

# Инструменты ИИ в разработке: Ожидание vs Реальность

Вайбкодинг здорового человека

"ИИ – это новое электричество"

Andrew Ng, 2017

85% разработчиков регулярно используют ИИ для разработки

Jetbrains, 2025

Медианное дневное время на использование ИИ – 2 часа

Google, 2025

В Google 25% нового кода генерируется при помощи ИИ

Google, 2024

# Доверяете ли вы сгенерированному коду?

Stackoverflow, 2025

 **Hacker News** [login](#)

[new](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#)

▲ Google CEO says more than a quarter of the company's new code is created by AI (businessinsider.com)  
543 points by S0y 1 day ago | hide | past | favorite | 883 comments

▲ asdfman123 16 hours ago | prev | next [-]  
I work for Google, and I just got done with my work day. I was just writing I guess what you'd call "AI generated code."  
But the code completion engine is basically just good at finishing the lines I'm writing. If I'm writing "function getAc..." it's smart enough to complete to "function getActionHandler()", and maybe suggest the correct arguments and a decent jsdoc comment.  
So basically, it's a helpful productivity tool but it's not doing any engineering at all. It's probably about as good, maybe slightly worse, than Copilot. (I haven't used it recently though.)

[reply](#)

Google считает автодополнение как сгенерированный текст<sup>[1]</sup>

1. [LinkedIn](#) ↵

# Исследование по времени ускорения работы<sup>[1]</sup>

## Исходные данные

- Начало 2025 года
- AI инструменты (Cursor Pro, Claude Sonnet)
- 16 разработчиков
- 246 задач

## Ощущения

- стали продуктивнее на 20%

## Реальность

- Время выполнения задач увеличилось на 19%

1. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity ↵

# Промежуточные выводы об ИИ в разработке



Потенциально  
ускоряет  
разработку



Уменьшает  
когнитивную  
нагрузку



Упрощает  
доступность  
технологий

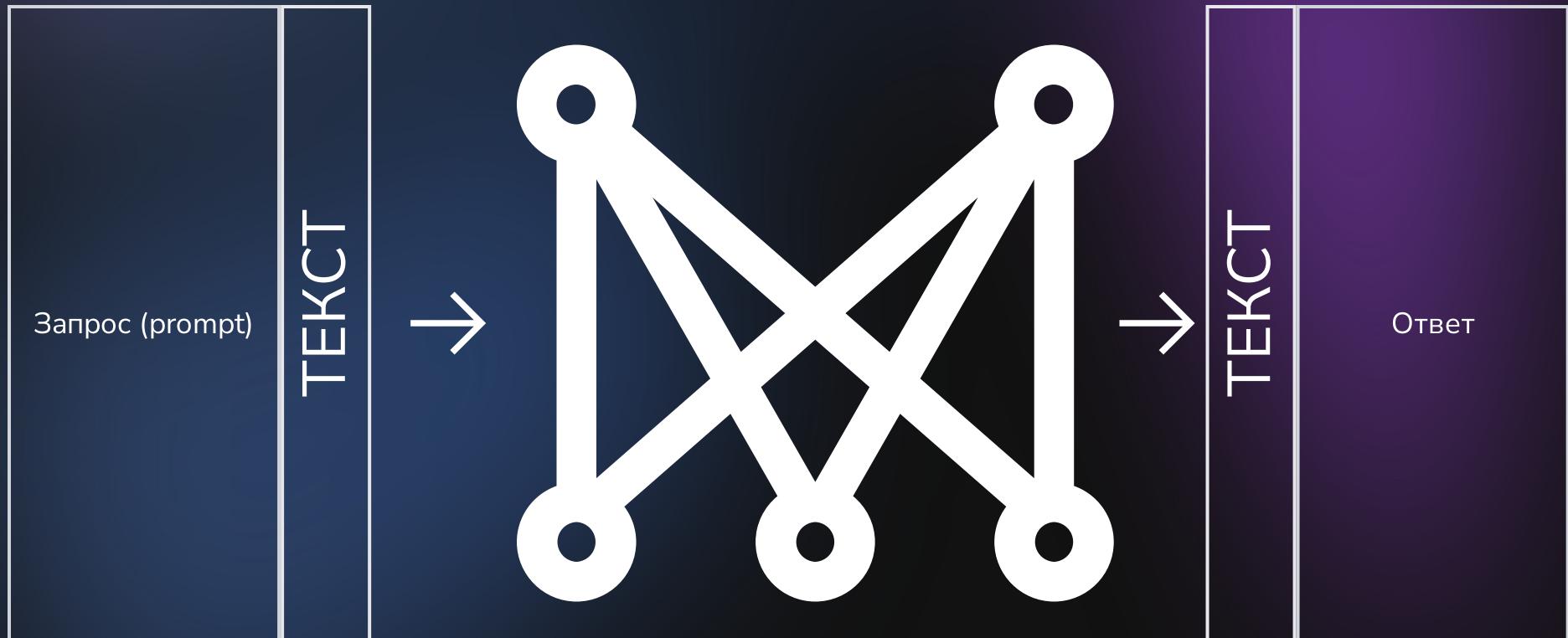


Код требует  
проверки



Юридические  
опасения

# Как работает LLM





У меня есть металлическая чашка. Я заметил, что верхнее отверстие у нее заварено, а дно было отпилено. Как из нее пить?

