**Добавить в актуальность:** Мы позиционируем наш Telegram-бот как полезного помощника школьникам, который поможет им успешно справляться с учебными задачами. В современном мире, где ученики сталкиваются с большим количеством информации, наш бот станет надежным сопровождающим в учебном процессе. Мы убеждены, что наш бот будет особенно полезен для тех учеников, которые имеют трудности с организацией своего времени и пропускают важные события, такие как олимпиады и конкурсы. С помощью нашего бота, школьники смогут получить свои расписания, задания и напоминания в удобном и быстром формате, что позволит им более эффективно управлять своим временем и достигать своих учебных целей.

В начале работы мы изучили API MOS.RU, который реализует два HTTP запроса:

1. <https://dnevnik.mos.ru/mobile/api/schedule>?student\_id={Id}&date={Дата}

Этот запрос возвращает расписание человека по его Id и конкретной дате.

1. <https://dnevnik.mos.ru/mobile/api/profile>

Этот запрос возвращает информацию о человеке. Это нужно для получения Id пользователя.

К каждому из этих запросов нужно также добавить токен аутентификации, который отправляется в заголовках к каждому HTTP запросу.

Рассмотрим процесс создания Telegram-бота:

**Создание Telegram-бота** может быть разделено на несколько шагов:

1. **Создание бота.** Для этого нужно использовать официального бота **@BotFather**. Чтобы создать нового бота, нужно написать ему сообщение "/newbot" и следовать дальнейшим инструкциям.

2. **Настройка бота.** После создания бота, необходимо настроить его функционал. Это можно сделать с помощью Bot API, предоставляемого Telegram. Например, можно добавить команды, которые бот будет выполнять при получении определенного сообщения, настроить автоматические ответы и т.д.

3. **Написание кода.** Для того чтобы создать Telegram-бота, необходимо написать код, который будет обрабатывать запросы, полученные от пользователей. Для этого можно использовать один из языков программирования, поддерживаемых **Telegram Bot API**, мы в данном проекте использовали C#, потому что нам уже знаком этот язык и он позволяет создать красивую архитектуру для проекта.

4. **Размещение бота.** После написания кода, бот необходимо разместить на сервере, чтобы он был доступен круглосуточно. Для этого можно использовать различные облачные сервисы, например, Amazon Web Services или Heroku, мы остановились на [Render](https://render.com), потому что он предоставляет бесплатный функционал для размещения сервисов.

5. **Подключение бота к Telegram.** После размещения бота на сервере, необходимо подключить его к Telegram. Для этого нужно указать API-ключ, который был выдан **@BotFather** при создании бота.

6. **Тестирование бота.** После всех предыдущих шагов, бот готов к работе. Необходимо провести тестирование, чтобы убедиться, что он работает корректно и выполняет все заданные функции.

На рисунке 1 представлен фрагмент кода, который отвечает за формирование HTTP запросов к API MOS.RU.



Рисунок 1. Класс для реализации MOS.RU API.

Ключевой составляющей любого бота является база данных, в которой хранится множество таблиц, необходимых для работы приложения. В частности, база данных нашего Telegram-бота содержит информацию о всех зарегистрированных пользователях бота, их напоминаниях, а также таблицу с информацией об олимпиадах. Благодаря этой базе данных, наш бот может быстро и эффективно обрабатывать запросы пользователей и предоставлять им актуальную информацию о проходящих мероприятиях или олимпиадах. Мы выбрали PostgreSQL так как уже имели опыт работы с этой системой управления баз данных.

