

**МОСКОВСКИЙ
ПОЛИТЕХ**

ПРОМТ-ИНЖИНИРИНГ

Игровая логика и искусственный интеллект

Цели и задачи

Освоить структуру: Формула идеального промта.

Понять логику: Как «думают» большие языковые модели (LLM) и почему они ошибаются.

Изучить инструменты: Особенности работы с GigaChat, YandexGPT и мировыми лидерами в РФ.

Применить на практике: Продвинутые техники (Few-Shot, CoT) для решения сложных задач.

ИИ не заменит вас. Вас заменит человек, использующий ИИ.

Новая грамотность: Промт-инжиниринг становится базовым навыком, как знание Excel или поиск в Google.

Рост эффективности: Сотрудники с ИИ выполняют рутинные задачи на 30–50% быстрее (исследования 2024–2025 гг.).

Российский контекст: Бурный рост отечественных LLM (Сбер, Яндекс) и их внедрение в корпоративный сектор.

Кто такой ИИ?

ИИ — это ваш гениальный стажер без жизненного опыта.

LLM (языковая модель) не думает. Она просто предсказывает следующее слово с максимальной вероятностью.

Относитесь к ИИ как к очень умному, но буквальному стажеру

✓ Хороший промт:

"Ты — SEO-специалист. Напиши 3 варианта заголовков для статьи о промт-инжиниринге. Каждый не более 60 символов."

✗ Плохой промт:

"Напиши заголовок для статьи"

Детализация = Качество результата

Формула идеального промта

Анатомия идеального промта: 5 компонентов успеха

РОЛЬ

Кто отвечает

КОНТЕКСТ

Доп. данные

ЗАДАЧА

Что сделать

ОГРАНИЧЕНИЯ

Рамки и условия

ФОРМАТ

Вид результата

ПРИМЕР: Ты — маркетолог. Наш продукт — платформа для обучения ИИ. Создай email-рассылку. Не более 150 слов, тон дружелюбный. Markdown с заголовком и СТА.

Роль

Задавая роль, вы активируете нужный «режим мышления» модели.
Роль отсекает лишнее и фокусирует модель на нужном сегменте знаний.

Юрист

Опытный юрист по трудовому праву

Разработчик

Senior Python-разработчик

Аналитик

Дата-аналитик (финансы)

Копирайтер

Креативный копирайтер (IT)

Преподаватель

Преподаватель математики

Ученый

Исследователь МО

Совет: Чем конкретнее роль, тем точнее ответ

Контекст и Данные:

Нет контекста — нет результата

Галлюцинации

Модель придумывает факты при недостатке информации.

Вымышленные даты и цифры.
Несуществующие источники.
Ложные утверждения.

Решение

Больше контекста = меньше ошибок

Прикладывайте документы.
Указывайте конкретные данные.
Описывайте ситуацию подробно.

Избыток информации лучше, чем недостаток!

Формат вывода: Управляем упаковкой

Явно указывайте нужный формат вывода для получения структурированного результата



Markdown

Ответь в Markdown с ##



Таблица

Выведи в виде таблицы



Список

Нумерованный список из 5

{ } JSON

Верни в JSON с ключами



Стиль

Деловой /
неформальный тон



Длина

Не более 100 слов

Конкретика в формате = Удобство использования результата

Few-Shot Prompting

Обучение на примерах: Не объясняй, а покажи

Zero-Shot (Без примеров): Мы просто просим сделать задачу. Результат часто усредненный.

Few-Shot (С примерами): Мы даем модели 1–3 примера «Вопрос — Идеальный ответ» перед нашей задачей.

Как это работает: Модель видит паттерн и копирует его логику, стиль и структуру.

2–5 примеров достаточно для точной классификации

Few-Shot Prompting(Пример):

Промт: «Преврати описание товара в рекламный слоган.



Chain of Thought

Цепочка рассуждений: Заставь модель «думать», прежде чем отвечать

LLM пытаются выдать ответ мгновенно, пропуская логические шаги (как ученик, выкрикивающий ответ без решения).

Решение: Принудительно разбить задачу на этапы.

Просите модель показать промежуточные шаги решения

Zero-Shot CoT

Магическая фраза: «Давай подумаем шаг за шагом»

Суть метода: Если у вас нет времени писать примеры, используйте триггерную фразу.



Магическая фраза: «Let's think step by step» (Давай подумаем шаг за шагом / пошагово).

