

## АРХИТЕКТУРА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ TELEGRAM-БОТА ДЛЯ АВТОМАТИЗАЦИИ МАТЕМАТИЧЕСКИХ ВЫЧИСЛЕНИЙ

Ильясова А.К., Яголичев А.В.

*Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Астраханский государственный технический университет», Астрахань,  
Российская Федерация, e-mail: Ilyasova77@mail.ru*

При всем разнообразии онлайн-приложений актуальной проблемой остается существование ограниченного количества бесплатных сервисов с поддержкой русскоязычного интерфейса, способных использовать интегрированный подход для комплексного решения математических задач. Целью исследования является создание multifunctional Telegram-бота, способного автоматизировать решение широкого спектра математических задач и повысить доступность вычислений для пользователей. Для реализации бота использованы языки программирования JavaScript и Python, а также специализированные библиотеки для символьных и численных вычислений, генерации графических представлений и организации взаимодействия между компонентами через WebSocket. В ходе исследования реализованы функции решения уравнений, вычисления производных и интегралов, построения графиков, выполнения матричных операций и статистического анализа данных. Особое внимание уделено удобству пользовательского интерфейса, поддержке ввода выражений в естественной форме, а также возможности сохранения истории запросов и результатов. Проведенная апробация показала эффективность предложенного подхода и перспективность дальнейшего развития интеллектуальных образовательных сервисов на основе мессенджер-платформ. Разработанный Telegram-бот может стать востребованным инструментом поддержки обучения и самообразования для студентов, инженеров и научных работников, способствуя развитию цифровых образовательных технологий.

**Ключевые слова:** Telegram-бот, автоматизация, математические вычисления, LaTeX, WebSocket, символьные вычисления, образовательные технологии, Python, JavaScript, пользовательский интерфейс, чат-бот, искусственный интеллект

## ARCHITECTURE AND SOFTWARE IMPLEMENTATION OF A TELEGRAM BOT FOR AUTOMATING MATHEMATICAL CALCULATIONS

Ilyasova A.K., Yagolichev A.V.

*Federal State Budgetary Educational Institution of Higher Education  
"Astrakhan State Technical University", Astrakhan, Russian Federation,  
e-mail: Ilyasova77@mail.ru*

Among the wide variety of online applications, the limited number of free services with Russian-language interfaces capable of using an integrated approach to comprehensively solving mathematical problems remains a pressing issue. The goal of this research is to create a multifunctional Telegram bot capable of automating the solution of a wide range of mathematical problems and increasing the accessibility of computations for users. The bot was implemented using JavaScript and Python programming languages, as well as specialized libraries for symbolic and numerical calculations, generating graphical representations, and organizing interactions between components via WebSocket. The research included functions for solving equations, calculating derivatives and integrals, plotting graphs, performing matrix operations, and statistical data analysis. Particular attention was paid to the user interface's usability, support for entering expressions in natural form, and the ability to save a history of queries and results. The pilot study demonstrated the effectiveness of the proposed approach and its potential for the further development of intelligent educational services based on messenger platforms. The developed Telegram bot could become a sought-after tool for students, engineers, and researchers, contributing to the development of digital educational technologies.

**Keywords:** Telegram bot, automation, mathematical computations, LaTeX, WebSocket, symbolic computation, Python, educational technologies, JavaScript, user interface, chatbot, artificial intelligence

### Введение

В современном мире представлен большой выбор приложений и веб-ресурсов, предназначенных для автоматизации решения широкого спектра математических задач различной сложности. Однако актуальной проблемой остается существование ограниченного количества сервисов, способных использовать интегрированный подход для комплексного их решения. Анализ существующих математических он-

лайн-приложений показал, что многие такие платформы предоставляют расширенные возможности только платно или не поддерживают интерфейс на русском языке. Создание Telegram-бота, способного решать широкий спектр математических задач, является актуальным и перспективным направлением в сфере образовательных технологий. Такой инструмент может стать персональным помощником для студентов, инженеров, научных работников и всех, кто

сталкивается с вычислениями в своей деятельности. Он позволит оперативно получать результаты, экономить время на поиск и использование разрозненных онлайн-сервисов и программных пакетов, а также повысить эффективность решения. А.И. Абрамова в своей статье «Использование Telegram-бота в образовательном процессе вуза» [1] отмечает, что подобный подход обладает значительными преимуществами и облегчает процесс обучения студентов. Аналогичное мнение выражено в статье Б.С. Горячкина, Д.А. Галичия, В.С. Цапия, В.В. Бурашникова, Т.Ю. Крутова «Эффективность использования чат-ботов в образовательном процессе» [2]. Авторы подчеркивают, что при грамотной разработке бота, включающей добавление необходимых студентам функций, определение стиля общения бота, обеспечение использования только проверенных сервисов и библиотек, можно создать эффективного и полезного помощника для студентов.