Рассмотрим функциональность нашего Telegram-бота. Он предоставляет следующие возможности:

1. Регистрация аккаунта. Пользователи могут безопасно зарегистрироваться в системе, предоставив свой логин и пароль от MOS.RU, либо используя свой аутентификационный токен от MOS.RU.
2. Регистрация олимпиады/конкурса. Администраторы могут добавлять новые мероприятия в список.
3. Получение школьного расписания. Пользователи могут быстро получить свое школьное расписание на конкретную дату или на сегодняшний день.
4. Получение домашнего задания. Пользователи могут получить домашнее задание на все предметы на определенную дату или на завтра.
5. Добавление напоминания. Пользователи могут сохранять важную информацию, такую как оповещения о дедлайнах или важных событиях, связанных с учебой.
6. Список напоминаний. Пользователи могут просмотреть список всех сохраненных напоминаний с полной информацией о них.
7. Помощь в изучении темы. Используя искусственный интеллект, наш бот помогает школьникам, которые стесняются задавать вопросы учителям при изучении какой-либо темы.

Благодаря своей функциональности и возможностям, наш бот помогает пользователям улучшать свою учебную деятельность и получать своевременную информацию о проходящих мероприятиях.

На рисунке 2 представлен процесс регистрации пользователя.

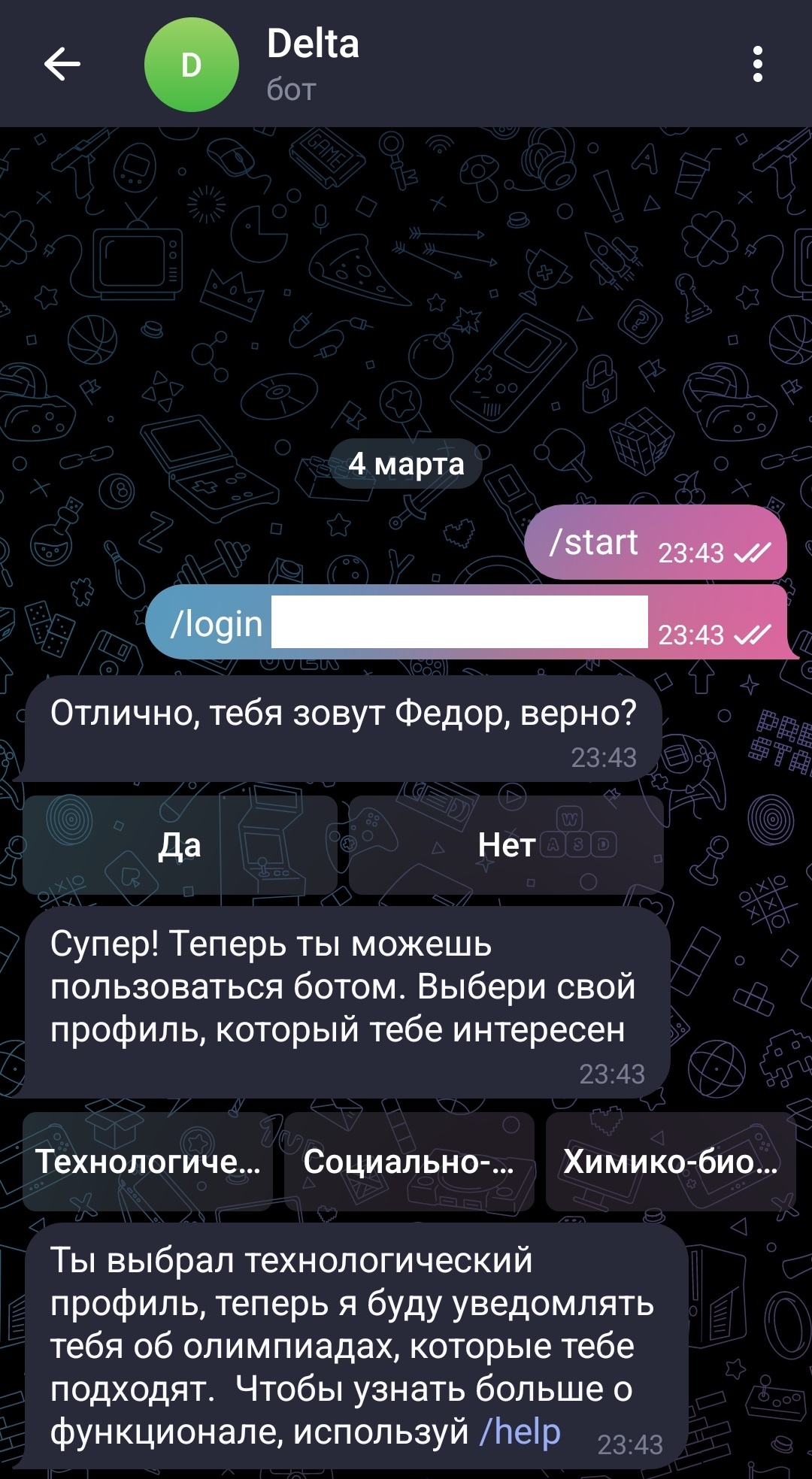


Рисунок 2. Регистрация в боте.

На рисунках 3 и 4 представлено взаимодействие с MOS.RU API: получение расписания и домашнего задания.

