

Региональная открытая конференция по научно-техническому
творчеству школьников «Лабиринты науки»

Секция

информационно-телекоммуникационные технологии

Индивидуальный проект

**Создание Телеграм-бота для тестирования по материалам
подготовки к ЕГЭ**

Выполнил:

учащийся 11 класса ЦО

«Великосельская СШ»

Шляпнев Иван Андреевич

Руководитель:

Сидорович Светлана

Николаевна - учитель

информатики высшей

квалификационной категории

ЦО «Великосельская СШ»

г. Ярославль,

2025г.

Оглавление

Введение.....	2
Глава 1. Теоретические основы разработки Telegram-ботов: понятие, эволюция и функциональные возможности	3
1.1. Понятие Телеграм-бота, классификация и недостатки.....	3
1.2. Суть и способы создания Телеграм-ботов	5
Глава 2. Практическая разработка Telegram-бота	7
1. Создание Телеграм-бота.....	7
2. Реализация, тестирование, обработка результатов Телеграм-бота.	8
Заключение	10
Библиографический список	10
Приложения	13

Введение

Понятие «телеграм-бот» относительно недавно интегрировалось в повседневный цифровой лексикон рядового пользователя. Несмотря на то, что значительная часть аудитории цифровых сервисов имеет опыт взаимодействия с подобными программными решениями — будь то выполнение утилитарных задач или получение релевантной информации, — фундаментальное понимание их архитектуры и механизмов функционирования остаётся за пределами внимания большинства пользователей.

Актуальность выбора данной темы обусловлена личной профессиональной ориентацией автора. В рамках текущего этапа профессионального самоопределения осуществлён целенаправленный переход к освоению языка программирования Python, который рассматривается как ключевая технологическая платформа для дальнейшей профессиональной реализации.

Цель: создание Телеграм-бота, способного провести тестирование пользователя с целью определения уровня его подготовки по предмету.

Задачи:

1. Изучить понятие и развитие Телеграм-ботов.
2. Разобраться в устройстве и работе Телеграм-ботов.
3. Написать Телеграм-бота и проверить его работоспособность.
4. Провести публичное испытание бота.
5. Собрать обратную связь у респондентов о работе Телеграм-бота.

Объект исследования: мессенджеры.

Предмет исследования: Телеграм-бот.

Гипотеза: интерфейс Telegram-бота, структурированный по привычному для пользователей принципу и мгновенной обратной связью, будет восприниматься респондентами как более удобный и мотивирующий, чем традиционные онлайн-тесты в браузере.

Теоретическая значимость: поможет углубить знания в сфере информационных технологий и создании Телеграм-ботов.

Практическая значимость: возможность на практике создать своего Телеграм-бота и проверка гипотезы исследования.

Методы исследования: анализ, синтез, сравнение, наблюдение, тестирование, анкетирование.

Глава 1. Теоретические основы разработки Telegram-ботов: понятие, эволюция и функциональные возможности

1.1. Понятие Телеграм-бота, классификация и недостатки.

Телеграм-боты — автоматизированные аккаунты в мессенджере Telegram, предназначенные для обработки и отправки сообщений по заданным алгоритмам. Взаимодействие с ботом происходит через чат (личный или групповой): пользователь отправляет команду — бот выполняет запрограммированное действие[21].

Работа бота обеспечивается за счёт API (application programming interface — программного интерфейса приложения), который позволяет программам обмениваться данными. Управление логикой бота осуществляется через HTTPS-запросы (защищённый протокол передачи данных) к API Telegram[6].

Ключевая особенность ботов — расширение функционала мессенджера: на основе текстовых команд пользователь может получать нужные данные или услуги прямо в чате[14].

