

УДК 004.852

ИСПОЛЬЗОВАНИЕ ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА ОПОВЕЩЕНИЯ РЕЗУЛЬТАТОВ ЖЕРЕБЬЕВКИ СТУДЕНТОВ ДЛЯ УЧАСТИЯ В ТУРНИРАХ ШАХМАТНОГО КЛУБА КУЗГТУ.

Клецов Н., студент гр. ИТб-212, IV курс

Научный руководитель: Протодияконов А.В., доцент (к.н.) кафедры ИиАПС

Кузбасский государственный технический университет имени

Т. Ф. Горбачева

г. Кемерово

В контексте современной парадигмы цифровизации, автоматизация рутинных операций в различных сферах деятельности, включая спортивные организации, приобретает первостепенное значение. В частности, в шахматном клубе КузГТУ, процедура формирования пар участников турниров традиционно выполняется в ручном режиме, что сопряжено с существенными затратами времени и высоким риском возникновения человеческих ошибок. С целью повышения эффективности и точности данного процесса, предложена концепция разработки интеллектуальной системы оповещения, базирующейся на использовании чат-бота, способного автоматизировать процедуру жеребьевки с минимальным участием персонала.

Чат-бот будет выполнять следующие функции:

1. Автоматизированная жеребьевка: Чат-бот будет извлекать список участников из базы данных, сортировать их по рейтингу и формировать пары для турнира. Это позволит избежать ошибок, связанных с вводом данных и обеспечит справедливое распределение участников.

2. Уведомления участников: После завершения жеребьевки чат-бот будет автоматически уведомлять студентов о результатах жеребьевки, сообщая им номера столов и ФИО оппонентов. Это значительно упростит процесс информирования и сделает его более оперативным.

3. Снижение нагрузки на организаторов: Автоматизация процесса жеребьевки позволит организаторам турниров сосредоточиться на других важных аспектах, таких как подготовка к турниру и необходимое взаимодействие с участниками, например, для объяснения правил игры.

Внедрение чат-бота в процесс жеребьевки студентов шахматного клуба КузГТУ не только упростит организацию турниров, но и повысит их качество. Система автоматизированного оповещения будет способствовать более эффективному управлению турнирами и улучшению взаимодействия между участниками и организаторами.

В рамках настоящей статьи рассматривается возможность интеграции чат-бота в информационную инфраструктуру турнира с целью оперативного оповещения участников.

Программная реализация чат-бота предполагает его функционирование в среде обмена сообщениями Telegram. Разработка осуществлялась с использованием языка C# и интегрированной среды разработки Visual Studio. Архитектура системы включает в себя реляционную базу данных, реализованную на основе СУБД PostgreSQL server, предназначенную для хранения структурированной информации об участниках, включающей в себя уникальный идентификатор (номер зачетной книжки), рейтинг Эло и полные персональные данные.

Для обеспечения бесперебойного доступа к актуальной информации предполагается непрерывное функционирование программного обеспечения, отвечающего за работу чат-бота, на выделенном сервере.

После того, как администратор турнира в роли оператора программы запустит необходимую команду на сервере, каждый из предварительно зарегистрированных студентов в базе данных PostgreSQL получит оповещение о пройденной жеребьевке и информацию по ней. Конечно, для начала, студента необходимо авторизовать в телеграм-боте. Диаграмма деятельности процесса авторизации пользователя представлена на рисунке 1.