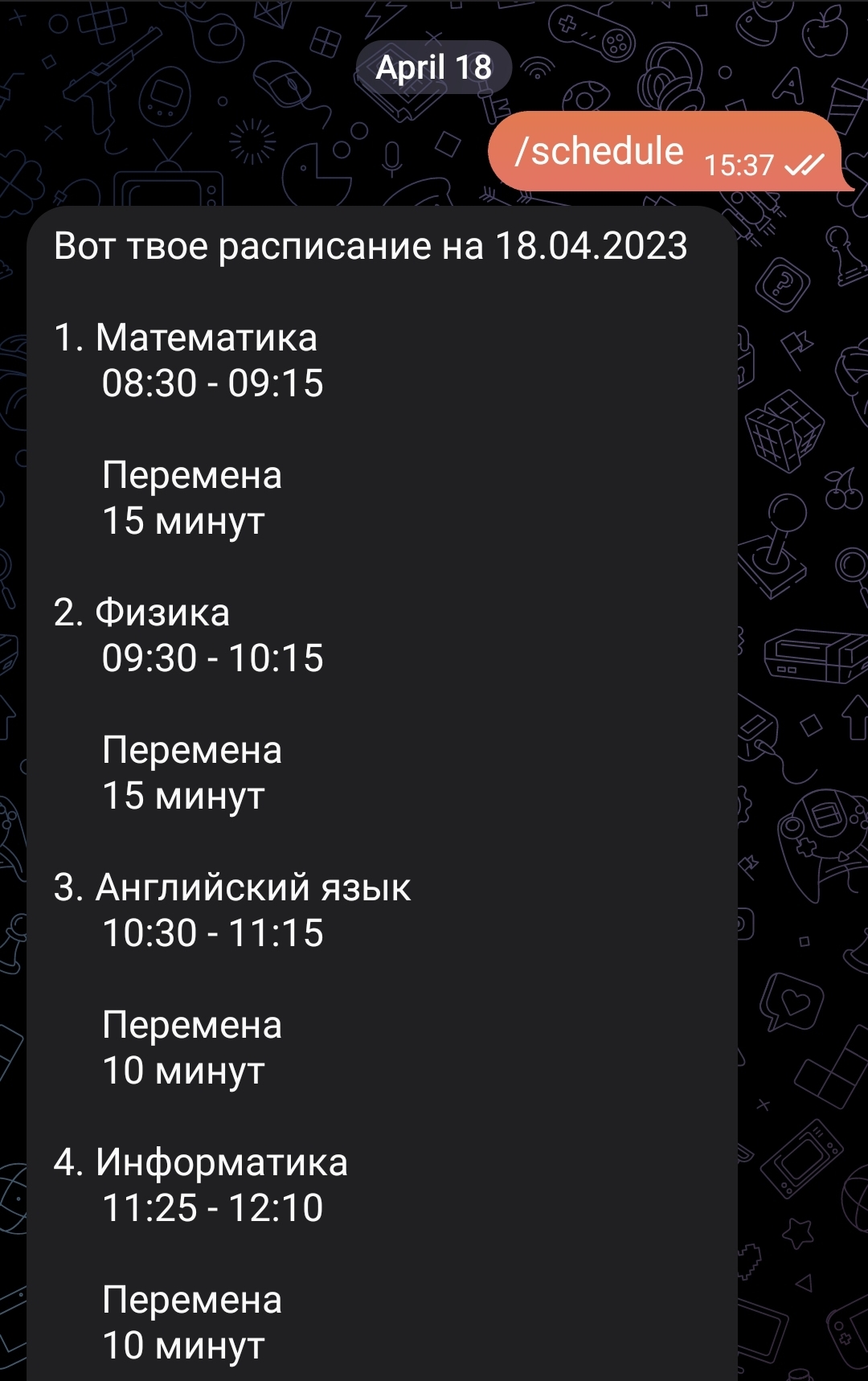


Рисунок 3. Расписание школьных предметов.