После модель сама включает режим логических рассуждений.

По данным исследований (Google/Stanford), эта одна фраза повышает точность решения математических задач с 17% до 78%.

Iterative Refinement

Итеративное улучшение

Ограниченность разового запроса: Ожидание идеального результата с первой попытки часто ведет к неудовлетворенности качеством вывода.

Принцип «Диалогового проектирования»: Работа с нейросетью — это процесс последовательных уточнений, а не разовое действие.

Алгоритм итерации:

1. **Прототипирование:** Получение базовой структуры ответа.
2. **Анализ и коррекция:** Оценка соответствия тону, объему и точности.
3. **Оптимизация:** Внесение уточняющих инструкций.

Negative Prompting X

Отрицательные промты

Метод исключения: Оптимизация через ограничения

Функция ограничений: Сокращение области поиска модели путем явного указания нежелательных элементов.

Применение в текстовых задачах:

- «Исключи вводные фразы и клише ("В современном мире", "Важно отметить")».
- «Не упоминай конкурентов и сторонние бренды».
- «Избегай использования сложной терминологии и деепричастных оборотов».

Проблема галлюцинаций ⚠

Доверяй, но проверяй: Феномен «Галлюцинаций»

Что это такое: Ситуация, когда модель уверенно генерирует правдоподобную, но ложную информацию (несуществующие законы, даты, цитаты).

Причина: Модель не «знает» факты, она предсказывает слова. Если фактов нет в её базе, она их додумывает.

Где риск максимален:

Ссылки на научные статьи и законы.

Биографии малоизвестных людей.

Расчеты (без использования инструмента калькулятора).

Золотое правило: Используйте ИИ для структуры и идей, но факты проверяйте вручную.

Верификация и Самопроверка (Self-Consistency)

Многоэтапная проверка: Внутренний аудит модели

Причина ошибок: Модели генерируют текст последовательно и не всегда могут оценить общую логику результата в момент написания.

Методика самопроверки: Запрос на повторный анализ собственного вывода.

Инструменты верификации:

- «Проверь приведенные расчеты на наличие арифметических ошибок».
- «Выяви возможные противоречия в аргументации».
- «Подтверди соответствие ответа заданным в первом шаге критериям».

Сфера применения: Критически важные задачи — аналитика, код, юриспруденция.

Ландшафт ИИ-инструментов в России 🇷🇺

Экосистема нейросетей: Что выбрать в 2026 году?

Отечественные лидеры (Доступность и РФ-контекст):

GigaChat (Сбер): Мультимодальность, отличная работа с русским культурным кодом, интеграция в бизнес-сервисы.

YandexGPT: Глубокая интеграция с поиском, экосистемой Яндекса и API для разработчиков.



●. **YandexGPT 5**

Глобальные модели

ChatGPT (OpenAI) / Claude (Anthropic):

Эталоны логики и программирования (требуют обхода ограничений).



DeepSeek:

Прорыв в соотношении цена/качество, открытость и высокая скорость.



GigaChat (Сбер) — Сила локализации

Преимущества:

- Понимание специфики российского законодательства и документооборота.
- Высокое качество генерации на русском языке без «машинного привкуса».
- Встроенная генерация изображений через Kandinsky.



Кейс применения: Автоматизация службы поддержки или написание текстов для локального рынка.

Доступность: Работает без VPN, официальная поддержка корпоративных клиентов.

YandexGPT – Скорость и интеграция

YandexGPT: Поиск, Аналитика, Масштабирование

Ключевые фишки:

- Интеграция в Яндекс Почту, Документы и Браузер (краткий пересказ видео и статей).
- Мощный API для встраивания в корпоративные продукты (Yandex Cloud).
- Режим «Помощник» для быстрой работы с информацией в реальном времени.

Сильная сторона: Обработка огромных пластов информации из интернета в режиме Live.

●. YandexGPT 5

Матрица выбора

Какой инструмент выбрать под вашу задачу?

Задача	Инструмент (Выбор редакции)	Почему?
Официальная переписка, Договоры (РФ)	GigaChat / YandexGPT	Знание законов, безопасность данных.
Кодинг, Сложная логика	DeepSeek / Claude 3.5	Лучшее понимание архитектуры кода.
Анализ больших документов	YandexGPT (Summarizer)	Быстрая работа с русским текстом.
Креатив (Картинки + Текст)	Kandinsky (в GigaChat)	Удобно все в одном месте.
Личные задачи (Travel, обучение)	ChatGPT / Copilot	Глобальная база знаний.

