

# Изучение понимания кода в научном программировании: предварительные выводы от исследователей

Дата: 2025-01-17 00:00:00

Ссылка на исследование: <https://arxiv.org/pdf/2501.10037>

Рейтинг: 65

Адаптивность: 75

## Ключевые выводы:

Исследование направлено на изучение понимания кода в научном программировании путем опроса 57 ученых-исследователей из различных дисциплин. Основные результаты показывают, что большинство ученых осваивают программирование самостоятельно или на рабочем месте, при этом 57.9% не имеют формального обучения написанию читаемого кода. Несмотря на признание важности читаемости кода для научной воспроизводимости, исследователи сталкиваются с проблемами недостаточной документации и плохих соглашений об именовании, а также наблюдается тенденция к использованию больших языковых моделей для улучшения качества кода.

## Объяснение метода:

Исследование выявляет конкретные проблемы с читаемостью кода (недостаточное комментирование, плохое именование, неудачная структура), которые пользователи могут учитывать при формулировании запросов к LLM и оценке результатов. Тенденция использования LLM для улучшения кода подтверждает ценность этого подхода для широкой аудитории.

**## Ключевые аспекты исследования:** 1. **Образовательный фон и практики программирования научных исследователей:** Исследование показало, что большинство ученых осваивают программирование самостоятельно или в процессе работы, а 57.9% не имеют формального обучения написанию читаемого кода. Основные языки - Python и R.

**Проблемы понимания научного кода:** Основные трудности включают недостаточное комментирование (44 участника), отсутствие документации (33), плохое наименование функций/переменных (31), и неудачную организацию структуры проекта (31).

**Проблемы с именами идентификаторов:** Наиболее частые проблемы - слишком

короткие или непонятные имена (40), несогласованные соглашения об именовании (30), имена, не отражающие назначение (29).

**Использование инструментов для улучшения кода:** 49.12% участников никогда не используют автоматизированные инструменты для улучшения качества кода. Среди использующих, многие обращаются к ИИ и LLM (особенно ChatGPT и Claude).

**Важность читаемости для воспроизводимости:** Все участники признают важность читаемости кода для обеспечения воспроизводимых научных результатов, 83.76% считают это очень или чрезвычайно важным.

## Дополнение: Исследование не требует дообучения или API для применения его выводов в стандартных чатах с LLM. Основные концепции и подходы исследования могут быть непосредственно использованы обычными пользователями в стандартных чатах.

Ключевые концепции, применимые в стандартном чате:

**Чеклист типичных проблем с кодом:** Пользователи могут использовать выявленные в исследовании проблемы (недостаточное комментирование, плохое именование, неудачная структура) как чеклист при запросе LLM проверить или улучшить их код. Например: "Проверь мой код на наличие следующих проблем: недостаточное комментирование, непонятные имена переменных, несогласованные соглашения об именовании."

**Улучшение именования:** Пользователи могут конкретно запрашивать улучшение именования в своем коде, указывая на проблемы, выявленные в исследовании: "Переименуй переменные и функции, избегая слишком коротких имен, общих терминов и несогласованных стилей."

**Структурирование документации:** Исследование показывает важность документации для понимания кода. Пользователи могут запрашивать: "Добавь к коду необходимые комментарии и создай README, объясняющий структуру и назначение программы."

**Критическая оценка генерируемого кода:** Понимая типичные проблемы с кодом, пользователи могут более критично оценивать код, сгенерированный LLM, и запрашивать конкретные улучшения.

Результаты от применения этих концепций: - Более читаемый и поддерживаемый код - Лучшее понимание собственного кода и кода, сгенерированного LLM - Более эффективное взаимодействие с LLM при работе с кодом - Повышение воспроизводимости результатов научных исследований - Сокращение времени на понимание и отладку кода

## Анализ практической применимости: 1. **Образовательный фон и практики программирования** - Прямая применимость: Понимание того, что большинство пользователей LLM без технического образования имеют схожие пробелы в знаниях

о написании качественного кода поможет формулировать запросы с учетом этих ограничений. - Концептуальная ценность: Осознание важности структурированного подхода к программированию даже при отсутствии формального образования. - Потенциал для адаптации: Пользователи могут запрашивать у LLM помощь в улучшении структуры и читаемости своего кода, понимая типичные проблемы.

**Проблемы понимания научного кода** Прямая применимость: Знание наиболее распространенных проблем позволяет пользователям формулировать более точные запросы к LLM для улучшения кода. Концептуальная ценность: Понимание ключевых аспектов, влияющих на читаемость кода, помогает оценивать качество генерируемого LLM кода. Потенциал для адаптации: Пользователи могут создавать чеклисты для проверки своего кода на основе выявленных проблем.

### **Проблемы с именами идентификаторов**

Прямая применимость: Конкретные проблемы с именованием могут служить руководством при запросе к LLM об улучшении именования в коде. Концептуальная ценность: Понимание важности качественного именования для понятности кода. Потенциал для адаптации: Пользователи могут использовать эти знания для создания собственных конвенций именования и запрашивать LLM следовать им.

### **Использование инструментов для улучшения кода**

Прямая применимость: Растущая тенденция использования LLM для улучшения кода подтверждает правильность такого подхода для широкой аудитории. Концептуальная ценность: Осознание необходимости критически оценивать код, сгенерированный LLM. Потенциал для адаптации: Пользователи могут комбинировать традиционные инструменты с LLM для достижения лучших результатов.

### **Важность читаемости для воспроизводимости**

Прямая применимость: Понимание важности читаемого кода мотивирует пользователей запрашивать у LLM более качественные решения. Концептуальная ценность: Осознание связи между качеством кода и воспроизводимостью результатов. Потенциал для адаптации: Пользователи могут применять эти знания при работе над долгосрочными проектами, где важна воспроизводимость.

## **Prompt:**

Применение Результаты Исследования о Понимании Кода в Научном Программировании для Промтов GPT ## Ключевые инсайты для использования в промтах

Исследование предоставляет ценные данные о том, как ученые работают с кодом и с какими проблемами сталкиваются. Эту информацию можно эффективно использовать при составлении промтов для GPT, особенно когда требуется помощь с научным программированием.

## ## Пример промта

[=====] Я исследователь в области [область науки], использующий Python для анализа данных. У меня нет формального образования в программировании, как и у 57.9% ученых согласно исследованиям. Помоги мне переработать следующий фрагмент кода с учетом лучших практик:

[код]

Пожалуйста: 1. Добавь подробные комментарии, так как недостаточная документация - главная проблема понимания научного кода 2. Улучши именование переменных для большей ясности 3. Реорганизуй код в логические модули 4. Объясни изменения, которые ты внес, простым языком 5. Предложи документацию, которую стоит добавить в README проекта [=====]

## ## Как работают знания из исследования в этом промте

**Учет образовательного фона:** Промт учитывает, что большинство ученых (57.9%) не имеют формального образования в написании читаемого кода, что помогает GPT адаптировать объяснения.

**Фокус на главных проблемах:** Промт направлен на решение основных проблем, выявленных в исследовании:

Недостаточность комментариев (16.18%) Плохая документация (12.13%) Неудачные соглашения об именовании

**Практическое применение:** Запрос включает конкретные действия из раздела "Практические применения" исследования:

Улучшение комментариев и документации Внедрение осмысленных соглашений об именовании Модульная организация кода

**Учет важности воспроизводимости:** Промт косвенно затрагивает вопрос научной воспроизводимости, который 83.76% ученых считают важным или чрезвычайно важным.

Такой подход к составлению промтов помогает получить от GPT более релевантную помощь, адаптированную к реальным потребностям научного сообщества.