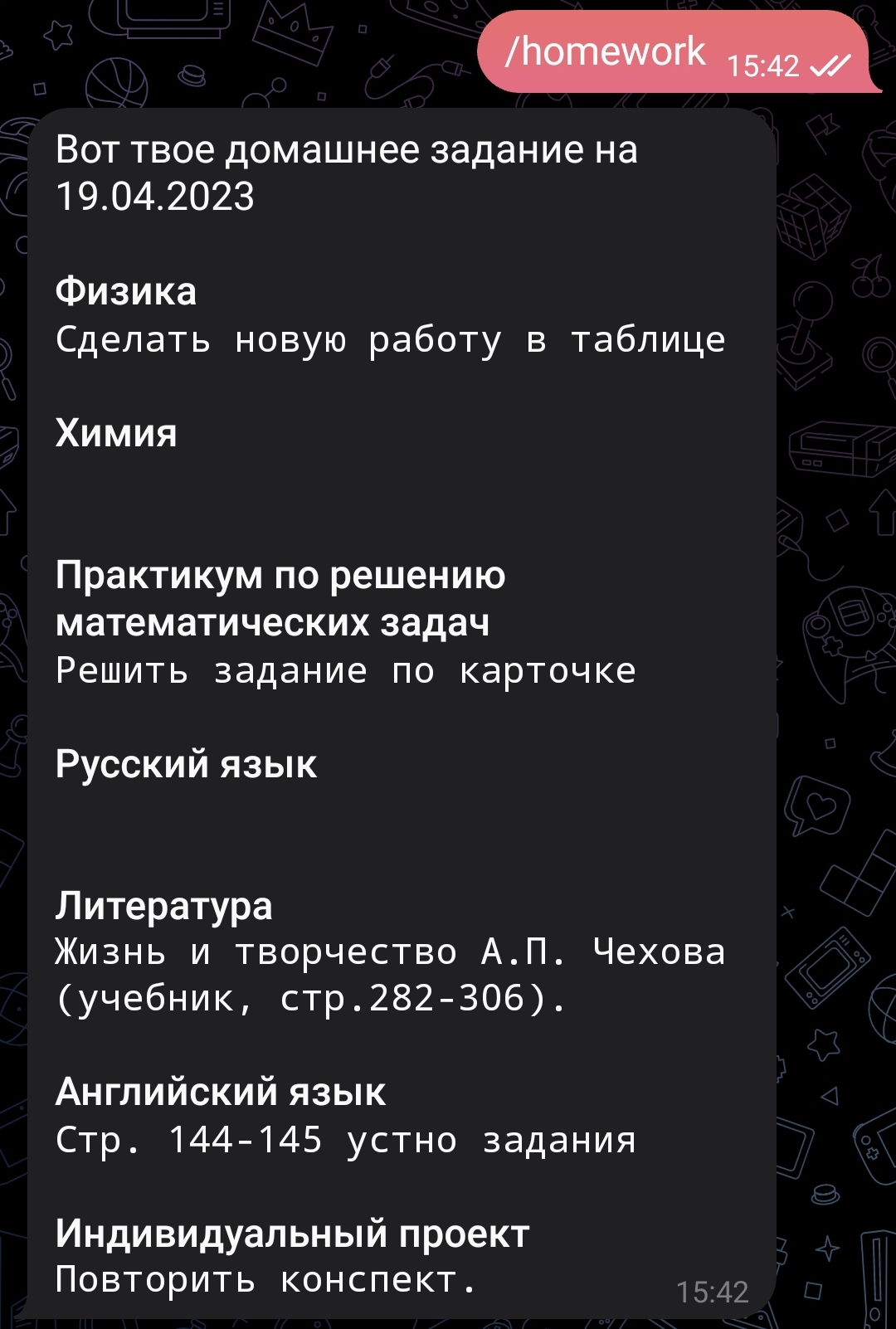


Рисунок 4. Домашнее задание.

На рисунках 5 и 6 продемонстрированы функции добавления напоминаний и олимпиад.

