

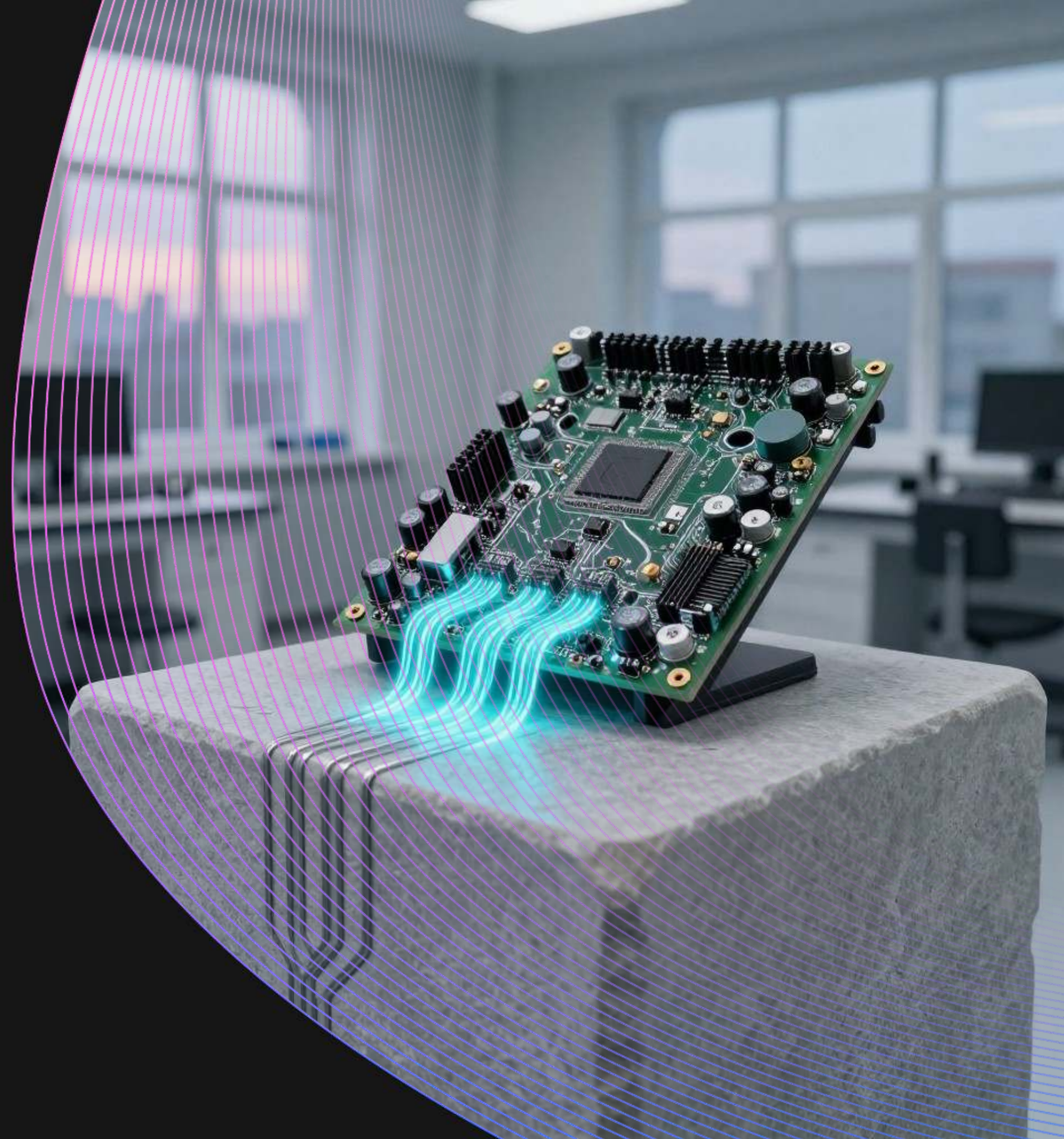
Код и кодирование: как информация превращается в данные

Кодирование переводит смысл в знаки,
чтобы хранить, передавать и
обрабатывать информацию.



Кодирование — основа всей информатики

Любая цифровая техника оперирует битами: 0 и 1. Исторически люди пользовались разными кодами — от письменности до Морзе. В информатике кодирование стало универсальным способом представить текст, числа, графику и звук для устройств связи.



