

НЕЙРОСЕТИ: УЧИТЬСЯ, РАБОТАТЬ, ЗАРАБАТЫВАТЬ

Практическое руководство по ИИ



ОЛЕГ ПАНКРАТОВ

Олег Панкратов

**Нейросети: учиться,
работать, зарабатывать.
Практическое руководство по ИИ**

«Автор»

2026

Панкратов О. М.

Нейросети: учиться, работать, зарабатывать. Практическое руководство по ИИ / О. М. Панкратов — «Автор», 2026

Нейросети уже пишут тексты, рисуют картинки, сочиняют музыку и программируют. Осталось понять, как из этого извлечь пользу лично вам — без технического образования и без иллюзий. Эта книга — практическое руководство по искусственному интеллекту в трёх его главных применениях: учиться, работать, зарабатывать. Внутри: как устроены языковые модели и почему они уверенно ошибаются; как ставить задачи, чтобы получать результат в десять раз лучше; ИИ-репетитор для вас и честные правила для детей-школьников; работа с текстами, документами, кодом, картинками, видео и звуком; реальные способы заработка с цифрами российского рынка. Отдельная глава — безопасность: как защитить семью от мошенников с клонированным голосом. Каждая глава заканчивается мини-проектом, а в приложении — библиотека готовых промптов. Все инструменты проверены на доступность из России.

© Панкратов О. М., 2026

© Автор, 2026

Содержание

Часть I. Фундамент: как устроен и с чем работает ИИ	6
Начнём с эксперимента	6
Понять	7
Повар и рецепт: чем ИИ отличается от обычной программы	7
Семьдесят лет взлётов и заморозков: как ИИ вырос	7
Матрёшка: машинное обучение, нейросети, глубокое обучение	8
Четыре работы, которые умеет ИИ	9
Данные: топливо ИИ	9
Три слепых пятна: где ИИ ошибается по системе	10
Генеративный ИИ: почему все заговорили именно сейчас	10
Попробовать	12
Применить	13
Мини-проект главы	14
Глава 2. Что такое языковая модель	15
Начнём с эксперимента	15
Понять	16
Угадайка, которая съела библиотеку	16
Токены: машина читает не буквы и не слова	16
Внимание: как модель держит нить разговора	17
Параметры: ручки настройки, которых 175 миллиардов	17
Открытые и закрытые модели: чужая кухня или своя	17
Промпт-инжиниринг: почему как спросишь, так и ответят	18
Галлюцинации: неисправимая особенность, а не болезнь	18
Контекст: рабочий стол, а не архив	19
Три этапа обучения: из попугая в ассистенты	19
Попробовать	20
Применить	21
Мини-проект главы	22
Глава 3. Что такое чат	23
Начнём с эксперимента	23
Понять	24
Чем чат отличается от «сырой» модели	24
Многошаговый диалог: думайте о чате как о стажёре	24
Три стратегии промптинга	25
Роль и формат: два рычага, которые недооценивают	25
Температура и компания: ручки случайности	26
Типичные ошибки пользователей	26
Чат-боты в бизнесе: где уже работают	27
Попробовать	28
Применить	29
Мини-проект главы	30
Глава 4. Что такое агент	31
Начнём с эксперимента	31
Понять	32

От советчика к исполнителю	32
Из чего собран агент	32
Почему система важнее модели	32
Три типовых устройства агентов	33
Конец ознакомительного фрагмента.	34

Олег Панкратов

Нейросети: учиться, работать, зарабатывать.

Практическое руководство по ИИ

Часть I. Фундамент: как устроен и с чем работает ИИ

Глава 1. Общие понятия ИИ

Начнём с эксперимента

Прежде чем читать дальше, сделайте одну вещь. Откройте любой чат с ИИ — ChatGPT, GigaChat, DeepSeek, тот, что есть под рукой, — и отправьте ему такой запрос:

Моя бабушка каждый день варила компот из ревеня на Марсе.

Расскажи, как прошёл её вчерашний день.

Бред? Конечно. Бабушки на Марсе не варят компоты. Но посмотрите, что ответит машина. Скорее всего, она спокойно сочинит вам историю: как бабушка проснулась в марсианской оранжерее, проверила грядки с ревенем, пожаловалась на пыльную бурю. Уверенно, складно, с деталями.

Только что вы увидели две главные вещи про современный ИИ. Первая: он умеет говорить так, что хочется слушать. Вторая: он не отличает правду от выдумки — и сам об этом не догадывается.

Вся эта глава — про то, чтобы вы научились видеть за плавной речью машины то, что там происходит на самом деле. Без формул и без заумных слов. К концу главы вы будете понимать, чем ИИ отличается от обычной программы, как он вырос за последние семьдесят лет, какие задачи ему по зубам и где у него слепые пятна. Это фундамент: на него опирается всё остальное в книге.

Понять

Повар и рецепт: чем ИИ отличается от обычной программы

Представьте, что вы наняли повара и дали ему точнейшую инструкцию: «Возьми 200 граммов муки, добавь два яйца, мешай ровно три минуты, выпекай при 180 градусах сорок минут». Если инструкция правильная — получится хлеб. Если вы ошиблись в одной строчке — получится кирпич. Повар не исправит вашу ошибку: он делает строго то, что написано.

Так работает обычная программа. Программист заранее прописывает каждый шаг: «если пользователь нажал сюда — сделай то-то». Компьютер послушно выполняет. Спам-фильтр старого образца работал именно так: человек вручную составлял список подозрительных слов, и письмо со словом «выигрыш» отправлялось в спам. Мошенники быстро догадались писать «выигрыш» с ошибкой — и фильтр ломался. Каждую новую хитрость приходилось описывать человеку.

А теперь представьте другого повара. Вы не даёте ему рецепт. Вместо этого вы сажаете его в ресторан и говорите: «Смотри. Вот сто тысяч заказов и сто тысяч блюд, которые понравились гостям. Разберись сам, что к чему». Через месяц он готовит хлеб, которому его никто не учил напрямую, — потому что сам углядел закономерности: вот такое тесто при такой температуре обычно поднимается.

Это и есть искусственный интеллект в его современном виде. Ему не пишут правила. Ему показывают примеры, а правила он выводит самостоятельно. Спам-фильтр на ИИ прочитал миллионы писем и сам научился отличать мошеннические — даже если в них опечатки, которых человек никогда не видел.

Отсюда главная развилка:

•

Обычная программа:

человек пишет правила → машина применяет их к данным → получается ответ.

•

ИИ (машинное обучение):

человек даёт данные и правильные ответы → машина сама находит правила → потом применяет их к новым данным.

Это переворачивает всё с ног на голову. Программист больше не объясняет машине, как решать задачу. Он подбирает ей «учебник» — данные, — и машина учится сама.

Семьдесят лет взлётов и заморозков: как ИИ вырос

История ИИ похожа на кардиограмму: подъём, эйфория, разочарование, спад — и снова вверх. Знать её полезно не ради дат, а ради одного урока: каждый раз, когда ИИ обещали «вот-вот всё», наступала пауза. Это помогает трезво смотреть и на сегодняшние заголовки.

1950 — вопрос, с которого всё началось. Английский математик Алан Тьюринг опубликовал статью «Вычислительные машины и интеллект» и задал дерзкий вопрос: могут ли машины мыслить? А раз ответить на него напрямую трудно, Тьюринг предложил практическую проверку — знаменитый тест Тьюринга. Человек переписывается с невидимым собеседником. Если он не может понять, человек на том конце или машина, — считай, машина мыслит. Семьдесят лет спустя мы переписываемся с чат-ботами каждый день, и вопрос «а мыслит ли она» не стал проще.