Типология Телеграм-ботов по функциональному назначению:

1. Информационные — предоставляют актуальные данные из внешних источников: прогноз погоды; курсы валют; новостные сводки; расписания и т. п. Отличительная черта — оперативное обновление информации и удобная визуализация[2].
2. Сервисные (помощники) — автоматизируют повседневные задачи: установка напоминаний; перевод и перефразирование текстов; проведение опросов. Упрощают рутинные операции для пользователя[2].
3. Бизнес-боты — решают корпоративные задачи: обслуживание клиентов (служба поддержки); продажа товаров и услуг; управление персоналом. Применимы как для малого бизнеса, так и для крупных компаний[10].
4. Развлекательные — создают контент и организуют досуг: игры и викторины; генерация изображений и текстов; подбор собеседников[2].

Таким образом, Телеграм-боты формируют многофункциональную экосистему, способную закрывать широкий спектр пользовательских и бизнес-потребностей.

Телеграм-боты, несмотря на функциональность, имеют ряд существенных недостатков. Во-первых, недостаточная защита данных может привести к утечке конфиденциальной информации пользователей. Во-вторых, злоумышленники нередко используют ботов для мошенничества — например, для фишинга или продажи личных данных. В-третьих, работа

бота полностью зависит от стабильного интернет-соединения, что создаёт проблемы в зонах с плохим покрытием. Наконец, технические ошибки в коде и отсутствие своевременной поддержки разработчика могут нарушить работоспособность бота и ухудшить пользовательский опыт[20].

Приходим к выводу, что создание Телеграм-бота носит большую ответственность. Любой недочёт со стороны разработчика может испортить опыт взаимодействия пользователя и поставить под угрозу сохранность личных данных. Также требуется поддержка бота, чтоб он не утратил свою работоспособность.

1.2. Суть и способы создания Телеграм-ботов

Создание Telegram-бота — многоэтапный процесс, сложность которого определяется целевым функционалом. Чем амбициознее задачи бота, тем больше ресурсов (временных, технических, квалификационных) потребуется на его реализацию[17].

Существует три основных подхода к разработке. Первый — собственноручное программирование: требует глубоких знаний языка, но даёт максимальную свободу в проектировании функционала. Второй — использование готовых шаблонов: не предполагает навыков кодинга, однако серьёзно ограничивает возможности кастомизации. Третий — работа в конструкторах: позволяет создавать ботов с расширенным функционалом при базовом понимании логики программирования, хотя выбор опций зависит от конкретного инструмента[22].

В рамках данного проекта выбран метод собственноручного написания кода, что предполагает прохождение трёх ключевых этапов. На стадии проектирования формулируются цели бота, определяются технологии разработки и стратегия дальнейшей поддержки. Программирование — наиболее трудоёмкий этап, в ходе которого реализуется функционал посредством кодирования. От качества проектирования напрямую зависит

эффективность этой фазы. Завершающий этап — отладка: выявление и устранение ошибок, влияющих на пользовательский опыт[19].

Для реализации бота выбран язык программирования Python. Его ключевые преимущества — простой синтаксис, лёгкость освоения и обширная экосистема инструментов. В отличие от более мощных, но сложных в освоении языков (например, C++), Python позволяет быстро создавать работоспособные решения, минимизируя затраты на разработку[15,16].

В качестве фреймворка использован Aiogram — асинхронный инструмент для работы с Telegram Bot API. Его достоинства: высокая производительность (обработка множества запросов одновременно); современный стек технологий (asyncio, aiohttp); активное сообщество и подробная документация[5,22].

Альтернативные фреймворки — PyTelegramBotAPI (подходит для простых проектов) и Telethon (оптимален для массовых рассылок) — в данном случае менее релевантны из-за ограничений по масштабируемости или специфичности функционала[5,6].

Для разработки выбрана интегрированная среда Visual Studio Code (VS Code). Её преимущества: кроссплатформенность и лёгкость установки; интуитивный интерфейс и широкие возможности кастомизации; встроенная поддержка Python и плагинов для работы с Telegram-ботами; наличие ИИ-помощника Copilot для ускорения написания кода[18].

Хотя PyCharm также популярен среди Python-разработчиков, VS Code предпочтительнее благодаря большей гибкости, бесплатному доступу и обширной библиотеке расширений.

