

Генерация псевдокода и его преобразование в программный код с использованием GPT-5.2

Кокухина Софья Александровна

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

В статье рассматриваются возможности использования языковой модели GPT-5.2 для генерации псевдокода по текстовому описанию алгоритмов и его последующего преобразования в программный код. Анализируется способность модели интерпретировать естественно-языковые запросы, формировать структурированные алгоритмические описания и адаптировать их к синтаксису различных языков программирования. Рассматривается влияние степени детализации промпта на корректность и полноту получаемых результатов.

Ключевые слова: GPT-5.2, псевдокод, программирование, генерация кода, алгоритмы, искусственный интеллект, автоматизация разработки.

Pseudocode generation and its transformation into program code using GPT-5.2

Kokukhina Sofya Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article explores the capabilities of the GPT-5.2 language model in generating pseudocode from textual descriptions and converting it into executable programming code. The study analyzes how the model interprets natural language input, produces structured algorithmic representations, and adapts them to different programming languages. The impact of prompt detail on the correctness and completeness of generated results is examined.

Keywords: GPT-5.2, pseudocode, programming, code generation, algorithms, artificial intelligence, software development.

1. Введение

1.1 Актуальность

Разработка программного обеспечения традиционно включает этап формализации алгоритма, который часто представляется в виде псевдокода. Псевдокод позволяет абстрагироваться от синтаксических особенностей конкретного языка программирования и сосредоточиться на логике решения задачи. Однако для начинающих разработчиков переход от текстового

описания задачи к формализованному алгоритму и затем к программному коду представляет значительную сложность.

Современные языковые модели, такие как GPT-5.2, способны обрабатывать естественный язык и генерировать структурированные алгоритмические описания, что открывает возможности автоматизации данного процесса. Использование таких моделей позволяет упростить этап проектирования алгоритмов, сократить время разработки и повысить доступность обучения программированию.

1.2 Обзор исследований

А.К. Ковалёв и А.И. Панов рассказали про подход, который использует способность больших языковых моделей, обученных на большом объеме неразмеченных данных, хранить знания общего назначения (common sense). Рассмотрели модели, использующие про данный подход в задачах воплощенного искусственного интеллекта [1]. К.А. Ромаданский представил две модели для работы с русским языком: генератор псевдослов и классификатор, оценивающий степень схожести введенной последовательности символов с настоящими словами. Классификатор использован для оценки результатов генератора. Обе модели основаны на рекуррентной нейронной сети с долгой краткосрочной памятью и обучены на датасете существительных русского языка [2]. Р.К. Лебедев провел исследование по возможности использовать ChatGPT как большую языковую модель, для генерации реалистичного случайного программного кода, который может быть использован в качестве кода-приманки для препятствования реверс-инжинирингу [3]. А.В. Куртукова и А.С. Романов проанализировали современные методы генерации исходного кода программы и выбрали наиболее эффективный из них с целью проведения исследований, направленных на разграничение авторства кода между человеком и машиной [4]. В статье А.А. Вяткина и В.Д. Олисеенко исследуется вопрос ответов на большие таблицы с помощью генерации pandas-кода. Приводится описание метода генерации кода, при котором в промпте, подаваемом LLM, описывается лишь часть таблицы - названия столбцов и примеры строк [5].

1.3 Цель исследования

Целью исследования является анализ возможностей языковой модели GPT-5.2 в генерации псевдокода по текстовому описанию задачи и его последующего преобразования в программный код.

2 Материалы и методы

В рамках исследования используется языковая модель GPT-5.2, применяемая для генерации псевдокода и программного кода на основе текстовых запросов. В качестве входных данных используются описания алгоритмических задач различной сложности, включающие операции над массивами, поиск значений и обработку числовых данных.

Методика исследования включает последовательное формирование текстового описания задачи, генерацию псевдокода, последующее преобразование полученного алгоритма в программный код и анализ корректности результатов

3. Результаты

3.1 Обработка массива: поиск максимального элемента

Для начала рассматривается базовая задача обработки массива.

Промпт:

«Сформируй псевдокод для поиска максимального элемента в массиве чисел, а затем преобразуй его в код на Python.»

Сгенерированный псевдокод корректно отражает логику поиска максимального значения, а программная реализация полностью соответствует алгоритму и может быть выполнена без дополнительной доработки (Рис. 1).