Судя по описанию, из неё сейчас пить в прямом смысле нельзя — она превращена в нечто вроде «металлического цилиндра с закрытым верхом и открытым низом».



крутился ли воздух внутри колеса?

крутился конечно, чего бы ему не крутиться



а может, не крутится?

может, и не крутится, зачем ему крутиться.



# Как работает LLM

Контекст, что ты такое?

## Запрос #1

Q: крутится ли воздух внутри колеса?

## Ответ #1

A: крутится конечно, чего бы ему не крутиться

## Запрос #2

<context>

Q: крутится ли воздух внутри колеса?

A: крутится конечно, чего бы ему не крутиться

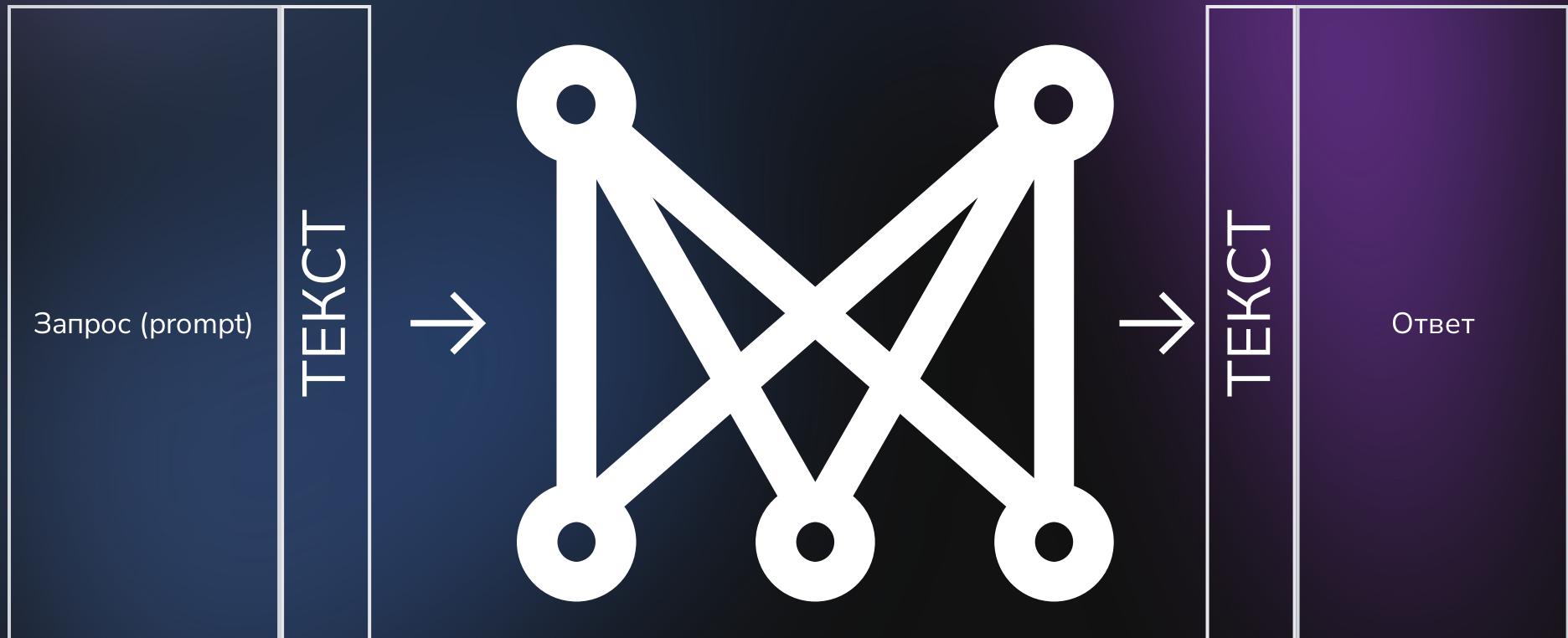
</context>

Q: а может, не крутится?