Таким образом, комбинация Python + Aiogram + VS Code формирует оптимальный технологический стек для создания Telegram-бота. Этот набор инструментов обеспечивает баланс между простотой реализации, гибкостью конфигурации и перспективами масштабирования проекта. Учитывая, что

обе среды программирования в виде Visual Studio Code и PyCharm приветливы к начинающим и неопытным программистам, для дальнейшей работы будет выбран VS Code, поскольку обладает большим функционалом и количеством различных документаций для пользователя.

Глава 2. Практическая разработка Telegram-бота

1. Создание Телеграм-бота

В ходе проектирования было определено, что бот будет проводить тестирование по четырём основным дисциплинам (русский язык, алгебра, геометрия, обществознание) и пяти модулям курса МХК для 10-го класса. Для каждой дисциплины подготовлены тесты: по 20 вопросов в формате ОГЭ для основной группы предметов и по 10 вопросов по главам учебника для МХК. По завершении тестирования пользователь получает итоговый балл, список допущенных ошибок с пояснениями и предложение пройти другой тест. Взаимодействие с ботом организовано через инлайн-кнопки, обеспечивающие удобное меню выбора и ответы на вопросы.

Для разработки были выбраны язык программирования Python, фреймворк Aiogram и среда Visual Studio Code. Наглядная блок-схема, представленная в Приложении 1, задаёт чёткую логику взаимодействия пользователя с ботом, что существенно упрощает процесс разработки.

Проект организован в виде взаимосвязанных модулей, каждый из которых отвечает за определённую часть функционала. В папке `handlers` находятся файлы, управляющие взаимодействием с пользователем: `start_handlers.py` обеспечивает приветствие и меню выбора тестов, а `test_handlers.py` реализует логику тестирования — от вывода вопросов до подсчёта баллов и выдачи результатов. Технический файл `init.py` необходим для корректной работы модуля.

Папка `keyboards` содержит файлы, отвечающие за инлайн-кнопки: `inline_buttons.py` формирует кнопки для меню и результатов, а `init.py` выполняет техническую функцию. В папке `services` расположены файлы

questions.py, где хранится база из 130 вопросов с ответами и пояснениями, и user_manager.py, обеспечивающий взаимодействие с Telegram API и обработку ошибок.

В корневой директории находятся ключевые файлы: bot.py служит точкой входа и отвечает за запуск бота; config.py хранит токен для связи с Telegram; data.json используется для логирования ошибок; states.py управляет состояниями диалога.

На этапе программирования была реализована заявленная логика работы бота. При запуске он приветствует пользователя и предлагает выбрать тест. Во время тестирования последовательно выводит вопросы с вариантами ответа, фиксирует выбор пользователя, подсчитывает баллы. По завершении теста выдаёт подробный результат, включая список ошибок с пояснениями, и предлагает перейти к другому тесту.

В процессе отладки были устранены типичные проблемы, повышающие удобство использования. В частности, реализована блокировка произвольных сообщений пользователя, чтобы не загромождать чат. Кроме того, вместо отправки новых сообщений бот теперь редактирует существующие, что делает интерфейс более чистым и удобным для восприятия.

В результате был успешно разработан и запущен Telegram-бот «ТестоВик» (@TestoV1k_bot). Он поддерживает тестирование по пяти дисциплинам, предоставляет развёрнутую обратную связь по результатам, обеспечивает удобное взаимодействие через инлайн-кнопки и демонстрирует устойчивость к типовым ошибкам ввода. Ссылка на бота: <https://web.telegram.org/a/#7746086154> (Приложение 7).

2. Реализация, тестирование, обработка результатов Телеграм-бота.