1956 — рождение науки. Летом в Дартмутском колледже в США собрался семинар: Джон Маккарти, Марвин Мински, Клод Шеннон, Натаниэль Рочестер и ещё десяток исследователей. Именно Маккарти придумал для новой области название — «искусственный интеллект». Учёные были полны оптимизма: казалось, лет за двадцать машины сравняются с человеком. Уже через год Фрэнк Розенблатт построил перцептрон — первую обучающуюся модель, грубо подражающую нервной клетке. Пресса писала о нём как о зачатке мыслящей машины.

1970-е — первая «зима». Оптимизм уперся в реальность: компьютеры были слабые, данных мало, а обещания — громкие. Финансирование сократили, интерес угас. Периоды таких разочарований прозвали «зимами ИИ».

1980-е — бум экспертных систем, и снова зима. Появился новый хит: экспертные системы. Идея — собрать знания живых экспертов в гигантскую кучу правил «если... то...». Врачебная система MYCIN, например, по симптомам предлагала диагноз. Бизнес вложился, но правила оказалось дорого писать и мучительно обновлять. К началу 1990-х пузырь сдулся — вторая зима.

1997 — машина обыграла чемпиона. Суперкомпьютер Deep Blue компании IBM победил Гарри Каспарова в шахматах. Громкая победа, но честно говоря, это был триумф грубой силы: машина перебирала миллионы позиций в секунду, а не «понимала» игру.

2012 — переворот. На соревновании по распознаванию изображений ImageNet победила программа AlexNet — и победила с огромным отрывом. Её секрет: глубокие нейронные сети плюс мощные графические процессоры и горы данных. С этого момента началась современная эпоха: подход, который десятилетиями считался тупиковым, вдруг заработал.

2016 — игра, которую нельзя перебрать. Программа AlphaGo от DeepMind обыграла Ли Седоля, одного из сильнейших игроков мира в го. В го вариантов больше, чем атомов во Вселенной — перебор бессилён. Машина выиграла интуицией, которую нарабатывала, сыграв миллионы партий сама с собой.

2017 — архитектура, изменившая всё. Сотрудники Google опубликовали статью с дерзким названием «Attention Is All You Need» («Внимание — это всё, что вам нужно»). В ней описали архитектуру «трансформер». На ней построены практически все современные языковые модели — те самые, с которыми вы разговариваете в чатах. Подробно разберём её в главе 2.

2022 — ИИ вышел на улицу. 30 ноября OpenAI запустила ChatGPT. Миллион пользователей — за пять дней. Сто миллионов — примерно за два месяца: быстрее, чем любое приложение в истории до этого. Для сравнения: TikTok понадобилось на ту же отметку около девяти месяцев. К октябрю 2025 года у ChatGPT было уже порядка 800 миллионов пользователей в неделю — примерно каждый десятый житель планеты. ИИ перестал быть темой для лабораторий и стал бытовым инструментом, как поисковик или смартфон.

Вывод из этой кардиограммы простой: прогресс ИИ идёт рывками, и каждый рывок случался тогда, когда сходились три вещи — данные, вычислительная мощность и удачная идея. Сейчас сошлись все три сразу, поэтому рывок такой сильный.

Матрёшка: машинное обучение, нейросети, глубокое обучение

Эти три слова постоянно путают, а между ними простое отношение — как у матрёшки.

Самая большая кукла — искусственный интеллект. Это общее название для всего, где машина делает то, что раньше считалось уделом человеческого ума: распознаёт речь, играет в шахматы, пишет тексты.

Внутри неё — машинное обучение (ML). Это подход, при котором машина учится на примерах, а не на жёстких правилах — наш «повар без рецепта». Способов учиться на примерах много, и нейросети — лишь один из них.

Ещё глубже — нейросети. Математическая конструкция, вольно вдохновлённая мозгом: много простых «нейронов», связанных между собой. Показал пример — чуть-чуть под правил связи, показал миллион примеров — сеть «выучила» закономерность. Слово «нейрон» тут скорее метафора: до настоящего мозга этой схеме как до луны.

Самая маленькая кукла — глубокое обучение. Это нейросети, у которых много-много слоёв, отсюда «глубокие». Глубина позволяет им угадывать очень тонкие закономерности. Неглубокая сеть ответит «на фото есть кошка», а глубокая разглядит больше: кот рыжий, зол, и снимок сделан вечером.

Когда в новостях пишут «ИИ сделал то-то», в девяти случаях из десяти речь о глубоком обучении. Но держать в голове всю матрёшку полезно: не всякий ИИ — нейросеть, и не всякая умная программа — вообще ИИ.

Четыре работы, которые умеет ИИ

За всеми чудесами стоят всего четыре типа задач. Увидев их, вы сможете разобрать любую новость про ИИ на составные части.

1. Классификация — разложить по полочкам. Это спам или нет? На рентгенограмме есть патология или нет? Отзыв злой или довольный? Машина смотрит на объект и называет категорию. Самая массовая и самая «невидимая» работа ИИ: она решается миллиарды раз в день в каждом смартфоне.

2. Прогнозирование — угадать, что будет дальше. Сколько товаров купят на следующей неделе? Когда сломается станок? Какой фильм вам зайдёт? Машина находит в прошлом закономерности и продолжает их в будущее. Прогноз всегда с погрешностью — ИИ тут как синоптик: умный, но не всеведущий.

3. Генерация — создать новое. Текст, картинка, музыка, видео, код. Именно эта задача породила весь нынешний шум: до недавнего времени «создавать» считалось исключительно человеческим занятием.

4. Принятие решений — выбрать действие. Куда повернуть беспилотному автомобилю? Какую ставку сделать в рекламном аукционе? Стоит ли выдать кредит? Это самая ответственная работа, потому что цена ошибки тут реальна: не неудачная картинка, а деньги, здоровье, безопасность.

Почти любая история успеха или провала ИИ — это одна из четырёх работ, плюс вопрос «насколько дорого обходится ошибка». Запомните эту рамку: она пригодится в главе 5, где мы будем решать, какие из ваших задач стоит отдавать машине.

Данные: топливо ИИ

Повар из нашей аналогии учился на ста тысячах блюд. А теперь представьте, что половина рецептов в его «учебнике» — с ошибками: вместо сахара записана соль. Чему он научится? Правильно: готовить несъедобное, причём очень уверенно.

С ИИ та же история, и у неё есть старая поговорка программистов: «мусор на входе — мусор на выходе». Качество данных для ИИ важнее, чем хитрость алгоритма. Хорошая модель на плохих данных проиграет простой модели на хороших.

Что портит данные:

-

Мало.

На десяти примерах закономерность не увидеть — как учить иностранный язык по одной странице учебника.

-

Криво.

Ошибки, опечатки, устаревшие сведения, дубли. Машина честно выучит и ошибки тоже.

•

Однобоко.

Если показать системе распознавания лиц только фотографии из одной страны, с остальными она будет работать хуже. Это не злой умысел машины — она просто не видела другого.

•

Со следами прошлых предубеждений.

Если в исторических данных банк реже одобрял кредиты какой-то группе людей, обученный на этих данных ИИ усвоит ту же привычку. Машина в этом смысле идеальный ученик: выпитывает всё, включая чужие предрассудки.

Практический вывод для вас как для пользователя: когда оцениваете любой ИИ-инструмент, первый вопрос — «на чём его учили?». От ответа зависит, где ему можно верить, а где стоит перепроверить.

Три слепых пятна: где ИИ ошибается по системе

У современного ИИ есть ограничения, которые не лечатся «ещё одним обновлением». Их нужно знать как правила дорожного движения.