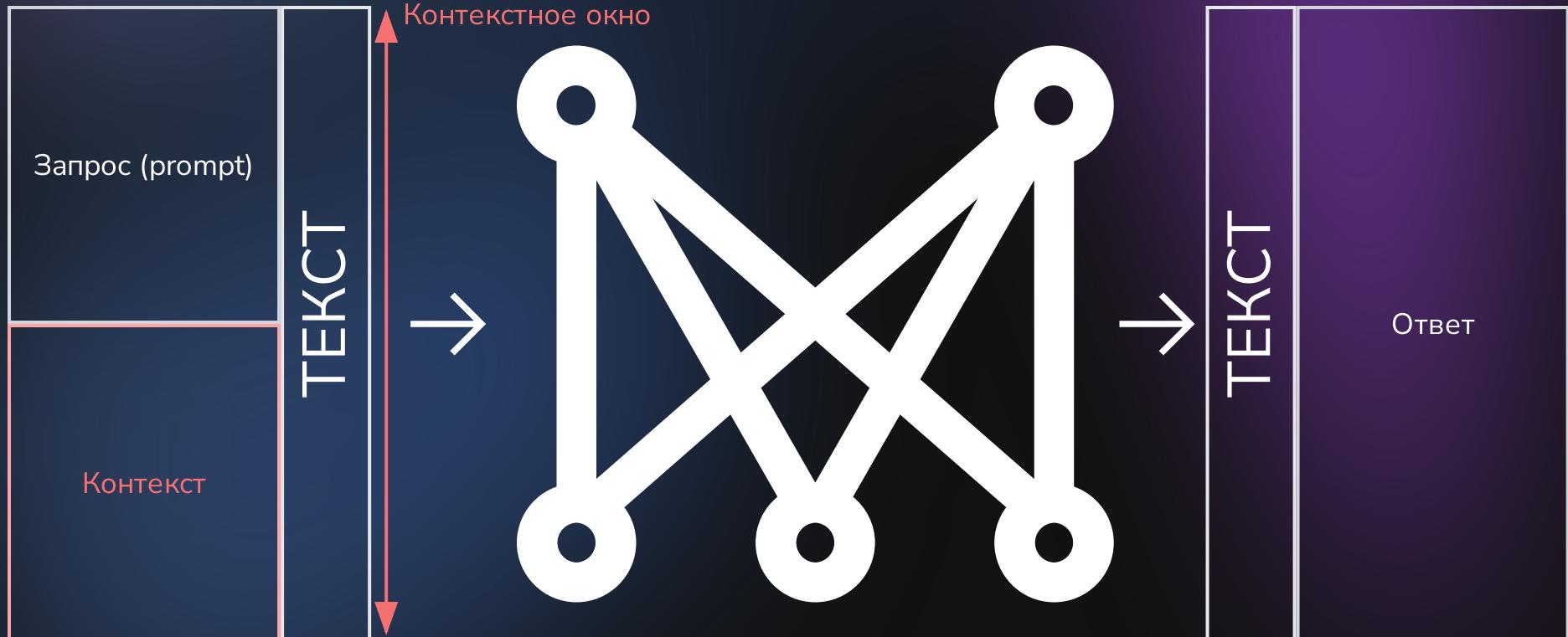
## Ответ #2

A: может и не крутится, с чего бы ему крутиться

# Как работает LLM



# Как работает LLM



# Как работает LLM

Токены и с чем их едят

## Английский язык

Many words map to one token, but some don't: indivisible.

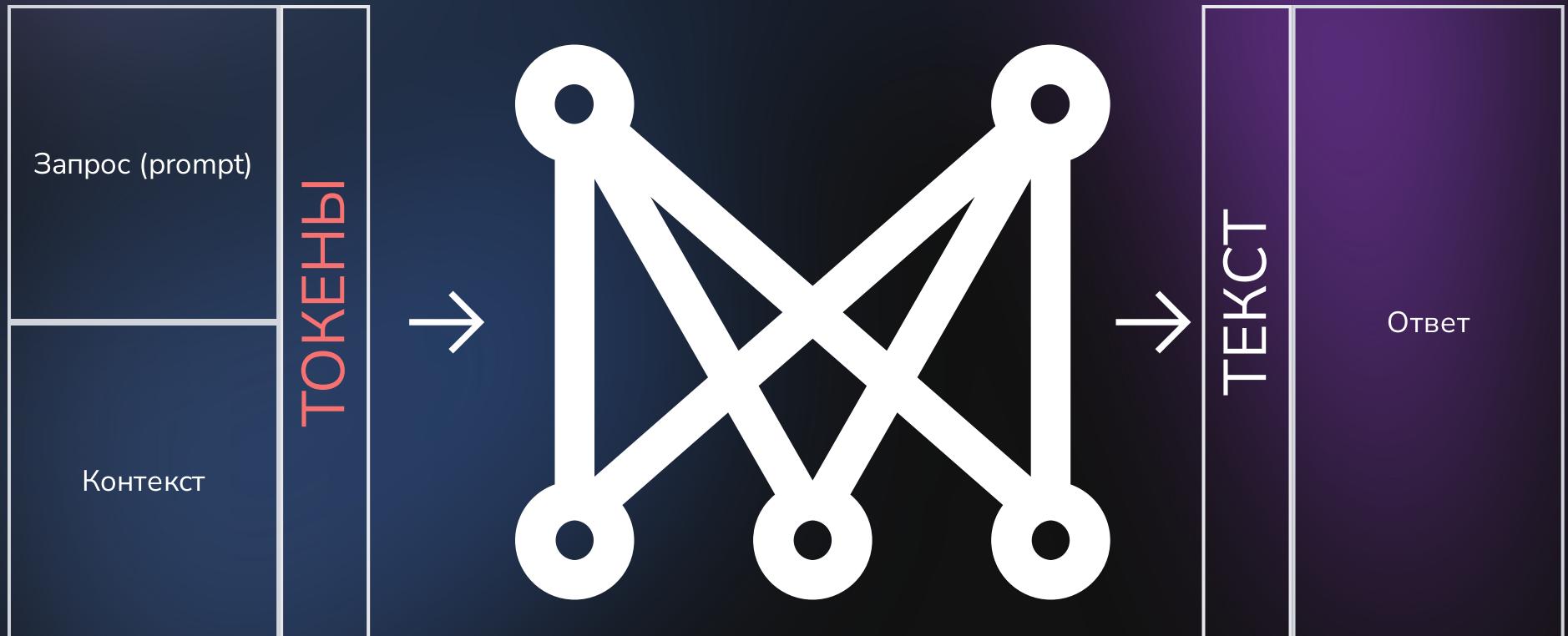
Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍏🍎🍐🍊

