

# Основные элементы маркированной структуры промптов

Техника маркированной структуры промптов включает в себя набор стратегий для визуальной организации запросов с помощью следующих элементов:

## 1. Ясная иерархическая организация

- **Заголовки и подзаголовки:** Разделение промпта на логические секции
- **Нумерованные списки:** Обеспечение последовательности шагов или пунктов
- **Маркированные списки:** Организация связанных, но не обязательно последовательных элементов
- **Отступы:** Создание визуальной иерархии информации

## 2. Явное разделение компонентов промпта

- **Разграничение контекста от инструкций**
- **Выделение ролей от задач**
- **Отделение примеров от основного запроса**
- **Четкое обозначение ожидаемого формата ответа**

## 3. Специальные маркеры и форматирование

- **Маркеры секций:** Использование символов-разделителей (например, [=====])
- **Выделение ключевых требований:** Капитализация, символы, жирный шрифт
- **Теги для обозначения логических блоков:** например, <think>, <answer>
- **Тематические разграничители:** ## для заголовков, ### для подзаголовков

# Почему маркированная структура работает: научное обоснование

Эффективность маркированной структуры объясняется несколькими когнитивными и технологическими факторами:

## 1. Улучшение внимания и обработки информации

Исследования показывают, что структурированные промпты улучшают механизм внимания языковых моделей, позволяя им лучше фокусироваться на

различных аспектах запроса. Согласно данным, это связано с архитектурой механизма внимания трансформеров.

## 2. Усиление обработки токенов

Маркеры, заголовки и визуальные разделители становятся якорями, к которым модель может "привязать" свое внимание в длинных промптах. Как отмечается в исследовании "Селективное привязывание подсказок", такие якоря усиливают внимание к ключевой информации.

## 3. Улучшение работы с рабочей памятью модели

Структурирование информации помогает моделям эффективнее использовать свой контекстный буфер, аналогично тому, как структурирование информации помогает человеческой рабочей памяти. Это особенно важно для сложных запросов.

## 4. Снижение когнитивной нагрузки

Четкая структура снижает "когнитивную нагрузку" на модель, позволяя ей лучше разделять и обрабатывать различные аспекты задачи, особенно в многозадачных промптах.

## 5. Синтаксические преимущества для токенизации

Маркеры и специальные символы часто токенизируются как отдельные токены, что делает их более заметными в векторном пространстве модели и улучшает разделение концептов.

# Ключевые типы маркированной структуры

## 1. Блочная структура с разделителями

```
[=====]
# Задание
Составить план контент-маркетинга для стартапа.

## Исходные данные
- Ниша: B2B SaaS для управления проектами
- Целевая аудитория: менеджеры среднего звена
- Бюджет: ограниченный
- Конкуренты: [список конкурентов]

## Что требуется
```

1. Стратегия контент-маркетинга на 6 месяцев
2. Основные темы для блога
3. Идеи для лид-магнитов
4. KPI для отслеживания результатов

## Формат ответа

- Для каждого пункта не более 5 ключевых идей
- Обоснование каждой идеи (1-2 предложения)
- Приоритизация по важности

[=====]

## 2. Структура "think-then-answer"

# Анализ финансового отчета компании

<think>

- Проанализировать ключевые показатели: прибыль, выручка, долг, маржа
- Определить тренды в сравнении с предыдущими периодами
- Сравнить с отраслевыми стандартами
- Выявить потенциальные риски и возможности
- Сформулировать ключевые наблюдения и рекомендации

</think>

<answer>

### 1. Анализ текущего финансового положения

- Прибыльность: ...
- Ликвидность: ...

### 2. Сравнение с предыдущими периодами

- Тренд выручки: ...
- Изменение маржинальности: ...

### 3. Рекомендации

- Краткосрочные действия: ...
- Долгосрочная стратегия: ...