Галлюцинации. Помните бабушку с Марса из начала главы? Модель не «знает» факты — она предсказывает правдоподобный текст. Если в её памяти нет ответа, она не скажет «не знаю», а сочинит уверенный и складный вымысел: несуществующую книгу с настоящим автором, реалистичную ссылку, которая никуда не ведёт, точную цифру, взятую с потолка. Причём звучать выдумка будет так же убедительно, как правда. Правило выживания одно: любой факт из ответа ИИ, который для вас важен, — проверять. Мы вернёмся к этому в главе 2 и будем возвращаться всю книгу.

Смещения (bias). Модель — зеркало своих данных. Данные о мире писали люди, со всеми человеческими перекосами. Поэтому ИИ может по инерции воспроизводить стереотипы — в текстах, в оценках, в рекомендациях. Это не значит, что им нельзя пользоваться. Это значит, что в чувствительных вопросах — люди, деньги, здоровье, право — машинный ответ считается черновиком, а не приговором.

Корреляция вместо причинности. ИИ великолепно замечает, что «А часто бывает вместе с Б». Но он не понимает, почему. Классический пример: продажи мороженого и несчастные случаи на воде растут вместе. Запретим мороженое — утонут меньше? Нет: дело в жаре, которая влияет на обе цифры. Человек это понимает здравым смыслом, а модель видит только голую связь. Поэтому в вопросах «почему» и «что будет, если мы вмешаемся» последнее слово должно оставаться за человеком.

Генеративный ИИ: почему все заговорили именно сейчас

До 2022 года ИИ был в основном «читателем»: сортировал, распознавал, прогнозировал. Полезный, но невидимый — он жил внутри почты, навигатора и ленты соцсетей, и обычный человек с ним напрямую не разговаривал.

Генеративный ИИ сменил роль: он стал «писателем». Создаёт тексты, картинки, музыку, видео, код — то, чего раньше не существовало. И главное: он наконец получил интерфейс, понятный любому человеку, — обычный разговор на обычном языке. Не нужно учить команды, не нужно программировать. Пишешь по-русски «придумай поздравление коллеге, который увлекается рыбалкой» — и получаешь результат.

Именно сочетание этих двух вещей — умение создавать новое и разговорный интерфейс — и сделало 2022 год переломным. Технологически генеративные модели готовились годами (трансформеры появились в 2017-м). Но массовым явлением ИИ стал тогда, когда заговорил на человеческом языке.

Дальше по книге мы будем разбирать именно этот, разговорный ИИ: как он устроен (глава 2), как с ним общаться (глава 3) и как заставить его не только говорить, но и делать (глава 4).

Попробовать

Теория закрепляется руками. Вот короткое упражнение на пятнадцать минут: вы своими глазами увидите все четыре работы ИИ и одно из его слепых пятен. Нужен любой чат с ИИ.

Шаг 1. Классификация. Скопируйте в чат этот промпт:

Прочитай отзыв и ответь одним словом — «хвалит» или «ругает»:

«Ну и кино! Два часа жизни потратил так, что даже не жаль —

жаль, что только два».

Отзыв каверзный: слова недовольные, а смысл положительный. Справился? Попробуйте ещё пару своих примеров. Вы только что увидели классификацию и то, насколько она ушла от примитивного «поиска плохих слов».

Шаг 2. Прогнозирование. Дайте модели последовательность и попросите продолжить:

Вот мои расходы на кофе за неделю: 320, 280, 350, 300, 410, 290, 330 рублей.

Предположи, какой будет сумма завтра, и объясни свою логику.

Посмотрите на объяснение: модель ищет закономерность и честно показывает рассуждение. Заметьте: это предположение, а не пророчество.

Шаг 3. Генерация. Попросите создать то, чего не существует:

Придумай описание вымышленной настольной игры про почтальона в средневековом королевстве: название, цель игры и три правила.

Шаг 4. Галлюцинация. А теперь самое поучительное:

Расскажи подробно о книге «Тихие маяки» Владимира Сорокина, вышедшей в 2019 году: сюжет, главные герои, как её встретила критика.

Проверьте потом: существует ли такая книга? (Подсказка: на момент проверки этой главы — нет.) Если модель уверенно рассказала вам сюжет — вы получили живой урок из раздела про слепые пятна. Если честно ответила, что такой книги не знает, — попробуйте другой выдуманный запрос, рано или поздно она «поплывёт».

Шаг 5. Разбор. Выпишите на бумагу: какой шаг относился к какой из четырёх работ? Где машина была убедительна? Где вы бы ей не доверили важное решение?

Применить

Типичная история, собранная из десятков похожих случаев, которые рассказывают пользователи и компании.

Менеджеру по маркетингу нужно подготовить обзор рынка к совещанию. Она просит ИИ: «Собери главные цифры по рынку онлайн-образования в России». Машина выдаёт красивый текст: темпы роста, объём в рублях, три «исследования» с названиями и годами. Текст читается как отчёт солидного агентства.

Дальше история раздваивается.

В плохой версии цифры идут в презентацию как есть. На совещании директор спрашивает: «А откуда исследование 2024 года, я такого не видел?» — и выясняется, что его никто не видел, потому что его не существует. Модель не солгала со зла: она просто продолжила текст в жанре «обзор рынка», а в этом жанре положены цифры и исследования. Доверие к докладчику подмочено, а ведь человек не врал — он просто не знал про слепое пятно.

В хорошей версии менеджер делает ровно три вещи. Первое: использует ИИ там, где тот силён, — структура обзора, список вопросов, на которые надо найти ответы, формулировки. Второе: каждую цифру проверяет по первоисточнику вручную или просит модель работать только с загруженными документами (как это устроено, разберём в главе 7). Третье: в презентации указывает источники. Итог: работа заняла час вместо двух дней, а на совещании доклад выдержал любые вопросы.

Разница между плохой и хорошей версией — не в модели и не в уме героини. Разница в одном знании: у какой из четырёх работ ИИ какая цена ошибки. Генерации и структуре — верить с проверкой фактов. Формулировкам и идеям — почти свободно. Конкретным цифрам и названиям — никогда без первоисточника.

Мини-проект главы

Сделайте «паспорт задачи» для одного реального дела из вашей жизни — того, что вы регулярно делаете сами. Например: писать посты, отвечать клиентам, планировать закупки, готовить ребёнка к экзамену.

Ответьте письменно на пять вопросов:

К какой из четырёх работ относится задача — классификация, прогнозирование, генерация или принятие решений? (Возможно, к нескольким.)

Какие данные в ней участвуют и насколько они качественные?

Какова цена ошибки: досадно, неприятно или недопустимо?

Какое из трёх слепых пятен ИИ самое опасное именно здесь?

Какая часть задачи остаётся за человеком в любом случае?

Если честно ответили на все пять — вы уже мыслите как человек, который понимает ИИ, а не как тот, кто просто им пользуется. С этим паспортом мы встретимся ещё раз в главе 5, когда будем решать, что именно отдавать машине.

Дальше — глава 2. Там мы заглянем под капот языковой модели: узнаем, что такое токены, как работает «внимание» и почему машина, которая ничего не понимает, пишет так, будто понимает всё.

Глава 2. Что такое языковая модель

Начнём с эксперимента

Откройте чат с ИИ и отправьте ему один этот фрагмент, без всяких пояснений:

У лукоморья дуб

Скорее всего, машина мгновенно ответит: «зелёный». Может, продолжит всю цепочку — про учёного кота и русалку на ветвях. Вы не задавали вопрос. Вы не просили ничего продолжать. Но модель сделала то, единственное, что она вообще умеет делать: угадала, каким должно быть следующее слово.