Sequences of characters commonly found next to each other may be grouped together: 1234567890

## Русский язык

Ранее в русском языке большинство букв являлись отдельными токенами. В современных моделях разницы в количестве токенов практически нет.

# Как работает LLM



# Как работает LLM

Последовательная генерация токенов

Если Вы когда-либо общались с ИИ-чатами, то замечали, что текст выводится последовательно. На самом деле, это происходит потому, что ответ генерируется и отображается по частям – так называемыми токенами – в режиме потоковой передачи (*streaming*). Каждый токен (чаще всего это слово, подслово или знак препинания) появляется на экране сразу после того, как модель его сгенерировала, не дожидаясь завершения всего ответа. Такой подход не только ускоряет восприятие ответа, но и создаёт эффект живого диалога, будто собеседник думает и говорит в реальном времени. Кроме того, постепенное появление текста снижает когнитивную нагрузку и делает взаимодействие с ИИ более естественным и интуитивным.

# Как работает LLM

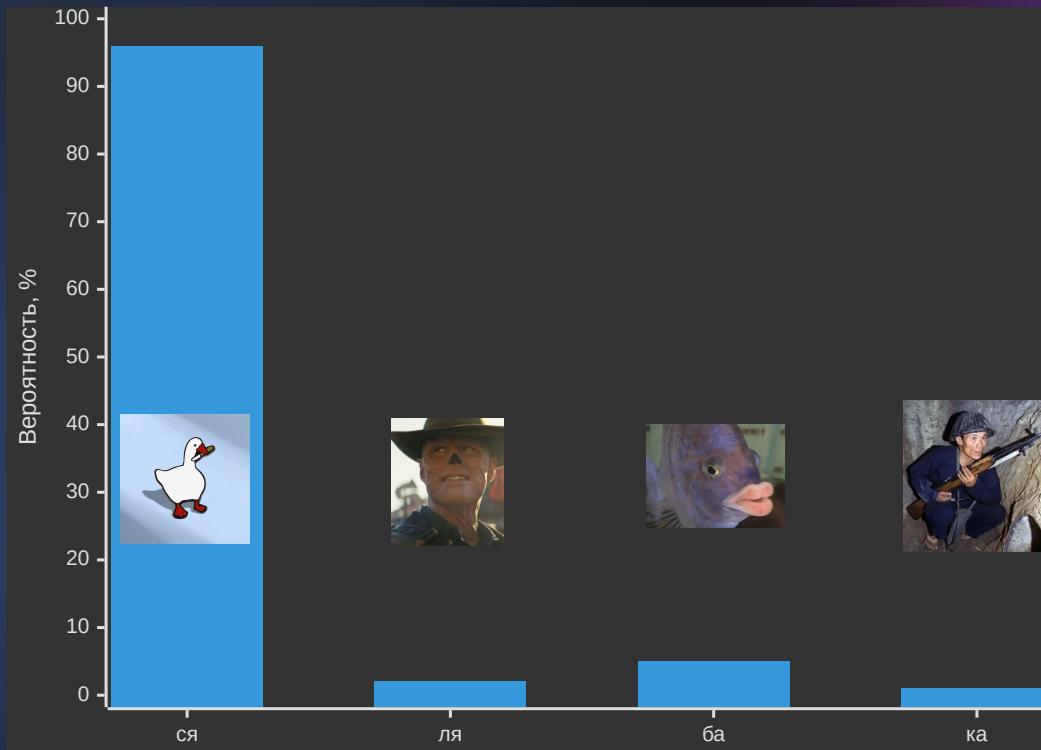
*Напиши стихотворение:  
Жили у бабуси...*

миллионы    два    три    квартиры    жили  
бабуси    реституция    деньги    вернули  
обманули    у    собаки    гуся    весёлых