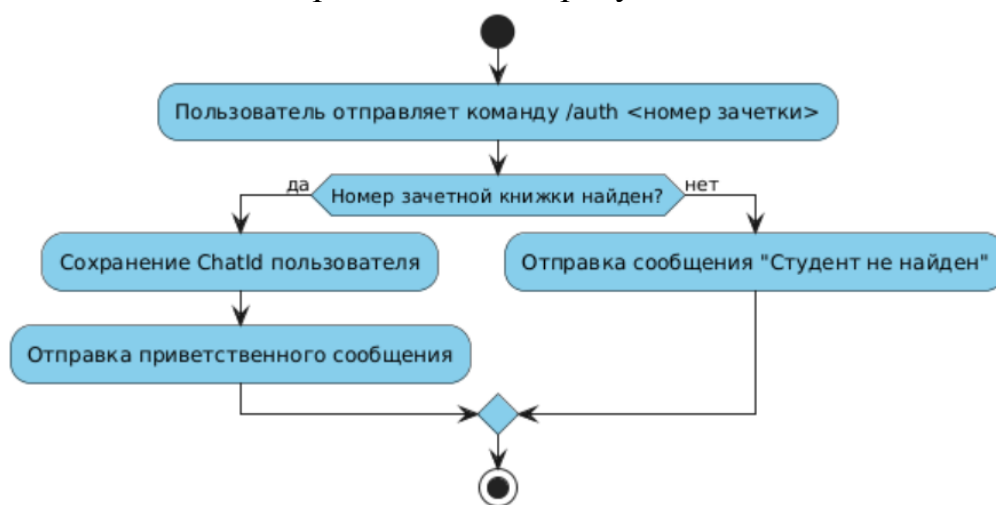


Рисунок 1 – Диаграмма деятельности процесса авторизации пользователя.

В силу ограничений, накладываемых платформой Telegram в целях предотвращения несанкционированной рассылки сообщений (спам), чат-боты не обладают функциональностью инициирования диалога с пользователем. В связи с этим, возникает необходимость в установлении соответствия между учетной записью студента, предварительно зарегистрированной в базе данных, и идентификатором пользователя, подключившего чат-бота. С целью реализации данной задачи, в приветственном сообщении пользователю предлагается ввести команду, содержащую номер его зачетной книжки, ранее

предоставленный администратору турнира при регистрации. После получения данной информации, чат-бот осуществляет обращение к базе данных с запросом на поиск записи о студенте с указанным номером зачетной книжки. В случае успешного обнаружения соответствующей записи, в атрибут, хранящий информацию о данном шахматисте, заносится идентификатор чата с пользователем. При положительном результате поиска, пользователю направляется автоматическое подтверждающее сообщение, содержащее обращение по имени и отчеству. В противном случае, пользователю отправляется уведомление о том, что студент с указанным номером зачетной книжки не был найден в базе данных.

После запуска администратором турнира команды по началу жеребьевки, авторизовавшийся студент получит данные жеребьевки. Процесс формирования пар участников соревнований, представленный в виде диаграммы деятельности на рисунке 2.

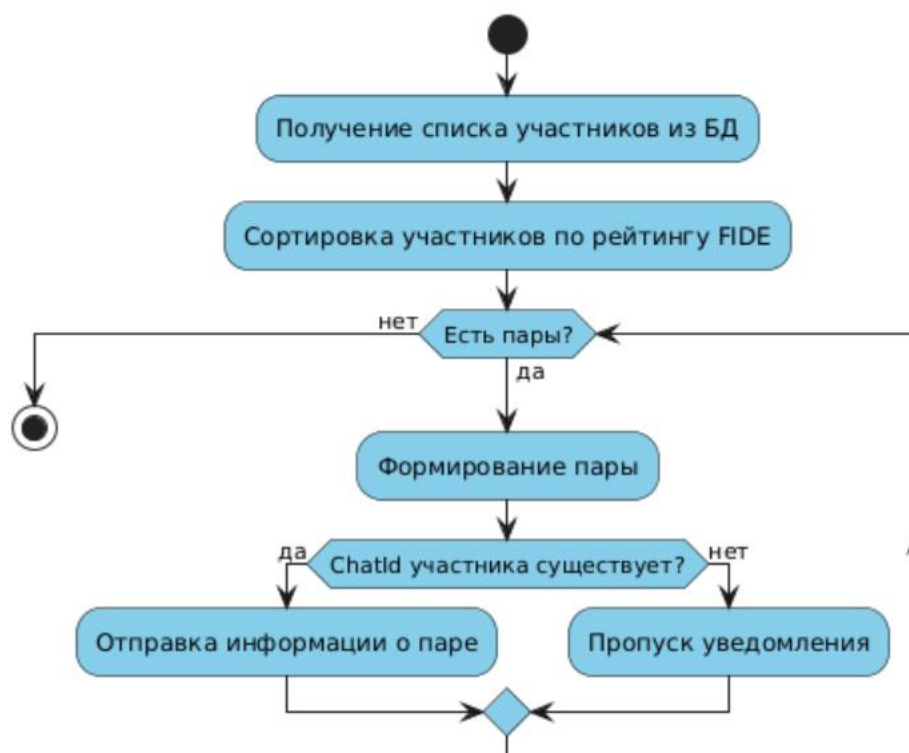


Рисунок 2 – Диаграмма деятельности процесса формирования пар участников соревнований.