Функционал предлагаемого приложения включает в себя решения алгебраических уравнений, вычисление производных и интегралов, построение графиков функций, выполнение матричных операций, решение задач линейной алгебры и статистического анализа данных [3–5]. Важным аспектом является возможность обработки математических выражений, вводимых пользователем в естественной форме, что значительно упрощает взаимодействие с ботом.

Разработка интуитивно понятного и удобного интерфейса является одним из ключевых аспектов. Бот должен предоставлять пользователю четкие инструкции по использованию, предлагать различные опции и параметры для решения задач и отображать результаты вычислений в структурированном виде. Также важно обеспечить возможность сохранения истории запросов и результатов, что позволит вернуться к предыдущим вычислениям для их сравнения. Создание такого приложения способствует развитию образовательных и научных инструментов, обеспечивая удобный и эффективный способ решения математических задач.

**Целью исследования** является разработка многофункционального Telegram-бота для автоматизации решения широкого спектра математических задач.

**Задачи исследования:**

- проанализировать и выбрать библиотеки Node.js и Python для реализации Telegram-бота;
- реализовать основные математические функции, включая решение уравнений, вычисление производных и интегралов;

- разработать удобный пользовательский интерфейс с поясняющими сообщениями;

- обеспечить вывод решений в формате LaTeX и интеграцию Python-библиотек для генерации изображений;

- организовать взаимодействие компонентов через WebSocket.

### Материалы и методы исследования

Реализация Telegram-бота осуществляется с использованием языков программирования JavaScript и Python, что обусловлено необходимостью интеграции инструментов для символьных вычислений, генерации графических представлений и обеспечения взаимодействия с пользователем. [6, 7]. Основная логика работы бота реализована на платформе Node.js, где посредством менеджера пакетов npm были установлены и использованы следующие библиотеки: Node.js Telegram Bot API, path, fs, Nerdamer, Algebrite, Math.js [8, 9]. Библиотека Node.js Telegram Bot API обеспечивает обработку запросов и взаимодействие с Telegram API, path и fs используются для работы с файловой системой, а Nerdamer, Algebrite и Math.js – для выполнения символьных и численных математических вычислений различной сложности [10, 11].

Для генерации изображений с решениями в формате LaTeX применяются библиотеки Python: Matplotlib, SymPy и ге. Matplotlib и SymPy обеспечивают построение и визуализацию математических выражений, а библиотека ге используется для обработки и валидации входных данных с помощью регулярных выражений.

Взаимодействие между компонентами, реализованными на JavaScript и Python, организовано с использованием технологии WebSocket. Для этого в Python-скрипте, отвечающем за создание изображений, были задействованы библиотеки asyncio и websockets, а на стороне Node.js – библиотека ws [12].

Алгоритм работы компонентов реализован следующим образом: после получения команды от пользователя и выполнения соответствующих вычислений на стороне Node.js, данные задачи и полученный результат передаются на Python-сервер посредством WebSocket. В Python-скрипте эти данные преобразуются в изображение, которое сохраняется на сервере. После завершения процесса генерации изображения Python-сервер отправляет уведомление о готовности файла, и Node.js-скрипт пересылает полученное изображение пользователю в Telegram, обеспечивая автоматизацию процесса и наглядное представление результатов вычислений, существенно повышая удобство использования разрабатываемого программного продукта [13].

Взаимодействие с Telegram-ботом реализуется посредством интуитивно понятного интерфейса, предусматривающего выбор одной из основных функциональных возможностей: решение алгебраических уравнений степеней I–III, вычисление производных функций, определенных и неопределенных интегралов, упрощение алгебраических выражений, построение графиков, выполнение матричных операций, а также решение задач линейной алгебры и статистического анализа данных. Для каждой из указанных задач предус-

мотрен пошаговый ввод исходных данных с последующей генерацией результатов в текстовом и графическом форматах. Такой подход обеспечивает широкую доступность сервиса для пользователей с различным уровнем подготовленности и способствует повышению эффективности выполнения математических вычислений [14, 15].

