

Литература

1. Dekun Tao MQTT in Python with Paho Client: Beginner's Guide 2025 – URL: <https://www.emqx.com/en/blog/how-to-use-mqtt-in-python> (дата обращения: 11.10.2025).

**СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
ЧАТ-БОТОВ НА БАЗЕ МЕССЕНДЖЕРА TELEGRAM****М. К. Залесский, О. Н. Андрейчук, И. А. Серeda***Национальный детский технопарк, г. Минск, Республика Беларусь*

Описана разработка инновационной системы автоматизированного проектирования и развертывания чат-ботов мессенджера Telegram. Ключевой особенностью системы является использование генеративных больших языковых моделей (LLM), что позволяет полностью автоматизировать процесс создания программного обеспечения на основе текстовых описаний функционала на естественном языке. Это решает проблему высокого порога входа, связанного с необходимостью специализированных знаний в программировании и DevOps.

Ключевые слова: автоматизация, искусственный интеллект, чат-бот, база данных, языковой запрос.

**A SYSTEM FOR AUTOMATED DESIGN OF CHATBOTS BASED
ON THE TELEGRAM MESSENGER****M. K. Zalessky, O. N. Andreichuk, I. A. Sereda***National Children's Technopark, Minsk, Republic of Belarus*

The project developed an innovative system for automated design and deployment of chatbots for the Telegram messenger. A key feature of the system is the use of generative large-scale language models (LLM), which enables fully automated software development based on natural-language descriptions of functionality. This solves the high barrier to entry associated with the need for specialized programming and DevOps knowledge.

Keywords: automation, artificial intelligence (AI), chatbot, database, natural language query.

Современный этап развития цифровых коммуникаций характеризуется растущим спросом на автоматизированные системы взаимодействия, в частности, Telegram-ботов, которые способны решать широкий спектр задач – от службы поддержки до автоматизации бизнес-процессов [1]. Однако процесс разработки и развертывания такого бота традиционно требует значительных временных затрат, специализированных знаний в области программирования и DevOps, что создает высокий порог входа для рядовых пользователей и малого бизнеса.

Цель исследования – разработка и внедрение автоматизированной системы генерации и деплоя Telegram-ботов на основе обработки естественно-языковых запросов пользователя с помощью технологий искусственного интеллекта.

Новизна проекта заключается в полной автоматизации цикла создания программного продукта – от интерпретации естественно-языкового запроса пользователя до генерации кода и его непосредственного развертывания на сервере. Это решение стало возможным только с недавним прогрессом в области генеративного ИИ и облачной инфраструктуры, что подчеркивает его технологическую новизну и актуальность для рынка, стремящегося к максимальной демократизации и ускорению процессов разработки.

Разрабатываемая система представляет собой платформу для автоматизированной генерации и развертывания Telegram-ботов на основе естественно-языковых за-

просов пользователей [2]. Основное функциональное назначение заключается в упрощении процесса создания программного обеспечения, позволяя пользователям без технических знаний реализовывать сложные автоматизированные решения через простое текстовое описание желаемого функционала.

Система обеспечивает полный цикл создания телеграм-бота: от интерпретации пользовательского запроса до генерации работоспособного кода и его развертывания на сервере. Пользователь получает возможность создавать ботов с индивидуальной логикой работы, интегрированных с внешними API и сервисами, поддерживающих сложные сценарии взаимодействия, без необходимости написания кода, изучения документации или настройки серверной инфраструктуры.

Ключевой функцией является интеграция с генеративными LLM, которые анализируют текстовые запросы пользователей и преобразуют их в готовый программный код на Python с использованием фреймворка aiogram [3]. Система автоматически проверяет сгенерированный код на синтаксические ошибки, обеспечивая его работоспособность перед развертыванием.

С технической стороны система реализована на асинхронной архитектуре, что обеспечивает стабильную работу при одновременной обработке множества запросов на генерацию и обновление ботов. Механизмы изоляции процессов гарантируют, что сбой в работе одного бота не влияют на работу всей системы или других созданных ботов.

Для реализации системы автоматической генерации Telegram-ботов был выбран язык программирования Python, который идеально подходит для данной задачи благодаря своей простоте, читаемости кода и богатой экосистеме библиотек. Python обеспечивает эффективную работу с искусственным интеллектом, асинхронным программированием и интеграцией с различными API, что критически важно для данного проекта.

