KROK AI

# AI для новичков

Практическое руководство по работе с нейросетями

16 уроков · 128 вопросов · практическое руководство

Электронное издание

Krok AI

KROK AI

# AI для новичков

Практическое руководство по работе с нейросетями

16 уроков · 128 вопросов · практическое руководство

# Содержание книги


1 Как думает нейросеть: токены, предсказание, контекстное окно

2 Первый рабочий промпт: задача + формат + ограничения

3 Итерации вместо перфекционизма

4 Роли, аудитория и тон: смена перспективы

5 Роль ≠ магия. Шаблон-инструкция: промпт с пропусками

6 Few-shot промптинг: обучение модели на примерах

7 Структурированный вывод: списки, таблицы, JSON, схемы

8 Цепочки шагов: декомпозиция и самопроверка

9 Тексты: письма, посты, статьи — от плана до вычитки

10 Длинные документы: тезисы, извлечение фактов, переработка

11 Изображения и презентации: формула промпта и структура слайдов

12 ИИ как учитель: объяснения, тренажёр, персональный план

13 Таблицы и данные без программирования

14 Автоматизация рутины: от поиска задач до связок без кода

15 Галлюцинации и проверка фактов

16 Безопасность и приватность: чек-лист перед отправкой

УРОК 1

# Как думает нейросеть: токены, предсказание, контекстное окно

Начнём с главного: нейросеть не «знает» ничего и не «думает» как человек. Она — очень мощная машина предсказания следующего слова. Когда вы пишете «Небо над городом было…», модель перебирает миллионы вариантов и выбирает наиболее вероятное продолжение: «серым», «ясным», «в дымке». Весь текст, который она генерирует, собирается по одному такому предсказанию за раз.

Модель работает не с буквами, а с токенами — кусочками текста. Одно слово обычно состоит из 1–3 токенов: «привет» может быть одним токеном, а «предсказание» распадётся на два-три. Именно поэтому модель иногда странным образом ошибается в подсчёте букв в слове: она видит не буквы, а целые куски. Это не глупость, а особенность устройства.

У модели есть контекстное окно — рабочая память, куда помещаются и ваши сообщения, и её ответы. В современных чатах это десятки и сотни страниц текста. В пределах одного чата модель «помнит» всё, что вы обсуждали. Но откройте новый чат — и она забудет всё: имя, задачу, договорённости. Поэтому важные детали либо держите в одном диалоге, либо заново вставляйте в промпт.

Модель не умеет говорить «я не знаю» так честно, как хотелось бы. Её задача — предсказать правдоподобное продолжение, а не проверить его на истинность. Если вы спросите несуществующую книгу, модель с высокой вероятностью придумает правдоподобное название, автора и год. Тон при этом будет такой же уверенный, как при ответе на простой вопрос. Это называется галлюцинацией, и с ней мы подробно разберёмся в уроке 15.

Практический вывод: относитесь к нейросети как к очень начитанному, но слегка самонадеянному помощнику. Она отлично пишет, объясняет, переформулирует и структурирует. Но цифры, даты, цитаты, ссылки и названия стоит перепроверять — особенно если от них зависит решение.

**ПРИМЕР ПРОМПТА**

```
«У меня есть вопрос об историческом событии. Если ты не уверен в точных датах или деталях — прямо скажи об этом, а не придумывай. Вопрос: когда открыли первый супермаркет в России?»
```

**ПРИМЕР ПРОМПТА**

```
«Объясни, как работает контекстное окно нейросети, так, будто я вообще не разбираюсь в технологиях. Максимум 5 предложений, без терминов вроде "эмбеддинг" и "инференс".
```

> 💡 Если ответ выглядит уверенно, но для вас важен — считайте его черновиком до

проверки. Уверенный тон модели — это стиль генерации, а не признак истины.

---

## ТЕСТ

**Чем является токен для нейросети?**

- ○ Целым предложением
- ○ Одной буквой текста
- ● **Кусочком текста, из которых состоит слово ✓**

**Почему так:** Токен — это фрагмент слова: слово обычно распадается на 1–3 токена, поэтому модель может ошибаться, например, при подсчёте букв.

**Что произойдёт с контекстом, если открыть новый чат?**

- ○ Модель запомнит разговор, но покажет позже
- ● **Модель начнёт с чистого листа и забудет всё ✓**
- ○ Модель перенесёт память из старого чата

**Почему так:** Память модели ограничена контекстным окном одного диалога; новый чат — это новая, пустая память.

**Почему нейросеть может уверенно назвать несуществующую книгу?**

- ● **Её задача — предсказать правдоподобный текст, а не проверять факты ✓**
- ○ Она не знает ни одной книги вообще
- ○ Она хочет вас обмануть специально

**Почему так:** Модель генерирует вероятностное продолжение. Уверенный тон — это стиль генерации, а не признак достоверности.

**Что помещается в контекстное окно?**

- ○ Все чаты пользователя за всё время
- ○ Только последнее сообщение пользователя
- ● **Ваши сообщения и ответы модели в текущем диалоге ✓**

**Почему так:** Контекстное окно — рабочая память одного диалога: и ваши сообщения, и ответы модели.

**Модель плохо считает количество букв «р» в слове «сверхдраматический». Почему?**

- ○ Она всегда ошибается в длинных словах

- [x] **Она видит токены-куски, а не отдельные буквы ✓**
- [ ] Она не знает русский язык

**Почему так:** Модель оперирует токенами, а не буквами, поэтому побуквенные задачи ей даются плохо.

**Чат засорился старыми обсуждениями, и ответы стали хуже. Что разумно сделать?**

- [x] **Начать новый чат и кратко пересказать задачу ✓**
- [ ] Написать капсом, что старое не важно
- [ ] Попросить модель «забыть» прошлые сообщения

**Почему так:** Новый чат с кратким пересказом очищает контекст от мусора. «Забудь» не удаляет сообщения из окна.

**Какой вывод про уверенный тон модели верен?**

- [ ] Модель всегда пишет неуверенно, если сомневается
- [ ] Уверенный тон означает, что модель проверила факт
- [x] **Уверенный тон — просто результат генерации, а не гарантия истины ✓**

**Почему так:** Тон задан обучением на уверенных текстах; он ничего не говорит о правильности ответа.

**Что из перечисленного нейросеть делает обычно хорошо?**

- [ ] Точно называет сегодняшнюю дату по памяти
- [x] **Переформулирует, объясняет и структурирует текст ✓**
- [ ] Гарантирует актуальность ссылок

**Почему так:** Работа с текстом — сильная сторона модели. Даты, ссылки и актуальные факты — типичные места ошибок.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Проведите два мини-эксперимента, чтобы почувствовать устройство модели. Эксперимент 1: в новом чате спросите «Сколько букв "о" в слове "молокоотсепараторный"?» и проверьте ответ вручную. Эксперимент 2: в одном чате задайте тему (например, «я планирую поездку в Казань»), напишите 3 сообщения, затем откройте новый чат и спросите, о чём вы говорили.**

### ЧЕК-ЛИСТ

- ☐ Выполнен эксперимент с подсчётом букв, ответ проверен вручную и записан

- ☐ В первом чате модель действительно использовала ранее упомянутые детали
- ☐ В новом чате модель подтвердила, что не знает о прошлом разговоре
- ☐ Записаны 1–2 собственных наблюдения о том, как «память» модели изменит ваши привычки в работе

УРОК 2

# Первый рабочий промпт: задача + формат + ограничения

Промпт — это просто задание, которое вы даёте модели. Разница между «напиши текст про кофе» и хорошим промптом — как разница между «сделай что-нибудь» и нормальным техническим заданием. Рабочий промпт строится из трёх блоков: задача, формат, ограничения.

Задача — что конкретно нужно сделать и зачем. Формат — как должен выглядеть результат: объём, структура, тип (список, таблица, письмо). Ограничения — рамки: для кого текст, чего избегать, какой стиль. Дополнительно полезно дать контекст — немного информации о вашей ситуации, чтобы модель не додумывала её сама.

Соберите промпт по схеме: (1) что сделать, (2) в каком виде, (3) с какими рамками. Например: «Составь список из 7 идей подарка коллеге на день рождения. Для каждой идеи — название и одна строка с объяснением. Бюджет до 3000 рублей, человек не любит кофе и спортивные товары». Один промпт — три блока, и результат сразу в разы точнее.

**ПРИМЕР ПРОМПТА**

```
«Напиши письмо клиенту с извинением за задержку доставки на 2 дня. Объём 3–4 предложения, тон вежливый и деловой, без излишних оправданий. В конце предложи скидку 5% на следующий заказ. Имя клиента — Ольга».
```

**ПРИМЕР ПРОМПТА**

```
«Объясни разницу между ипотекой и арендой жилья человеку, который никогда этого не сравнивал. Формат: таблица из 5 строк — критерий, ипотека, аренда. Критерии: первоначальный взнос, ежемесячные платежи, гибкость, накопление активов, риски. Без финансовых советов, только сравнение».
```

Первая — нечёткая задача: «напиши что-нибудь про здоровье». Вторая — отсутствие формата: модель выбирает сама, и почти всегда не так, как вам нужно. Третья — противоречивые требования: «подробно, но в двух предложениях». Четвёртая — ожидание, что модель угадает контекст: не сказано, для кого текст и куда он пойдёт. Пятая — одна попытка вместо итерации: получив не то, человек бросает, хотя достаточно было уточнить.

Хорошая привычка: перед отправкой перечитайте промпт глазами исполнителя, который видит текст впервые. Всё ли понятно? Что бы вы спросили на месте модели? Всё, что вы на месте модели хотели бы уточнить, — добавьте в промпт сразу.

> 💡 Держите под рукой личный шаблон: «Задача: … Формат: … Ограничения: … Контекст: …». Заполните поля — промпт готов.

## ТЕСТ

**Из каких трёх блоков состоит базовый рабочий промпт?**

- ○ Приветствие, вопрос, прощание
- ● **Задача, формат, ограничения ✓**
- ○ Тема, объём, дедлайн

**Почему так:** Рабочий промпт отвечает на три вопроса: что сделать, в каком виде, в каких рамках. Контекст добавляется по необходимости.

**«Напиши текст про пользу спорта» — в чём главная проблема промпта?**

- ● **Нет формата и рамок: объём, аудитория, стиль не заданы ✓**
- ○ Модель не знает слова «польза»
- ○ Слишком короткое слово «спорт»

**Почему так:** Без формата и ограничений модель сама выбирает объём и стиль — и почти наверняка промахнётся.

**Что относится к блоку «формат»?**

- ○ «Текст для внутреннего чата команды»
- ○ «Не используй канцелярит»
- ● **«Сделай таблицу из 4 колонок» ✓**

**Почему так:** Формат — это вид результата: таблица, список, объём. Стилевые запреты — ограничения, назначение — контекст.

**Какая из ошибок — противоречивое требование?**

- ○ «Используй список»
- ● **«Подробно раскрой тему, но уложись в 2 предложения» ✓**
- ○ «Напиши письмо для клиента»

**Почему так:** Подробность и жёсткий лимит в 2 предложения конфликтуют; модель вынуждена выбирать, и вы не получите предсказуемый результат.