Информационная безопасность

Не рассказывайте ИИ лишнего

Важно знать: Всё, что попадает в чат (особенно в бесплатные версии), сохраняется на серверах разработчика и может быть использовано для обучения.

Что нельзя отправлять в ИИ:

- ✗ Личные данные:** ФИО клиентов, телефоны, адреса.
- ✗ Секреты компании:** Финансовые планы, пароли, закрытый код.
- ✗ Договоры:** Документы с грифом «Конфиденциально».

Как работать безопасно:

Удаляйте имена и названия компаний (заменяйте на «Клиент X»).

Используйте ИИ для черновиков, а реальные данные вставляйте сами в готовый файл.

Маркетинг и Тексты

Примеры эффективных промтов

Задача: Рерайтинг и адаптация контента под разные площадки (Telegram, VK, E-mail).

«Перепиши этот сухой пресс-релиз для нашего Telegram-канала.

Роль: Ты — ироничный редактор с отличным чувством юмора.

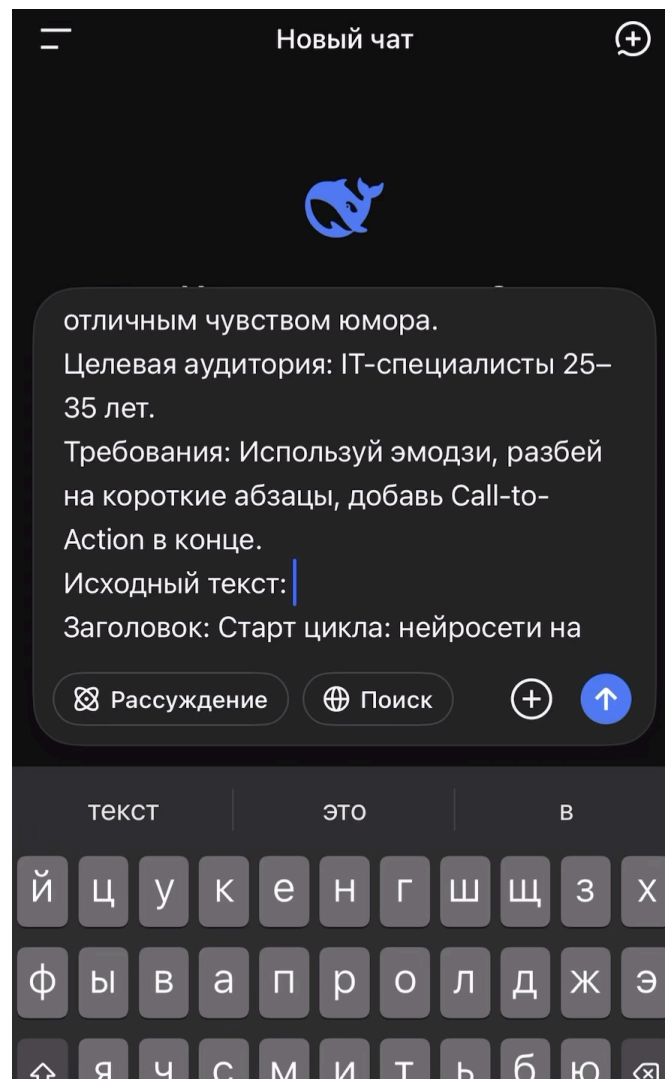
Целевая аудитория: IT-специалисты 25–35 лет.

Требования: Используй эмодзи, разбей на короткие абзацы, добавь Call-to-Action в конце.

Исходный текст: [Вставить текст]»

Результат: Живой пост вместо бюрократической новости.

Видео-пример:



Программирование и IT

Примеры эффективных промтов

Поиск багов и логических ошибок:

Роль: Ты — Senior QA Automation Engineer с 10-летним опытом поиска критических уязвимостей в коде.

Контекст: Данный фрагмент кода (Python) отвечает за обработку платежных транзакций в нашем интернет-магазине. Пользователи жалуются на двойное списание средств.

Задача: Проанализируй код, найди логическую ошибку, которая приводит к дублированию транзакций, и предложи исправленный вариант.