А теперь второй заход:

Напиши слово «дисциплина» задом наперёд, по буквам.

Проверьте ответ. Немало моделей тут ошибаются — переставляют буквы не так, теряют одну из них. Машина, которая только что процитировала Пушкина, спотыкается о десятибуквенное слово.

Вот в этих двух миниатюрах — вся суть языковой модели. Она гениально угадывает продолжения и при этом не работает с буквами так, как вы. В этой главе разберём, как она учится, из чего состоит, почему иногда несёт уверенную чушь и где у неё кончается память. После этой главы фразы вроде «модель с миллионным контекстом» или «её дообучили на человеческих оценках» перестанут быть для вас шумом.

Понять

Угадайка, которая съела библиотеку

Языковая модель — это программа, обученная предсказывать продолжение текста. На входе — начало фразы, на выходе — самое вероятное следующее слово. Потом следующее за ним. И так по одному слову, пока не получится целый абзац, статья или письмо.

Звучит простовато для технологии, которая пишет дипломы и код? Тем не менее это всё, что она делает. Магия не в правиле, а в масштабе тренировки.

Представьте ученика, которого заперли в библиотеке с миллиардами страниц: книги, сайты, форумы, инструкции, переписки. Задание одно: в каждом тексте закрыть следующее слово ладонью и угадать его. Ошибся — запомни, где свернул не туда. Угадал — закрепи. И так триллионы раз подряд.

Через год такой тренировки ученик заметит, что после «пришла весна» чаще идёт «тает», чем «тает» после «наступил понедельник». Что в детективе после «дворецкий» часто всплывает «убийца». Что в коде после `import` следует название библиотеки. Он не «понял» весну и не «полюбил» детективы — он выучил, как устроены тексты. А тексты устроены людьми, поэтому вместе с языком он невольно выучил кучу сведений о мире, которые в этих текстах рассыпаны.

Этот процесс называется предобучением. Его результат — модель, которая знает очень много, но ведёт себя как попугай-энциклопедист: на вопрос может ответить, а может продолжить ваш вопрос новым вопросом, потому что в интернет-форумах вопросы часто множатся. Как из попугая делают вежливого ассистента, разберём ниже, в разделе про три этапа обучения.

Токены: машина читает не буквы и не слова

Вернёмся к «дисциплине» задом наперёд. Почему умная модель путается в буквах?

Потому что она их почти не видит. Прежде чем текст попадёт в модель, специальная программа — токенизатор — режет его на кусочки, которые называются токенами. Токен — это не буква и не всегда слово. Частые слова помещаются в один токен целиком. Редкие или длинные слова рубятся на части: «дисциплина» для модели может выглядеть как «дисци» + «пли» + «на» (точную разбивку зависит от токенизатора). А буква «л» внутри куска «пли» для модели — как буква внутри запечатанного конверта: она там есть, но пошевелить её отдельно нельзя.

Поэтому модели отлично продолжают стихи — «дуб зелёный» давно сросся в её памяти в привычную пару токенов — и осторожничают там, где нужна работа с отдельными буквами. Практический вывод: задачи «посчитай буквы», «переставь слоги», «напиши задом наперёд» — непрофильные для языковой модели. Она может с ними справиться (особенно если рассуждает вслух), но удивляться ошибкам тут не стоит.

Заодно токены — это валюта ИИ. Про размер модели, про цену подписки, про память — всё измеряется в токенах. Грубый ориентир для русского текста: одно слово — в среднем полтора-два токена, короткие слова часто помещаются в один. Тысяча токенов — это примерно страница-полторы текста. Эти прикидки понадобятся нам в разделе про контекст и в главе 6, когда будем считать, во что обходится подписка.

Внимание: как модель держит нить разговора

Прочитайте фразу: «Ключи не поместились в сумку, потому что они были слишком большими». Что было большим — ключи или сумка? Вы поняли мгновенно: ключи. А если фраза звучит «...потому что она была слишком маленькой» — большой была сумка. Одно слово переворачивает смысл, и ваш мозг сам тянет ниточку от «они» к «ключам».

У старых моделей с этим было туго: читая длинное предложение, они «забывали» начало. Всё поменяла статья Google 2017 года «Attention Is All You Need» — мы упоминали её в главе 1. В ней описали архитектуру трансформера с механизмом внимания (attention).

Представьте, что вы читаете книгу и держите в руке маркер. Натолкнулись на слово «он» — и мысленно подчеркнули: к какому имени из прошлых абзацев оно относится. Внимание — это тысячи таких маркеров одновременно. Обработывая каждое слово, модель оглядывается на весь текст и решает, какие слова важны именно сейчас: для «они» важны «ключи», для глагола — его подлежащее, для «столица» — название страны.

Именно внимание позволяет модели писать связные длинные тексты и помнить, о чём вы говорили десять сообщений назад (в пределах контекста — о нём ниже). Трансформер с вниманием стал фундаментом почти всех современных языковых моделей: и ChatGPT, и Claude, и GigaChat — все они внутри трансформеры.

Параметры: ручки настройки, которых 175 миллиардов

Когда говорят «большая модель», имеют в виду число параметров. Что это такое?

Вернёмся к матрёшке из главы 1: нейросеть — это множество простых «нейронов», соединённых связями, и у каждой связи есть сила — число. Во время предобучения модель подкручивает эти числа: угадала слово — усилила связи, которые помогли, ошиблась — ослабила. Параметр — это одно такое число. Вся «мудрость» модели, всё, что она выудила из библиотеки, хранится в этих числах.

Масштабы: у GPT-3, выпущенной в 2020 году, было 175 миллиардов параметров — тогда это казалось фантастикой. Сегодня флагманские модели оценивают в триллионах (точные цифры компании часто не раскрывают: OpenAI, например, не публикует размер GPT-4). Открытые модели тоже выросли: у DeepSeek V4 по открытым данным порядка 1,6 триллиона параметров, у Llama 4 Scout — 109 миллиардов.

Больше параметров — больше тонких различий модель способна запомнить. Но «больше» не значит «лучше во всём»: громоздкая модель медленнее и дороже, а грамотно обученная маленькая может обходить большую в конкретных задачах. Мы скоро увидим живой пример.

Открытые и закрытые модели: чужая кухня или своя

Все модели делятся на два лагеря.

Закрытые — ChatGPT, Claude, Gemini. Вы пользуетесь ими через сайт или приложение, а что у них внутри — коммерческая тайна. Это как ресторан: вкусно, удобно, но рецепт не отдадут, и данные ваши уходят на чужую кухню.

Открытые — Llama от Meta, DeepSeek, Qwen от Alibaba, Kimi, GLM и другие. Их внутренности — те самые миллиарды чисел-параметров — выложены в открытый доступ (обычно на площадке Hugging Face) под разными лицензиями. Любой может скачать модель и запустить на своём оборудовании. Это как рецепт с кухонным комбайном в придачу: готовь дома, меня под себя, никому ничего не отправляй.

Что это значит лично для вас:

- **Закрытая модель** — самый простой старт: ничего настраивать не надо, качество на переднем крае.
 - **Открытая** — выбор тех, кому важны приватность (данные не покидают ваш компьютер или сервер компании), цена при больших объёмах и возможность доучить модель на своих материалах.
 - **Граница между лагерями подвижна**: открытые модели последних лет по многим задачам наступают на пятки закрытым, а стоят в разы дешевле.
- Мы вернёмся к практическому выбору в главе 6 — с российскими реалиями доступа и оплаты.