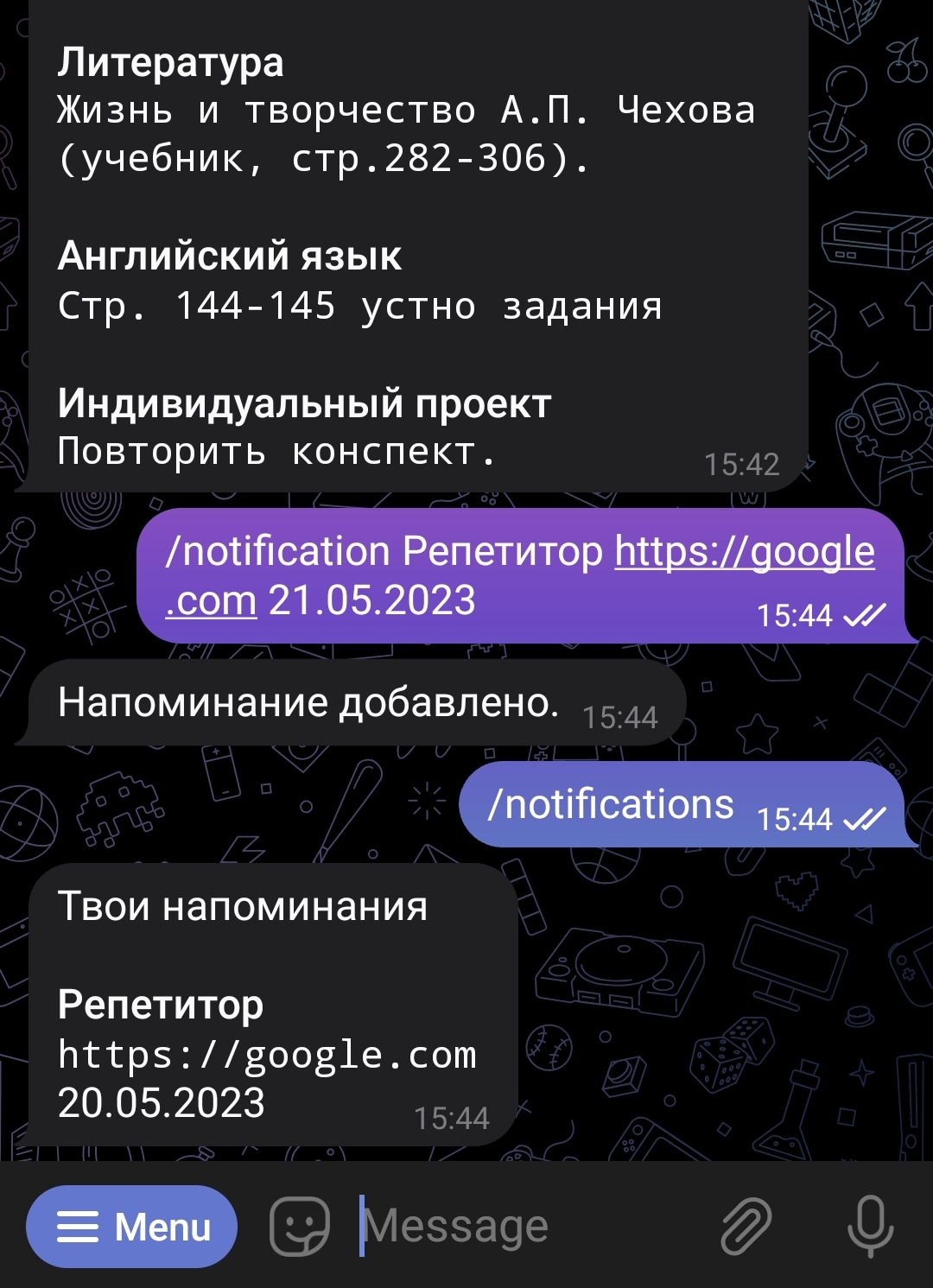


Рисунок 5. Создание напоминания и получение их списка.

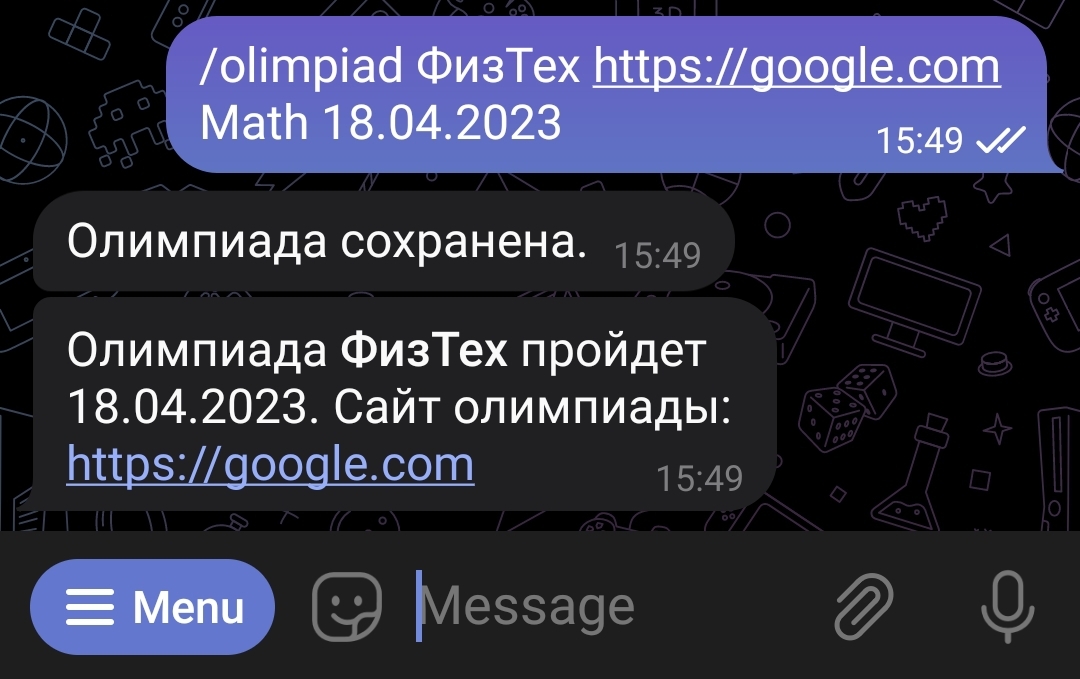


Рисунок 6. Добавление олимпиады, а также уведомление пользователей о ней.

На рисунке 7 представлен искусственный интеллект, который поможет школьникам в изучении различных тем.