**Зачем в промпте контекст о ситуации?**

- ● **Чтобы модель не додумывала вашу ситуацию сама ✓**
- ○ Контекст нужен только программистам
- ○ Чтобы модель отвечала дольше

**Почему так:** Без контекста модель строит предположения; пара строк о ситуации резко повышает точность ответа.

**Вы получили «не совсем то», но направление верное. Что разумнее всего?**

- ○ Смириться и редактировать всё вручную
- ○ Начать новый чат с нуля
- ● **Уточнить промпт: что именно изменить в ответе ✓**

**Почему так:** Если направление верное — это задача для итерации: попросите скорректировать конкретные места.

**Какой промпт написан по формуле «делай → в виде → с учётом»?**

- ○ «Расскажи про инвестиции»
- ● **«Объясни простыми словами, что такое вклад, 3 предложения, для подростка, без рекламы банков» ✓**
- ○ «Напиши хорошо и красиво про деньги»

**Почему так:** Есть задача (объяснить вклад), формат (3 предложения) и ограничения (для подростка, без рекламы).

**Что стоит сделать перед отправкой промпта?**

- ○ Проверить промпт антивирусом
- ○ Добавить как можно больше слов, чтобы модель точно поняла
- ● **Перечитать его глазами исполнителя, который видит задачу впервые ✓**

**Почему так:** Чтение «глазами модели» выявляет недостающие детали — их лучше добавить сразу, чем потом итерациями.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите реальную задачу из вашей жизни или работы (письмо, список покупок, объяснение темы ребёнку, пост для соцсетей). Напишите промпт по формуле «задача + формат + ограничения + контекст», получите ответ. Затем найдите у результата минимум 2 недостатка и уточните промпт один раз.**

### ЧЕК-ЛИСТ

- ☐ Задача взята из вашей реальной жизни, а не выдуманная
- ☐ Промпт явно содержит блоки «задача», «формат», «ограничения»

- ☐ Найдены и записаны минимум 2 недостатка первого ответа
- ☐ Второй промпт-уточнение заметно улучшил результат — сохраните обе версии для сравнения

УРОК 3

# Итерации вместо перфекционизма

Новичок стремится написать идеальный промпт с первого раза и расстраивается, когда не выходит. Опытный пользователь работает иначе: первый ответ — это черновик, а промпт — это диалог. Средняя удачная работа с моделью — это 2–4 уточнения, и это норма, а не провал.

Ключевая мысль: не переписывайте промпт заново, а правьте результат. Модель помнит весь диалог, поэтому достаточно коротких команд: «сделай короче в два раза», «добавь конкретный пример во второй пункт», «перепиши тон теплее, без канцелярита». Так вы с каждым шагом приближаетесь к цели, а не начинаете сначала.

Полезные универсальные команды: «сделай проще» (уходят сложные слова), «дай 3 варианта» (появляется выбор), «конкретнее, с примерами» (уходит вода), «сохрани структуру, измени только третий пункт» (точечная правка), «что ты не учёл из моего запроса?» (самопроверка модели). Запомните три-четыре из них — и 80% ситуаций закрыты.

**ПРИМЕР ПРОМПТА**

«Хорошо, но сделай короче: оставь только главную мысль в каждом абзаце, максимум 3 предложения на абзац. Стиль и структуру не меняй».

**ПРИМЕР ПРОМПТА**

«Дай 3 разных варианта этого письма: первый — деловой и сдержанный, второй — тёплый и личный, третий — очень короткий, 2 предложения. Пронумеруй их».

Ещё один приём — попросить модель самокритику: «Оцени свой ответ: что в нём слабо и что стоит улучшить?» Часто модель сама находит воду, пропущенные детали и слабые места — а вы решаете, что править.

Итерации хороши, когда направление верное. Но бывает, что ответ «уплыл»: вы уже 10 сообщений правите текст, и он стал хуже, чем был. Или задача изменилась по сути. Или модель «застряла» на неудачном варианте, и каждая правка тянет за собой старые огрехи. Это сигналы открыть новый чат: сформулируйте задачу заново с учётом того, что вы поняли, — и потратьте 2 промпта вместо 10.

> 💡 Если правка нужна только в одном месте — прямо так и скажите: «измени только заголовок, остальное оставь как есть». Иначе модель может переписать весь текст.

---

## ТЕСТ

**Какой стиль работы с моделью обычно самый результативный?**

- [x] **Относиться к первому ответу как к черновику и уточнять ✓**
- [ ] Принимать любой первый ответ как финальный
- [ ] Написать идеальный промпт с первого раза

**Почему так:** 2–4 итерации — норма. Первый ответ задаёт направление, уточнения доводят его до цели.

**Модель помнит предыдущие сообщения в чате. Что из этого следует?**

- [ ] Модель не примет уточнение после первого ответа
- [ ] Нужно каждый раз переписывать промпт с нуля
- [x] **Достаточно коротких уточнений типа «сделай короче» ✓**

**Почему так:** Пока диалог в контексте, работают короткие команды правки результата без пересказа задачи.

**Что делает команда «сделай проще»?**

- [ ] Удаляет половину текста наугад
- [x] **Убирает сложные слова и упрощает изложение ✓**
- [ ] Меняет тему текста

**Почему так:** Это универсальное уточнение про упрощение языка; объём и структуру задают отдельными командами.

**Вам нужен выбор из нескольких подходов. Какое уточнение подойдёт?**

- [x] **«Дай 3 варианта: сдержанный, тёплый и очень короткий» ✓**
- [ ] «Напиши ещё раз то же самое»
- [ ] «Ответь да или нет»

**Почему так:** Просьба о пронумерованных вариантах с разными характеристиками даёт материал для выбора и гибрида.

**Зачем просить модель «оцени свой ответ, что слабо»?**

- [ ] Это не работает никогда
- [ ] Чтобы она удалила ответ
- [x] **Она часто сама находит воду и слабые места, а вы решаете, что править ✓**

**Почему так:** Самокритика — быстрый способ выявить недостатки ответа без внешнего редактора.

**Какой из сигналов говорит, что пора открыть новый чат?**

- [ ] Первый ответ не понравился

- [x] **Вы уже 10 сообщений правите текст, и он стал хуже ✓**
- [ ] Вы хотите задать ещё один уточняющий вопрос

**Почему так:** Когда правки тянут за собой старые огрехи или задача изменилась — новый чат с кратким пересказом выгоднее.

**Нужно изменить только заголовок письма. Как сформулировать правку?**

- [ ] «Начни диалог заново»
- [ ] «Перепиши письмо»
- [x] **«Измени только заголовок на ..., остальное оставь как есть» ✓**

**Почему так:** Явное указание «остальное не трогай» защищает готовые части текста от ненужных переписываний.

**Что делает команда «прежде чем отвечать, задай мне 3 уточняющих вопроса»?**

- [ ] Модель отказывается отвечать
- [x] **Модель собирает недостающие детали до выполнения, снижая риск промаха ✓**
- [ ] Модель задаёт вопросы после ответа

**Почему так:** Так вы передаёте модели право собирать требования — это экономит несколько раундов правок.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите любую задачу (лучше ту же, что в уроке 2). Выполните минимум три итерации: первый ответ → уточнение формата («сделай короче/проще») → уточнение содержания («добавь пример/конкретику») → при желании запрос трёх вариантов. Затем сравните финальную и первую версии и запишите, что изменилось.**

### ЧЕК-ЛИСТ

- ☐ Сделано минимум 3 итерации в одном чате
- ☐ Использованы минимум 2 разных типа уточнений (про формат и про содержание)
- ☐ Финальная версия сохранена рядом с первой — видно улучшение
- ☐ Записан вывод: какие уточнения сработали именно у вас лучше всего

УРОК 4

# Роли, аудитория и тон: смена перспективы

Один из самых мощных приёмов — задать модели роль и аудиторию. Роль — это «кем» она будет: «ты опытный редактор», «ты вежливый менеджер поддержки». Аудитория — «для кого» результат: для ребёнка, для директора, для соседа, который ничего не смыслит в теме. Эти две строчки меняют словарь, глубину и тон ответа сильнее, чем десять уточнений стиля.

Почему это работает: в обучающих текстах у каждого типа писателя и читателя свой характерный язык. Роль и аудитория «включают» нужный слой. «Ты налоговый консультант, объясни вычет директору небольшой фирмы» и «...объясни студенту первокурснику» дадут два очень разных текста.

Тот же вопрос, заданный под разными аудиториями, — отличный инструмент понимания. Сначала попросите «объясни, как 10-летнему» — получите суть без жаргона. Затем «объясни, как эксперту для эксперта» — увидите настоящую терминологию и тонкости. Разница между двумя ответами сама показывает, где в теме лежат сложные места.

**ПРИМЕР ПРОМПТА**

```
«Ты — редактор популярного онлайн-журнала. Проверь мой текст: исправь ошибки, убери канцелярит, сохрани мой стиль и смысл. Не переписывай с нуля, покажи правки списком: было → стало».
```

Тон задаётся словами-образцами: «дружелюбно, как в переписке с хорошим знакомым», «официально, как в письме банка», «живо, как у хорошего лектора». Чем конкретнее образ, тем точнее тон. Можно дать и негативное ограничение: «без восклицательных знаков», «без слов "уникальный" и "инновационный"».

И важный приём из урока 3, который здесь особенно уместен: «прежде чем отвечать, задай уточняющие вопросы». Когда вы дали роль («ты мой нутрициолог»), модель имеет право выяснить контекст — цели, ограничения, привычки. Хороший «исполнитель» сначала спрашивает, а не бросается писать.

**ПРИМЕР ПРОМПТА**

```
«Ты — нутрициолог, работающий с занятыми людьми. Прежде чем дать рекомендации по питанию на неделю, задай мне 3 уточняющих вопроса о моих целях, ограничениях и привычках. Дождись моих ответов».
```

> 💡 Роль — не магия и не универсальный усилитель. Она настраивает стиль и уровень детализации, но не даёт модели новых знаний или доступа к вашей информации. Подробнее — в уроке 5.

> 💡 Комбинируйте: роль + аудитория + тон + формат в одном промпте. Например: «Ты редактор… Для новичков… Дружелюбно… Список из 5 пунктов…».

---

## ТЕСТ

**Что сильнее всего влияет на словарь и тон ответа?**

- ○ Время суток
- ○ Длина промпта
- ● **Заданные роль и аудитория ✓**

**Почему так:** Роль задаёт «голос» писателя, аудитория — уровень языка читателя. Вместе они перестраивают весь текст.

**«Объясни, как 10-летнему» — зачем этот приём?**

- ○ Он уменьшает длину ради экономии
- ● **Он убирает жаргон и показывает суть темы простыми словами ✓**
- ○ Он нужен только для детей

**Почему так:** Детский уровень вырезает терминологию и обнажает суть — полезно и для понимания, и как основа для объяснений другим.

**Что даёт сравнение объяснений «для новичка» и «для эксперта»?**

- ● **Видно, где в теме лежат сложные места и терминология ✓**
- ○ Второе объяснение всегда правильнее
- ○ Ничего, это одно и то же

**Почему так:** Разница между двумя уровнями показывает, какие детали требуют подготовки, а что можно говорить просто.

**Какая строка задаёт тон, а не роль?**

- ○ «Аудитория — студенты»
- ○ «Ты опытный юрист»
- ● **«Дружелюбно, как в переписке с хорошим знакомым» ✓**

**Почему так:** Тон — это окраска текста (образцы и запреты), роль — кем модель выступает, аудитория — для кого.

**Правда ли, что роль «эксперт по безопасности» делает модель лучше в защите данных?**

- ○ Да, роль добавляет новые знания
- ● **Нет: роль настраивает стиль и глубину, но не даёт новых знаний ✓**
- ○ Да, но только в платных тарифах

**Почему так:** Роль активирует нужный слой текстов, но не добавляет модели возможностей или информации.

**Зачем просить модель задать уточняющие вопросы после назначения роли?**

- ○ Чтобы проверять вашу грамотность
- ○ Чтобы потянуть время
- ● **Чтобы исполнитель собрал контекст до работы и не додумывал сам ✓**

**Почему так:** Хороший исполнитель выясняет требования до старта: это заменяет несколько раундов переделок.

**Какая формулировка — негативное ограничение тона?**

- ○ «Пиши тепло и живо»
- ● **«Без восклицательных знаков, без слов "уникальный" и "инновационный"» ✓**
- ○ «Ты редактор журнала»

**Почему так:** Негативные ограничения прямо запрещают конкретные элементы — их легко проверить глазами после ответа.

**Модель отвечает слишком сухо и официально. Какое уточнение точнее всего?**

- ● **«Перепиши дружелюбнее, как в переписке с хорошим знакомым, смысл не меняй» ✓**
- ○ «Отвечай короче»
- ○ «Пиши лучше»

**Почему так:** Слово-образец («как в переписке со знакомым») даёт модели конкретный ориентир стиля.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Выберите тему, которую вам реально нужно объяснить кому-то (родственнику, коллеге, ребёнку, клиенту). Получите от модели три версии объяснения: как 10-летнему, как обычному взрослому без подготовки и как эксперту. Из трёх версий соберите одну финальную (можно гибрид) под вашего конкретного слушателя.**

### ЧЕК-ЛИСТ

- ☐ Тема выбрана под реального человека из вашей жизни

- ☐ Получены все 3 версии с разными уровнями аудитории
- ☐ Записано, чем версии различаются (это карта сложных мест темы)
- ☐ Финальная версия собрана под вашего слушателя и проверена: понятно ли ему

УРОК 5

# Роль ≠ магия. Шаблон-инструкция: промпт с пропусками

После урока 4 легко решить: «назначу модель шеф-поваром — получу ужин». Нет. Роль — это линза, а не волшебная палочка. Модель с ролью «юрист» отвечает на языке юриста, но не проверяет актуальные законы и не несёт ответственности. Роль не даёт доступа к вашим документам, не помнит прошлого чата и не заменяет специалиста в серьёзных вопросах — деньги, здоровье, право.