Понятие кода, кодирования и декодирования

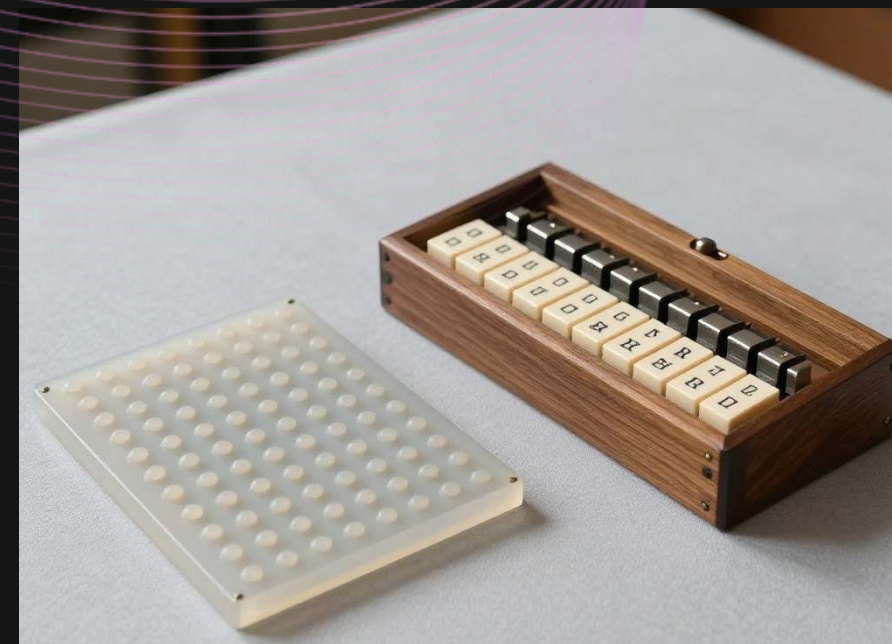
- Код — это набор условных обозначений, по которому мы понимаем, что именно за сообщение передаётся. Без кода одни и те же символы не имеют однозначного смысла.
- Кодирование — преобразование исходной информации в формальный вид, пригодный для хранения или передачи. Компьютеру важно, чтобы данные были записаны по правилам конкретного кода.
- Декодирование — обратный процесс: из закодированных данных восстанавливается исходное сообщение. Один объект можно представить по-разному — например, числом, текстом, графикой или звуком.

Исторические и наглядные примеры кодов



Морзе: сообщение вместо речи

Азбука Морзе превращала текст в последовательности точек и тире. Радиосвязь получала «рисунок времени», который можно было передавать на расстояние и принимать по тем же правилам кода. Важно, что смысл зависел от договорённости о знаках.



