают операции И и ИЛИ: например, вместо «нужно оба условия» записывают «достаточно одного». Тогда итог меняется на некоторых наборах переменных, и выражение перестаёт соответствовать задаче.
- Ещё одна причина ошибок — пропуск НЕ, неправильные скобки и игнорирование приоритета операций. Если считать выражение слева направо без правил, можно получить неверный результат, который таблица истинности быстро покажет.



Сколько строк нужно в таблице истинности



Количество строк зависит от числа переменных: для каждой новой переменной наборы удваиваются. Поэтому таблица быстро растёт, и важно заранее оценивать её размер перед вычислениями.

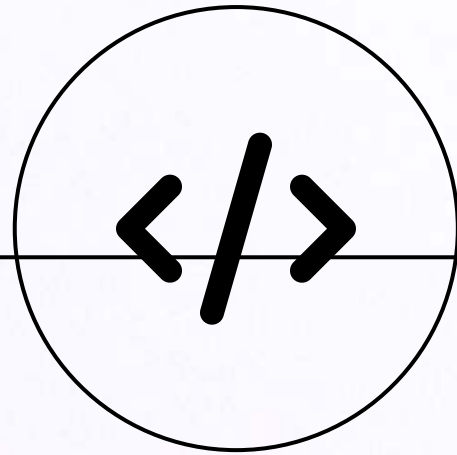
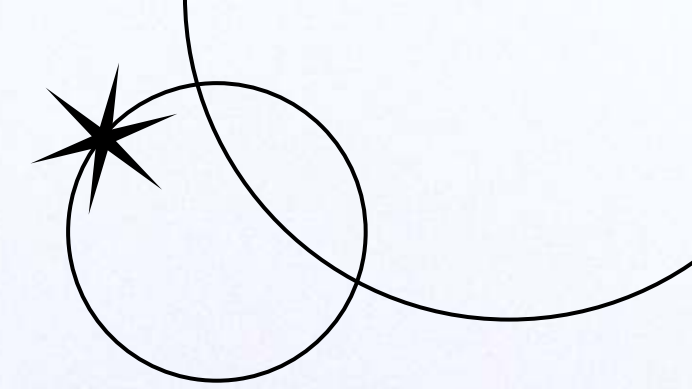
Переменные	Число строк	Пример наборов
1	2	0; 1
2	4	00,01,10,11
3	8	000...111

Чем больше переменных в логическом выражении, тем внимательнее нужно строить таблицу: рост количества строк экспоненциальный, из-за чего возрастает риск ошибки.



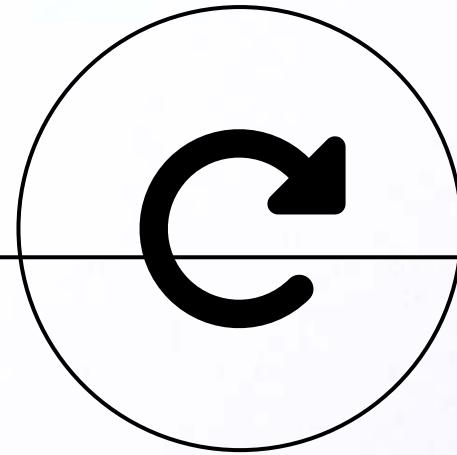
Школьная методика построения таблиц истинности: 2^n строк при n переменных

Логика в программировании: условия, проверки, фильтры



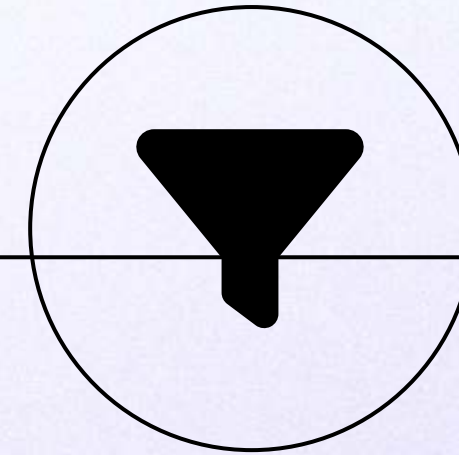
Ветвления: выбор между вариантами

Условия в программах работают по той же логике, что и выражения в алгебре: true/false определяют ветвление выполнения. Это удобно при проверке признаков, например «допустить или отказать».



Циклы с проверкой условий

Во многих алгоритмах выполнение цикла зависит от логического условия. Логические операции позволяют сформулировать, когда цикл продолжается, а когда завершится, без лишних проверок.



Фильтрация данных и поисковые запросы

При фильтрации часто используют комбинации критериев: одни данные подходят, если выполнено условие, и не подходят иначе. Знание И, ИЛИ, НЕ помогает точнее описывать правила отбора.

Ключевые выводы по алгебре логики

Алгебра логики описывает высказывания и правила объединения условий с помощью И, ИЛИ и НЕ. Таблицы истинности проверяют результат для всех наборов, а законы логики помогают упрощать формулы без потери смысла.