Зато роли отлично решают другую задачу: экономят ваши объяснения. Если вы раз в день пишете посты в одном стиле, вместо пересказа стиля каждый раз можно зафиксировать его в роли: «Ты — автор коротких постов для Telegram-канала маленькой пекарни. Тёплый тон, короткие фразы, всегда одна конкретная деталь из жизни пекарни».

Паттерн «шаблон» — самый практичный инструмент для повторяющихся задач. Вы один раз пишете инструкцию с пропусками в фигурных скобках, а потом меняете только значения. Модель подставляет значения и выполняет инструкцию. Фигурные скобки важны: они явно показывают, где переменная, а где обычный текст.

**ПРИМЕР ПРОМПТА**

```
«Ты — автор писем поддержки интернет-магазина. Шаблон: "Здравствуйте, {имя}! Ваш заказ № {номер} {статус}. {Действие клиента}. Если есть вопросы — просто ответьте на это письмо." Подставь: имя = Анна, номер = 45128, статус = передан в службу доставки сегодня, действие = придёт 12 ноября, отслеживать можно по ссылке в личном кабинете».
```

**ПРИМЕР ПРОМПТА**

```
«Ты готовишь пост для Telegram-канала кофейни по шаблону: "Тема: {тема}. Зацепка: {вопрос или факт}. Польза: {что читатель узнает}. Призыв: {что сделать}". Заполни шаблон для темы "новый сезонный напиток — облепиховый раф"».
```

У шаблона три больших плюса. Первый: стабильность — результат одинаково устроен от раза к разу. Второй: скорость — заполняете три поля вместо пересказа задачи. Третий: делегирование — шаблон можно передать коллеге, и он получит тот же результат без обучения.

Шаблоны стоит копить в отдельном файле: шаблон письма клиенту, шаблон поста, шаблон разбора задачи, шаблон ответа на отзыв. Это ваша личная «аптечка» типовых задач. Со временем вы начнёте видеть шаблонную структуру в любой повторяющейся работе — это самый заметный скачок в эффективности.

> 💡 Не усложняйте шаблон сразу: начните с 2–3 переменных. Если переменных больше пяти, заполнение само становится медленным.

> 💡 Храните шаблоны с реальными значениями отдельно от шаблонов с пропусками — и никогда не подставляйте в публичный бот личные данные клиентов.

---

## ТЕСТ

**Что на самом деле даёт модели назначенная роль?**

- ○ Новые знания и доступ к интернету
- ● **Переключение стиля, словаря и уровня детализации ✓**
- ○ Гарантию профессионального качества ответа

**Почему так:** Роль — это линза стиля. Она не добавляет знаний и не заменяет реального специалиста.

**В каких вопросах роль «ваш юрист» не заменяет настоящего юриста?**

- ● **В серьёзных решениях: право, деньги, здоровье ✓**
- ○ Нигде, роль полностью заменяет специалиста
- ○ В написании приветствия

**Почему так:** Модель без проверки фактов и ответственности не должна быть финальной инстанцией в критичных вопросах.

**Что означает {переменная} в шаблонном промпте?**

- ○ Спецкоманду для интернета
- ○ Системную ошибку модели
- ● **Место, куда вы подставляете конкретное значение ✓**

**Почему так:** Фигурные скобки помечают пропуск: каждый раз вы меняете только значения, инструкция остаётся той же.

**Главный плюс шаблона-инструкции?**

- ○ Модель отвечает технически быстрее
- ● **Стабильный, повторяемый результат при минимальных усилиях ✓**
- ○ Шаблоны работают только у программистов

**Почему так:** Шаблон даёт повторяемость и экономию времени, а также легко передаётся коллегам.

**Сколько переменных разумно иметь в первом шаблоне?**

- ○ Не меньше 10 для точности

- ○ Число переменных не имеет значения
- ● **1–3, чтобы заполнение было быстрым ✓**

**Почему так:** Слишком много полей замедляет работу; начните с малого и добавляйте по необходимости.

**Какой из примеров — шаблонный промпт?**

- ○ «Напиши письмо»
- ● **«Шаблон: "Здравствуйте, {имя}! Заказ №{номер} {статус}". Подставь: Анна, 45128, доставляется» ✓**
- ○ «Ты юрист»

**Почему так:** Есть инструкция с пропусками и конкретные значения для подстановки.

**Почему шаблоны удобно хранить в отдельном файле?**

- ● **Это личная библиотека: повторяющиеся задачи закрываются за секунды ✓**
- ○ Без файла шаблоны не работают
- ○ Файлы ускоряют нейросеть

**Почему так:** Накопленные шаблоны — главный персональный актив: они экономят время и стандартизируют работу.

**Вы пишете еженедельную сводку для руководителя в одном формате. Что сделать один раз, чтобы экономить время?**

- ○ Попросить модель запомнить формат навсегда
- ○ Каждый раз просить модель «напиши сводку»
- ● **Зафиксировать шаблон со структурой и переменными и заполнять его еженедельно ✓**

**Почему так:** «Запомни навсегда» не работает между чатами; шаблон решает задачу надёжно.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Найдите в вашей работе или жизни задачу, которую вы делаете минимум раз в неделю в похожем виде (письмо, отчёт, пост, ответ на типовой вопрос). Создайте под неё шаблон-инструкцию с 2–4 переменными в фигурных скобках и проверьте его на трёх разных наборах значений.**

### ЧЕК-ЛИСТ

- ☐ Задача повторяется минимум раз в неделю

- ☐ Шаблон содержит инструкцию и 2–4 переменные в фигурных скобках
- ☐ Шаблон проверен на 3 разных наборах значений — результаты одинаково устроены
- ☐ Шаблон сохранён в отдельный файл с понятным названием

УРОК 6

# Few-shot промптинг: обучение модели на примерах

«Few-shot» — это когда вы показываете модели 2–3 примера желаемого результата прямо в промпте, а она продолжает закономерность для нового случая. Это самый естественный способ показать модели «нужно вот такое»: словами описать сложно, а показать легко. Один пример — «одно-shot», несколько — «фью-шот» (few-shot), ноль примеров — «нуль-шот».

Пример — лучший способ задать формат. Попробуйте словами описать нужный вид ответа на отзыв клиента: вы устанете писать правила. А три пары «отзыв → ответ» модель считывает мгновенно и делает четвёртый ответ в том же духе.

**ПРИМЕР ПРОМПТА**

```
«Превращай отзывы клиентов в короткие ответы поддержки. Тон тёплый, без оправданий,
всегда предложение решения.
Пример 1. Отзыв: "Курьер опоздал на час". Ответ: "Извините за ожидание — опоздание
недопустимо. Отправим промокод на следующий заказ в личный кабинет."
Пример 2. Отзыв: "Товар пришёл с вмятиной". Ответ: "Сочувствуем — заменим товар без
волокиты. Курьер привезёт новый завтра, старый заберём."
Теперь отзыв: "Долго не могли дозвониться"
```

Когда примеры особенно помогают: нестандартный или фирменный формат, конкретный тон и стиль, разбор по схеме («вопрос → категория → причина → действие»), задачи с редкой терминологией.

Плохие примеры работают хуже, чем их отсутствие: модель скопирует и ошибки. Если в примерах одинаковая структура, модель слепо повторит её даже там, где это неуместно. Если все примеры слишком похожи, модель может заузить задачу. И осторожно с чувствительными данными: не показывайте в примерах реальные переписки клиентов.

Формат примеров важнее их количества. Два аккуратных, единообразных примера работают лучше пяти разнобойных. Соблюдайте в примерах тот же вид, который хотите на выходе: если хотите ответ без кавычек — не берите ответы в кавычки в примерах. И разделяйте примеры явно: «Пример 1… Пример 2… Теперь: …».

**ПРИМЕР ПРОМПТА**

```
«Классифицируй обращения по схеме: "Обращение → категория → срочность". Категории:
доставка, оплата, возврат, другое. Срочность: высокая/средняя/низкая.
Пример 1: "Не пришла посылка, ждать 5 дней" → доставка → высокая.
Пример 2: "Вопрос по бонусам на будущее" → другое → низкая.
Теперь: "Оплатил дважды, верните деньги"».
```

> 💡 Начните с 2 примеров. Если модель сбивается — добавьте третий или «закрепите» правило словами после примеров: «Всегда сохраняй длину ответа как в примерах».

> 💡 Примеры можно комбинировать с шаблоном: инструкция + шаблон со скобками + 2 примера заполнения — самая надёжная конструкция для регулярных задач.

---

## ТЕСТ

**Что такое few-shot промпт?**

- [x] **Промпт с 2–3 примерами желаемого результата ✓**
- [ ] Промпт, отправленный несколько раз
- [ ] Промпт с ограничением длины ответа

**Почему так:** Few-shot — обучение в промпте: показываем примеры, модель продолжает закономерность.

**Зачем показывать примеры, если можно описать требования словами?**

- [ ] Словами нельзя описать ничего
- [ ] Примеры не нужны никогда
- [x] **Пример точнее и быстрее передаёт формат и тон, особенно сложные ✓**

**Почему так:** Формат и тон словами описывать громоздко; пары «вход → выход» модель схватывает сразу.

**Что опаснее для результата?**

- [ ] Два аккуратных единообразных примера
- [x] **Два примера с ошибками и разнобоем ✓**
- [ ] Один хороший пример

**Почему так:** Модель копирует закономерности из примеров, включая ошибки; плохие примеры хуже отсутствия примеров.

**Все примеры у вас слишком похожи друг на друга. Что может произойти?**

- [ ] Модель автоматически сделает примеры разнообразнее
- [ ] Модель ничего не заметит
- [x] **Модель может заузить задачу и повторять структуру даже там, где не надо ✓**

**Почему так:** Однотипные примеры задают слишком узкий шаблон поведения.

**Вы хотите ответ без кавычек. Как оформить примеры?**

- ○ Взять ответы в кавычки для красоты
- ● **Дать ответы в примерах тоже без кавычек ✓**
- ○ Без разницы, кавычки модель игнорирует

**Почему так:** Формат примеров — сильнейший сигнал: модель воспроизведёт вид примеров, включая мелочи.

**Какой элемент делает примеры удобными для модели?**

- ● **Явная нумерация и разделение: «Пример 1… Пример 2… Теперь: …» ✓**
- ○ Порядок букв внутри слов
- ○ Включение личных данных клиентов

**Почему так:** Явное разделение помогает модели отделить примеры от рабочего случая и инструкции.

**Модель сбивается даже с двумя примерами. Что сделать первым шагом?**

- ○ Удалить все примеры
- ○ Отказаться от few-shot
- ● **Добавить третий пример или закрепить правило словами после примеров ✓**

**Почему так:** Усиление сигнала — ещё один пример или словесное правило — обычно выправляет поведение.

**Какая конструкция самая надёжная для регулярных задач?**

- ○ Только длинная инструкция словами
- ● **Инструкция + шаблон со скобками + пара примеров заполнения ✓**
- ○ Только примеры без инструкции

**Почему так:** Комбинация задаёт и правила, и структуру, и образец — минимум свободы для случайных трактовок.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Выберите типовой «вход» из вашей практики: отзывы, письма, комментарии, заявки. Составьте few-shot промпт с 2–3 парами «вход → желаемый выход» и обработайте 5 реальных входов. Затем замените один пример на плохой (с разнобоем) и сравните — почувствуйте силу качества примеров.**

### ЧЕК-ЛИСТ

- ☐ Входы взяты из вашей реальной практики (обезличенные)

- ☐ В промпте 2–3 единообразные пары «вход → выход»
- ☐ Обработано минимум 5 новых входов, выходы соответствуют стилю примеров
- ☐ Проведён эксперимент с «плохим» примером и записан вывод о разнице

УРОК 7

# Структурированный вывод: списки, таблицы, JSON, схемы

Когда результат нужно куда-то вставить — в таблицу, в форму, в программу, — вольный текст не годится. Нужен структурированный вывод: список, таблица или даже JSON — простой машиночитаемый формат «ключ: значение». Хорошая новость: модель отлично умеет возвращать структуру, если её об этом прямо попросить.