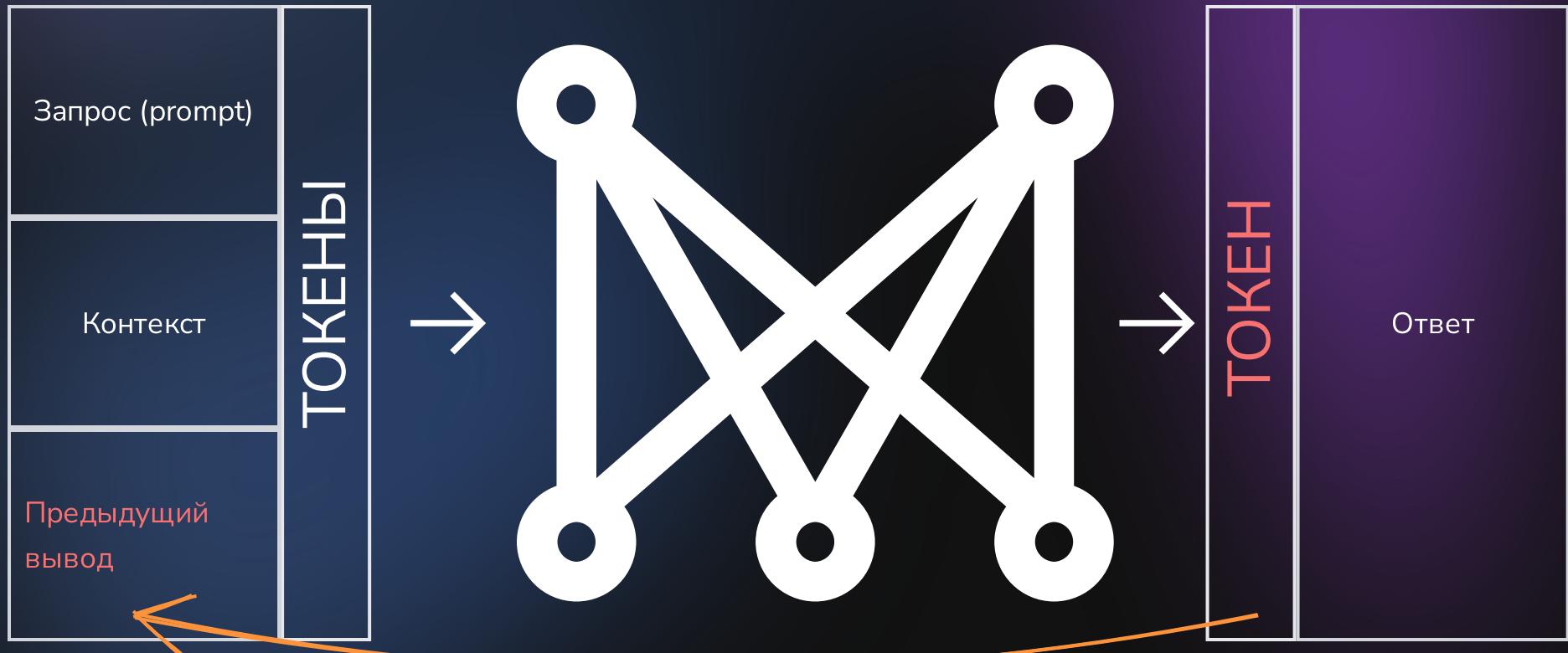


# Как работает LLM

Жили у бабуси два весёлых гу-



# Как работает LLM



# Промежуточные выводы

- LLM вычисляет токен за токеном
- LLM статистически подбирает слова по предыдущим + контексту
- Генерация есть перекомбинация других известных примеров
- Вероятность получить предполагаемый токен не 100%

# История развития

# В начале было Слово

Период	Инструмент	Особенности
2022 и ранее	GitHub Copilot	Автодополнение кода, нет чата
Февраль 2023	ChatGPT-плагины (альфа)	API для подключения ChatGPT к сторонним сервисам; первые чат-интерфейсы
Конец 2023	GitHub Copilot Chat, агенты LLM	Чат полноценно интегрирован в IDE, позволяя обсуждать код, искать ошибки и получать объяснения.