### Результаты исследования и их обсуждение

Фрагмент части кода с описанием.

#### Решение кубических уравнений

```
function thirdDegree(a, b, c, d) {
  if (isNaN(+a) || isNaN(+b) || isNaN(+c) || isNaN(+d)) {
    return "Неправильный ввод данных";
  } else {
    const p = (3 * a * c - b ** 2) / 3 * a ** 2;
    const q = (2 * b ** 3 - 9 * a * b * c + 27 * a ** 2 * d) / 27 * a ** 3;
    const Q = (p / 3) ** 3 + (q / 2) ** 2;
    const alpha = math.cbrt(math.sum(-q / 2, math.sqrt(Q)));
    const beta = math.cbrt(math.subtract(-q / 2, math.sqrt(Q)));
    const y1 = math.sum(alpha, beta);
    const y2 = math.sum(math.divide(math.multiply(-1, y1), 2), math.multiply(math.sqrt(3), math.
multiply(math.divide(math.subtract(alpha, beta), 2), math.complex(0, 1))));
    const y3 = math.subtract(math.divide(math.multiply(-1, y1), 2), math.multiply(math.sqrt(3), math.
multiply(math.divide(math.subtract(alpha, beta), 2), math.complex(0, 1))));
    let x1, x2, x3;
    if (Q < 0) {
      x1 = y1.re - b / (3 * a);
      x2 = y2.re - b / (3 * a);
      x3 = y3.re - b / (3 * a);
    } else if (Q.toFixed(5) === 0) {
      x1 = y1 - b / (3 * a);
      x2 = y2.re - b / (3 * a);
      x3 = y3.re - b / (3 * a);
    } else {
      x1 = y1 - b / (3 * a);
      x2 = math.subtract(y2, math.divide(b, math.multiply(3, a))).toString();
      x3 = math.subtract(y3, math.divide(b, math.multiply(3, a))).toString();
    }
    const answer = [x1, x2, x3];
    for (let i = 0; i < answer.length; i++) {
      if (typeof answer[i] === "string") {
        const number = answer[i].slice(0, answer[i].length - 1);
        if (number.includes(".") && number.split(".").pop().length >= 5 && (number.split(".").pop().
includes("99999") || number.split(".").pop().includes("00000"))) {
          answer[i] = math.round(+number) + "i";
        }
      } else {
        const stringRoot = String(answer[i]);
        const fixedStringRoot = String(answer[i].toFixed(5));
        if (stringRoot.includes(".") && stringRoot.split(".").pop().length >= 5 && (fixedStringRoot.
split(".").pop().includes("99999") || fixedStringRoot.split(".").pop().includes("00000"))) {
          answer[i] = math.round(answer[i]);
        }
      }
    }
  }
  return answer;
}
```

Функция `thirdDegree` реализует алгоритм для вычисления всех корней кубического уравнения. Входные параметры – коэффициенты уравнения проверяются на корректность: если хотя бы один из них не является числом, функция сообщает о неправильном вводе данных. Основная часть алгоритма основана на методе Кардано, который позволяет аналитически находить корни кубического уравнения. Сначала вычисляются параметры, выражающие уравнение в депрессированном виде, а также дискриминант, определяющий тип корней (действительные или комплексные). Проверяется его знак, и на этом основании вычисляются корни уравнения (выполняется обратная замена). Если величина имеет отрицательный знак, то все три корня являются действительными, а значит, достаточно будет взять только действительную часть

переменных (в таком случае мнимая часть будет равна нулю). Если равна нулю, то имеем дело с повторяющимися действительными корнями. Важно отметить, что из-за погрешности может оказаться не непосредственно нулем, а крайне близким к нему числом, поэтому в условии проверяется на соответствие нулю только целая часть числа и первые пять знаков после запятой. Если величина больше нуля, то в результате получаем один вещественный корень и два сопряженных комплексных корня. По этой причине в вычислении их значений используются функции из библиотеки `Math.js`. В конце остается лишь округлить числа, имеющие в десятичной записи большое количество чисел 0 или 9.