Шрифт Брайля и шифр Цезаря: код под задачу

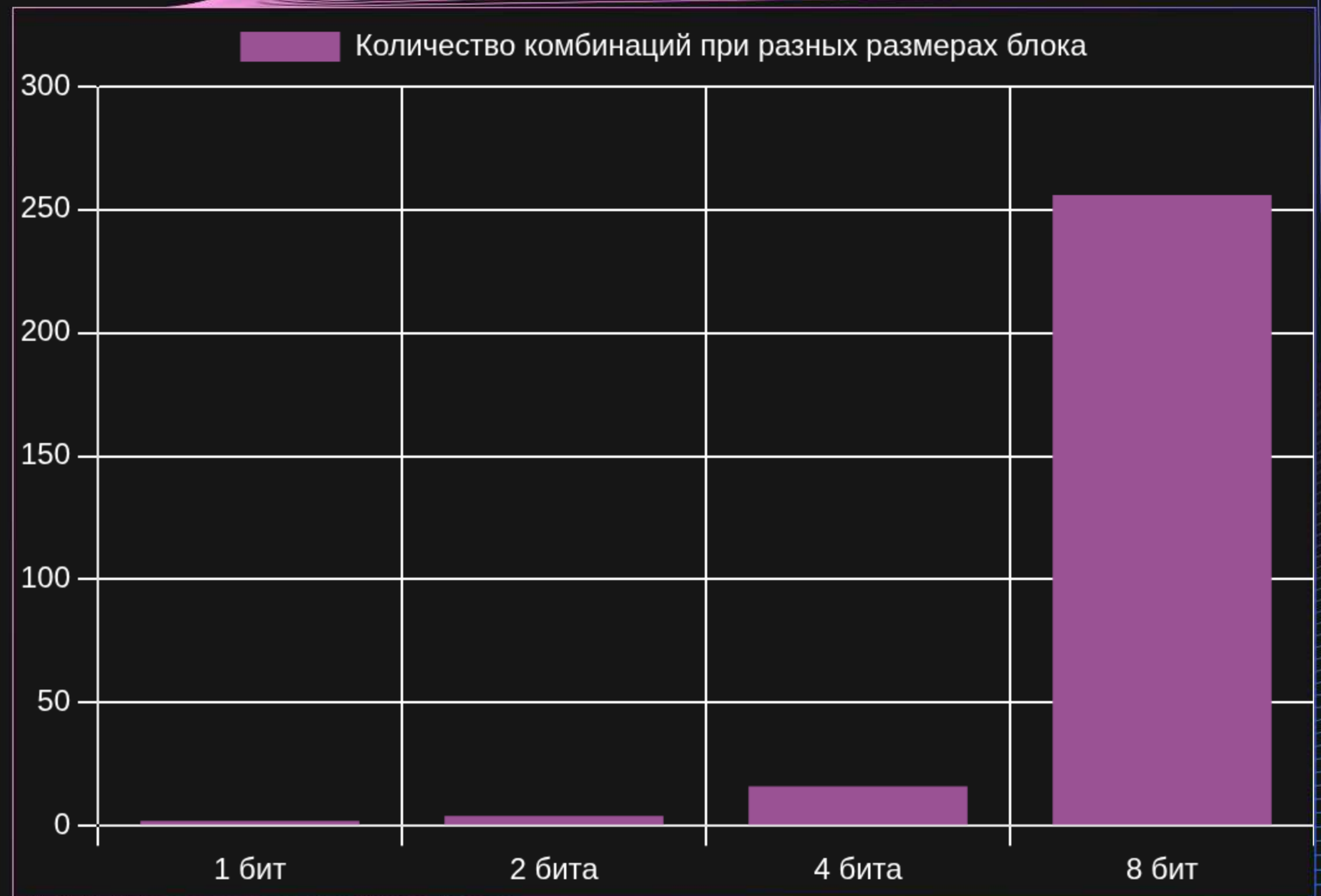
Шрифт Брайля использует рельефные точки, которые читаются на ощупь, и тем самым даёт отдельным людям доступ к письменной информации. А шифр Цезаря меняет буквы по правилу сдвига: код становится инструментом скрытности или эксперимента с правилами замены.

Единицы информации и рост числа комбинаций

Чем больше бит в группе, тем точнее можно различать варианты: увеличивается «алфавит» возможных состояний.

Диаграмма показывает экспоненциальный рост вариантов: даже небольшое увеличение числа бит резко расширяет возможности хранения и кодирования.

Собственная сводка: формула числа комбинаций для n бит — 2^n (теоретическое соотношение, учебные материалы по информатике).



Основные виды кодирования информации

Вид	Пример	Идея кода
Текст	Unicode	Каждый символ получает числовой код
Числа	двоичное представление	Значение записывают в битах
Графика	пиксельные данные	Цвет и позицию раскладывают на параметры
Звук	отсчёты амплитуды	Сигнал измеряют и записывают дискретно

— Сравним, как по-разному представляют текст, числа, изображения и звук, чтобы ими было удобно управлять в цифровых системах.

Выбор кода определяется задачей: прочесть знак, выполнить вычисление, передать изображение или уменьшить объём данных при сохранении нужного качества.

Обобщение по школьному курсу информатики: базовые принципы представления данных в ЭВМ.