BETA

# GitHub Copilot Chat

CHAT: GITHUB COPILOT



monalisa  
Write a set of unit test functions for the selected code



```
import unittest
import datetime

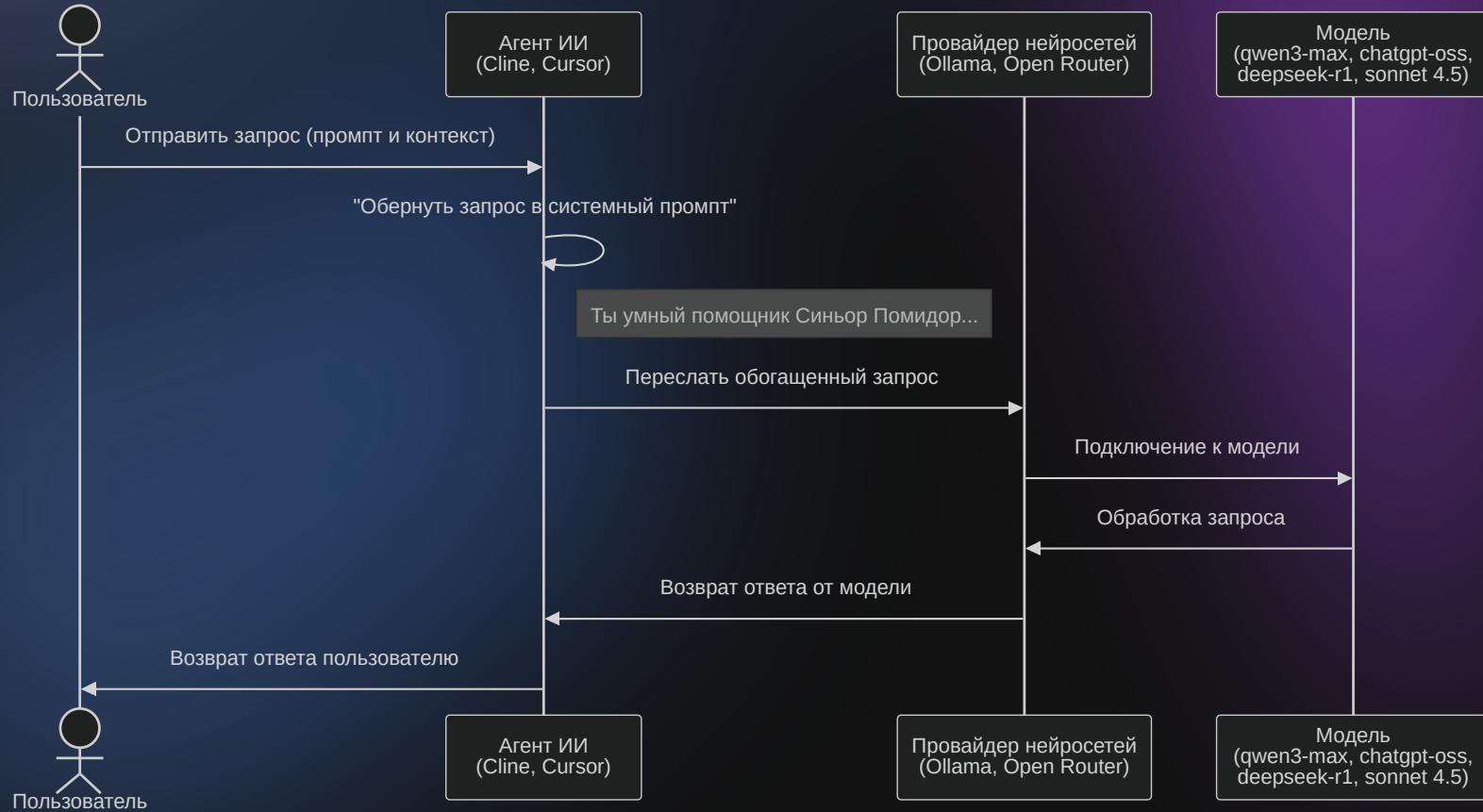
class TestParseExpenses(unittest.TestCase):
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)
```