Для решения квадратных уравнений написана функция `secondDegree`, использующая формулу дискриминанта.

```
function secondDegree(a, b, c) {
  if (isNaN(+a) || isNaN(+b) || isNaN(+c)) {
    return "Неправильный ввод данных";
  } else {
    const d = b ** 2 - 4 * a * c;
    if (d > 0) {
      if (Math.sqrt(d) % 1 === 0) {
        return ['${(-b - Math.sqrt(d)) / 2 * a}', '${(-b + Math.sqrt(d)) / 2 * a}'];
      } else {
        return [Algebrite.run(`${-b} - sqrt(${d}) / 2 * ${a}`), Algebrite.run(`${-b} + sqrt(${d}) / 2 * ${a}`)];
      }
    } else if (d === 0) {
      const x = -b / 2 * a;
      return [x];
    } else {
      if (Math.sqrt(d) % 1 === 0) {
        return ['${(-b - Math.sqrt(-d)) / 2 * a}i', '${(-b + Math.sqrt(-d)) / 2 * a}i'];
      } else {
        return [Algebrite.run(`${-b} - sqrt(${d}) / 2 * ${a} * i`), Algebrite.run(`${-b} + sqrt(${d}) / 2 * ${a} * i`)];
      }
    }
  }
}
```

Также осуществляется проверка возможности извлечения корня из дискриминанта. В случаях, когда извлечение корня невозможно, применяется функция упрощения выражения из библиотеки `Algebrite`.

Разработанный бот функционирует на персональном сервере, его идентификатор – `@UniversalMathBot`. Полученные результаты подтверждают эффективность предложенного подхода и демонстрируют перспективность дальнейшего развития интеллектуальных образовательных сервисов на основе мессенджер-платформ.

## Заключение

В ходе проведенного исследования была реализована программная архитектура многофункционального бота, предназначенного для автоматизации решения широкого спектра математических задач. Разработанный бот сочетает в себе современные алгоритмы символьных и численных вычислений, интуитивно понятный интерфейс и возможность визуализации решений в формате `LaTeX`. Интеграция технологий `Node.js` и `Python`, а также использование `WebSocket` для межязыкового взаимодействия обеспе-

чили гибкость и расширяемость программного продукта.

В перспективе планируется расширение функционала бота за счет внедрения модулей машинного обучения для распознавания рукописных формул и интеллектуального анализа ошибок пользователей. Особое внимание будет уделено обеспечению безопасности данных и масштабируемости системы для поддержки большого количества одновременных пользователей. Кроме того, планируется интеграция с внешними образовательными платформами и создание API для сторонних разработчиков, что позволит использовать бота в различных учебных и исследовательских проектах. Таким образом, реализованный Telegram-бот может стать эффективным инструментом поддержки обучения и самообразования в области математики, способствуя развитию цифровых образовательных технологий.