Кодирование чисел в компьютере

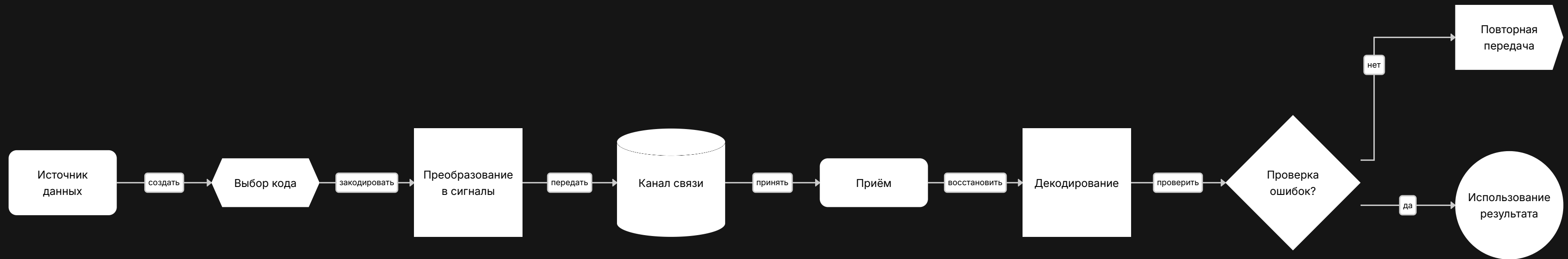
Целые числа в компьютере обычно записываются в двоичной системе счисления. Поскольку там только 0 и 1, любое значение представляют конечной последовательностью бит.

— Чтобы различать положительные и отрицательные числа, используют специальные схемы, например знаковый разряд и варианты представления вроде прямого, обратного и дополнительного кода. Правило выбирает формат.

— Десятичные дроби требуют особого подхода: используется фиксированная или плавающая точка. Точность вычислений зависит от разрядности и от того, как именно кодируют дробную часть.

Путь информации от источника к получателю

Цепочка цифровой связи: данные кодируют, передают сигналом, затем декодируют и проверяют корректность.



Кодирование текста и таблицы символов

А Таблица символов

Компьютер хранит текст не «буквами», а кодами: каждому символу соответствует число. Это позволяет одинаково обрабатывать знаки в разных программах и устройствах.

ASCII как ранний стандарт

Ранние кодировки, например ASCII, поддерживали ограниченный набор символов. Поэтому для разных языков часто не хватало кодов, и часть текста могла отображаться неверно.

Unicode и охват письменностей

Unicode задуман как универсальный стандарт: в нём содержатся коды для большинства письменностей мира. Благодаря этому на разных платформах буквы и знаки отображаются согласованно.

Эмодзи и специальные символы

Unicode позволяет кодировать не только буквы и цифры, но и знаки препинания, математические символы и эмодзи. Поэтому один и тот же текст выглядит одинаково на разных устройствах при корректной настройке.

Почему один байт позволяет кодировать до 256 символов

— Для n бит число вариантов равно 2^n : каждый дополнительный бит удваивает число возможных состояний. Это и делает байт универсальной «единицей охвата».

Значит, для алфавитов с буквами, цифрами и служебными знаками нужен достаточно большой набор кодов, иначе символы начинают не помещаться или путаются.

Теоретическая зависимость: число комбинаций для n бит — 2^n ; байт содержит 8 бит (общепринято в учебных курсах информатики).



Кодирование изображений: пиксели и цветовые модели

- Изображение на экране складывается из пикселей: каждый элемент хранит собственный цветовой код. Компьютер не «рисует красками», а подбирает набор значений, которые вместе формируют картинку для вашего глаза.
- Чаще всего используют модель RGB: цвет задают три канала — красный, зелёный и синий. Чем выше разрешение и глубина цвета, тем больше данных требуется для точной передачи, поэтому растёт и размер файла.



Кодирование звука в цифровой форме

Дискретизация

Сигнал измеряют в отдельные моменты времени: получается последовательность отсчётов вместо непрерывной волны. Чем выше частота дискретизации, тем плотнее «сеточка» измерений и тем точнее воспроизведение.

Квантование

Каждый отсчёт округляют до значения из конечного набора. Разрядность определяет, насколько мелко делится диапазон амплитуд, а значит — величину возможной ошибки квантования.