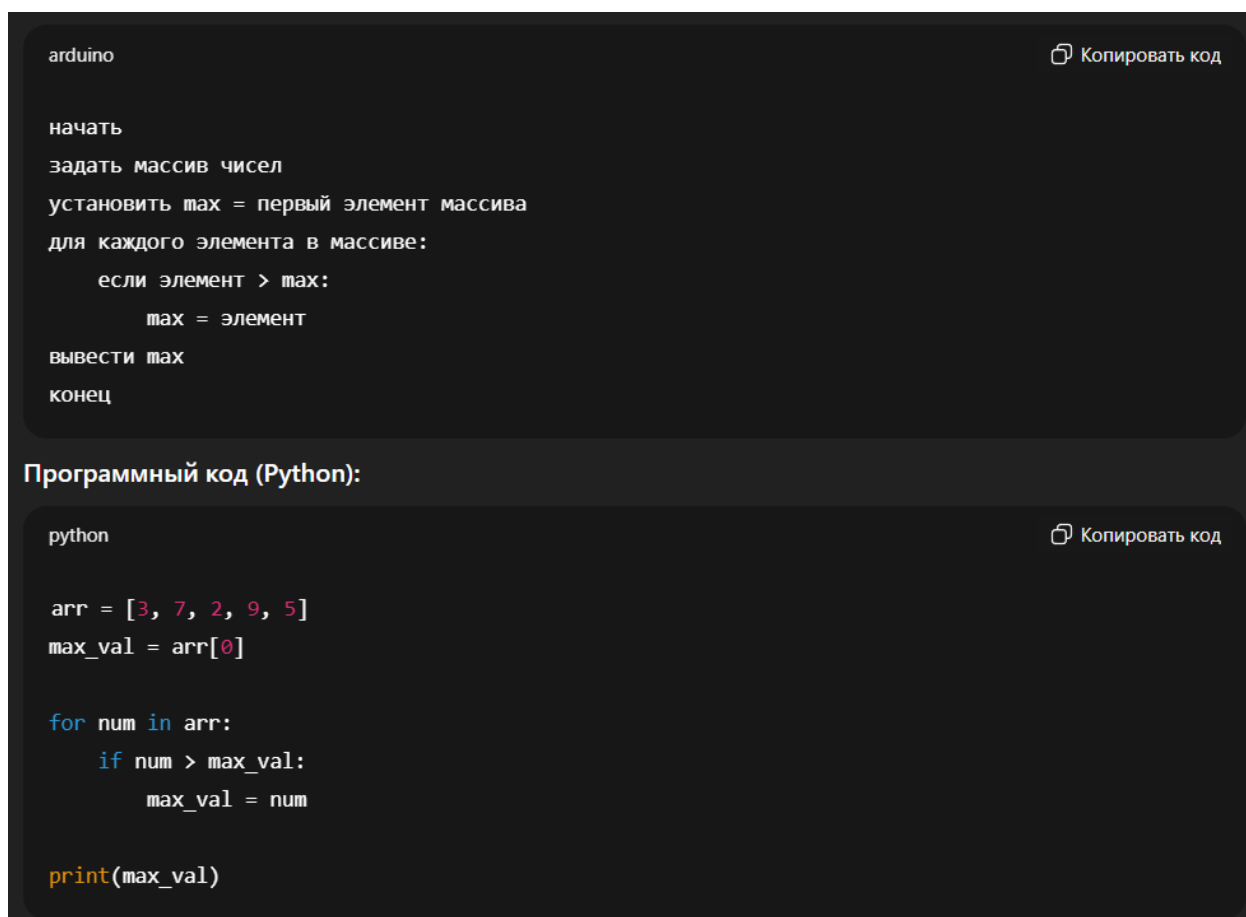


Рисунок 1 – Псевдокод обработки массива и его перевод на Python

3.2 Условные конструкции: проверка чётности числа

На втором этапе проверяется способность модели работать с простыми логическими условиями.

Промпт:

«Опиши псевдокод алгоритма проверки, является ли число чётным, и напиши код на Python.»

Модель корректно использовала оператор остатка от деления и сформировала понятную структуру условия, что делает результат пригодным для обучения базовым конструкциям программирования (Рис. 2).