Начните с самых частых видов. Список: «верни нумерованный список, один пункт — одна строка». Таблица: «верни таблицу с колонками: название, цена, срок». Для таблиц удобно просить Markdown — чат покажет её красиво, а скопировать в Excel можно без потерь.

**ПРИМЕР ПРОМПТА**

```
«Прочитай список товаров ниже и верни таблицу Markdown с колонками: Товар | Цена | Категория. Без пояснений до и после таблицы.
Товары: молоко 89 р (молочка); хлеб 45 р (выпечка); стиральный порошок 320 р (бытовая химия); яблоки 150 р/кг (овощи и фрукты)».
```

Для строгого формата используют приём «верни строго в формате...». Вы описываете схему — набор полей и их тип — и просите вернуть только это. Простой пример схемы: «Верни строго JSON: {"товар": строка, "цена_руб": число, "в_наличии": true|false}. Без текста до и после». JSON нужен, когда результат пойдёт дальше в автоматизацию (урок 14) или в таблицу с формулами.

**ПРИМЕР ПРОМПТА**

```
«Из текста объявления ниже извлеки данные и верни строго JSON по схеме: {"тип": "квартира|дом", "район": строка, "площадь_м2": число, "цена_млн": число, "мебель": true|false}. Только JSON, без комментариев.
Объявление: "Продам 2-комн. квартиру 54 м2 в Головинском районе, 8.2 млн, мебель частично остаётся"».
```

Что делать, если структура поехала: лишний текст вокруг, «болтающиеся» запятые в JSON, вместо числа — «около 8 млн»? Во-первых, повторите жёстче: «верни только JSON, без вводных фраз и без текста после». Во-вторых, проверьте схему: возможно, вы не указали тип значения. В-третьих, дайте модели битый фрагмент и попросите исправить: «вот JSON с ошибкой — исправь, чтобы он был валидным».

Полезная привычка — просить самопроверку: «после генерации проверь, что вывод строго соответствует схеме, и если нет — исправь». А если данные важные, проверяйте выборочно вручную: структура может быть идеальной, а значение внутри — выдумкой.

> 💡 Для таблиц просите Markdown, для автоматизации — JSON, для чтения человеком —

список с подзаголовками. Выбор вида — половина успеха.

💡 Если модель вставляет «примерно», добавьте правило: «Если значение в тексте отсутствует — ставь null, не придумывай». Это резко снижает количество выдуманных данных.

---

## ТЕСТ

**Зачем нужен структурированный вывод?**

- ○ Структурированный вывод всегда точнее по фактам
- ○ Чтобы ответ выглядел длиннее
- ● **Чтобы результат можно было вставить в таблицу, форму или обработать программно ✓**

**Почему так:** Структура — про удобство дальнейшей работы, а не про достоверность содержимого.

**Что такое JSON простыми словами?**

- ○ Язык программирования
- ● **Машиночитаемый формат «поле: значение» в фигурных скобках ✓**
- ○ Вид шрифта

**Почему так:** JSON — текстовый формат со строгими правилами; его читают таблицы, автоматизации и программы.

**Какой приём делает вывод максимально строгим?**

- ○ Просьба ответить капсом
- ○ Просьба «постарайся»
- ● **Формулировка «верни строго в формате...» с явной схемой полей ✓**

**Почему так:** Слово «строго» плюс схема — явный контракт: набор полей, типы, запрет лишнего текста.

**В JSON модель написала «около 8 млн» вместо числа. Что сделать?**

- ○ Смириться
- ● **Уточнить тип поля в схеме и попросить исправить битый фрагмент в валидный ✓**
- ○ Удалить всё поле

**Почему так:** Тип «число» в схеме + просьба исправить фрагмент — стандартный ремонт структуры.

**Что означает правило «если значения нет — ставь null, не придумывай»?**

- [x] Значения, которых нет в исходнике, помечаются пустым значением вместо выдумки ✓
- [ ] Модель перестанет отвечать
- [ ] Модель будет удалять текст

**Почему так:** Это главный анти-галлюцинационный приём при извлечении данных.

**Для чего просить Markdown-таблицу, а не JSON?**

- [ ] Markdown надёжнее для программ
- [ ] Разницы нет вообще
- [x] Markdown красивее в чате и легко копируется в Excel для человека ✓

**Почему так:** Markdown — для человека и ручной работы, JSON — для машин и автоматизации.

**Структура идеальная, но значение цены подозрительное. Что это значит?**

- [ ] Структура защищает и значения
- [x] Структура и факты — разные вещи: содержимое нужно проверять отдельно от оформления ✓
- [ ] Значения всегда верные, если структура валидна

**Почему так:** Модель может идеально оформить выдумку. Важные значения проверяются по источнику.

**Вывод обернут в лишние фразы «Конечно! Вот ваш JSON:». Что сделать?**

- [x] Повторить жёстче: «только JSON, без текста до и после» ✓
- [ ] Попрощаться с моделью
- [ ] Всегда просить Markdown

**Почему так:** Явный запрет на текст до и после — самое прямое решение обрамляющего мусора.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите 5–10 однотипных текстовых фрагментов из вашей практики (объявления, заявки, письма, отзывы). Постройте схему извлечения (4–6 полей) и получите таблицу Markdown. Затем повторите то же со строгим JSON и правилом «нет данных → null». Сравните оба результата.**

### ЧЕК-ЛИСТ

- ☐ Взяты минимум 5 реальных однотипных фрагментов
- ☐ Схема содержит 4–6 полей с указанием типов
- ☐ Получены и таблица Markdown, и строгий JSON с правилом null
- ☐ Проведена выборочная ручная проверка значений по исходникам, ошибки зафиксированы

УРОК 8

# Цепочки шагов: декомпозиция и самопроверка

Сложная задача «с наскока» проваливается даже у людей. Секрет работы с моделью такой же, как с исполнителями: режьте задачу на шаги. Декомпозиция — это не про «поумнеть», а про контроль: на каждом шаге вы видите результат и можете поправить его до того, как он потянул за собой следующие.

Приём «решай по шагам» (иногда пишут «думай пошагово») прост: попросите модель сначала выписать план или рассуждение, а потом ответ. Когда модель «проговаривает» шаги, качество на задачах с логикой и расчётами заметно растёт. Ещё лучше — управлять шагами самому.

**ПРИМЕР ПРОМПТА**

```
«Не отвечай сразу. Сначала перечисли 5 шагов, которые нужны, чтобы составить бюджет домашнего ремонта. Дождись моего "дальше". Потом пройдись по шагам и составь черновик бюджета с примерными вилками цен».
```

Классическая цепочка для текстов и проектов: (1) план из пунктов → вы правите план; (2) черновик по плану → вы правите подход; (3) финальная версия с учётом правок. На шаге 1 переделки стоят секунды, на шаге 3 — минуты. Так вы покупаете качество дёшево.

**ПРИМЕР ПРОМПТА**

```
«Задача: подготовить сравнение трёх тарифов мобильных операторов для семьи из 4 человек. Шаг 1: составь список критериев сравнения (5–7 штук) и покажи мне. Не пиши само сравнение, пока я не одобрю критерии».
```

Вторая половина урока — «проверь свой ответ». Модель может перепроверить себя: пересчитать цифры, сопоставить вывод с условием, найти противоречия. Полезные формулировки: «реши, а затем проверь решение, шаг за шагом», «перечитай свой ответ и найди ошибки», «проверь ответ на соответствие всем пунктам условия и перечисли, какие условия учтены».

Чек-лист вместо туманной просьбы «сделай хорошо» работает отлично: перечислите модели критерии качества («в ответе: указан срок, есть сумма, есть подпись, нет канцелярита») и попросите проверить ответ по каждому пункту. Это превращает размытое качество в измеримое.

**ПРИМЕР ПРОМПТА**

```
«Сначала реши задачу. Затем отдельно перечисли все условия задачи и отметь, как твой ответ учитывает каждое. Если какое-то условие не учтено — исправь ответ до отправки».
```

> 💡 Сложную задачу делите на несколько сообщений, а не в один абзац: «шаг 1 — только план». Править план в 5 раз дешевле, чем готовый документ.

---

## ТЕСТ

**В чём главный смысл декомпозиции задачи?**

- ○ Задача становится короче по тексту
- ● **Контроль результата на каждом шаге, правки дешевле ✓**
- ○ Модель работает технически быстрее

**Почему так:** Пошаговый подход позволяет ловить ошибки рано — править план в разы дешевле, чем финальный документ.

**Что делает приём «решай по шагам»?**

- ○ Уменьшает число токенов
- ○ Запрещает модели отвечать
- ● **Просит сначала выписать рассуждение или план, затем ответ — качество логики растёт ✓**

**Почему так:** «Проговаривание» шагов помогает модели на задачах с логикой и расчётами.

**Классическая цепочка для текстов — это...**

- ○ Три случайных вопроса подряд
- ● **План → правки → черновик → правки → финал ✓**
- ○ Сразу финал, потом план

**Почему так:** Сначала план (дёшево править), потом черновик, потом финальная версия с учётом правок.

**Зачем просить модель перечислить условия задачи и проверить их по ответу?**

- ● **Так обнаруживаются пропущенные условия до того, как вы это заметите ✓**
- ○ Так модель считает быстрее
- ○ Это украшение ответа

**Почему так:** Самопроверка по списку условий — практичный способ поймать пропуски.

**Чем чек-лист лучше просьбы «сделай хорошо»?**

- ○ Он отключает галлюцинации

- ○ Он короче
- ● **Он превращает размытое качество в конкретные проверяемые пункты ✓**

**Почему так:** Перечисленные критерии дают модели (и вам) измеримый стандарт проверки.

**На каком шаге правки дешевле всего?**

- ○ На финальной версии
- ● **На этапе плана из пунктов ✓**
- ○ Правки стоят одинаково

**Почему так:** Поменять пункт плана — секунды; переделать готовый текст под новый план — минуты.

**Какая формулировка — прямая просьба о самопроверке?**

- ● **«Реши, затем перечитай и найди ошибки, исправь» ✓**
- ○ «Думай как эксперт»
- ○ «Ответь подробно»

**Почему так:** Это двухфазный промпт: решение, потом отдельная фаза проверки с исправлением.

**Когда шаги лучше разнести по разным сообщениям?**

- ○ Только в уроках
- ○ Никогда, всё надо писать одним сообщением
- ● **В сложных задачах — чтобы между шагами вставлять ваши правки ✓**

**Почему так:** Раздельные сообщения дают точку контроля после каждого шага.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите сложную для вас задачу (сравнение покупок, планирование события, структура доклада, разбор бюджета). Проведите её через цепочку «план → черновик → финал» в трёх отдельных сообщениях. После финала попросите модель проверить результат по вашему чек-листу из 4–5 критериев.**

### ЧЕК-ЛИСТ

- ☐ Задача действительно сложная (одним промптом в лоб не решается)
- ☐ План согласован до черновика, правки плана внесены
- ☐ Черновик и финал — отдельные шаги, различия видны

- ☐ Модель проверила финал по вашему чек-листу, найденные недочёты исправлены

УРОК 9

# Тексты: письма, посты, статьи — от плана до вычитки

Текст — самая частая задача новичков, и здесь модель раскрывается полностью. Главное правило: у каждого вида текста своя цель. Письмо — чтобы получатель сделал что-то или узнал главное за 10 секунд. Пост — чтобы зацепить и удержать. Статья — чтобы ответить на вопрос читателя целиком. Скажите модели цель, и она подберёт структуру.