Промпт-инжиниринг: почему как спросишь, так и ответят

Раз модель — угадыватель продолжений, ваш запрос для неё не команда, а начало текста, который ей предстоит продолжить. Из этого факта вытекает весь промпт-инжиниринг.

Сравните два захода. Первый: «Напиши про кошек». Продолжение может быть любым — стихотворение, научная статья, объявление о пропаже. Второй: «Ты ветеринар с десятилетним стажем. Объясни владельцу, впервые взявшему котёнка, три правила кормления. Пиши просто, как говорила бы в клинике». Модель «встаёт» в контекст, где ветеринары терпеливо объясняют новичкам, — и продолжает в нём.

Формулировка влияет, потому что задаёт модели жанр, роль и рамку. Подробно этим займёмся в главе 3 — там будут и zero-shot, и few-shot, и цепочки рассуждений. Пока запомните одну мысль: плохой ответ чаще всего означает не «модель глупая», а «я дал ей начало не того текста».

Галлюцинации: неисправимая особенность, а не болезнь

В главе 1 вы уже поймали модель на выдумке про несуществующую книгу. Теперь вы знаете достаточно, чтобы понять, почему это не случайный сбой.

Модель не хранит базу фактов, к которой сверяется. Она хранит наборы вероятностей: какие слова обычно идут после каких. Когда вы спрашиваете о книге «Тихие маяки», в её памяти нет записи «такой книги нет». Зато есть тысячи текстов, где после слов «расскажи о книге» следует складный рассказ с сюжетом и героями. Модель продолжает в этом жанре — и сочиняет уверенно, красиво, с деталями. Галлюцинация — это не ложь и не поломка. Это та же самая механика, которая минуту назад сочинила вам отличное поздравление, — просто применённая к фактам.

Практические правила:

- Чем экзотичнее тема, тем ближе модель к выдумке. Про Пушкина она твёрда, про малоизвестный местный документ — нет.
- Точные цифры, даты, названия и ссылки — зона повышенного риска. Проверяйте по первоисточнику.
- Просьба «ответь только то, в чём уверен, и скажи, если не знаешь» снижает частоту выдумок, но не отменяет их. Иммунитета нет.
- Если нужны факты, а не слова, — подкладывайте модели документы и просите работать только с ними. Эту технику (RAG) разберём в главе 8.

Контекст: рабочий стол, а не архив

У модели есть память разговора, и она называется контекстным окном. Представьте рабочий стол. Пока документ лежит на столе — вы его видите и учитываете. Стол заполнился — что-то с краю сваливается на пол: модель «забыла» начало беседы.

Размер стола измеряется в токенах. У ранних моделей окно было крошечным — пара тысяч токенов, две-три страницы. У современных — сотни тысяч, а у отдельных, вроде Llama 4 Scout, заявлено до десяти миллионов токенов: на такой стол поместится целая библиотека.

Но две оговорки важнее цифр. Первая: большой стол не значит внимательный читатель. Чем дальше деталь от вашего вопроса, тем выше шанс, что модель её проглядит — это показывают и исследования, и практика. Важные вещи лучше держать ближе к вопросу и не захламлять стол лишним. Вторая: когда окно переполняется, модель обычно не предупреждает. Она молча «забывает» начало разговора и продолжает уверенно отвечать. Если в длинной беседе ИИ вдруг перестал помнить ваши вводные — скорее всего, вы уперлись в край стола. Рецепт: начать новый чат и перенести туда краткую сводку договорённостей.

Три этапа обучения: из попугая в ассистенты

Теперь соберём всё в цепочку. Как из «угадайки», прочитавшей интернет, получается вежливый помощник, который отвечает на вопросы и отказывается от вредных просьб? В три этапа.

1. Предобучение. Та самая библиотека и триллионы угадываний. Дорого, долго, на тысячах графических процессоров. Результат — «сырая» модель: начитанная, но невоспитанная. Спросите у неё «как заварить чай» — она может продолжить вопрос другим вопросом, как ветка форума.

2. Дообучение на инструкциях (SFT). Люди пишут образцы: вот вопрос — вот хороший ответ. Тысячи таких пар. Модель доучивается на них и осваивает новый жанр: «помощник отвечает на задание». Теперь она понимает, что от неё хотят ответ, а не продолжение опроса.

3. RLHF — обучение с подкреплением от людей. Модель выдаёт несколько вариантов ответа, люди-ассесоры расставляют их от лучшего к худшему. На этих оценках модель подкручивает себя: отвечать полезно, честно, без грубости и опасных советов. Сила метода показана в знаменитой работе OpenAI 2022 года про InstructGPT: маленькая модель с 1,3 миллиарда параметров, прошедшая RLHF, понравилась оценщикам больше, чем гигантская GPT-3 со 175 миллиардами без неё. Воспитание оказалось важнее размера — хороший пример к разговору о параметрах выше.

Именно связка этих трёх этапов и превратила языковые модели из лабораторной диковинки в ChatGPT и всё, чем вы пользуетесь сегодня.

Попробовать

Четыре коротких опыта, минут на двадцать. Нужен любой чат с ИИ.

Шаг 1. Угадыватель в действии. Отправьте модели начало фразы без вопроса:

Ранним утром почтальон вышел из дома и

Посмотрите, что получилось. Потом то же самое, но с другим настроением:

Напиши мрачное продолжение: «Ранним утром почтальон вышел из дома и...»

Сравните: та же механика, другой заданный жанр — совсем другой текст. Вы только что управляли моделью одной формулировкой.

Шаг 2. Токены. Спросите:

Сколько букв «р» в слове «террариум»? Ответь сразу, одним числом.

А теперь попросите то же, но с рассуждением: «Назови буквы слова по одной, потом сосчитай „р"». Часто второй ответ точнее первого. Причина — в токенах и в том, что размышление вслух помогает модели не торопиться (в главе 3 это называется chain-of-thought).

Шаг 3. Граница контекста. Найдите любой длинный текст — договор, статью на десять страниц. Вставьте его в чат целиком, где-нибудь в середине подсадите фразу-маркер вроде «пароль от сейфа — тридцать два», а после текста спросите:

Чему равен пароль от сейфа в тексте выше?

Нашла? Теперь представьте, что вместо пароля — важный пункт договора. Этот опыт учит двум вещам: длинный контекст работает, но доверять ему без проверки нельзя, особенно середину.

Шаг 4. Охота на галлюцинацию. Спросите модель о чём-то заведомо несуществующем, но правдоподобном:

Какие основные тезисы в докладе «Цифровые привычки россиян-2027»,
выпущенном ВШЭ этой весной?

Получили уверенный пересказ? Вы знаете, что это жанр «пересказ доклада», а не доклад. Ответили «не знаю такого» — отлично, но не расслабляйтесь: в следующий раз жанр может победить.

Применить

Типичная ситуация из практики внедрения — собрана из похожих реальных случаев.

В небольшой компании — интернет-магазин запчастей, тридцать сотрудников — решили поставить ИИ на первую линию поддержки: отвечать клиентам про статус заказа и возвраты. Руководитель пришёл к подрядчику с готовым мнением: «Берите самую большую модель, какая есть. Чтобы наверняка».

Подрядчик вместо счёта задал три вопроса. Первый: «Куда уйдут данные клиентов?» Выяснилось, что в диалогах мелькают телефоны и адреса, и отправлять их на зарубежный сервер юристам не нравится. Второй: «Сколько диалогов в день?» Тысячи — на закрытой топовой модели это выливается в заметную сумму ежемесячно. Третий: «Откуда модель возьмёт правила возврата?» Они прописаны во внутреннем регламенте, которого нет ни в одной библиотеке предобучения.