</answer>

## 3. Иерархическая структура с вложенными списками

# План разработки нового продукта

## 1. Исследование рынка

- \* 1.1. Анализ конкурентов
  - Прямые конкуренты (список)
  - Косвенные конкуренты (список)
- \* 1.2. Потребительские тренды
  - Основные потребности целевой аудитории
  - Неудовлетворенные потребности

## ## 2. Концепция продукта

- \* 2.1. Ключевые особенности
- \* 2.2. Уникальное торговое предложение
- \* 2.3. Позиционирование

## ## 3. Дорожная карта разработки

- \* 3.1. Фаза прототипирования
  - Сроки: ...
  - Ресурсы: ...
- \* 3.2. MVP
  - Минимальный набор функций
  - Критерии успеха
- \* 3.3. Полноценный релиз

## 4. Категоризированные примеры с структурированным запросом

### # Перевод технических текстов

#### ## Категория 1: Программное обеспечение

Пример 1:

- Английский: The software utilizes a distributed architecture.
- Русский: Программное обеспечение использует распределенную архитектуру.

Пример 2:

- Английский: API endpoints must be secured with authentication.
- Русский: Конечные точки API должны быть защищены аутентификацией.

#### ## Категория 2: Машиностроение

Пример 1:

- Английский: The hydraulic system operates under high pressure.
- Русский: Гидравлическая система работает под высоким давлением.

Пример 2:

- Английский: Bearing failure is caused by inadequate lubrication.

- Русский: Выход подшипника из строя вызван недостаточной смазкой.

## Переведите следующий текст:

The load balancer distributes incoming network traffic across multiple servers.

## Практические рекомендации по использованию маркированной структуры

### 1. Правила создания эффективной маркированной структуры

#### Принцип иерархической организации

- Размещайте наиболее важную информацию сверху
- Группируйте связанную информацию в логические блоки
- Используйте не более 3 уровней вложенности для сохранения ясности

#### Принцип визуального разделения

- Используйте пустые строки между секциями
- Применяйте последовательную систему отступов
- Выбирайте различные маркеры для разных типов информации

#### Принцип информационной плотности

- Стремитесь к краткости в каждой точке
- Избегайте избыточных объяснений внутри структуры
- Используйте ключевые слова и фразы вместо длинных объяснений

### 2. Контрольный список для проверки промпта

- ✓ Имеет ли промпт четкую визуальную структуру?
- ✓ Разделены ли различные типы информации (контекст, инструкции, примеры)?
- ✓ Используются ли заголовки для обозначения основных секций?
- ✓ Применяются ли маркированные/нумерованные списки для связанных элементов?
- ✓ Обозначен ли ясно ожидаемый формат ответа?
- ✓ Выделены ли визуально ключевые требования или ограничения?

### 3. Адаптация для различных типов задач

#### Для аналитических задач

- Структурируйте отдельно: исходные данные, вопросы, требуемые выводы
- Используйте нумерованные шаги для процесса анализа

#### Для творческих задач

- Разделите требования к содержанию и к формату
- Используйте маркированные списки для характеристик, которые должен содержать результат

#### Для задач классификации или оценки

- Структурируйте критерии оценки с четкими рубриками
- Используйте вложенные списки для подкритериев

#### Для диалоговых агентов

- Разделите блоки для внутреннего мышления и внешних ответов
- Структурируйте правила коммуникации в маркированных списках

## Примеры эффективных маркированных промптов

### Пример 1: Структурированный промпт для медицинской консультации

# Медицинская консультация по симптомам

## Контекст и инструкции

Вы — медицинский ассистент, обученный на обширной медицинской литературе. Ваша задача — предоставить информативный, точный и хорошо структурированный ответ на медицинский запрос пациента.