Деловое письмо: приветствие → суть в первом абзаце → детали → чёткий запрос действия → подпись. Отказ: благодарность → короткое «нет» без вранья → при желании альтернатива → доброжелательная концовка. Запрос информации: кто вы → что именно нужно → зачем и к какому сроку → чем упростите жизнь ответом.

**ПРИМЕР ПРОМПТА**

```
«Напиши вежливый отказ на заявку фрилансера, который предложил сотрудничество с ценой выше нашего бюджета. Структура: благодарность → честное "нет" из-за бюджета → предложение обратиться через полгода → тёплый тон. Максимум 5 предложений, без канцелярита».
```

**ПРИМЕР ПРОМПТА**

```
«Напиши письмо-запрос поставщику: просим актуальный прайс и сроки поставки на март. Контекст: работаем с ними второй год, объём около 200 единиц в месяц. Тон деловой, тёплый. В конце — конкретный вопрос с дедлайном: ответ нужен до пятницы».
```

Для постов и статей работает цепочка из урока 8: план → части → сборка. Сначала попросите 3–5 вариантов заголовков и план поста, выберите и поправьте. Потом генерируйте по одному разделу за раз — так каждый кусок получает внимание. Финал — сборка и единый проход по стилю.

Два приёма для качества. «3 варианта и гибрид»: попросите три разных варианта вступления или поста целиком и склейте свой из лучших кусков — это почти всегда лучше любого одного варианта. И «вычитка»: дайте модели свой готовый текст с ролью редактора — она найдёт опечатки, канцелярит, слабые места и покажет правки списком.

**ПРИМЕР ПРОМПТА**

```
«Ты — редактор. Вот мой текст (ниже). Проверь: грамотность, канцелярит, повторы, логику. Покажи правки списком "было → стало" с кратким объяснением каждой. Текст не переписывай целиком. Текст: "…"
```

> 💡 Пишите модели, куда пойдёт текст и кто его увидит: «пост для Telegram-канала для мам дошкольников». Это меняет всё: длину, сленг, примеры.

> 💡 Всегда просите показывать цифры, имена и обещания отдельным списком после текста — так вы за минуту проверите самое рискованное.

---

## ТЕСТ

**Что нужно назвать модели в первую очередь при работе над текстом?**

- ○ Цвет шрифта
- ○ Длину в символах
- ● **Цель текста и его аудиторию ✓**

**Почему так:** Цель определяет структуру, аудитория — словарь и примеры. Длина — следствие.

**Как устроен скелет делового письма?**

- ○ Детали → суть → приветствие
- ● **Суть в первом абзаце → детали → чёткий запрос действия ✓**
- ○ Рассуждение → вывод → намёк на действие

**Почему так:** Деловое письмо экономит время получателя: суть сразу, действие — конкретное.

**Как правильно строить вежливый отказ?**

- ● **Благодарность → честное «нет» → при желании альтернатива ✓**
- ○ Молчание — тоже ответ
- ○ Долгое оправдание и обещания

**Почему так:** Честное «нет» без вранья и, по возможности, альтернатива сохраняют отношения.

**Что даёт приём «3 варианта и гибрид»?**

- ○ Гарантию, что все три варианта одинаковы
- ○ Модель работает медленнее
- ● **Лучший результат: вы склеиваете свой вариант из сильных кусков ✓**

**Почему так:** Варианты показывают развилки, гибрид собирает лучшее — обычно сильнее любого одиночного.

**Какой порядок работы над статьёй рекомендован в уроке?**

- ○ Сразу весь текст целиком
- ● **План → правки → части по одной → сборка ✓**

- ○ Сначала финал, потом план

**Почему так:** Цепочка шагов (урок 8) применена к тексту: править план дёшево, части получают внимание отдельно.

**Зачем просить модель вынести цифры и обещания отдельным списком после текста?**

- ● **Чтобы за минуту проверить самые рискованные места текста ✓**
- ○ Так модель лучше пишет
- ○ Для красоты

**Почему так:** Цифры и обещания — главные места ошибок и выдумок; отдельный список упрощает проверку.

**Модель прислала готовый текст с выдуманной статистикой «по исследованиям». Что делать?**

- ○ Добавить ещё цифр
- ○ Публиковать, звучит солидно
- ● **Удалить или заменить: потребовать либо источник, либо формулировку без выдуманных цифр ✓**

**Почему так:** Непроверенные цифры — типичная галлюцинация; публикация без проверки опасна для репутации.

**Какой промпт — просьба о вычитке, а не переписывании?**

- ○ «Напиши лучше мой текст»
- ● **«Покажи правки списком "было → стало", целиком не переписывай» ✓**
- ○ «Сделай то же самое, но красивее»

**Почему так:** Формат «было → стало» сохраняет ваш текст и даёт точечные, объяснимые правки.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Напишите с помощью модели реальный текст из трёх разных жанров: деловое письмо (запрос или отказ), пост для соцсетей по цепочке «план → части → сборка» и вычитку существующего вашего текста в роли редактора. Для одного из жанров используйте «3 варианта и гибрид».**

### ЧЕК-ЛИСТ

- ☐ Созданы все три жанра текстов для реальных задач
- ☐ Для поста использована цепочка план → части → сборка

- ☐ Получены 3 варианта минимум для одного текста и собран гибрид
- ☐ Проведена вычитка: правки показаны в формате «было → стало»

УРОК 10

# Длинные документы: тезисы, извлечение фактов, переработка

Длинные документы — где ИИ экономит часы. Три базовые операции. Summarize (тезисы): сжать документ до главного. Extract (извлечение): вытащить конкретные факты в таблицу. Rewrite (переработка): переписать под другую аудиторию или цель. Это три разных промпта к одному и тому же тексту — и три разных результата.

Summarize работает хорошо, если вы скажете, для чего тезисы: «для решения, идти ли на встречу», «для доклада руководителю», «чтобы решить, читать ли целиком». От назначения зависит, что попадёт в выборку. Полезная форма — «5 тезисов + 1 риск + 1 открытый вопрос».

**ПРИМЕР ПРОМПТА**

```
«Вот текст договора-оферты (ниже). Сделай выжимку для владельца небольшого магазина, который решает, подписывать ли: 5 главных тезисов, отдельно — 3 пункта с обязательствами и штрафами, отдельно — что происходит при расторжении. Без юридической оценки, только факты из текста».
```

Extract — самая недооценённая операция. Из договора можно вытащить суммы и сроки в таблицу, из отчёта — все метрики по месяцам, из 40 отзывов — все упоминания проблем. Здесь работает связка с уроком 7: задайте колонки таблицы явно. И главное правило — «только факты из текста»: если чего-то нет, ставьте «нет в тексте», а не додумывайте.

**ПРИМЕР ПРОМПТА**

```
«Из текста ниже извлеки все обязательства сторон в таблицу: Обязательство | Кто | Срок | Последствия нарушения. Только информация из текста; если поле не указано — пиши "нет в тексте". Текст: "…"
```

Rewrite — переработка под аудиторию. Технический отчёт → резюме для руководителя в 5 пунктах. Инструкция → памятка для новичка. Статья → сценарий для короткого видео. Промпт задаёт: исходник, кто читатель, цель, вид результата. Это отличается от summarize: тут меняется форма и адресат, а не только объём.

Про длину и части. Если документ больше 50–100 страниц, делите его на куски и обрабатывайте последовательно, затем просите модель свести частичные результаты в общую сводку. И держите в голове: чем длиннее документ, тем легче модели «потерять» деталь из середины — важные места проверяйте поиском по исходнику.

**ПРИМЕР ПРОМПТА**

```
«Перепиши этот технический отчёт для руководителя без технического бэкграунда: 5 пунктов — что сделано, что не сделано, что нужно от него. Без жаргона, каждая строка — один факт. Отчёт: "…"
```

> 💡 Всегда указывайте: «факты — только из текста, если данных нет — так и пиши». Это самое дешёвое средство от выдумок в выжимках.

---

## ТЕСТ

**Чем extract отличается от summarize?**

- ○ Ничем
- ● **Extract вытаскивает конкретные факты в структуру, summarize сжимает до главных мыслей ✓**
- ○ Summarize точнее по фактам

**Почему так:** Тезисы — про суть и смысл; извлечение — про конкретные поля и данные, обычно таблицей.

**Зачем говорить модели, для чего вам тезисы?**

- ● **От назначения зависит, какие факты попадут в выжимку ✓**
- ○ Это формальность
- ○ Чтобы модель ответила быстрее

**Почему так:** «Для решения» и «для доклада» — разные выборки главного; назначение управляет отбором.

**Что означает правило «только факты из текста»?**

- ○ Модель должна цитировать весь текст
- ○ Модель должна отвечать короче
- ● **Данные берутся только из документа, при отсутствии — пометка «нет в тексте» ✓**

**Почему так:** Правило блокирует додумывание — главный риск при извлечении данных.

**Документ на 120 страниц. Как обрабатывать?**

- ○ Отправить целиком и один раз
- ● **Разделить на части, обработать последовательно, затем свести в общую сводку ✓**
- ○ Взять только первую страницу

**Почему так:** Крупные документы делят на куски; частичные результаты потом сводятся вместе.

**Rewrite — это...**

- ● **Переработка текста под другую аудиторию, цель и форму ✓**
- ○ Проверка орфографии
- ○ То же, что summarize

**Почему так:** Rewrite меняет адресата и форму (отчёт → резюме, инструкция → памятка), а не только сокращает.

**В таблице обязательств из договора у пункта стоит «нет в тексте». Это...**

- ○ Повод переделать всё
- ○ Ошибка модели
- ● **Правильное поведение по правилу отсутствия данных ✓**

**Почему так:** Явная пометка отсутствия лучше выдуманного значения — её легко заметить и дозаполнить вручную.

**Почему в длинных документах важные места стоит проверять поиском по исходнику?**

- ○ Модель обязана врать
- ● **Модель может «потерять» деталь из середины длинного текста ✓**
- ○ Так быстрее читать

**Почему так:** На длинных документах детали из середины чаще выпадают из внимания модели.

**Какой первый шаг полезен перед выжимкой большого документа?**

- ○ Удалить половину текста
- ○ Просить сразу полный пересказ
- ● **Спросить карту: сколько в документе о нужной теме и в каких разделах ✓**

**Почему так:** Карта документа ориентирует: какие части обрабатывать детально, какие пропустить.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите реальный длинный документ (договор, отчёт, инструкция, статья на 10+ страниц). Выполните все три операции: тезисы под конкретное решение, таблица-извлечение фактов (5–6 колонок) и переработка фрагмента под другую аудиторию. Проверьте минимум 3 извлечённых факта по исходнику.**

### ЧЕК-ЛИСТ

- ☐ Документ взят реальный и длинный

- ☐ Тезисы сформулированы под конкретное решение или задачу
- ☐ Таблица извлечения содержит 5–6 колонок и правило «нет в тексте»
- ☐ Минимум 3 факта проверены по исходнику, расхождения отмечены

УРОК 11

# Изображения и презентации: формула промпта и структура слайдов

Генерация изображений подчиняется той же логике, что и текст: чем конкретнее задание, тем лучше результат. Рабочая формула промпта для картинок: объект + стиль + композиция + детали. Объект — что изображено. Стиль — как: фотография, иллюстрация, акварель, плоский вектор, 3D-рендер. Композиция — план кадра: крупный план, вид сверху, объект по центру. Детали — свет, цветовая гамма, настроение.