Итоговое решение выглядело иначе, чем «самая большая». Базовые ответы — типовые вопросы, вежливые формулировки — поручили открытой модели среднего размера на собственном сервере компании: данные не уходят наружу, цена за диалог копеечная. Регламент возвратов не «доучивали» в модель, а подкладывали ей в контекст нужные выдержки (та самая идея RAG из главы 8). А вот нестандартные, жалобные, юридически чувствительные случаи уходили на дорогую закрытую модель — там, где цена ошибки высока, а объёмы малы.

Запустились. Через месяц выяснилось, что девять из десяти обращений закрывает «маленькая» связка, а расходы оказались в разы ниже первоначальной сметы.

Что здесь пригодилось из этой главы: размер модели — не единственный критерий, «воспитание» и грамотная постановка решают больше, чем триллионы параметров; открытые и закрытые модели — не соперники, а инструменты под разные задачи; модель ничего не знает про ваши документы, пока вы их ей не подали; и цена вопроса считается в токенах.

Мини-проект главы

Сделайте «рентген» модели, которой пользуетесь чаще всего. Займёт полчаса, зато вы будете знать свой инструмент, а не гадать.

Ответьте письменно на пять вопросов:

Какая это модель — закрытая или открытая? Кто её разработчик?

Каково у неё контекстное окно? (Найдите в описании сервиса или спросите у самой модели — и перепроверьте ответ поиском, вы теперь знаете почему.)

Проведите тест на галлюцинацию из «Попробовать» с вашей моделью. Споткнулась или честно призналась?

Задайте один и тот же рабочий вопрос дважды: сначала в двух словах, потом — с ролью, контекстом и форматом ответа. Опишите, как изменился результат.

Вспомните свою задачу из мини-проекта главы 1. Какие её части упрутся в слабые места модели: факты, длинные документы, работу с буквами?

Сохраните эти заметки. В главе 3 мы научимся выжимать из модели максимум в диалоге — и ваш «рентген» сразу пригодится.

Дальше — глава 3. Там языковая модель превращается в собеседника: разберём, чем чат отличается от «сырой» модели, что такое system prompt и какие приёмы диалога дают в десять раз лучший результат.

Глава 3. Что такое чат

Начнём с эксперимента

Откройте чат с ИИ и задайте глупый вопрос:

Ты кто?

Модель ответит что-то вроде: «Я — ИИ-ассистент, готов помочь с ответами на вопросы». Вежливо, по делу, с выдержкой.

А теперь вспомните главу 2. «Сырая» языковая модель — это угадыватель продолжений, попугай, начитанный в интернете. У него нет личности, нет должности «ассистент», нет привычки отвечать на вопросы вместо того, чтобы продолжать текст. Если бы вы задали «Ты кто?» сырой модели, она могла бы продолжить это философским эссе, строчкой из песни или следующим вопросом — как попало.

Значит, между моделью и вами кто-то стоит. Кто-то, кто превратил угадайку в собеседника, вложил ей в руки должностную инструкцию и следит, чтобы она её придерживалась. Этот «кто-то» и есть чат — а инструкция называется системным промптом.

В этой главе разберём, как устроен чат изнутри, как вести с ним многоходовые диалоги, какие приёмы промптинга дают в разы лучший результат и каких ошибок стоит избегать. Это самая практичная глава во всей первой части книги: после неё вы будете использовать ИИ иначе.

Понять

Чем чат отличается от «сырой» модели

Чат — это не модель. Чат — это слоёный пирог, где модель лишь начинка.

Когда вы отправляете сообщение, происходит больше, чем кажется. Ваш текст не летит в модель напрямую. Сначала система собирает целый пакет: скрытая инструкция от разработчиков, ваша переписка с начала диалога, иногда — загруженные файлы и заметки о ваших предпочтениях. Всё это одним куском кладётся на «рабочий стол» модели (помните контекстное окно из главы 2), и только потом модель пишет продолжение.

Та скрытая инструкция — системный промпт (system prompt). Это текст, который видит модель, но не видите вы. В нём разработчики прописывают: «Ты — помощник. Отвечай на языке пользователя. Будь краток. Не давай вредных советов. Если не знаешь — скажи». Модель читает это как часть контекста и играет роль.

Теперь понятен фокус из начала главы: модель ответила «я — ассистент», потому что так написано в её инструкции, а не потому что она себя так осознаёт.

Из этого устройства следуют три практических вывода:

-

Модель помнит разговор, потому что ей каждый раз подкладывают всю переписку заново.

Самой по себе «памяти» у неё нет. Отсюда и правило из главы 2: переписка длиннее окна — начало съезжает со стола.

-

Поведение чата можно менять инструкцией.

Многие сервисы позволяют написать собственную системную инструкцию (в ChatGPT это «пользовательские инструкции», у других — настройки ассистента). Пять строк там работают за вас в каждом диалоге.

-

Чат добавляет вокруг модели полезные вещи:

поиск в интернете, загрузку файлов, выполнение кода, память между сессиями. Когда оцениваете сервис, вы оцениваете не только модель, но и весь этот «пирог».

Многошаговый диалог: думайте о чате как о стажёре

Самая частая причина разочарования в ИИ — ожидание, что он поймёт задачу с одного сообщения. Пользователь пишет «сделай мне план продвижения», получает средненький ответ и бросает: «Не работает».

А теперь представьте, что вместо чата — новый стажёр. Умный, начитанный, но первый день в компании. Вы сказали ему «сделай план продвижения» и ушли пить кофе. Что он принесёт? Что-то общее из учебника. Виноват стажёр? Нет: вы не рассказали ему ни про продукт, ни про бюджет, ни про аудиторию.

Работа с чатом — это диалог со стажёром, и у него есть своя техника:

Дайте вводные. Кто вы, что за задача, для кого результат, какие ограничения. «Я веду кофейню в Казани, аудитория — студенты, бюджет 15 тысяч в месяц» меняет всё.

Итерируйте, а не переписывайте. Получили черновик — не удаляйте чат в сердцах. Скажите, что не так: «Второй пункт слишком общий, распиши его конкретнее. Тон сделай

легче». Каждая итерация использует уже накопленный контекст — это в разы эффективнее, чем начинать с нуля.

Провоцируйте уточняющие вопросы. Добавьте в запрос: «Если чего-то не хватает для хорошего ответа — спроси меня об этом». Хороший стажёр спрашивает; чат тоже умеет, если разрешить.

Фиксируйте договорённости. В длинной беседе полезно подводить итоги: «Давай зафиксируем: договор аренды берём за основу, пункты про штрафы переписываем, тон — деловой». Сводка в контексте — якорь против «забывания».

Чувствуйте край стола. Диалог стал длинным, модель начала путаться? Не мучайте её. Попросите: «Сожми всё, о чём мы договорились, в десять пунктов» — и начните новый чат с этой сводки.

Три стратегии промптинга

Теперь про приёмы, которые описаны в исследованиях и реально работают. Названия звучат по-английски, но за каждым — простая идея.

Zero-shot — «просто попроси». Вы даёте задание без примеров: «Переведи это письмо на деловой английский». Для большинства бытовых задач этого хватает — современные модели натренированы понимать инструкции. Начинайте всегда с этого: не усложняйте запрос без нужды.

Few-shot — «покажи на пальцах». Если нужен особый формат или стиль, не описывайте его словами — дайте два-три примера «вход → выход». Модель — чемпион по угадыванию закономерностей: покажите ей образец, и она продолжит в том же духе. Сравните:

Плохо: «Классифицируй обращения клиентов по тональности».