Первым этапом реализации проекта является процедура регистрации чат-бота в среде Telegram. Процесс регистрации всех создаваемых чат-ботов осуществляется посредством взаимодействия с официальным ботом – BotFather. С использованием интерфейса BotFather задаются атрибуты чат-бота, такие как отображаемое имя, предназначенное для визуализации пользователям, и уникальный идентификатор, используемый для поиска бота в системе, который должен соответствовать установленному шаблону (завершаться суффиксом "bot", например, @chesskuzstu_bot). Помимо конфигурирования

внешних параметров, также предоставляется возможность настройки функциональных характеристик чат-бота.

Взаимодействие с BotFather, аналогично другим чат-ботам, осуществляется посредством отправки и обработки команд. Перечень самых основных команд, доступных в интерфейсе BotFather:

- /newbot – инициирование процесса создания нового чат-бота;
- /token – получение уникального токена, необходимого для авторизации бота.

После успешного назначения имени и уникального идентификатора, система генерирует токен – секретную последовательность символов, используемую для аутентификации приложения и требующую обеспечения конфиденциальности.

Информация о созданном боте Telegram представлена на рисунке 3.

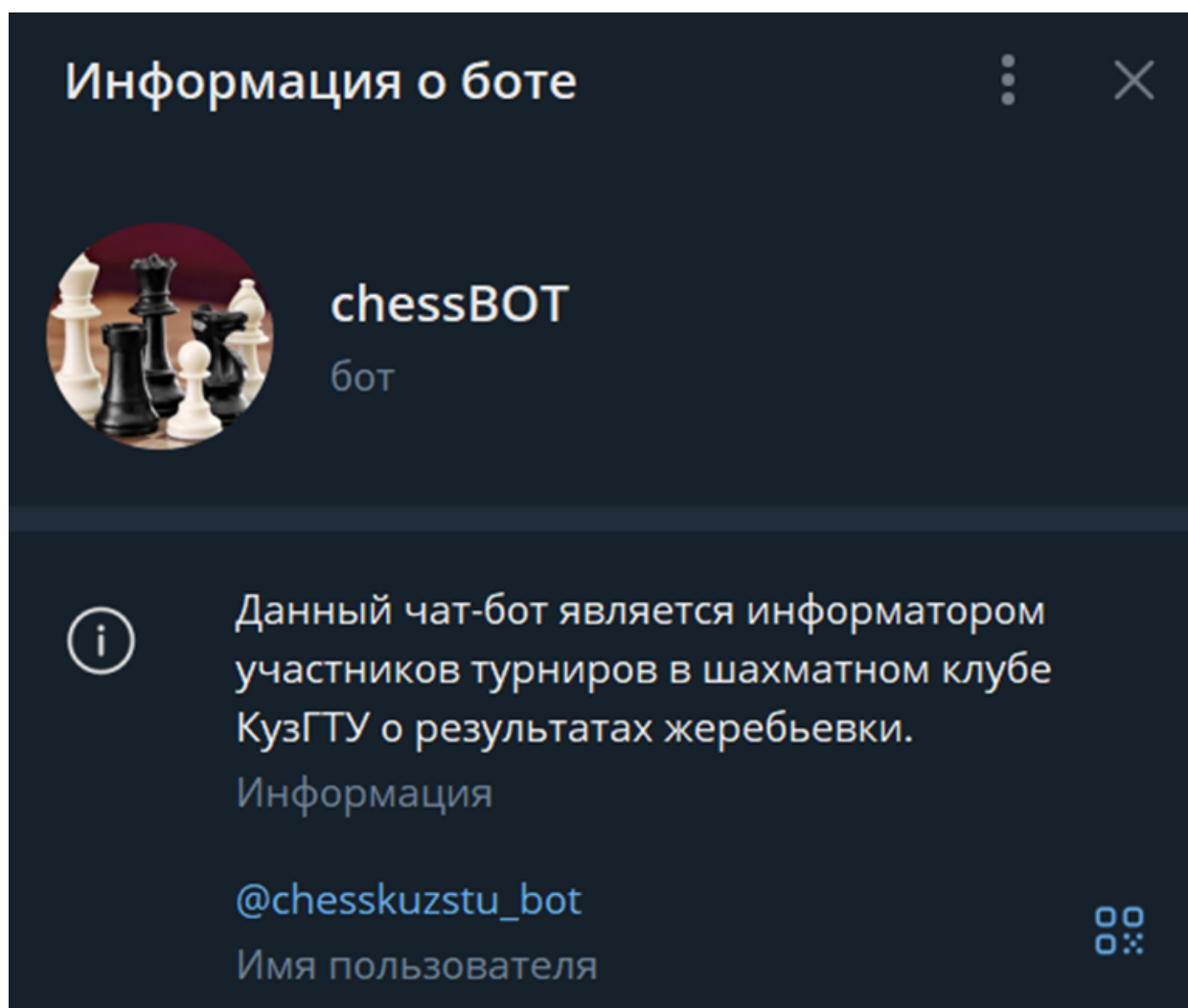


Рисунок 3 – Информация о созданном боте Telegram.