**ПРИМЕР ПРОМПТА**

«Фотография уютной кофейни ранним утром: бариста за деревянной стойкой наливает кофе в керамическую чашку. Стиль: естественная фотография, тёплый утренний свет из окна. Композиция: средний план, объект чуть правее центра. Детали: мягкие тени, тёплая гамма — бежевый и коричневый, атмосфера тишины».

Ключевой принцип итераций для картинок: меняйте одну переменную за раз. Не получилось — поменяйте только стиль. Не то настроение — только детали освещения. Если менять всё сразу, вы не узнаете, что именно помогло. Ведите мысленный список: объект/стиль/композиция/детали — и крутите по одному.

Слова на изображениях — слабое место генераторов: буквы могут искажаться, особенно на русском. Правило: минимум текста на картинке, короткие слова крупным шрифтом, текст в кавычках в промпте — «надпись "ОТКРЫТО"». Если надпись критична — сгенерируйте фон без текста и добавьте надпись в редакторе. Это надёжнее.

**ПРИМЕР ПРОМПТА**

«Плоская векторная иллюстрация для обложки презентации о финансах: растущий график, монеты, лупа. Стиль: минимализм, 3 цвета — тёмно-синий, бирюзовый, белый. Композиция: элементы в правой половине, слева чистое поле под заголовок. Без текста на изображении».

Для презентаций ИИ — прежде всего структурщик. Он хорошо делает каркас: тезисы слайдов, логику повествования, вопросы для аудитории. А вот дизайн и финальные файлы пока надёжнее делать в обычных редакторах, взяв у модели содержание и структуру.

**ПРИМЕР ПРОМПТА**

«Составь структуру презентации из 8 слайдов на тему "Как мы снизили время доставки на 20%". Для каждого слайда: заголовок (до 6 слов), 3 тезиса-буллета, идея для визуала. Аудитория — руководители отделов, цель — показать результат и заручиться поддержкой».

> 💡 Сохраняйте удачные промпты картинок вместе с результатом: связка «промпт → картинка» — ваша личная библиотека стилей.

> 💡 Для слайдов просите «идею визуала» к каждому тезису — потом эти идеи превращаются в промпты для картинок. Так текст и графика рождаются в одной связке.

---

## ТЕСТ

**Из каких четырёх частей состоит формула промпта для картинок?**

- ● **Объект, стиль, композиция, детали ✓**
- ○ Цвет, шрифт, размер, формат
- ○ Тема, заголовок, текст, подпись

**Почему так:** Что изображено, как по стилю, как по кадру и какие детали — четыре рычага управления.

**Почему итерации по картинкам надо делать по одной переменной?**

- ○ Модель не принимает несколько изменений
- ○ Так быстрее технически
- ● **Иначе непонятно, какая правка дала эффект ✓**

**Почему так:** Одна переменная за раз — управляемый эксперимент: видно, что именно изменилось.

**Почему текст на картинках лучше добавлять в редакторе?**

- ○ Редактор придаёт картинке стиль
- ● **Генераторы часто искажают буквы, особенно кириллицу ✓**
- ○ Так дешевле

**Почему так:** Надёжный путь: генерируем фон без текста, надпись — в редакторе.

**Как в промпте картинки задать надпись?**

- ● **Взять текст в кавычки: надпись "ОТКРЫТО", коротким словом и крупно ✓**
- ○ Написать надписи капсом без кавычек
- ○ Описать её смысл

**Почему так:** Кавычки плюс короткое слово и крупный размер повышают шанс корректной отрисовки.

**Для чего ИИ в презентациях наиболее надёжен?**

- ○ Показывать презентацию аудитории
- ○ Рисовать финальный дизайн слайдов
- ● **Делать каркас: структуру, тезисы, идеи визуалов ✓**

**Почему так:** Модель сильна в содержании и структуре; финальный дизайн надёжнее делать в редакторе.

**Что должен содержать промпт для структуры презентации?**

- ○ Только тему
- ● **Тему, число слайдов, аудиторию, цель и формат тезисов ✓**
- ○ Имена участников

**Почему так:** Аудитория и цель управляют отбором тезисов; формат задаёт единообразие слайдов.

**На обложку нужно чистое место под заголовок слева. Как это заказать?**

- ○ Ничего, модель сама догадается
- ○ «Напиши заголовок слева»
- ● **В промпте: элементы в правой половине, слева чистое поле под заголовок ✓**

**Почему так:** Композиция заказывается словами про расположение элементов и свободные зоны.

**Картинка получилась близкой, но настроение холодноватое. Что менять?**

- ○ Объект целиком
- ● **Только детали: свет и цветовую гамму на тёплые ✓**
- ○ Стиль на 3D

**Почему так:** Настроение — это уровень «детали»; смена объекта или стиля уничтожила бы удачное.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Создайте визуальный комплект для реальной задачи: одна картинка по формуле «объект + стиль + композиция + детали» (для соцсетей, обложки или объявления) и структура презентации из 6–8 слайдов с тезисами и идеями визуалов. Сделайте минимум 3 итерации картинки, каждый раз меняя только одну переменную, и запишите, что именно изменилось.**

### ЧЕК-ЛИСТ

- ☐ Промпт картинки явно содержит все 4 части формулы
- ☐ Сделано минимум 3 итерации с изменением только одной переменной за раз
- ☐ Структура презентации содержит заголовки, тезисы и идеи визуалов для каждого слайда
- ☐ Записан вывод: какие изменения переменных сработали лучше всего

УРОК 12

# ИИ как учитель: объяснения, тренажёр, персональный план

Недооценённый сценарий использования нейросети — личный учитель, доступный круглосуточно. В отличие от статьи или видео, модель подстраивается под вас: переспрашивает, упрощает, приводит примеры из вашей сферы. Базовый приём — объяснение слоями: сначала «простыми словами», потом «чуть глубже», потом «технически точно».

**ПРИМЕР ПРОМПТА**

```
«Объясни, что такое инфляция, тремя уровнями: (1) простыми словами через пример из жизни, (2) глубже — с механизмами и связями, (3) технически точно с терминами. После каждого уровня спроси, переходить ли к следующему».
```

Второй приём — метафоры из вашей области. Скажите модель свою профессию: «Я бухгалтер. Объясни, как работает VPN, через аналогии из бухгалтерии и документооборота». Знакомая предметная область делает новое понятным в разы быстрее, чем абстрактные примеры.

Самый сильный учебный приём — перевернуть роли. Вместо «расскажи мне» — «спрашивай меня». Модель последовательно задаёт вопросы по теме, проверяет ваши ответы, поправляет ошибки и поднимает сложность. Это активное вспоминание — проверенный способ усвоения, несравнимо лучше пассивного чтения.

**ПРИМЕР ПРОМПТА**

```
«Ты — мой преподаватель по основам вёрстки писем. Задавай мне по одному вопросу за раз, от простого к сложному. После каждого моего ответа: верно/неверно, короткое объяснение, следующий вопрос. Через 10 вопросов дай оценку и список тем, где я слаб».
```

**ПРИМЕР ПРОМПТА**

```
«Проведи со мной разбор моей ошибки. Я сделал так: (описание). Сначала спроси меня 3 вопроса, чтобы понять мой ход мыслей, потом укажи, где именно ошибка, и дай похожую задачу для тренировки».
```

Третий сценарий — персональный план обучения. Опишите стартовый уровень, цель и сколько времени в неделю вы реально готовы тратить. Модель соберёт план по неделям с темами и практикой. Дальше план живёт: каждую неделю вы возвращаетесь, отчитываетесь, что получилось, и корректируете.

> 💡 После разбора темы просите: «Дай 3 вопроса, на которые я должен уметь ответить, если

понял тему». Проверка себя — лучшая метрика понимания.

---

## ТЕСТ

**В чём главное преимущество ИИ перед статьёй или видео при изучении темы?**

- ○ Модель всегда приводит источники
- ○ Модель отвечает длиннее
- ● **Модель подстраивается: упрощает, переспрашивает, меняет примеры под вас ✓**

**Почему так:** Адаптивность — ключ: объяснение подстраивается под ваши вопросы и уровень.

**Что такое «объяснение слоями»?**

- ○ Несколько разных тем подряд
- ● **Один вопрос с разной глубиной: просто → глубже → технически ✓**
- ○ Объяснение через таблицы

**Почему так:** Слои позволяют остановиться на комфортной глубине и не тонуть сразу в терминологии.

**Зачем просить метафоры из вашей профессии?**

- ● **Знакомая область делает новое понятным быстрее ✓**
- ○ Модель любит бухгалтеров
- ○ Так ответ длиннее

**Почему так:** Аналогия из вашей сферы переносит структуру нового знания на то, что вы уже знаете.

**Что делает приём «тренажёр-опрос»?**

- ○ Модель оценивает вас по шкале
- ○ Модель рассказывает лекцию
- ● **Модель задаёт вам вопросы, проверяет ответы и поднимает сложность ✓**

**Почему так:** Активное вспоминание: вопросы к вам + обратная связь — сильнее пассивного чтения.

**Как перевернуть роли в промпте?**

- ○ «Расскажи мне про тему»
- ● **«Задавай мне по одному вопросу за раз, проверяй ответы» ✓**
- ○ «Сделай конспект темы»

**Почему так:** Модель из рассказчика становится спрашивающим — вы тренируетесь, а не читаете.

**Что нужно указать для персонального плана обучения?**

- ○ Имя и возраст
- ○ Только цель
- ● **Стартовый уровень, цель и реальное время в неделю ✓**

**Почему так:** Без уровня и реального времени план будет либо слишком сложным, либо невыполнимым.

**Какая просьба — хорошая проверка понимания после разбора темы?**

- ○ «Дай конспект»
- ● **«Дай 3 вопроса, на которые я должен уметь ответить, если понял тему» ✓**
- ○ «Похвали меня»

**Почему так:** Попытка ответить на контрольные вопросы сразу показывает, где понимание дырявое.

**Разбор ошибки через 3 вопроса о вашем ходе мыслей нужен, чтобы...**

- ● **Найти, где именно сломалась логика, а не просто показать правильный ответ ✓**
- ○ Усложнить диалог
- ○ Потянуть время

**Почему так:** Диагностика хода мыслей точнее, чем готовое решение: исправляется причина, а не симптом.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Выберите тему, которую давно хотели понять (финансы, технология, правила, язык). Проведите сессию: (1) объяснение тремя слоями, (2) метафора из вашей профессии, (3) тренажёр из 10 вопросов с оценкой в конце. Сохраните финальную оценку и список слабых мест.**

### ЧЕК-ЛИСТ

- ☐ Тема выбрана реальная и актуальная лично для вас
- ☐ Получены все 3 слоя объяснения и метафора из вашей сферы
- ☐ Проведён тренажёр минимум из 10 вопросов с обратной связью
- ☐ Записаны оценка и 2–3 слабых места — они станут темой следующей сессии

УРОК 13

# Таблицы и данные без программирования

Вам не нужно уметь программировать, чтобы работать с данными. Нейросеть умеет писать формулы за вас: вы описываете словами, что нужно посчитать, она даёт готовую формулу для Excel или Google Таблиц с объяснением. Ваша задача — правильно описать структуру таблицы и проверять результат.