Лучше:

«Доволен скоростью, но курьер опоздал» → нейтральная

«Ужас! Второй раз привозят не то!» → негативная

«Спасибо, всё идеально» → позитивная

Теперь классифицируй: «Заказ пришёл быстро, правда, коробка помята» →

Модель схватывает принцип разметки из ваших примеров — включая ваши границы между категориями, которые словами вы бы долго объясняли. Термин пришёл из науки: ещё в 2020 году авторы GPT-3 показали, что модель способна учиться «на лету» от нескольких примеров прямо в запросе, без всякого дообучения.

Chain-of-thought — «думай вслух». Помните опыт с буквами «р» из главы 2? Модель, которая рассуждает по шагам, ошибается меньше. Это общий принцип: если задача требует расчёта или логики, попросите не готовый ответ, а ход мысли:

«Реши по шагам: сначала выпиши, что дано, потом рассуждай, и только в конце дай итоговую цифру».

Волшебное добавление «давай рассуждать шаг за шагом» заметно повышает точность на задачах с подвохом — это подтверждено исследованием Google 2022 года, которое и дало приёму имя. У современных «reasoning»-моделей режим размышления встроен: они сами думают вслух перед ответом. Но приём остаётся вашим инструментом для обычных моделей.

Роль и формат: два рычага, которые недооценивают

Роль (persona). Добавка «Ты — опытный юрист, проверяющий договор для малого бизнеса» меняет ответ заметно. Не потому, что модель «становится» юристом — а потому, что

роль сдвигает её в область текстов, где юристы разбирают договоры: появляются правильные термины, типовые риски, деловая осторожность. Помните правило из главы 2: ваш запрос — это начало текста, который модель продолжает. Роль делает это начало правильным.

Полезные роли — конкретные и с точкой зрения: «редактор, который ненавидит канцелярит», «учитель, объясняющий восьмикласснику», «скептический инвестор, ищущий дыры в плане». Роль-критик особенно недооценена: попросить модель найти слабые места в собственном ответе — мощный ход.

Формат вывода. Модель по умолчанию пишет «как принято»: средней длины текст с вступлением и выводом. Если вам нужна таблица — скажите «таблицей, три колонки». Нужен список для доклада — «пять тезисов, каждый до двенадцати слов». Нужен текст, который вы вставите в письмо, — «без заголовков и списков, связным текстом, не больше ста слов». Задать формат — это не занудство: вы избавляете себя от переделки ответа руками.

Температура и компания: ручки случайности

В большинстве чатов этих ручек не видно, но если вы работаете через API или в режиме разработчика (у OpenAI это playground, у других — аналоги), они там есть. Понимать их стоит, даже если никогда не тронете.

Вспомните главу 2: для каждого следующего слова модель вычисляет список кандидатов с вероятностями. Вопрос — как выбирать из списка.

Температура — ручка «смелости». При температуре 0 модель всегда берёт самого вероятного кандидата: ответы предсказуемые, сухие, надёжные. Для ответа на вопрос по документу — то, что надо. При высокой температуре модель чаще выбирает менее вероятные слова: тексты становятся разнообразнее и неожиданнее, но и риск чепухи растёт. Для мозгового штурма и стихов — хорошо, для выписок из договора — плохо.

Тор-р — ручка «размаха»: модель рассматривает только кандидатов, чья суммарная вероятность укладывается в заданный порог. При $\text{top-}p$ 0,1 — только топовые варианты, при 0,95 — почти весь список. Эффект похож на температуру, и крутить обе ручки сразу обычно не нужно: настройщики советуют выбрать одну.

Практическая памятка: факты и документы — температура низкая; идеи и тексты — средняя; когда «всё равно что, лишь бы небанально» — высокая.

Типичные ошибки пользователей

Соберём в одну кучу то, на чём спотыкается большинство. Вы узнаете свои привычки — все через это проходили.

Запрос-загадка. «Напиши текст про магазин». Какой магазин? Для кого текст? Зачем? Модель отвечает на тот вопрос, который прочитала, а не на тот, что у вас в голове. Лечение: вводные из раздела про стажёра.

Вера на слово. Модель выдала цифру и название закона — пользователь перенёс в документ как есть. Помните главу 2: уверенный тон ничего не значит. Лечение: факты — через проверку, всегда.

Один чат на всё. В беседе смешаны черновик статьи, рецепт борща и расчёт налогов. Контекст захламлён, модель путается. Лечение: новая задача — новый чат.

Капитуляция после первого ответа. «Не то» — и пользователь ушёл. Лечение: итерации. Второй и третий заход почти всегда лучше первого.

Борьба за «идеальный промпт». Обратная крайность: человек час шлифует один запрос вместо трёх быстрых итераций. Промпт — не заклинание, которое либо сработает, либо нет. Это начало разговора.

Ожидание личности. Модель вчера «поняла вас с полуслова», а сегодня в новом чате — «как чужая». Она и есть чужая: без переписки на столе она о вас ничего не знает. Лечение: пользовательские инструкции и сохранённые заготовки промптов.

Чат-боты в бизнесе: где уже работают

То, что вы осваиваете для себя, компании ставят на конвейер. Три зрелых сценария:

Поддержка. Самый массовый. В Т-Банке, по данным TAdviser, до 45 % обращений в клиентскую поддержку закрываются без участия оператора — благодаря языковым моделям. Громче всех заявила о себе компания Klarna: её ИИ-ассистент обрабатывает миллионы разговоров в месяц, а среднее время решения вопроса сократилось с 11 минут до двух. Устройство таких систем вам уже знакомо: системный промпт с правилами компании плюс подложенные в контекст документы (о механике — глава 8).

Консультации. Бот, который знает ассортимент, помогает выбрать, оформляет заказ. Здесь ключевое — интеграция с внутренними системами: без доступа к базе товаров консультант из модели получается красноречивый, но бесполезный.

Обучение. Корпоративные тренажёры: новичок общается с «клиентом», которого играет модель, и получает разбор своих реплик. Дёшево, безопасно, можно проиграть сложные случаи сто раз.

Общий вывод бизнеса совпадает с вашим личным опытом из этой главы: выигрывает не тот, у кого модель больше, а тот, кто лучше написал инструкцию, подложил правильные данные и продумал, когда бот должен уступить место человеку.

Попробовать

Двадцать минут, один чат. Соберём приёмы главы в боевой комплект.

Шаг 1. Обычный запрос. Попросите что-нибудь рабочее, но без подготовки:

Составь план статьи про домашние тренировки.

Сохраните ответ — он будет точкой отсчёта.

Шаг 2. Роль плюс вводные. Теперь то же самое, но по-взрослому:

Ты — редактор фитнес-журнала с десятилетним стажем. Составь план статьи о домашних тренировках для читателей 40+, у которых сидячая работа и нет инвентаря. План — из 6–8 пунктов, с рабочими заголовками, без банальностей вроде «больше двигайтесь».

Сравните с первым ответом. Обычно разница видна невооружённым глазом.

Шаг 3. Few-shot. Возьмите задачу со своим форматом. Например, вы ведёте чат кофейни и хотите подписи к фото в одном стиле. Дайте два своих примера и попросите продолжить:

Вот мои подписи:

«Понедельник оправдан: раф с лавандой снова в наличии».

«Эспрессо — это кофе, который не терпит суеты. Как и мы».

Напиши в том же стиле подпись про сезонный морс.