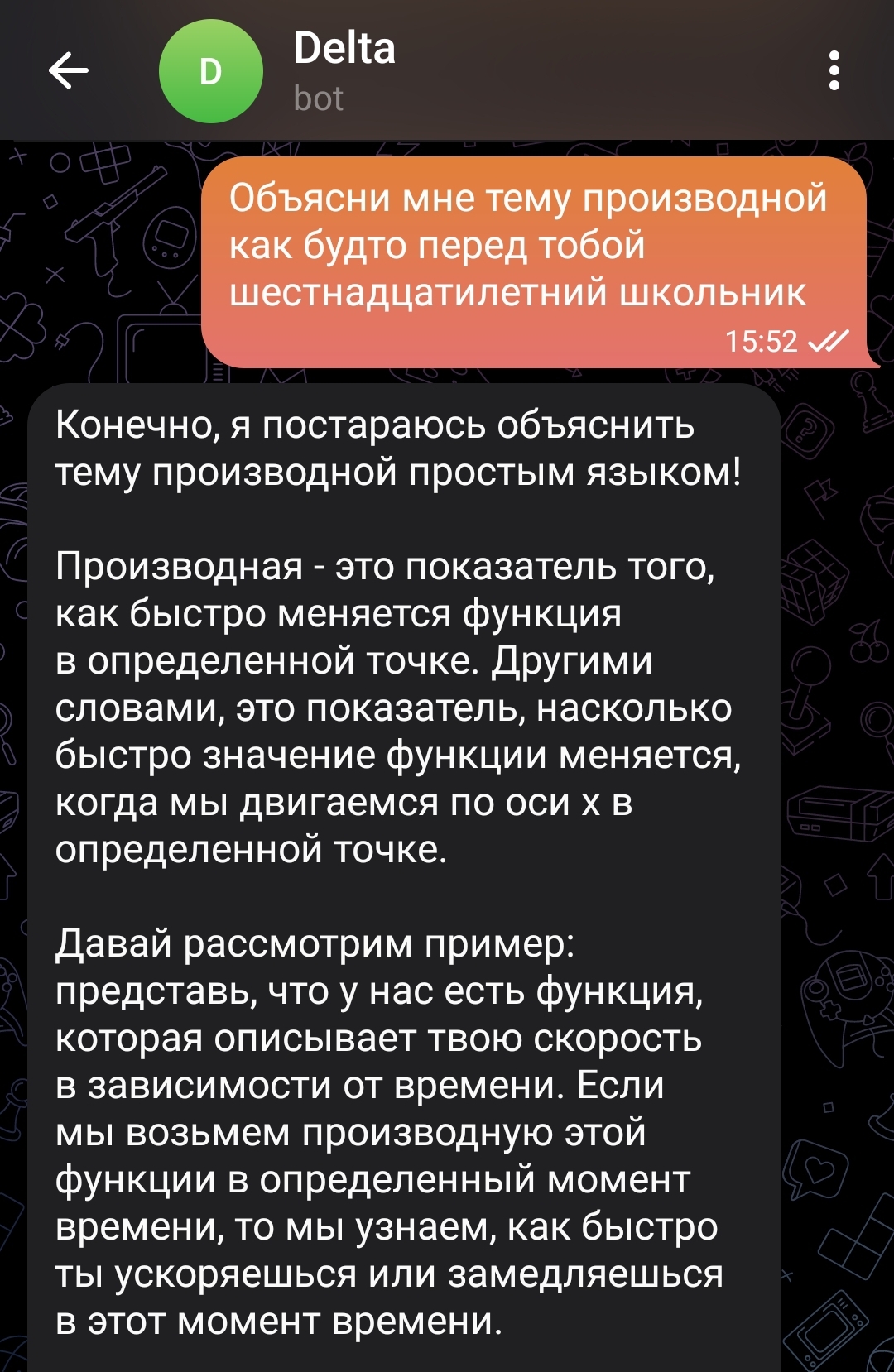


Рисунок 7. Искусственный интеллект – помощник школьников.

// Улучши данный текст, он будет использоваться для составления реферата.