**ПРИМЕР ПРОМПТА**

```
«У меня таблица Google Sheets. A — дата, B — менеджер, C — сумма продажи, D — возврат (да/нет). Напиши формулу, которая считает сумму продаж менеджера Иванова за март без возвратов. Дай формулу и объясни по частям, что она делает».
```

Реальные таблицы грязные: «Иванов» и «иванов » — один человек, даты в трёх форматах, суммы с пробелами и буквами. ИИ отлично помогает привести всё к одному виду. Приём: вставьте кусок грязных данных (обезличенный!) и попросите правила приведения — и их текстом, и готовой формулой, если она существует.

**ПРИМЕР ПРОМПТА**

```
«Вот фрагмент столбца с именами: "иванов ", "Иванов", "ИВАНОВ", "Петров-Водкин", "петров-водкин". Предложи правило приведения к единому виду и напиши формулу для Google Sheets, которая делает это автоматически. Объясни, какие случаи формула может не поймать».
```

Второй сценарий — сводные и выводы. Вставьте таблицу (или её обезличенную часть) и спрашивайте как аналитик: «Какие три месяца были сильнее по выручке и на сколько процентов от среднего?», «Какой товар даёт 80% продаж?», «Что странно в этих данных — выбросы, пропуски, дубли?». Модель посчитает и, главное, объяснит ход расчёта.

**ПРИМЕР ПРОМПТА**

```
«Вот таблица продаж по месяцам: (вставь данные). Посчитай: (1) среднюю выручку за год, (2) месяцы выше среднего с отклонением в процентах, (3) три самых больших падения подряд. Покажи расчёты по шагам, чтобы я мог проверить. Отдельно укажи, где данных не хватает для выводов».
```

Набор типовых аналитических вопросов, который стоит держать в голове: «что здесь самое большое/маленькое и на сколько», «какая динамика — рост или падение», «есть ли выбросы и что их объясняет», «какие 20% дают 80% результата», «каких данных не хватает для уверенного вывода». Последний вопрос — самый ценный: он защищает от поспешных выводов.

Правила гигиены. Убирайте личные данные перед вставкой в чат (урок 16). Всегда просите показывать расчёт по шагам — так вы проверите арифметику. И помните: модель отлично считает

по данным, которые лежат в промпте, но не «видит» ваш файл на компьютере, пока вы его не вставите или не подключите.

> 💡 Если формула выдала ошибку в таблице, скопируйте текст ошибки и вставьте модели: «вот формула и вот ошибка — исправь». Это быстрее, чем разбираться самому.

---

## ТЕСТ

**Что нужно описать модели, чтобы она написала формулу?**

- ○ Имя файла
- ● **Структуру таблицы (какие данные в каких столбцах) и что нужно посчитать ✓**
- ○ Версию Windows

**Почему так:** Модель не видит вашу таблицу; точное описание столбцов и задачи — основа правильной формулы.

**Что значит «грязные данные»?**

- ● **Одни и те же значения в разных форматах: регистры, пробелы, даты ✓**
- ○ Данные о бракованных товарах
- ○ Данные с вирусами

**Почему так:** Разнобой форматов мешает считать; чистка приводит значения к единому виду.

**Зачем просить модель «укажи, где данных не хватает для выводов»?**

- ○ Модель обязана это делать сама
- ○ Чтобы ответ был длиннее
- ● **Это защита от поспешных выводов на неполных данных ✓**

**Почему так:** Явный вопрос о пробелах в данных — самый дешёвый способ не сделать ложный вывод.

**Модель отлично считает по данным, которые…**

- ○ Есть на вашем компьютере
- ● **Вставлены в промпт или подключены к чату ✓**
- ○ Вы задумаете позже

**Почему так:** Модель работает только с тем, что есть в контексте: файл нужно вставить или подключить.

**Какой вопрос к таблице — типовой аналитический?**

- ○ «Перепиши таблицу красиво»
- ○ «Нравится ли тебе моя таблица?»
- ● **«Какие 20% позиций дают 80% результата?» ✓**

**Почему так:** Вопрос про концентрацию результата (принцип Парето) — классика анализа данных.

**Формула в таблице выдала ошибку. Что сделать первым шагом?**

- ○ Переписать таблицу заново
- ● **Вставить модели текст ошибки и формулу с просьбой исправить ✓**
- ○ Удалить ошибочный столбец

**Почему так:** Текст ошибки содержит причину; передача его модели — самый быстрый ремонт.

**Зачем просить «покажи расчёт по шагам»?**

- ● **Чтобы можно было проверить арифметику и логику ✓**
- ○ Модель без этого отказывается считать
- ○ Для отчётности

**Почему так:** Пошаговый расчёт превращает чёрный ящик в проверяемую цепочку.

**Перед вставкой клиентской таблицы в публичный чат нужно...**

- ○ Зашифровать таблицу паролем
- ○ Ничего, это безопасно
- ● **Убрать или обезличить личные данные клиентов ✓**

**Почему так:** Личные данные не должны попадать в публичные боты — об этом подробно урок 16.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Возьмите реальную таблицу (продажи, расходы, список задач, личный бюджет). Уберите личные данные. Сделайте три вещи: (1) попросите правило чистки для самого «грязного» столбца, (2) получите 2–3 формулы из описания словами, (3) задайте минимум 3 аналитических вопроса с просьбой показать расчёт по шагам.**

### ЧЕК-ЛИСТ

- ☐ Таблица реальная, личные данные удалены или обезличены
- ☐ Для «грязного» столбца получено правило чистки (и формула, если применимо)

- ☐ Минимум 2 формулы получены из описания и проверены в таблице
- ☐ Заданы 3+ аналитических вопроса, расчёты показаны по шагам, выводы записаны

УРОК 14

# Автоматизация рутины: от поиска задач до связок без кода

Автоматизация начинается не с инструментов, а с поиска кандидатов. Простой фильтр: частота × время. Выпишите задачи, которые вы делаете регулярно, и оцените: сколько раз в неделю и сколько минут за раз. Всё, что повторяется часто и съедает 10+ минут, — кандидат. Правило «двух недель»: если задача вернётся минимум 5 раз — шаблон или связка окупятся.

Важная оговорка: не всё стоит автоматизировать. Задачи с высокой ценой ошибки (платежи, юридические отправки), уникальные задачи и те, где важен человеческий контакт, — оставьте себе. Автоматизируйте черновики и подготовку, а решение и отправку оставляйте человеку: «ИИ готовит — человек проверяет и жмёт кнопку».

Первый уровень автоматизации — уже знакомые вам шаблоны (урок 5). Шаблон письма клиенту, шаблон ответа на отзыв, шаблон еженедельной сводки. Это «полуавтомат»: вы заполняете переменные, модель делает остальное. Этого хватает, чтобы срезать 60–80% времени на типовых текстах.

**ПРИМЕР ПРОМПТА**

```
«Собери еженедельную сводку для руководителя по шаблону: "Неделя {номер}: главное — {3 факта с цифрами}; проблемы — {что пошло не так}; решения — {что делаем}; нужно от вас — {какие решения}." Держи деловой тон, максимум 10 строк, без оценочных слов».
```

Второй уровень — сервисы-конструкторы автоматизаций: n8n, Make, Zapier и отечественные аналоги. Логика простая: «когда случилось А — сделай Б». Например: пришла новая заявка из формы → текст заявки уходит в нейросеть → модель классифицирует и черновик ответа кладёт в таблицу или чат. Вы собираете это из готовых блоков, без программирования — как схему из кубиков.

**ПРИМЕР СХЕМЫ ДЛЯ КОНСТРУКТОРА**

```
Триггер: новая строка в Google-форме заявок. Шаг 2: отправить текст заявки в нейросеть с промптом: "Классифицируй заявку: срочная/обычная, верни JSON {"категория": "...", "срочность": "...", "черновик_ответа": "..."}". Шаг 3: записать результат в таблицу. Шаг 4: если срочная — уведомление в чат
```

Третий сценарий — чат-бот для типовых вопросов. Соберите 10–20 самых частых вопросов клиентов (или коллег) и ответы на них. Из этого получается база знаний для бота: он отвечает на типовое сразу, а сложное передаёт вам. Начните с малого: один канал, один сценарий, ручной разбор исключений.

> 💡 В любой связке оставляйте человека в цикле: пусть автоматизация готовит черновик, а отправка — за вами. Это спасает от массовых ошибок.

---

## ТЕСТ

**По какой формуле искать задачи-кандидаты на автоматизацию?**

- [x] **Частота × время ✓**
- [ ] Цена × риск
- [ ] Сложность × красота

**Почему так:** Часто повторяющиеся и долгие задачи дают максимальную экономию при минимальной настройке.

**Что НЕ стоит автоматизировать в первую очередь?**

- [ ] Еженедельную сводку
- [ ] Ответы на типовые вопросы
- [x] **Задачи с высокой ценой ошибки и уникальные решения ✓**

**Почему так:** Платежи, юридику и нестандартные решения оставляют человеку; автоматизируют черновую рутину.

**Что такое связка без кода типа «форма → ИИ → таблица»?**

- [ ] Программа на Python
- [x] **Готовые блоки в конструкторе (n8n, Make, Zapier): триггер → обработка → результат ✓**
- [ ] Макрос в Word

**Почему так:** Конструкторы собираются из блоков: триггер, шаг с нейросетью, запись результата — без программирования.

**Правильный порядок внедрения автоматизации?**

- [ ] Сначала бот, потом понимание задачи
- [ ] Сразу настроить связку
- [x] **Неделя руками с записью шагов → шаблон → автоматизация ✓**

**Почему так:** Сначала выучить задачу руками, потом шаблон, потом автоматизация — иначе автоматизируется хаос.

**Зачем оставлять «человека в цикле» в автоматизации?**

- ○ Чтобы работа выглядела занятой
- ● **Автоматизация готовит черновик, а проверку и отправку делает человек — защита от массовых ошибок ✓**
- ○ Так требует закон

**Почему так:** Одна ошибка промпта при массовой отправке умножается на все сообщения; человек — предохранитель.

**Для чат-бота типовых вопросов нужно прежде всего...**

- ● **Собрать 10–20 частых вопросов и ответы на них ✓**
- ○ Научиться программировать
- ○ Купить дорогую подписку

**Почему так:** База типовых вопросов и ответов — сердце бота; её собирают из реальных обращений.

**Настройка займёт 4 часа, экономия — 20 минут в неделю. Вывод?**

- ○ Автоматизировать и усложнить ещё
- ○ Настроить обязательно
- ● **Возможно, хватит шаблона — окупаемость дольше месяца ✓**

**Почему так:** Считайте окупаемость честно; иногда простой шаблон даёт 80% выгоды за 1% усилий.

**Что делает ИИ в связке классификации заявок?**

- ○ Отвечает клиентам без контроля
- ● **Возвращает структуру: категория, срочность, черновик ответа — в JSON для следующего шага ✓**
- ○ Печатает заявку за клиента

**Почему так:** Связка с уроком 7: строгий JSON позволяет следующему блоку конструктора записать данные в таблицу.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Выберите одну регулярную рутинную задачу из вашей работы. Ведите её «журнал» неделю: сколько раз и сколько минут, какие шаги, какие решения. Затем составьте шаблон-инструкцию (урок 5) с промптом для этой задачи и посчитайте честную окупаемость. Опишите в 3–4 шагах, как выглядела бы связка в конструкторе для этой задачи.**

**ЧЕК-ЛИСТ**

- ☐ Задача реальная и регулярная, ведён журнал повторов и времени
- ☐ Составлен рабочий шаблон-инструкция с переменными
- ☐ Посчитана окупаемость: настройка ÷ (минуты × повторы)
- ☐ Описана гипотетическая связка: триггер → шаг с ИИ → результат, с человеком в цикле

УРОК 15

# Галлюцинации и проверка фактов

Галлюцинация — когда модель уверенно сообщает что-то неверное, придумав правдоподобное. Это не «баг», который скоро починят, а свойство генерации: модель строит вероятное продолжение, а не проверяет истину. Ваша задача — знать, где она ошибается чаще, и проверять именно там.