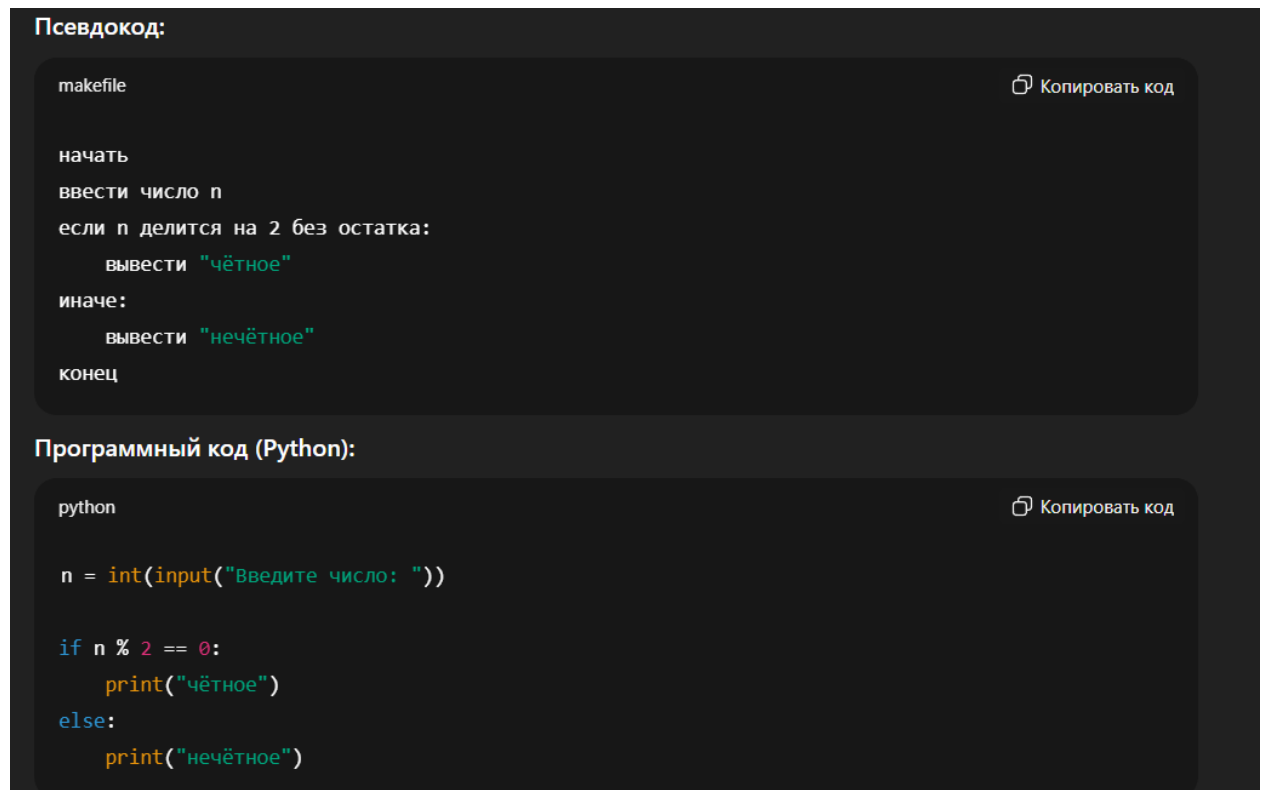


Рисунок 2 – Псевдокод и его интерпретация для проверки чётности

3.3 Работа с циклами: вычисление суммы элементов

На третьем этапе анализируется генерация алгоритмов с использованием циклов.

Промпт:

«Сгенерируй псевдокод для нахождения суммы элементов массива и преобразуй его в код на Python.»

Полученные результаты демонстрируют корректное использование итеративного подхода и аккуратную трансляцию псевдокода в синтаксис языка Python (Рис. 3).