## Симптомы пациента

- Головная боль (пульсирующая, в основном в правой части головы)
- Светочувствительность
- Тошнота
- Симптомы появляются 2-3 раза в неделю
- Продолжительность эпизода: около 4-6 часов

### ## Формат ответа

1. **\*\*Возможная причина\*\***: Наиболее вероятное объяснение симптомов
2. **\*\*Дополнительные возможные причины\*\***: Другие состояния, которые следует рассмотреть
3. **\*\*Рекомендации\*\***:
  - Неотложные меры (что делать сейчас)
  - Долгосрочные рекомендации
  - Когда следует обратиться к врачу
4. **\*\*Предостережения\*\***: Что не следует делать/принимать
5. **\*\*Дополнительные вопросы\*\***: Что еще нужно уточнить для более точной оценки

### ## Важно

- НЕ ставьте окончательный диагноз
- Укажите, что это информационная консультация, а не замена визита к врачу
- Используйте простой язык, понятный неспециалисту

## Пример 2: Маркированный промпт для разработки бизнес-стратегии

# Разработка стратегии выхода стартапа на международный рынок

### ## КОНТЕКСТ

- \* Стартап: SaaS-решение для автоматизации HR-процессов
- \* Текущий рынок: Россия (2 года присутствия)
- \* Финансы: привлечено \$1.5M инвестиций, месячный оборот \$80K
- \* Команда: 25 человек (15 разработчиков, 5 маркетинг/продажи, 5 операции)
- \* Ключевые отличия продукта: ИИ для подбора персонала, интеграция с местными job-порталами

### ## ЦЕЛЬ АНАЛИЗА

Разработать стратегию выхода на рынки Юго-Восточной Азии в течение следующих 12 месяцев.

### ## ТРЕБУЕМЫЕ РАЗДЕЛЫ СТРАТЕГИИ

1. Приоритизация стран для выхода
  - Топ-3 наиболее перспективных рынка
  - Критерии выбора
  - Потенциальный объем каждого рынка
2. Go-to-market стратегия
  - Рекомендуемая модель выхода (представительство/партнерство/полностью)

удаленно)

- Необходимые адаптации продукта
- Ценовая стратегия
- Каналы привлечения клиентов

### 3. Ресурсный план

- Необходимые инвестиции (приблизительно)
- Дополнительный найм
- Timeline основных этапов

### 4. Анализ рисков

- Топ-5 ключевых рисков
- Стратегии митигации

## ## ФОРМАТ ОТВЕТА

- \* Для каждого раздела: краткое вступление, затем структурированные пункты
- \* Каждый ключевой вывод должен сопровождаться кратким обоснованием
- \* Конкретные рекомендации вместо общих фраз
- \* В конце: Executive Summary (не более 200 слов)

## Пример 3: Промпт с think-then-answer структурой для анализа кода

# Анализ и оптимизация кода Python

<context>

Функция для поиска самых часто встречающихся слов в тексте:

```
```python
def find_most_common_words(text, n=10):
    words = text.lower().split()
    word_dict = {}
    for word in words:
        if word in word_dict:
            word_dict[word] += 1
        else:
            word_dict[word] = 1

    sorted_words = []
    for word, count in word_dict.items():
        sorted_words.append((word, count))
```



```
sorted_words.sort(key=lambda x: x[1], reverse=True)
```

```
result = []
```

```
for i in range(min(n, len(sorted_words))):
```

```
    result.append(sorted_words[i][0])
```

```
return result
```

</context>

<think>

- Проанализировать текущую реализацию
- Определить возможности для оптимизации (производительность, читаемость, надежность)
- Рассмотреть использование встроенных функций и библиотек Python
- Проверить обработку крайних случаев
- Подумать о возможных улучшениях алгоритма
- Оценить временную и пространственную сложность </think>

<answer>

## Анализ кода

### Проблемы текущей реализации:

1. **Неоптимальное использование структур данных** - ручное создание и сортировка словаря
2. **Отсутствие предобработки текста** - нет удаления знаков препинания, специальных символов
3. **Ограниченная обработка крайних случаев** - нет проверки на пустой ввод
4. **Избыточные операции** - лишние циклы для сортировки и создания результата

### Оптимизированный код:

```
from collections import Counter
```

```
import re
```

```
def find_most_common_words(text, n=10):
```

```
    if not text:
```

```
        return []
```

```
# Очистка текста от знаков препинания и приведение к нижнему регистру
cleaned_text = re.sub(r'[^\w\s]', '', text.lower())

# Использование Counter для подсчета и сортировки слов
word_counts = Counter(cleaned_text.split())

# Возврат n самых частых слов
return [word for word, _ in word_counts.most_common(n)]
```

### Преимущества оптимизированного решения:

1. **Эффективность:** Использование Counter из стандартной библиотеки значительно оптимизирует подсчет и сортировку
2. **Читаемость:** Код стал короче и выразительнее
3. **Надежность:** Добавлена обработка пустого ввода
4. **Функциональность:** Добавлена очистка текста от знаков препинания

### Временная и пространственная сложность:

- Временная сложность:  $O(n \log n)$ , где  $n$  - количество слов
- Пространственная сложность:  $O(n)$

### Пример 4: Промпт с категоризированной структурой для сложного запроса

## Сравнительный анализ облачных сервисов для ML-проектов

### Требования к анализу

- Сравнить AWS, Azure и Google Cloud для ML-проектов
- Фокус на возможностях для обучения и деплоя крупных языковых моделей
- Учитывать как технические, так и экономические аспекты
- Ориентация на средний бизнес с ограниченным бюджетом

# Категории сравнения

## 1. Базовая инфраструктура

- Доступность и типы GPU
- Возможности масштабирования
- Стабильность и SLA

## 2. Специализированные ML-сервисы

- Готовые решения для NLP
- Инструменты для разметки данных
- API для языковых моделей

## 3. Экономика и ценообразование

- Модели оплаты
- Стоимость обучения моделей
- Стоимость инференса

## 4. Экосистема и интеграции

- Интеграция с популярными ML-фреймворками
- CI/CD для ML-моделей
- Мониторинг и аналитика

## Формат результатов

- Для каждой категории: таблица сравнения по ключевым параметрам
- Рейтинг от 1 до 5 для каждого параметра с кратким обоснованием
- В конце каждой секции: краткие выводы о лидере в данной категории
- Заключение с общими рекомендациями по выбору платформы для разных сценариев использования

## Комбинирование с другими техниками промпт-инжиниринга

Маркированная структура отлично комбинируется с другими методами промпт-инжиниринга:

### 1. Маркированная структура + Chain-of-Thought

Структурирование цепочки рассуждений с помощью маркеров и заголовков делает

процесс мышления более организованным и прослеживаемым:

# Решение задачи комбинаторной оптимизации

## Шаг 1: Понимание проблемы

- Задача: [описание задачи]
- Что нам известно:
  - Ограничение 1
  - Ограничение 2
- Что нужно найти:

## Шаг 2: Формализация

- Математическая модель:
  - Переменные:
  - Целевая функция:
  - Ограничения:

## Шаг 3: Выбор метода решения

- Возможные подходы:
  - Подход 1: [преимущества/недостатки]
  - Подход 2: [преимущества/недостатки]
- Выбранный метод и обоснование:

## Шаг 4: Решение

- Применение выбранного метода:
  1. [шаг 1]
  2. [шаг 2] ...

## Шаг 5: Проверка и оптимизация

- Проверка ограничений:
- Возможности для улучшения:

## Итоговое решение

[Результат]

### 2. Маркированная структура + Few-shot Learning

Структурирование примеров в категории с четкой маркировкой улучшает их усвоение моделью:

## Классификация отзывов клиентов по эмоциональной окраске

### Примеры положительных отзывов

1. "Отличный сервис, всем рекомендую!" -> ПОЛОЖИТЕЛЬНЫЙ
2. "Очень доволен качеством продукта." -> ПОЛОЖИТЕЛЬНЫЙ

### Примеры нейтральных отзывов

1. "Продукт соответствует описанию." -> НЕЙТРАЛЬНЫЙ