Сохранение в формате

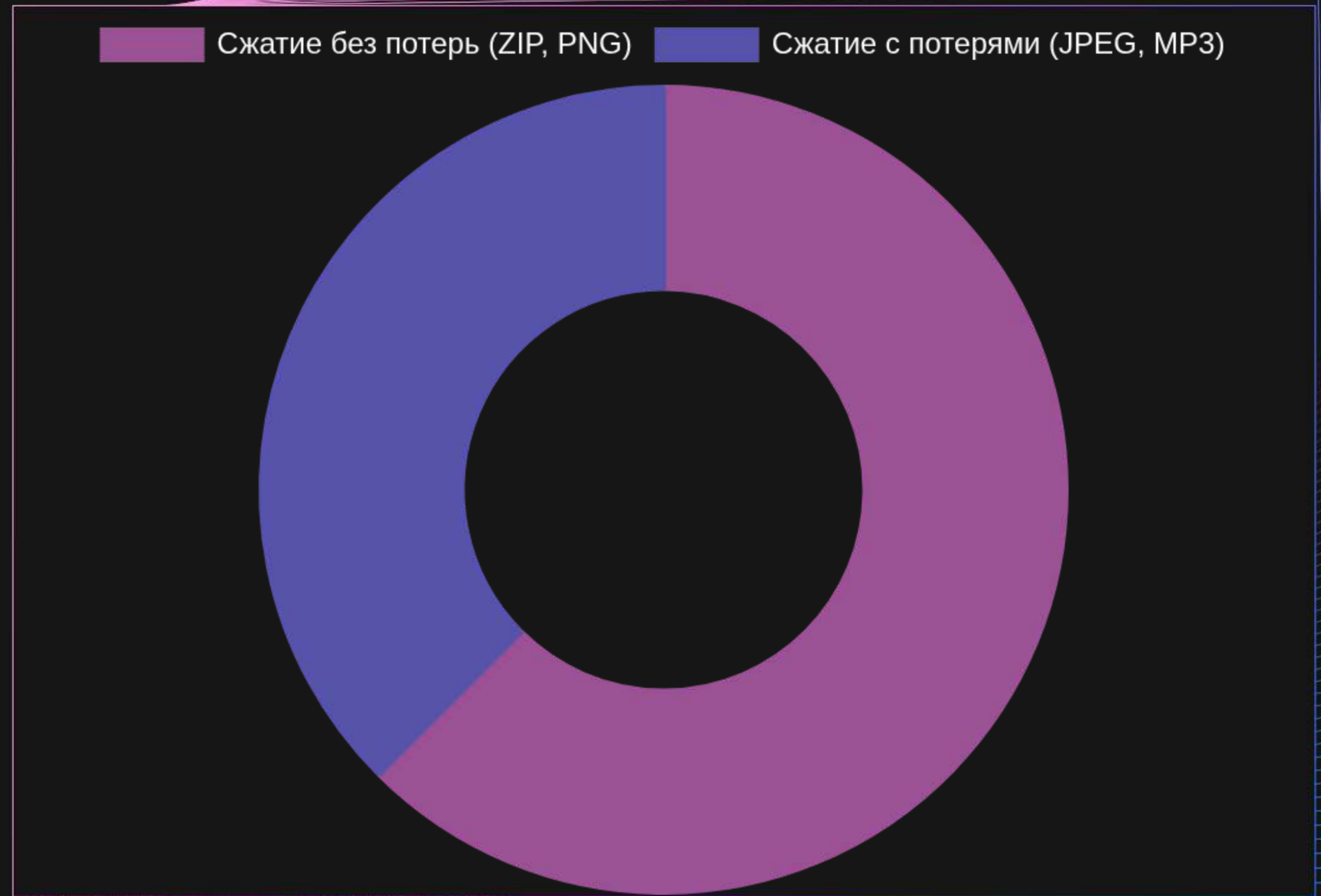
При записи звука формируется файл. WAV, как правило, хранит данные почти без потерь, а MP3 использует сжатие: часть информации убирается, чтобы уменьшить размер и сделать воспроизведение эффективнее.

Сжатие данных: без потерь и с потерями

Сжатие с потерями особенно эффективно там, где человеческое восприятие не замечает мелких искажений. Сжатие без потерь пригодно для файлов, где ошибка даже в одном бите недопустима.

Диаграмма отражает компромисс: чем сильнее сжатие, тем выше риск потери точности. Поэтому выбор формата определяют назначение файла и требования к качеству.