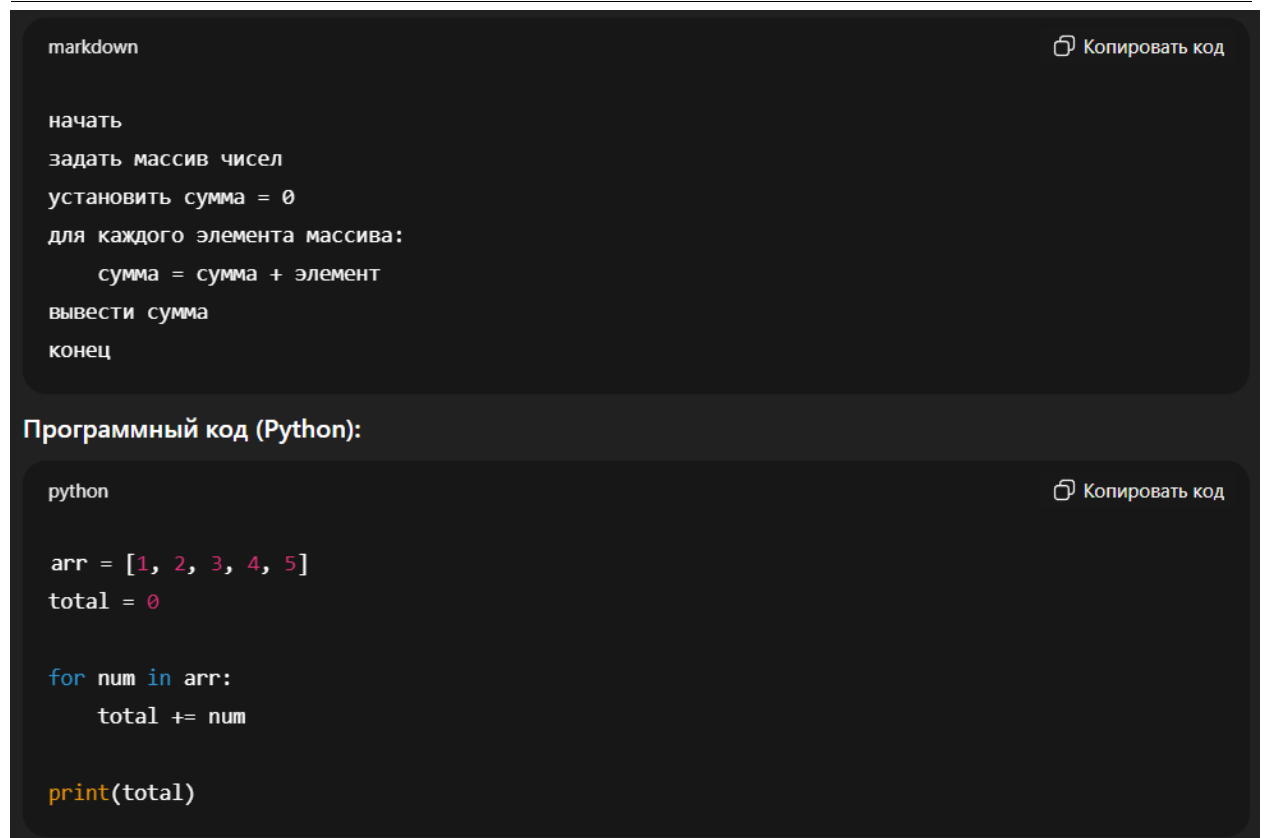


Рисунок 3 – Результат запроса для вычисления суммы

3.4 Генерация алгоритма с функцией: вычисление факториала

На заключительном этапе рассмотрена задача более высокого уровня сложности, включающая использование функций.

Промпт:

«Создай псевдокод алгоритма вычисления факториала числа с использованием функции, затем напиши реализацию на Python.»

Модель продемонстрировала способность формировать рекурсивные алгоритмы и корректно переносить их в программный код, что свидетельствует о более глубоком понимании структуры алгоритмов (Рис.4).