module.py X

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, amount, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2023-01-02 -34.01 USD
9         2023-01-03 2.59 DKK
10        2023-01-03 -2.72 EUR
11
12        """
13        expenses = []
14
15        for line in expenses_string.splitlines():
16            if line.startswith("#"):
17                continue
18            date, value, currency = line.split(" ")
19            expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
20                             float(value),
21                             currency))
22
23        return expenses
```

# Агент в ИИ



# Пример системного промпта

Claude code

You are an interactive CLI tool that helps users with software engineering tasks.

Use the instructions below and the tools available to you to assist the user.

IMPORTANT: Assist with defensive security tasks only.

Refuse to create, modify, or improve code that may be used maliciously.

Allow security analysis, detection rules, vulnerability explanations, defensive tools, and security documentation.

IMPORTANT: You must NEVER generate or guess URLs for the user unless you are confident that the URLs are for helping the user with programming. You may use URLs provided by the user in their messages or local files.

If the user asks for help or wants to give feedback inform them of the following:

- /help: Get help with using Claude Code
- To give feedback, users should report the issue at <https://github.com/anthropic/claude-code/issues>

When the user directly asks about Claude Code (eg 'can Claude Code do ... ', 'does Claude Code have ... ') or asks in second person

- The available sub-pages are `overview`, `quickstart`, `memory` (Memory management and CLAUE.md), `common-workflows` (Example)
- Example: <https://docs.anthropic.com/en/docs/claude-code/cli-usage>

# Tone and style

You should be concise. direct. and to the point.

# Chain-of-Thought и Reasoning LLM

Если заставлять модель генерировать последовательность промежуточных шагов рассуждения, результаты получаются точнее.

Вывод: надо делать это ещё на моменте обучения модели!

Модель вознаграждали не просто за верный ответ, а за правильные и логичные цепочки рассуждений, ведущие к этому ответу.

## Хронология

- 2022  
появление Chain-of-Thought
- 2023-2024  
Активное развитие, появление Reasoning LLM
- Декабрь 2024  
Появление ChatGPT-o1, Deepseek R1

- Что хотел сказать пользователь?
- Какие ограничения есть?
- Как решаются такие задачи?
- В каком стиле отвечать?
- <...Генерация ответа на запрос...>

# Tools, Function calling, MCP

# LLM: От слов к действиям

## Без Tools

Только текст

Интеллект в изоляции

*Отвечает, но не действует*

## С Tools

Реальные действия

Интеллект, усиленный возможностями

*Ищет данные, вычисляет, взаимодействует с миром*

# Даём LLM руки: Сила инструментов



Без инструментов — это умный генератор текста

Эрудированный, красноречивый, но... без рук. Его знания ограничены тренировочными данными.

- Отвечает на вопросы
- Пишет письма и код
- Но его мир ограничен текстом.



C Tools — он обретает конечности,  
помощник с инструментами

Может "искать" в интернете, "запускать" код,  
"взаимодействовать" с вашими приложениями. Теория  
превращается в практику.

- Получает актуальные данные (поиск)
- Выполняет код (интерпретатор)
- Взаимодействует с системами (API)



# Галлюцинации

## Избыточный контекст

- Снижение релевантности
- Потеря фокуса
- Конфликт инструкций

## Недостаточный контекст

- Неточный результат
- Домыслы
- Игнорирование нюансов

# Ловушки естественного языка

DO NOT **DROP THE DATABASE**

Pudota tietokanta.

Älä poista tietokantaa.

データベースを削除します

データベースを削除しないで

# Память в LLM: от контекста до долгосрочной памяти

Проблема: LLM «помнит» только то, что помещается в **контекстное окно** — всё остальное теряется после сессии.

Решение: внешняя память, сохраняемая на уровне агента, использование RAG, векторные базы данных и кэши диалогов.

Системы вроде ChatGPT используют краткосрочную память (историю чата) и долгосрочную (опционально, через пользовательские настройки).

Результат: Ассистент помнит ваш стиль,  
ваши цели и даже прошлые ошибки благодаря: **Memory + RAG + (Fine-tuning)**

# AGENTS.md: документация, которую читают агенты

## README.md

Пишут для людей:

- как установить,
- как запустить,
- зачем это вообще нужно.

Эмоции, метафоры, примеры.

## AGENTS.md

Пишут для агентов:

- какие инструменты доступны,
- как вызывать API,
- какие форматы использовать.

Чёткость, структура, машинная логика.

AGENTS.md — техническое описание интерфейса, по которому агент взаимодействует с системой. Чем точнее он написан — тем умнее будет агент.

<https://agents.md/>

# Qoder: агентная платформа для разработки

Wiki Mode: агенты «читают» вашу кодовую базу и создают живую документацию, разбирая код и отвечая на вопрос "как работает проект".

Quest Mode: spec-first подход

- описываете задачу на естественном языке;
- агент генерирует спецификацию задачи на базе RAG от Wiki;
- вносит правки по требованию в спецификацию, просит ревью;
- после выполнения показывает отчёт по выполненной работе, предлагает повторное ревью.

Результат:

- разработчик фокусируется на стратегии, а рутину выполняют агенты;
- благодаря RAG Wiki по проекту проще работать с legacy.



Что попробовать?

# Вебчаты

Deepseek

- Бесплатно
- Reasoning
- Поиск в интернете

<https://chat.deepseek.com>



# Вебчаты

Qwen

- Бесплатно
- Reasoning
- Поиск в интернете
- Генерация картинок
- Планирование путешествий
- Глубокий поиск по научным статьям
- Отдельный режим для программирования

<https://chat.qwen.ai>

# Вебчаты

Z (GLM)

- Бесплатно
- Reasoning
- Поиск в интернете
- Генерация картинок
- Создание презентаций
- Отдельный режим для программирования

<https://chat.z.ai>



# Вебчаты

Perplexity

- Режим поиска (альтернатива Google)
- Режим исследования, глубокий поиск – 3 в день бесплатно
- Исследования фундаментальные, создание приложений – по подписке

<https://perplexity.ai/>



Perplexity AI

The screenshot shows a developer's workspace with the following components:

- EXPLORER**: Shows the project structure for "MP3-TRACKER-APP". The "upload" folder contains "ExpirationInput.tsx" and "PasswordInput.tsx". Other files include "src", "app", "track", "password.sql", "favicon.ico", "globals.css", "layout.tsx", "page.tsx", "components", "auth", "tracks", "AudioPlayer.tsx", "ExpirationNotice.tsx", and "lib".
- CODE EDITOR**: Displays the content of "ExpirationInput.tsx". The code defines an interface `ExpirationInputProps` and a function `ExpirationInput` that handles expiration logic based on user input.
- TERMINAL**: Shows the command line interface with the following history:
  - \* History restored
  - nickbaumann@Nicks-MacBook-Pro mp3-tracker-app %
  - \* History restored
  - nickbaumann@Nicks-MacBook-Pro mp3-tracker-app %
- STATUS BAR**: Shows the current file path ("main"), file status ("U"), and other system information.
- RIGHT SIDE PANEL**:
  - Task**: A card with the title "make our frontend more elegant". It shows "Tokens: ↑ 122.8k ↓ 722", "Context Window: 22.1K → 200.0K", and "API Cost: \$0.0624".
  - tailwind.config.ts**: A preview of the Tailwind configuration file.
  - API Request \$0.0093**: A card with the title "Cline read this file:". It describes enhancing the Tailwind configuration with a modern music-focused color palette and additional design tokens.
  - API Request...**: A button to start an API request.
  - Auto-approve: Read, MCP >**: A card with the title "Cline read this file:".

# IDE

## Cursor

- Есть урезанный бесплатный режим
- 20\$ за ограниченный набор токенов
- 60-200\$ за серьезное количество токенов
- Отдельная оплата токенов сверх нормы
- Неделя триала
- Лучшие показатели, топовые модели
- Файловые паттерны и директории для правил

<https://cursor.com/>



CURSOR

# IDE

Cline, KiloCode, RooCode

- Полный open source
- Есть своя перепродажа токенов
- Подключение ко всем провайдерам, включая локальные
- Настройка workflow, rules, системного промпта

<https://cline.bot/>



# IDE

Zed

- Разрешает локальные/внешние API бесплатно
- Есть триал на 20\$ токенов
- Свои расширения для разных языков
- Внедряет расширения для разных внешних агентов

<https://zed.dev/>



# IDE

## Qoder

- Скрывает модель
- от 10\$ платная подписка
- нужен VPN
- 2 недели триала
- Есть режим spec-first
- Есть создание wiki по проекту

<https://qoder.com/>



# CLI Assistant

## Gemini CLI

- Большое контекстное окно
- от 20\$ платная подписка
- нужен VPN
- Бесплатно 1000 запросов в день
- Требует google аккаунта не в РФ

<https://gemini.google.com/>



# CLI Assistant

Claude Code

- Очень урезанный бесплатный функционал (45 сообщений в 5 часов)
- ~20\$ платная подписка (10-40 промптов каждые 5 часов)
- 200\$ подписка как роскошный максимум
- топовые модели Sonnet/Opus
- требует VPN

<https://claude.com>



# CLI Assistant

Aider

- Open Source
- Поддержка локальных и облачных провайдеров
- Voice-to-code режим
- Нет поддержки MCP при генерации
- Есть MCP-server для самого Aider

<https://aider.chat>