Перед реализацией Телеграм-бота была проведено пробное тестирование, чтоб узнать мнение аудитории 9, 10 и 11 классов об телеграм-

ботах. Учащиеся заполнили Яндекс форму (Приложение 3), которая включала различные вопросы о Telegram и телеграм-ботах. Результаты тестирования отображены в виде столбиков (Приложение 4). Тестирование, которое прошли 25 человек, показало следующее:

1. Большинство (84 %) пользуется Telegram.
2. 72 % знакомы с телеграм-ботами, но в то же время 72 % пользователей никогда не пользовались ими
3. Остальные 28 % использовали Телеграм-ботов для покупок и прослушивания музыки.
4. Большому числу опрошенных (68 %) интересно получить аналог РешуГИА в Telegram, который выполнен будет в виде бота.

В рамках реализации Телеграм-бота было проведено тестирование учащихся 9-11 классов. В рамках этого тестирования учащиеся ознакомились с функционалом бота, пройдя пару тестирований, узнав свой результат. После ознакомления ученики заполнили Яндекс форму, которая состояла из 4 обязательных вопросов. Всего было опрошено 26 человек[23]. (Приложение 5) Результаты тестирования отображены в виде диаграммы. (Приложение 6) Согласно полученным данным, можно сделать следующие выводы:

1. 88,5 % пользователей получили положительный опыт взаимодействия.
2. 80,8 % опрошиваемых отметили, что использовать Телеграм-бота проще, чем РешуГИА в виде экзаменатора.
3. 88,5 % готовы посоветовать бота своему знакомому.
4. 85,6 % высказались за использования подобных ботов.

Результаты анкетирования и тестирования подтвердили востребованность и перспективность образовательного Telegram-бота. Это свидетельствует о высокой приемлемости инструмента среди школьников и его потенциале для интеграции в учебный процесс.

Заключение

Цель достигнута, в рамках проекта создан Telegram-бот для тестирования учащихся по пяти дисциплинам. Решение проводит тесты в формате ОГЭ, анализирует ответы и даёт обратную связь с пояснением ошибок. Все поставленные задачи решены: изучение теории (принципы работы Telegram, обзор языков программирования), разработку плана, программирование, отладку и апробацию на учащихся 9–11-х классов.

Результаты подтвердили гипотезу: интерфейс Telegram-бота, структурированный по привычному для пользователей принципу и мгновенной обратной связью, воспринимается респондентами как более удобный и мотивирующий, чем традиционные онлайн-тесты в браузере. Анкетирование показало позитивное восприятие инструмента и готовность применять его в учёбе.

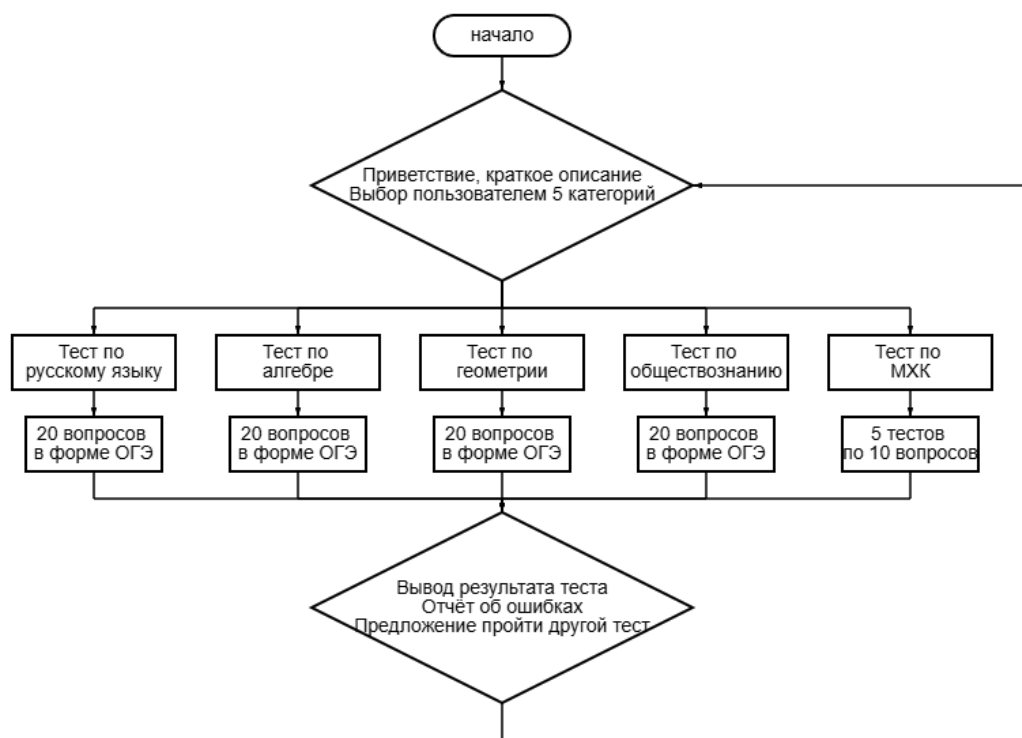
Бот готов к использованию в школьной программе. Перспективы развития — расширение дисциплин, внедрение интерактивных элементов и проработка монетизации.