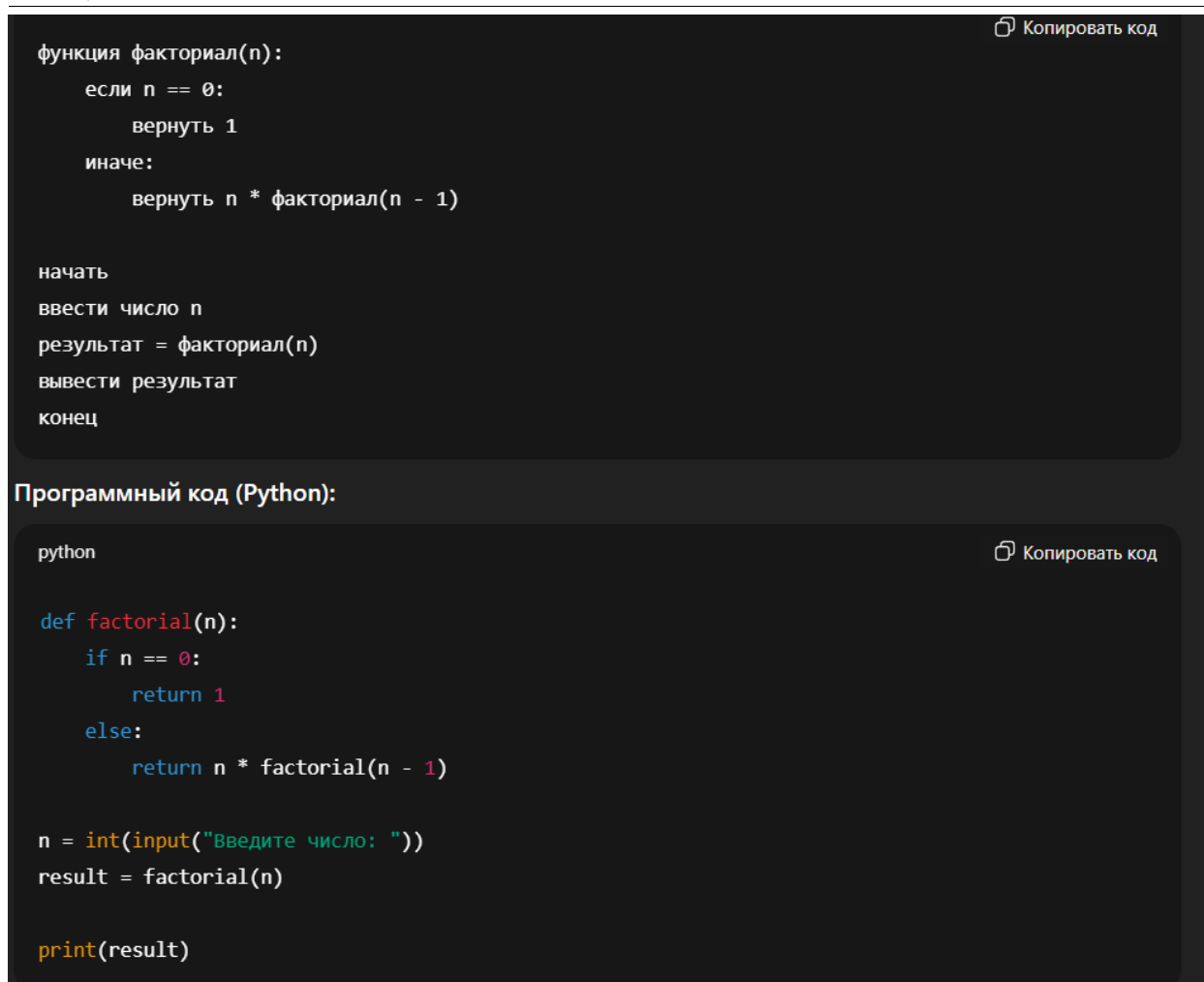


Рисунок 4 – Псевдокод и его перевод для вычисления факториала

4 Выводы

Проведённые эксперименты показывают, что GPT-5.2 успешно справляется с генерацией псевдокода для задач различной сложности и его преобразованием в программный код. Полученные алгоритмы отличаются логической последовательностью, корректной структурой и соответствием исходным условиям задач.

Качество результатов зависит от точности формулировки промпта, однако даже при относительно кратких запросах модель демонстрирует устойчиво высокий уровень генерации. Практическая часть подтверждает возможность использования GPT-5.2 как инструмента обучения алгоритмизации и программированию, а также как средства ускорения разработки простых программных решений.

Библиографический список

1. Ковалев А. К., Панов А.И. Применение предобученных больших языковых моделей в задачах воплощенного искусственного интеллекта // Доклады Российской академии наук. Математика, информатика, процессы управления. 2022. Т. 508, № 1. С. 94-99.

2. Ромаданский К. А., Ахаев А.Е., Гилязов Т.Р. Создание генератора псевдослов и классификация их схожести со словами словаря русского языка методами машинного обучения // Электронные библиотеки. 2025. Т. 28, № 1. С. 145-162.
3. Лебедев Р. К. Использование ChatGPT для генерации реалистичного случайного программного кода для затруднения реверс-инжиниринга // Информационные технологии. Научный инжиниринг: Материалы 61-й Международной научной студенческой конференции. Новосибирский национальный исследовательский государственный университет. 2023. С. 10.
4. Куртукова А. В., Романов А.С. Проблема искусственно сгенерированных исходных кодов в задаче идентификации автора программы // Сборник избранных статей научной сессии ТУСУР. 2022. № 1-2. С. 131-134.
5. Вяткин А. А., Олисеенко В.Д. Генерация pandas-кода для ответов по большим таблицам с помощью LLM // Международная конференция по мягким вычислениям и измерениям. 2025. Т. 1. С. 195-198.