Разработка функциональности чат-бота и его интеграция с платформой Telegram требует установки специализированного пакета Telegram.Bot в среде Visual Studio. Этот пакет, доступный через систему управления пакетами NuGet, предоставляет необходимые программные интерфейсы (API) для взаимодействия с Telegram, позволяя реализовать отправку и получение сообщений, обработку команд и другие функции. Использование NuGet упрощает процесс установки и обновления пакета, обеспечивая совместимость с используемой версией Visual Studio и другими библиотеками.

Для администрирования и мониторинга работы чат-бота может быть использовано консольное приложение. Этот тип приложения характеризуется простотой разработки и низкими требованиями к ресурсам, что делает его удобным инструментом для выполнения административных задач. Консольное приложение может предоставлять возможность запуска и остановки бота, просмотра журналов работы, управления параметрами конфигурации и выполнения других операций, необходимых для поддержания работоспособности системы.

Функционал консольного приложения представлен на рисунке 4.

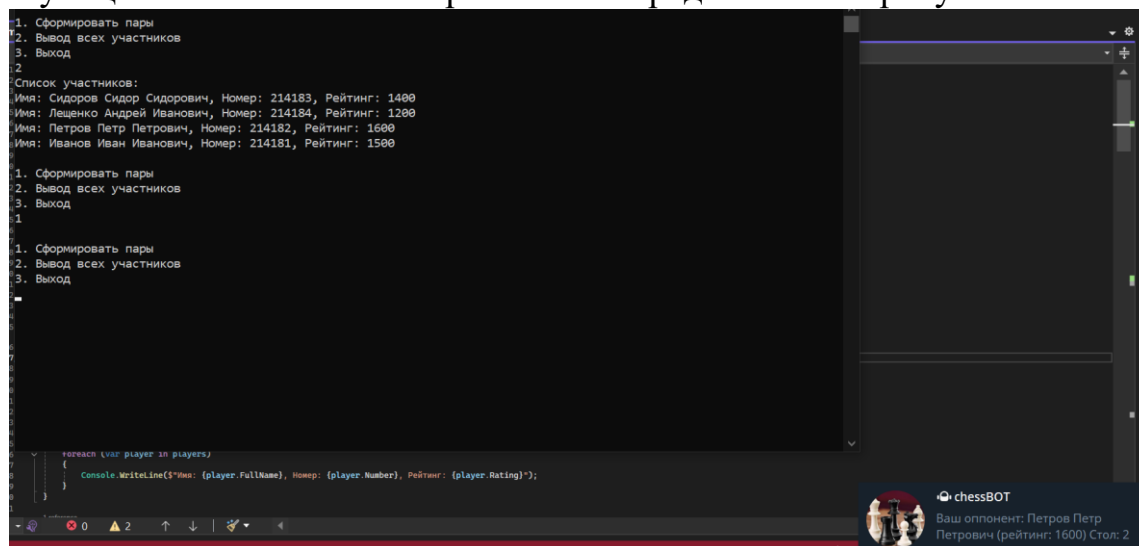


Рисунок 4 – Функционал консольного приложения для жеребьевки студентов для участия в турнирах шахматного клуба КузГТУ.

Для обеспечения взаимодействия пользователя с чат-ботом предусмотрен следующий набор команд:

- /start – данная команда автоматически иницируется при первом обращении пользователя к чат-боту. В ответ на данную команду система предоставляет пользователю подробные инструкции по прохождению процедуры аутентификации, необходимой для получения доступа к функциональности чат-бота.