### Список литературы

1. Абрамова А.И. Использование telegram-бота в образовательном процессе вуза // Вестник науки. 2022. № 1 (46). Т. 3. С. 150–153. URL: <https://www.vestnik-nauki.pf/article/5148> (дата обращения: 14.10.2025). ISSN: 2712-8849.
2. Горячкин Б.С., Галичий Д.А., Цапий В.С., Бурашников В.В., Крутов Т.Ю. Эффективность использования чат-ботов в образовательном процессе // E-Scio. 2021. № 4 (55). URL: <https://cyberleninka.ru/article/n/effektivnost-ispolzovaniya-chat-botov-v-obrazovatelnom-protsesse> (дата обращения: 14.10.2025).
3. Ураев Д.А. Классификация и методы создания чат-бот приложений // International scientific review. 2019. № LXIV. URL: <https://cyberleninka.ru/article/n/klassifikatsiya-i-metody-sozdaniya-chat-bot-prilozheniy> (дата обращения: 14.10.2025).
4. Шилова С.А., Крючкова А.А. Лингводидактический потенциал чат-ботов // Иностранные языки в контексте межкультурной коммуникации. 2021. № XIII. URL: <https://cyberleninka.ru/article/n/lingvodidakticheskiy-potentsial-chat-botov> (дата обращения: 14.10.2025).
5. Трашкова С.М. Информационные технологии в образовании: теоретико-правовые аспекты // Современные образовательные технологии в мировом учебно-воспитательном пространстве. 2016. № 3. URL: <https://cyberleninka.ru/article/n/informatsionnye-tehnologii-v-obrazovanii-teoretiko-pravovye-aspekty> (дата обращения: 14.10.2025).
6. Федотов В.А. Разработка информационной системы для оптового склада на языке программирования Python // Форум молодых ученых. 2021. № 2 (54). URL: <https://cyberleninka.ru/article/n/razrabotka-informatsionnoy-sistemy-dlya-optovogo-sklada-na-yazyke-programmirovaniya-python> (дата обращения: 14.10.2025).
7. Байдыбеков А.А., Гильванов Р.Г., Молодкин И.А. Современные фреймворки для разработки web-приложений // Интеллектуальные технологии на транспорте. 2020. № 4 (24). URL: <https://cyberleninka.ru/article/n/sovremennye-freymvorki-dlya-razrabotki-web-prilozheniy> (дата обращения: 14.10.2025).
8. Винокурова Д.В. Выбор оптимального языка программирования для генерации математических задач // Вестник НГУ. Серия: Информационные технологии. 2024. № 3. URL: <https://cyberleninka.ru/article/n/vybor-optimalnogo-yazyka-programmirovaniya-dlya-generatsii-matematicheskikh-zadach> (дата обращения: 14.10.2025).
9. Акишин Б.А. Особенности решения математических задач в среде Python // ДМ. 2019. № 49. URL: <https://cyberleninka.ru/article/n/osobennosti-resheniya-matematicheskikh-zadach-v-srede-python> (дата обращения: 14.10.2025).
10. Тестов В.А., Попков Р.А. Исследовательское обучение математике и системы компьютерной алгебры // Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика. 2024. № 4 (53). URL: <https://cyberleninka.ru/article/n/issledovatel'skoe-obucheniye-matematike-i-sistemy-kompyuternoy-algebry> (дата обращения: 14.10.2025).
11. Валинурова А.А., Балабанова Н.В., Маценков И.А. Алгоритм разработки telegram-бота – продуктивного помощника современного бизнеса // Современные наукоемкие технологии. Региональное приложение. 2023. № 2 (74). URL: <https://cyberleninka.ru/article/n/algoritm-razrabotki-telegram-bota-produktivnogo-pomoschnika-sovremennogo-biznesa> (дата обращения: 14.10.2025).
12. Lykova K.G., Schuchka T.A., Gnezdilova N.A. The use of stochastic approaches in computational modeling of professional tasks in the training of future personnel in the field of physical education and sports // Theory and Practice of Physical Culture. 2025. № 4. URL: <https://cyberleninka.ru/article/n/the-use-of-stochastic-approaches-in-computational-modeling-of-professional-tasks-in-the-training-of-future-personnel-in-the-field> (дата обращения: 14.10.2025).
13. Макаров И.С., Ларин Д.В., Воробьева Е.Г., Емелин Д.П., Карташов Д.А. Влияние асинхронных и многопоточных моделей обработки запросов на производительность серверных веб-приложений // Программные системы и вычислительные методы. 2025. № 1. URL: <https://cyberleninka.ru/article/n/vliyanie-asinhronnyh-i-mnogopotochnyh-modeley-obrabotki-zaprosov-na-proizvoditelnost-servernykh-veb-prilozheniy> (дата обращения: 14.10.2025).
14. Васькин В.А., Шибайкин С.Д., Никулин В.В. Разработка и оптимизация бота в Telegram для эффективного управления задачами на предприятии // Вестник ДГТУ. Технические науки. 2024. № 2. URL: <https://cyberleninka.ru/article/n/razrabotka-i-optimizatsiya-bota-v-telegram-dlya-effektivnogo-upravleniya-zadachami-na-predpriyatii> (дата обращения: 14.10.2025).
15. Одилов З.Р. Изучение информационных технологий и использование современного компьютерного программного обеспечения // E-Scio. 2021. № 4 (55). URL: <https://cyberleninka.ru/article/n/izuchenie-informatsionnyh-tehnologiy-i-ispolzovanie-sovremennogo-kompyuternogo-programmnogo-obespecheniya> (дата обращения: 14.10.2025).

**Конфликт интересов:** Авторы заявляют об отсутствии конфликта интересов.

**Conflict of interest:** The authors declare that there is no conflict of interest.