Библиографический список

1. Aiogram: официальная документация [Электронный ресурс]. — URL: <https://docs.aiogram.dev> (дата обращения: 10.09.2024).
2. API Telegram Bot: документация [Электронный ресурс]. — URL: <https://core.telegram.org/bots/api> (дата обращения: 30.10.2024).
3. Habr: статья о Telegram Mini Apps. URL: <https://habr.com/ru/articles/886864/>
4. Python. Официальный сайт языка программирования [Электронный ресурс]. — URL: <https://python.org> (дата обращения: 17.01.2025).
5. Skillbox: словарь IT, определение Фреймворк. — URL: <https://blog.skillfactory.ru/glossary/framework/> (дата обращения: 05.01.2025)
6. Skillbox: статья о API, определение API. — URL: https://skillbox.ru/media/code/chto_takoe_api (дата обращения: 05.01.2025)

7. Большая Российская энциклопедия: статья о Мессенджере. — URL: <https://bigenc.ru/c/messendzher-320bdd> (дата обращения: 05.01.2025)
8. ГОСТ Р 7.0.100–2018. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. — М.: Стандартинформ, 2018. — IV, 65 с.
9. Документация Aiogram, определение Aiogram. — URL: <https://docs.aiogram.dev/en/v3.20.0.post0/> (дата обращения: 05.01.2025)
10. Документация Telegram: Боты, определение боты. — URL: <https://tigrm.ru/docs/bots> (дата обращения: 05.01.2025)
11. Знание.Вики: статья о C++, определение C++. — URL: <https://znanierussia.ru/articles/C%2B%2B> (дата обращения: 05.01.2025)
12. Знание.Вики: статья о HTTPS, определение HTTPS. — URL: <https://znanierussia.ru/articles/HTTPS> (дата обращения: 05.01.2025)
13. Знание.Вики: статья о Python, определение Python. — URL: <https://znanierussia.ru/articles/Python> (дата обращения: 05.01.2025)
14. Знание.Вики: статья о Telegram, определение Telegram. — URL: <https://znanierussia.ru/articles/Telegram> (дата обращения: 05.01.2025)
15. Знание.Вики: статья о Си (язык программирования), определение C. — URL: [https://znanierussia.ru/articles/%D0%A1%D0%B8_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F](https://znanierussia.ru/articles/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (дата обращения: 05.01.2025)
16. Иванов, А. В. Программирование на Python: от новичка до профессионала / А. В. Иванов. — СПб.: Питер, 2023. — 544 с. — ISBN 978-5-4461-1234-5.