- /auth [номер зачетной книжки] – данная команда предназначена для аутентификации пользователя в системе. Для прохождения аутентификации пользователю необходимо ввести свой номер зачетной книжки, который используется для проверки его наличия в базе данных. В случае успешной

аутентификации, система отправляет пользователю персонализированное приветственное сообщение, содержащее его ФИО. Дополнительно может быть реализована система двухфакторной аутентификации для повышения безопасности.

— /deauth — данная команда позволяет пользователю завершить текущий сеанс работы с чат-ботом и выйти из системы. После выполнения данной команды пользователь должен будет повторно пройти процедуру аутентификации для получения доступа к функциональности чат-бота.

Результаты тестирования разработанного чат-бота «chessBOT», демонстрирующие корректность работы основных функций, представлены на рисунке 4.

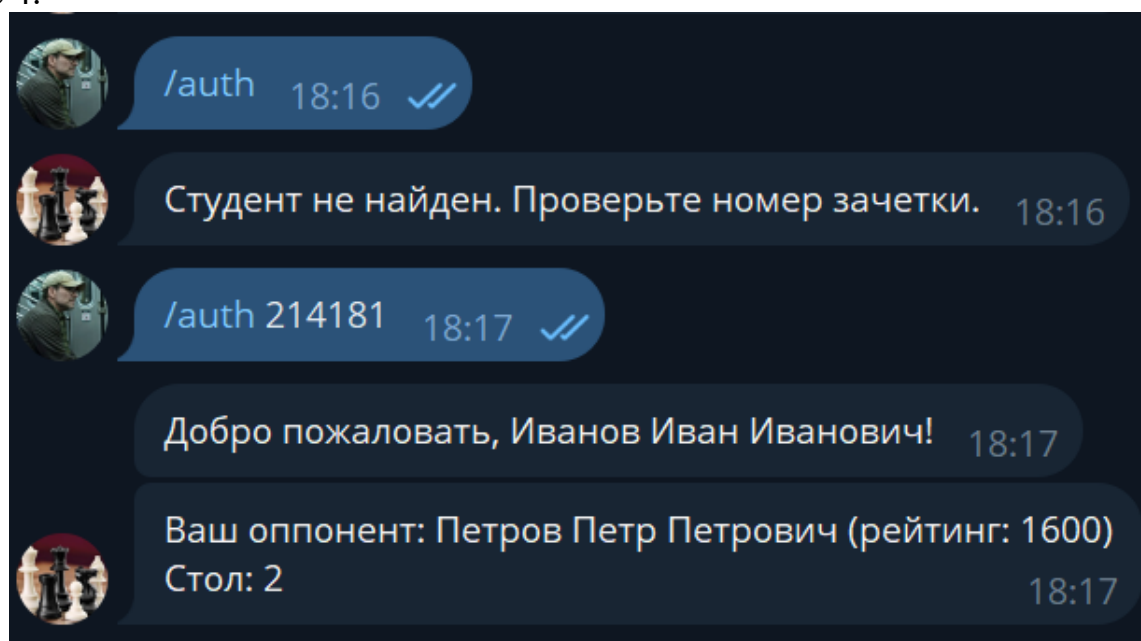


Рисунок 4 – Результаты тестирования разработанного бота «chessBOT» в Telegram.

Интеграция чат-бота в деятельность шахматного клуба КузГТУ позволит отказаться от практики персонального оповещения участников турниров, что приведет к значительному снижению нагрузки на судейский корпус и минимизации влияния человеческого фактора на процесс организации соревнований. Автоматизация процесса оповещения позволит судьям сосредоточиться на более важных задачах, таких как контроль за соблюдением правил и разрешение спорных ситуаций.

Кроме того, разработанное программное решение обладает потенциалом масштабирования и адаптации для использования в других спортивных дисциплинах и организациях, что позволяет расширить область его применения и повысить экономическую эффективность разработки. Возможность адаптации к другим видам спорта обеспечивается за счет модульной архитектуры системы и использования стандартных протоколов обмена данными.

Список литературы:

1. Документация Telegram: Боты. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots> (дата обращения: 23.03.2025).
2. Справочник по Telegram Bot API. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots/api> (дата обращения: 23.03.2025).
3. Пишем бота telegram на C#. [Электронный ресурс] URL: <http://aftamat4ik.ru/pishem-bota-telegram-na-c/> (дата обращения: 24.03.2025).
4. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: <http://sp.cs.msu.ru/ooap/exer2017.html#exer34> (дата обращения: 26.03.2025).