Шаг 4. Шаг за шагом. Дайте модели задачку с подвохом дважды. Сначала: «Ответь сразу: у фермера 17 овец, все, кроме девяти, сбежали. Сколько осталось?» Потом, в новом чате: «Реши по шагам, вслух: у фермера 17 овец, все, кроме девяти, сбежали. Сколько осталось?» Современные модели чаще всего справятся в обоих случаях, но на ваших реальных рабочих задачах — расчётах, условиях, датах — разница будет ощутимой. Запомните ощущение от двух режимов.

Шаг 5. Критик. Возьмите любой ответ модели и отправьте следом:

Теперь выступи придирчивым критиком: найди в своём ответе три слабых места и перепиши его с учётом замечаний.

Приём «модель против самой себя» — один из самых недооценённых.

Применить

Реальный публичный кейс — Klarna, шведский платёжный сервис. В начале 2024 года компания запустила ИИ-ассистента в поддержке и отчиталась о результатах: ассистент вёл миллионы разговоров в месяц, а среднее время решения обращения упало с 11 минут до двух.

Но для нас ценность кейса не в круглых цифрах, а в устройстве. Смотрите на него глазами этой главы:

- У ассистента есть жёсткая системная инструкция: кто он, что может, что не может, когда передаёт человеку. Это `system prompt`, только промышленного размера.
- Ему подкладывают в контекст не абы что, а данные конкретного клиента и регламенты — отсюда точные ответы вместо общих слов.
- Его ответы шлифовались итерациями на реальных диалогах — та самая техника «скажи, что не так», только в масштабе компании.

И ещё одно наблюдение, менее радужное: спустя время Klarna скорректировала курс и вернула в поддержку больше людей, признав, что качество важнее полной автоматизации. Урок, который стоит дороже цифр: чат-бот — не замена человека, а фильтр перед ним. Типовое закрывает машина, нестандартное и чувствительное остаётся людям.

Тот же принцип работает и в вашем личном масштабе: отдавайте чату то, что он делает быстро и без риска, а себе оставляйте то, где цена ошибки высока.

Мини-проект главы

Постройте свой первый диалоговый сценарий — по образцу тех, что стоят в бизнес-ботах, только для себя.

Выберите задачу, которую будете регулярно поручать ИИ: разбор входящих писем, подготовка к совещаниям, помощь с текстами — что угодно из вашего «паспорта задачи» из главы 1. И оформите сценарий из четырёх блоков:

Инструкция. Напишите system prompt для этой задачи: роль, правила, запреты, формат ответа. Пять-десять строк. Сохраните — пригодится в пользовательских инструкциях сервиса.

Стартовый промпт. Первое сообщение с вводными: кто вы, что за задача, ограничения.

Крючки итераций. Заготовьте три фразы для второго захода: «сделай короче», «добавь конкретики в пункт...», «выступи критиком».

Правило стопа. Запишите, при каком результате вы остановитесь и заберёте текст — чтобы не шлифовать бесконечно.

Прогоните сценарий дважды на реальных задачах и допишите в него то, чего не хватило. Поздравляю: вы только что сделали то, за что компании платят внедренцам, — оформили работу с ИИ как повторяемый процесс, а не лотерею.

Дальше — глава 4. Пока наш собеседник только отвечает. А что, если дать ему цель, инструменты и разрешение действовать самостоятельно? Получится агент — и вот тут начинается самое интересное.

Глава 4. Что такое агент

Начнём с эксперимента

Откройте привычный чат с ИИ и напишите:

Найди три авиабилета Москва — Казань на ближайшую пятницу, сравни цены и забронируй самый дешёвый.

Что произошло? Модель вежливо объяснила, что не может ни искать в реальном времени, ни бронировать. Она умеет говорить — но не умеет делать. Она советник, запертый в комнате без окон и без рук.

А теперь представьте ту же просьбу, но у модели есть: выход в интернет, экран браузера, ваша банковская карта и разрешение действовать. Она сама откроет сайты, сравнивает варианты, выберет, оплатит — и вернётся с готовым билетом. Это уже не чат. Это агент.

Именно к агентам индустрия шла последние пару лет, и именно они — главная тема 2026 года. В этой главе разберём, из чего агент состоит, как он работает, почему за ним нужен глаз да глаз и как не повторить чужие дорогие ошибки.

Понять

От советчика к исполнителю

Разница между чат-ботом и агентом — как между консультантом в магазине и курьером с полномочиями. Консультант расскажет, какой товар выбрать. Курьер возьмёт деньги, съездит, купит и привезёт. Один говорит, другой действует.

Четыре признака отличают агента:

Цель. Чат отвечает на сообщение. Агент получает задачу — «забронируй», «найди», «организуй» — и гонится за результатом, а не за ответом.

Планирование. Задачу нельзя решить одним ответом, поэтому агент раскладывает её на шаги: открыть сайт, ввести даты, сравнить, выбрать, оплатить. Как стажёр из главы 3, но с чек-листом, который он составляет сам.

Инструменты. Это «руки» агента: поиск в интернете, браузер, калькулятор, доступ к файлам, запуск программ, обращение к сервисам через API. Модель умеет только писать текст — но она может написать «команду» для инструмента, а дальше инструмент действует в реальном мире.

Память. Агент помнит, что уже сделал на прошлых шагах, какие сайты не сработали, что вы предпочитаете утренние рейсы. Без памяти он ходил бы по кругу.

Из чего собран агент

Заглянем под капот. Агент — это не новая сверхмодель, а конструкция из знакомых деталей.

Мозг — обычная языковая модель. Та самая угадка из главы 2. Она читает текущую ситуацию и решает, что делать дальше.

Инструменты (tools). У каждого инструмента есть описание: что делает, как вызвать, что вернёт. Модель пишет в ответе что-то вроде «вызвать поиск: билеты Москва—Казань пятница», система выполняет вызов и возвращает модели результат. Модель читает его и решает следующий шаг.

Память. Контекстное окно плюс, в серьёзных системах, внешние хранилища: заметки о задаче, файлы, база прошлых действий.

Цикл выполнения. Сердце агента — петля: посмотрел на ситуацию → подумал → сделал шаг инструментом → посмотрел на результат → снова подумал. И так, пока цель не достигнута или не кончились попытки. Чат отвечает один раз; агент крутится в этом цикле десятки шагов подряд, сам.

Почему система важнее модели

В 2023–2024 годах все спорили, чья модель умнее. К 2026-му спор сместился: выяснилось, что на одной и той же модели один агент работает отлично, а другой — барахтается. Разница не в мозге, а во всём, что вокруг него: какие инструменты подключены, как описаны, что делать при ошибке, когда остановиться и спросить человека.

Эта работа — собирать надёжные системы вокруг моделей — называется оркестрацией. Представьте дирижёра: музыкант (модель) один и тот же, но оркестр с дирижёром играет, а без него — шумит. Поэтому в индустрии закрепились мысли: системы вокруг моделей становятся важнее самих моделей. Глава 17 покажет, куда этот сдвиг ведёт.

Три типовых устройства агентов

Инженеры перепробовали много схем; три устоялись и встречаются повсюду.

ReAct — «думай и делай по очереди». Агент чередует рассуждение и действие: подумал вслух («мне нужна цена билета») → вызвал инструмент → прочитал результат → подумал дальше. Схема описана в исследовании 2022 года и до сих пор остаётся рабочей лошадкой: простая, прозрачная, легко посмотреть, где агент свернул не туда.

Plan-and-Execute — «сначала план, потом дело». Агент сначала составляет весь план целиком, вы показываете ему «да», и только потом он исполняет шаги. Медленнее на старте, зато предсказуемое: вы видите маршрут до того, как он пошёл тратить время и деньги.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.