Ограничения: Не используй сторонние библиотеки, решение должно быть на стандартном Python. Будь краток и лаконичен.

Формат: Сначала объяснение причины ошибки (2-3 предложения), затем блок исправленного кода.

Лучшие инструменты: DeepSeek, Claude 3.5, ChatGPT.

Видео-пример:

```
Анализ и исправление двойно...  
# Имитация успешного  
ответа от платежной системы  
return True  
  
# Пример использования  
if __name__ == "__main__":  
    processor =  
    PaymentProcessor("test.db")  
    # Симулируем два  
    одновременных запроса на один  
    и тот же заказ  
  
    processor.process_payment(order  
_id=1001, amount=1500.0)  
  
    processor.process_payment(order  
_id=1001, amount=1500.0) #  
Потенциальное двойное  
списание!
```

SQL

Примеры эффективных промтов

Генерация SQL запроса.

Роль: Ты — ведущий аналитик данных (Data Analyst) со специализацией на работе с PostgreSQL.

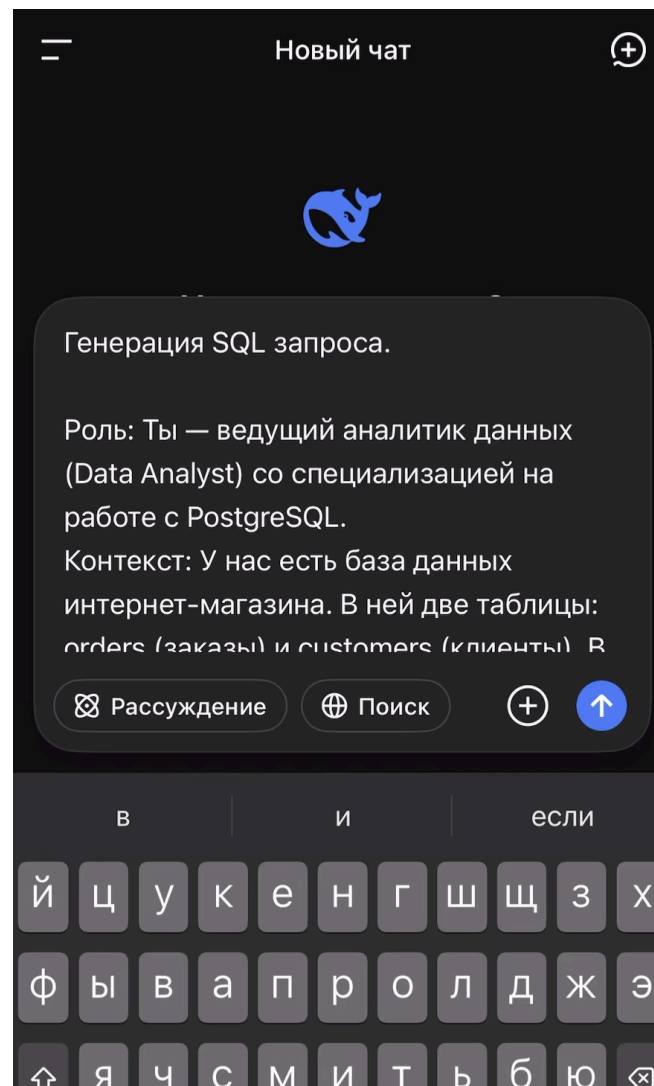
Контекст: У нас есть база данных интернет-магазина. В ней две таблицы: orders (заказы) и customers (клиенты). В orders есть колонки customer_id, order_date и total_amount. В customers есть id и city.

Задача: Напиши SQL-запрос, который выведет список городов и общую сумму продаж по каждому городу за последний месяц (декабрь 2025 года).

Ограничения: Отсортируй результат по убыванию суммы. Используй только стандартный синтаксис SQL.

Формат: Оформи код в блок с подсветкой синтаксиса и добавь краткий комментарий, за что отвечает каждая строка.

Видео-пример:



Заключение

ИИ не заменит человека, но человек с ИИ станет в два раза быстрее.

Промт-инжиниринг — это не магия, а умение четко ставить задачи.

Экспериментируйте, ошибайтесь и дорабатывайте (итерации — это ключ).

Главное: Оставайтесь пилотом. ИИ — это мощный двигатель, но штурвал всегда в ваших руках.

**МОСКОВСКИЙ
ПОЛИТЕХ**

Спасибо за внимание