Такая комбинация функциональных возможностей и технических решений делает систему универсальным инструментом для быстрого прототипирования и развертывания автоматизированных решений в мессенджере Telegram, обеспечивая доступ к технологии создания ботов для широкого круга пользователей – от предпринимателей и маркетологов до преподавателей и организаторов сообществ.

Для начала работы с ботом-конструктором пользователь отправляет команду /start, после этого необходимо следовать инструкциям на каждом этапе создания.

Процесс создания нового бота инициируется командой /create_bot. В результате система генерирует и развертывает бота, который начинает работать автономно. Пользователь может управлять своими ботами через главное меню (/menu), вызываемое в боте-конструкторе. Для созданных ботов представлено управление, вызываемое командой /upgrade – улучшение существующего бота; /del – удаление неактуальных ботов; /my_bots – просмотр информации о своих ботах. Для модификации бота используется команда /upgrade. Пользователь выбирает бота из списка и описывает желаемые изменения, после чего система генерирует и запускает обновленную версию. Удаление бота осуществляется командой /del через выбор из списка и подтверждение действия.

В результате выполнения проекта была успешно разработана инновационная платформа для автоматизированного создания Telegram-ботов через естественноязыковые запросы. Поставленная цель достигнута – создана система, позволяющая пользователям без технических знаний разрабатывать функциональных ботов.

Гипотеза исследования подтвердилась. Использование генеративного ИИ позволяет полностью автоматизировать процесс создания программного обеспечения, делая технологию создания ботов доступной для неподготовленных пользователей. Система показала высокую эффективность в реальных условиях эксплуатации.

Литература

1. Курпатов, Я. Ю. Создание Telegram-ботов на Python / Я. Ю. Курпатов. – СПб. : БХВ-Петербург, 2022. – 320 с.
2. Искусственный интеллект и обработка естественного языка: современные подходы и перспективы / под ред. В. К. Иванова. – М. : Техносфера, 2023. – 415 с.
3. Aiogram. Documentation. // Aiogram. – 2024. – URL: <https://docs.aiogram.dev/en/latest/> (дата обращения: 10.10.2025).

ЭВОЛЮЦИЯ ИНТЕРФЕЙСОВ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: ОТ КОМАНДНОЙ СТРОКИ К ГРАФИКЕ**А. М. Морковский, О. М. Дерюжкова***Гомельский государственный университет имени Франциска Скорины,
Республика Беларусь*

Статья посвящена анализу эволюции интерфейсов системного программного обеспечения. В работе прослеживается переход от текстовых команд к визуальной навигации, а также тенденции интеграции искусственного интеллекта, голосовых помощников и адаптивных интерфейсов. Статья актуальна в контексте стремительного развития информационных технологий и роста требований к удобству и персонализации системного ПО.

Ключевые слова: интерфейс, командная строка, графический интерфейс, пользователь, эволюция, программное обеспечение.

**EVOLUTION OF SYSTEM SOFTWARE INTERFACES:
FROM THE COMMAND LINE TO GRAPHICS****A. M. Morkovskij, O. M. Deryuzhkova***Francisk Skorina Gomel State University, Republic of Belarus*

The article is devoted to the analysis of the evolution of system software interfaces. The work traces the transition from text commands to visual navigation, as well as trends in the integration of artificial intelligence, voice assistants and adaptive interfaces. The article is relevant in the context of the rapid development of information technology and the growing requirements for the convenience and personalization of system software.

Keywords: interface, command line, GUI, user, evolution, software.

Интерфейс системного программного обеспечения играет ключевую роль в организации взаимодействия между пользователем и компьютерной системой. От того, насколько он удобен, понятен и функционален, зависит эффективность работы, скорость освоения новых технологий и уровень доступности цифровых решений для различных категорий пользователей.

Цель данной работы – проследить эволюцию интерфейсов системного программного обеспечения от командной строки (CLI) к графическим интерфейсам (GUI), выявить причины перехода между этапами, а также проанализировать современные тенденции в развитии гибридных и интеллектуальных интерфейсов.

Актуальность темы обусловлена стремительным развитием информационных технологий и ростом требований к удобству и адаптивности интерфейсов. Современные пользователи ожидают от системного ПО не только стабильности и производительности, но и интуитивной визуальной среды, способной подстраиваться под их уровень подготовки и задачи. Понимание исторического пути развития интерфейсов позволяет не только оценить текущие решения, но и прогнозировать будущие направления в проектировании человеко-машинного взаимодействия.