Модель чаще всего врёт в: цифрах и статистике («по исследованиям 73%...»), датах и хронологии, прямых цитатах и авторстве, ссылках (существуют правдоподобные несуществующие адреса), номерах законов и статей, названиях книг и фильмов, актуальном состоянии мира (цены, курсы, новости после обучения модели). Всё, что требует точной памяти об идентификаторах, — зона риска.

Заметьте закономерность: точные факты-идентификаторы (номер статьи, URL, год) — рискованно; общее понимание темы и работа с вашим текстом — надёжно. Не потому, что модель «умнеет», а потому, что предсказывать общий текст проще, чем вспоминать точный идентификатор.

**ПРИМЕР ПРОМПТА**

```
«Ответь на вопрос: какие штрафы предусмотрены за это нарушение? Для каждого утверждения укажи уровень уверенности: уверен / не уверен. Если не уверен — не приводи номер статьи и точные суммы, а скажи, где проверить».
```

**ПРИМЕР ПРОМПТА**

```
«Перечисли 5 утверждений из своего предыдущего ответа, в которых ты наименее уверен, и объясни почему. Не сглаживай: мне нужна честная оценка, а не спокойствие».
```

Техники проверки. Первая — другой чат или другая модель: независимый ответ снижает шанс одинаковой выдумки. Вторая — источник: поисковая система или официальный сайт. Третья — «перекрёстный допрос»: попросите модель найти слабые места собственного ответа и перечислить, что именно нужно проверить руками.

Практическая шкала. Можно верить почти всегда: переформулировки, объяснения, работа с вашим текстом, структурирование, идеи и черновики. Проверять выборочно: расчёты по вашим данным, обобщения по таблицам. Проверять всегда: цифры, даты, цитаты, ссылки, законы, конкретные юридические и медицинские советы. Не доверять без источника: «свежие» новости, статистика «из головы», точные расценки.

> 💡 Просите модель указывать, из какого фрагмента вашего текста взято каждое утверждение: выдумке труднее спрятаться, если каждое утверждение должно иметь адрес в источнике.

## ТЕСТ

**Что такое галлюцинация нейросети?**

- ○ Слишком длинный ответ
- ○ Ошибка соединения с сервером
- ● **Уверенное сообщение неверной, но правдоподобной информации ✓**

**Почему так:** Модель генерирует вероятный текст, а не проверенный: правдоподобное может быть выдумкой.

**В чём модель ошибается чаще всего?**

- ○ В переформулировке вашего текста
- ● **В точных идентификаторах: цифры, даты, ссылки, номера законов ✓**
- ○ В структурировании списков

**Почему так:** Точный идентификатор требует точной памяти; общий текст предсказать проще.

**Почему переформулировка вашего текста — надёжная зона?**

- ○ Ваш текст всегда правильный
- ○ Модель отключает выдумки в этом режиме
- ● **Модель здесь не генерирует новое, а работает с данным ей текстом ✓**

**Почему так:** Когда исходник в контексте, модели не нужно вспоминать: она преобразует то, что видит.

**Как работает техника «другой чат»?**

- ○ Это резервная копия
- ● **Независимый ответ второй модели снижает шанс одинаковой выдумки ✓**
- ○ Второй чат отвечает быстрее

**Почему так:** Две независимые генерации редко выдумывают одно и то же — расхождение сигналит о риске.

**Какой промпт повышает честность модели?**

- ● **«Если не уверен — скажи прямо, не выдумывай» ✓**
- ○ «Отвечай кратко»
- ○ «Отвечай увереннее»

**Почему так:** Разрешение на неуверенность + запрет на выдумку смещают поведение к пометкам сомнительного.

**Модель дала ссылку на исследование. Что делать?**

- ○ Попросить ещё ссылок
- ○ Вставлять ссылку в документ
- ● **Проверить: открыть ссылку; несуществующие ссылки — классическая галлюцинация ✓**

**Почему так:** Ссылки — красная зона: правдоподобный адрес может не существовать.

**Какая зона — «проверять всегда»?**

- ○ Идеи для постов
- ● **Цитаты, ссылки, законы, конкретные юридические и медицинские советы ✓**
- ○ Стиль текста

**Почему так:** Там, где ошибка дорога и факт точный, проверка обязательна независимо от уверенного тона.

**Модель утверждает: «по последним данным, курс вырос на 4%». Категория риска?**

- ● **Красная зона: свежие цифры «из головы» — проверять всегда ✓**
- ○ Это расчёт по вашим данным
- ○ Можно верить

**Почему так:** Актуальные цифры без источника — типичная выдумка: модель не знает текущего момента.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Проведите «аудит доверия». Попросите модель ответить на 5 фактических вопросов из вашей сферы (даты, цифры, законы, ссылки). Затем: попросите модель пометить уровень уверенности по каждому ответу, проверьте все ответы по независимым источникам и запишите таблицу: ответ модели → уверенность модели → правда по источнику.**

### ЧЕК-ЛИСТ

- ☐ Задано 5+ фактических вопросов из вашей реальной сферы
- ☐ Модель пометила уверенность по каждому ответу
- ☐ Все ответы проверены по независимым источникам
- ☐ Составлена таблица сравнения и записан вывод: какие типы вопросов у вашей модели надёжнее

УРОК 16

# Безопасность и приватность: чек-лист перед отправкой

Всё, что вы отправляете в публичный чат-бот, уходит на серверы компании-разработчика. По умолчанию не стоит отправлять туда: паспортные данные и документы клиентов, финансовые реквизиты, медицинскую информацию, коммерческую тайну (цены, клиентские базы, непубличные планы), пароли и доступы, личные данные третьих лиц без их согласия.

Хорошая новость: для работы модели почти всегда не нужны настоящие имена. Замените «Ирина Петровна, ИНН 7712345678, зарплата 145 000» на «клиент А., ИНН — 10 цифр, зарплата — около 150 тыс.». Модель выполнит задачу так же, а риска нет. Обезличивание — привычка на одну минуту, которая снимает 90% рисков.

**ПРИМЕР ПРОМПТА**

```
«Клиентка А., 34 года, ИП, доход около 120 тыс. в месяц, двое детей. Нужно составить письмо об изменении условий обслуживания. (Все данные обезличены — реальные имена, ИНН и контакты в промпт не вставлены)»
```

Второй слой защиты — тарифы и настройки. У большинства сервисов есть настройки использования данных: в бесплатных версиях переписка чаще используется для обучения моделей, в платных обычно можно отключить сохранение и обучение (в истории или настройках конфиденциальности). Стоит один раз зайти в настройки и включить это осознанно. Корпоративные тарифы дают дополнительные гарантии — для рабочих задач уточняйте правила у своей компании.

Третий слой — дисциплина инструментов. Не вставляйте в чат файлы, права на передачу которых у вас нет. Не смешивайте рабочий и личный аккаунты без понимания правил. И осторожнее с «подключить данные»: перед подключением облачного диска или почты к боту поймите, к чему получит доступ сервис.

Четыре вопроса за десять секунд: (1) Есть ли здесь личные данные — чьи-нибудь имена, документы, контакты? (2) Есть ли коммерчески чувствительное — цены, базы, планы, реквизиты? (3) Что будет, если этот текст увидит посторонний? (4) Можно ли заменить реальное на обезличенное и сохранить задачу? Если на первые два — «да», обезличьте или не отправляйте.

**ПРИМЕР ПРОМПТА**

```
«Проверь мой черновик перед отправкой клиенту: найди в нём любые личные данные, реквизиты и коммерчески чувствительные сведения и предложи, как их обезличить или убрать, не потеряв смысл. Черновик: "…"
```

> 💡 Сделайте обезличивание автоматической привычкой: перед Ctrl+V — взгляд на текст. Один прогон глазами «есть ли здесь чужие данные?» — и вы в порядке.

## ТЕСТ

**Что из этого НЕ стоит отправлять в публичный чат-бот?**

- ○ Обезличенное описание задачи
- ● **Паспортные данные клиентов ✓**
- ○ Абстрактный вопрос по стилю текста

**Почему так:** Личные данные клиентов — запретная категория для публичных ботов: это и риск утечки, и нарушение закона.

**Почему обезличивание почти не вредит качеству задачи?**

- ○ Обезличивание ускоряет ответ
- ○ Модель не замечает замен
- ● **Для работы модели обычно не нужны настоящие имена и реквизиты — важна структура ситуации ✓**

**Почему так:** «Клиент А., доход около 120 тыс.» несёт ту же смысловую нагрузку, что и реальные данные.

**Где обычно настраивается использование ваших данных для обучения?**

- ○ В настройках браузера
- ● **В настройках конфиденциальности/истории сервиса ✓**
- ○ Это невозможно отключить нигде

**Почему так:** У крупных сервисов есть переключатели истории и обучения на данных; их стоит включить осознанно.

**Чем обычно отличаются платные тарифы в плане приватности?**

- ● **Чаще можно отключить сохранение переписки и использование для обучения ✓**
- ○ Данные платных пользователей продают дороже
- ○ Ничем

**Почему так:** Платные тарифы обычно дают опции конфиденциальности и корпоративные гарантии.

**Второй вопрос чек-листа перед отправкой:**

- ○ Достаточно ли длинный промпт?
- ○ Красиво ли выглядит промпт?
- ● **Есть ли коммерчески чувствительное — цены, базы, планы, реквизиты? ✓**

**Почему так:** Коммерческие секреты — вторая запретная категория после личных данных.

**Что сделать перед подключением почты или облачного диска к ИИ-сервису?**

- ○ Ничего, это всегда безопасно
- ● **Понять, к чему именно получит доступ сервис и зачем ему это ✓**
- ○ Подключить все папки сразу

**Почему так:** Право доступа — тоже риск; подключайте минимально необходимое.

**Коллега просит вставить в чат клиентскую базу с телефонами «просто для анализа». Ваши действия?**

- ● **Обезличить или обработать средствами, разрешёнными компанией; телефоны убрать ✓**
- ○ Вставить только половину базы
- ○ Вставить, это удобно

**Почему так:** Личные данные третьих лиц не должны попадать в публичный бот ни в каком объёме.

**Зачем вопрос «что будет, если текст увидит посторонний»?**

- ○ Так проще писать промпты
- ○ Это паранойя
- ● **Он мгновенно оценивает цену утечки содержимого промпта ✓**

**Почему так:** Мысленный эксперимент с посторонним читателем — самый быстрый тест чувствительности содержимого.

---

## ДОМАШНЕЕ ЗАДАНИЕ

**Проведите «аудит приватности» своей работы с ИИ. Вспомните последние 5–10 отправленных промптов (или просмотрите историю). Отметьте, были ли там личные данные, реквизиты или коммерческая чувствительность. Обезличьте 2 реальных прошлых промпта и сравните: потерялось ли качество ответа? Проверьте и настройте конфиденциальность в вашем ИИ-сервисе.**

### ЧЕК-ЛИСТ

- ☐ Просмотрены минимум 5 прошлых промптов, чувствительные места отмечены
- ☐ 2 промпта обезличены и отправлены повторно — качество ответа сравнено
- ☐ Конфиденциальность в ИИ-сервисе проверена: история/обучение настроены осознанно

- [ ] Записан личный чек-лист из 4 вопросов перед отправкой — и он размещён на видном месте

# Что дальше

Эта книга — печатная версия интерактивного курса «AI для новичков» от Krok AI. В интерактивной версии вы отвечаете на вопросы прямо в браузере, видите свой прогресс по урокам и получаете обратную связь на каждый ответ.

Пройдите курс онлайн — это бесплатно и займёт около часа:

**Пройти интерактивный курс →**