17. Кузнецов, С. Л. Разработка чат-ботов: теория и практика / С. Л. Кузнецов. — М.: ДМК Пресс, 2022. — 320 с. — ISBN 978-5-93700-123-4.
18. Официальный сайт Visual Studio Code [Электронный ресурс]. — URL: <https://code.visualstudio.com> (дата обращения: 15.11.2024).
19. Петров, Н. А. Современные мессенджеры: архитектура и возможности интеграции / Н. А. Петров // Информационные технологии. — 2023. — № 4. — С. 45–52.
20. Сидоров, В. М. Безопасность мессенджеров и чат-ботов / В. М. Сидоров // Защита информации. Инсайд. — 2022. — № 5. — С. 67–74.
21. Телеграмм: официальный сайт [Электронный ресурс]. — URL: <https://telegram.org> (дата обращения: 17.01.2025).
22. Фёдоров, Д. С. Фреймворки для разработки Telegram-ботов: сравнительный анализ / Д. С. Фёдоров, Е. А. Морозова // Программирование и компьютерные науки. — 2023. — Т. 15, № 2. — С. 112–120.
23. Яндекс Формы: официальная документация [Электронный ресурс]. — URL: <https://yandex.ru/support/forms> (дата обращения: 20.02.2025).



```
handlers
    start_handlers.py
    test_hendlers.py
    __init__.py
keyboards
    inline_buttons.py
    __init__.py
services
    questions.py
    user_manager.py
    __init__.py
bot.py
config.py
data.json
states.py
```

Приложение 3

Пользуетесь ли вы мессенджером Telegram?

- 1) Да
- 2) Нет

Знакомы ли с Телеграм-ботами?

- 1) Да
- 2) Нет

Пользовались ли вы Телеграм-ботами?

- 1) Да
- 2) Нет

Если вы пользовались Телеграм-ботами, то напишите для чего. (Если нет, то поставьте -)
(Вариант пользователя)

Интересно ли вам было бы получить аналог РешуГИА в Telegram?

- 1) Да
- 2) Нет

Приложение 4

Пользуетесь ли вы мессенджером Telegram?

25 ответов



Знакомы ли с Телеграм-ботами?

25 ответов



Пользовались ли вы Телеграм-ботами?

25 ответов



Если вы пользовались Телеграм-ботами, то напишите для чего. (Если нет, то поставьте -)

7 ответов

Интересно ли вам было бы получить аналог РешуГИА в Telegram?

25 ответов



Приложение 5

Оцените опыт взаимодействия с Telegram ботом Тестовик:

- 1) Доволен
- 2) Смешанные чувства
- 3) Недоволен

Проще ли было вам использовать Telegram, чем какой-нибудь РешуГИА в виде экзаменатора?

- 1) Проще
- 2) Нет
- 3) Показалось труднее

Посоветовали бы данного бота своему знакомому?

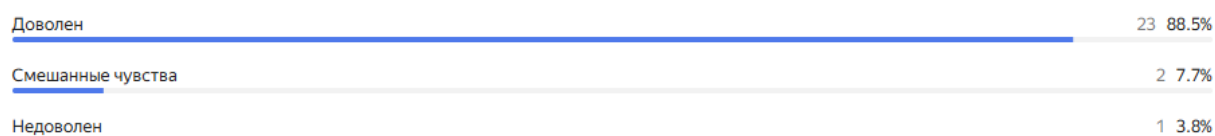
- 1) Да
- 2) Нет

Хотели бы вы использовать подобных ботов чаще в различных целях?

- 1) Да
- 2) Нет

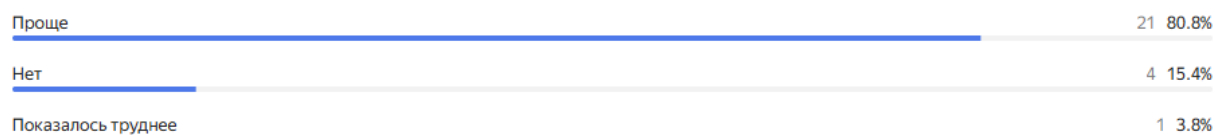
Оцените опыт взаимодействия с Telegram ботом ТестоВик

26 ответов



Проще ли было вам использовать Telegram, чем какой-нибудь РешуГИА в виде экзаменатора?

26 ответов



Посоветовали бы данного бота своему знакомому?

26 ответов



Хотели бы вы использовать подобных ботов чаще в различных целях?

26 ответов



Приложение 7

Название — ТестоВик (@TestoV1k_bot).

Ссылка — <https://web.telegram.org/a/#7746086154>