Обобщение принципов сжатия; примеры форматов: ZIP, PNG, JPEG, MP3 (общепринятая классификация).



Коды в повседневной жизни школьника

Код	Что кодируется	Где применяется	Преимущества	Риски
QR-код	ссылка или данные	оплата, обучение, вход	быстрое открытие информации	ошибки считывания, подмена QR
Штрихкод	идентификатор товара	магазины и склады	быстрый учёт и поиск	неполное считывание, несоответствие товара
PIN-код	секретный набор цифр	банковские операции	подтверждение личности	утечка PIN, угадывание при слабой защите
Пароль	секрет для входа	сервисы и аккаунты	контроль доступа	фишинг и повторное использование паролей
GPS-координаты	местоположение	навигация и трекинг	точный поиск точки	ошибки маршрута, приватность

QR-коды, штрихкоды, PIN и пароли, а также GPS-координаты помогают быстрее находить информацию и подтверждать действия, но требуют аккуратности при считывании и защиты.

Коды ускоряют действия за счёт однозначной цифровой записи, но безопасность и корректность результата зависят от качества считывания и уровня защиты от подделок и утечек.

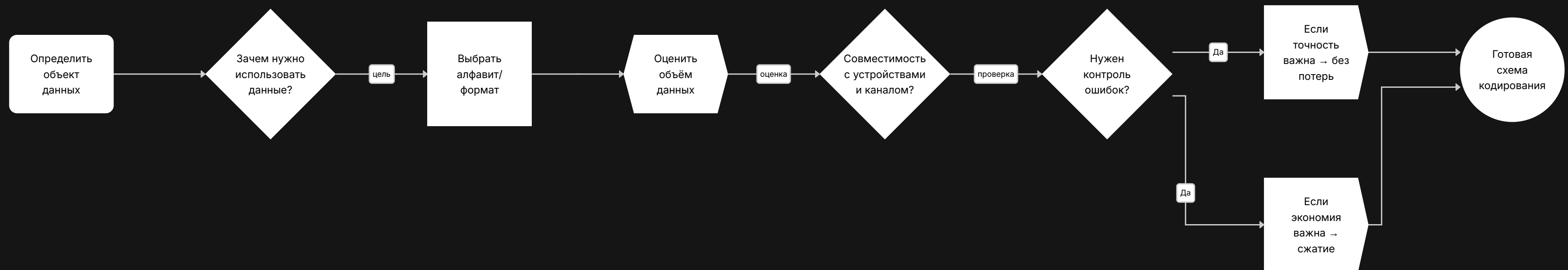
Обобщение типичных сценариев применения кодов в повседневных цифровых сервисах.

Ошибки передачи данных и контроль целостности

- При передаче данных помехи могут менять отдельные биты: вместо нуля приходит единица или наоборот. Тогда получатель получает другой набор значений и должен понять, что данные искажены.
- Для более сложных ситуаций применяются коды исправления ошибок. Они помогают восстановить часть повреждений, не запрашивая повтор всей передачи, что особенно важно при нестабильной связи.
- Чтобы обнаружить ошибки, используют биты чётности и контрольные суммы: они позволяют сравнить «подпись» сообщения с ожидаемой. Несовпадение сигнализирует, что при передаче что-то пошло не так.
- В сетях и памяти ценится не только факт отправки, но и гарантированная целостность. Поэтому после получения данных выполняют проверку и, если нужно, восстановление до использования результата.

Как выбирают способ кодирования

Логика выбора кодирования в зависимости от цели и требований к данным.



Коды в цифровых технологиях вокруг нас



Чтение вместо набора: мир через QR

Представьте, что вместо длинного адреса вы показываете телефоном QR-код. Магазин, школа или сервис мгновенно «узнают», что вам нужно. Такой код хранит компактную запись данных и ускоряет вход, оплату и получение информации.



Товарный штрих-код и «невидимая» связь с базой

Штрихкод на упаковке выглядит как простые линии, но за ними скрывается цифровой идентификатор. Сканер считывает код, система находит товар в базе и мгновенно обновляет учёт. Это пример точной передачи данных без участия человека.