2. "Доставка заняла указанное время." -> НЕЙТРАЛЬНЫЙ

### Примеры отрицательных отзывов

1. "Ужасное качество, не рекомендую." -> ОТРИЦАТЕЛЬНЫЙ
2. "Разочарован сервисом, больше не обращусь." -> ОТРИЦАТЕЛЬНЫЙ

## Отзывы для классификации

1. [первый отзыв]
2. [второй отзыв] ...

### 3. Маркированная структура + Самопроверка

Создание структуры для самопроверки ответов:

# Проверка фактической точности утверждений

## Утверждение для проверки

"Альберт Эйнштейн разработал теорию относительности в 1915 году во время работы в Швейцарии."

## Процесс проверки

### 1. Разбивка на фактические компоненты:

- Компонент 1: Эйнштейн разработал теорию относительности
- Компонент 2: Это произошло в 1915 году
- Компонент 3: В это время он работал в Швейцарии

### 2. Проверка каждого компонента:

- Компонент 1: [проверка]
- Компонент 2: [проверка]
- Компонент 3: [проверка]

### 3. Выявление неточностей:

- Неточность 1: [описание]
- Неточность 2: [описание]

### 4. Исправление утверждения:

- Исправленное утверждение: [корректная версия]

## Обоснование исправлений

- Исправление 1: [источники и пояснение]
- Исправление 2: [источники и пояснение]

## Когда и как использовать маркированную структуру

### Оптимальные сценарии применения

### 1. \*\*Сложные многоэтапные запросы\*\*

- Когда задача требует нескольких этапов анализа или действий

- Когда нужно явно разделить разные аспекты проблемы

2. **\*\*Запросы, требующие структурированных ответов\*\***

- Создание планов, стратегий, аналитических отчетов
- Когда важна организация информации в конечном результате

3. **\*\*Задачи с многоуровневыми требованиями\*\***

- Когда есть основные и второстепенные требования
- Когда необходима иерархия приоритетов

4. **\*\*При использовании примеров (few-shot learning)\*\***

- Для четкого разделения примеров и основного запроса
- Для категоризации различных типов примеров

### ### Потенциальные ограничения

1. **\*\*Перегруженность структурой\*\***

- Слишком сложная структура может отвлекать от сути запроса
- Рекомендуется ограничиваться 3-4 уровнями вложенности

2. **\*\*Избыточное форматирование\*\***

- Добавление форматирования без содержательной необходимости
- Важно, чтобы структура служила содержанию, а не наоборот

3. **\*\*Потеря естественности запросов\*\***

- Чрезмерно структурированные запросы могут звучать механически
- Для некоторых задач более подходит свободный формат

### ## Заключение

Маркированная структура промптов (Bulleted Structure) — это не просто визуальное форматирование, а мощный инструмент, который значительно улучшает качество взаимодействия с языковыми моделями. Исследования показывают, что хорошо структурированные промпты могут повысить точность ответов на 15-30% благодаря улучшению механизмов внимания модели, более эффективному использованию контекстного буфера и снижению когнитивной нагрузки.

Ключевые преимущества этой техники:

- **\*\*Улучшение понимания\*\*** сложных многоэтапных запросов
- **\*\*Повышение вероятности соблюдения\*\*** всех требований промпта
- **\*\*Структурирование мышления\*\*** модели и получаемых ответов
- **\*\*Снижение вероятности упущения\*\*** важных деталей
- **\*\*Улучшение организации информации\*\*** в ответах

Эффективная маркированная структура использует иерархию, визуальное разделение и информационную плотность для создания промптов, которые максимально эффективно задействуют возможности языковых моделей.

Внедрение этой техники в вашу практику работы с языковыми моделями может значительно повысить качество получаемых результатов, особенно для сложных аналитических, творческих и многоэтапных задач.