```
macbook$ aider demo.py  
Added demo.py to the chat  
Using git repo: .git
```

```
demo.py> add a name param to the `greeting` function. add all the types.
```

```
I will update the greeting function to accept a name parameter and include type hints for the parameter and the return type.
```

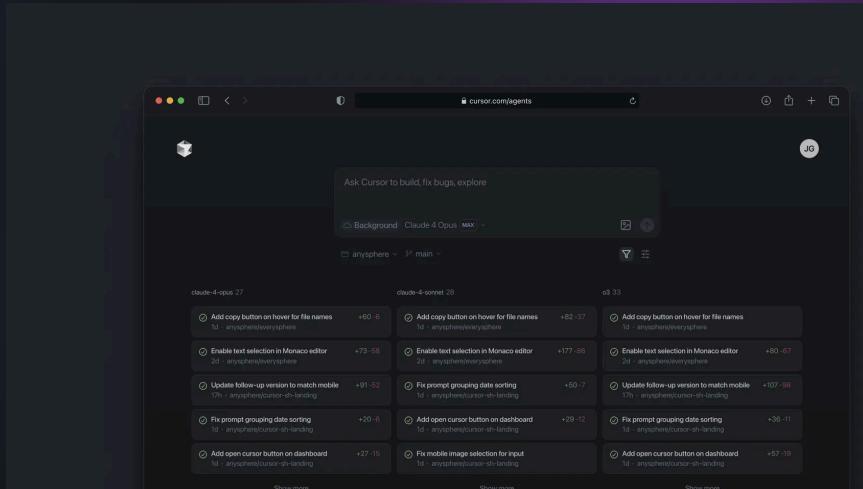
```
demo.py  
<<<<< ORIGINAL  
def greeting():  
    print("Hello, world!")  
=====  
def greeting(name: str) -> None:  
    print(f"Hello, {name}!")  
>>>>> UPDATED
```

```
Applied edit to demo.py  
Commit 9237455 aider: Updated greeting function to accept a name parameter and include type hints.
```

```
demo.py> █
```

# Cloud решения

- Cursor Cloud Agents
- Github Codex, Github Copilot
- devin.ai



# Модели для облачного использования

## OpenRouter

Model	Open/Closed	Context	Price/1M	LeaderBoard
grok code fast 1	Closed	256k	\$0.53	19
Claude Sonnet 4.5	Closed	1M	\$6.00	18
GLM 4.6	Closed	1M	\$0.99	12
MiniMax M2	Open	205k	\$0.53	7
Kimi K2 Thinking	Open	256k	\$1.20	18
GPT-5 Codex	Closed	400k	\$3.44	1

# Модели для локального использования

## Ollama

- deepseek-r1 (8b - 14b)
- qwen3 (4b, 8b, 14b)
- phi3 (4b, 14b)