Где кодирование даёт наибольший эффект

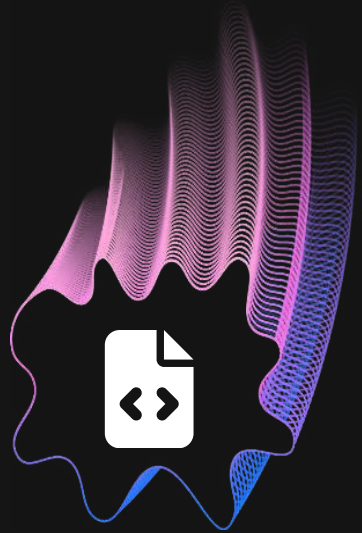
Кодирование особенно эффективно, когда данные нужно передавать быстро и однозначно. Тогда форматы, таблицы символов и проверка ошибок снижают количество повторов и недопонимание между системами.

Наивысшие значения приходится на сеть и файлы: там любая лишняя двоичная запись увеличивает время и стоимость передачи. Поэтому ценят компактные представления и устойчивость к искажениям.

Сводная качественная оценка областей применения (без привязки к конкретной измерительной статистике).



Практическое значение темы для десятиклассника



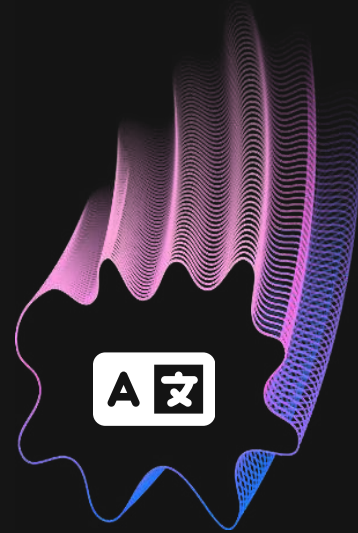
Файлы и их «вес»

Понимание кодировок помогает оценивать размер файлов и объяснять, почему одно изображение «тяжелее» другого. Вы лучше будете замечать, где растут данные и как это связано с битами.



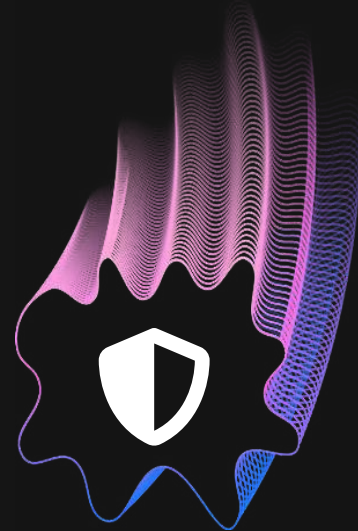
Интернет и передача данных

Вы начнёте различать, что именно передаётся по сети: текст, изображения или звук, и почему возможны задержки или искажения. Тема помогает осознанно читать сообщения об ошибках.



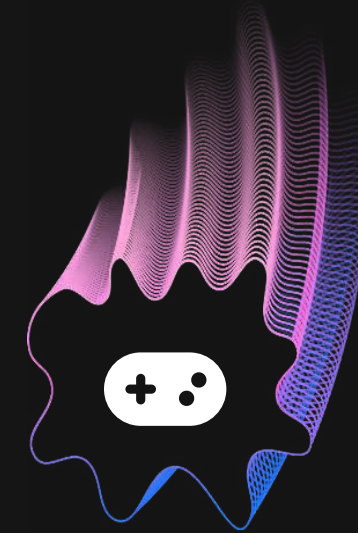
Текст и «кракозябры»

Когда неверно выбрана кодировка, символы отображаются как случайные знаки. Знание таблиц символов и форматов текста помогает понимать причины ошибок и выбирать правильные настройки.



Надёжные пароли

Биты и представления данных тесно связаны с безопасностью: сильный пароль труднее перебрать. Вы сможете лучше оценить риски слабых паролей и повторного использования.



Игры и сетевые сообщения

В играх и онлайн-сервисах кодирование влияет на скорость обмена данными и реакцию интерфейса. Понимание двоичных представлений делает поведение сервисов более предсказуемым.

Кодирование как универсальный язык цифрового мира

Кодирование переводит разнообразную информацию в формальные знаки, понятные человеку и машине. Биты, таблицы символов, пиксели, звуковые отсчёты и контроль ошибок образуют единую систему работы с данными.